Gefördert vom
bmb+f
Bundesministerium für
Bildung, Wissenschaft,
Forschung und Technologie

Verbmobil
Verbundvorhaben

From Word Hypotheses to Logical Form: An Efficient Interleaved Approach

W. Kasper
H.-U. Krieger
J. Spilker
H. Weber

DFKI Saarbrücken, University Erlangen-Nürnberg

Vm
Report 142
Juli 1996

Juli 1996

W. Kasper
H.-U. Krieger
J. Spilker
H. Weber

DFKI, Stuhlsatzenhausweg 3, 66123 Saarbrücken
University Erlangen-Nürnberg, Am Weichselgarten 9, 91058 Erlangen

Tel.: (0681)302 - 5252
Fax: (0681)302 - 5341
e-mail: {kasper|krieger}@dfki.uni-sb.de
{spilker|weber}@immd8.informatik.uni-erlangen.de

Gehört zum Antragsabschnitt: 15.7 Architektur integrierter Parser für gesprochene Sprache
15.8 Nichtsyntaktische Information für die semantische Auswertung

Die vorliegende Arbeit wurde im Rahmen des Verbundvorhabens Verbmobil vom Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMBF) unter dem Förderkennzeichen 01 IV 101 G gefördert. Die Verantwortung für den Inhalt dieser Arbeit liegt bei den Autoren.

From Word Hypotheses to Logical Form: An Efficient Interleaved Approach*

Walter Kasper, Hans-Ulrich Krieger, Jörg Spilker, Hans Weber

Abstract

This paper revisits word lattice search whose task is to find a plausible semantic interpretation for a given utterance. Our approach of interleaved search and analysis is designed to break the frontier of "toy" applications. The framework is implemented in two interacting modules, running in parallel. Instead of simply parsing a word lattice, we rather do tree decoding with a probabilistic approximation of a given grammar, employing a beam search strategy. Logical form is build up in tandem according to the decoded derivation histories, using a codescriptive HPSG grammar for dialog turns. The proposed architecture only uses the knowledge necessary in every processing step, the key aspect being an asynchronous coupling of the two specialized modules.

1 Introduction

As many scientists feel after having worked in the field for a while, processing speech, especially spontaneous speech, and deep linguistic analysis do not seem to fit very good together—for a couple of good reasons: efficiency, robustness and coverage.

In VERBMOBIL's special architecture subproject TP-15, we have gone even further by building a version of the spontaneous speech VERBMOBIL system, requiring that processing must be *time synchronous*. This, of course, presupposes that such a system (i) starts processing at the very begin of an utterance, (ii) keeps track with the speech signal, and (iii) performs word recognition, parsing, interpretation and translation all in parallel and all incrementally from left to right—the INTARC system.

The domain of VERBMOBIL consists of a collection of negotiation dialogs, where only a minor part consists of "sentences" in a strict linguistic sense (cf. Wahlster 1993 and Kay et al. 1994). A large amount of the dialog steps are long turns made of sequences of fragments, sentences and interjections. Well-trained bigram models for these dialogs show a perplexity of more than 100.

*This paper is also published in: *Proceeding of the 3rd KONVENS*, Oct. 7–9, 1996.

For the top-best chain, the best results obtained by acoustic decoding and bigram lie between 65 and 75 percent word accuracy using NIST scoring. Thus, building an analysis on the top-best chain will in general not lead to proper translations—either, leading to many total fails in case of a deep analysis or to wrong translations when using only a flat approach.[1].

To be more robust, a word lattice interface is used between word recognition and parsing (here as a synonym for tree recognition). The latter is passed incrementally (end point wise) to the lattice parsing module as a contribution to the time synchronous processing.[2] In order to guarantee that all of the uttered words are inside the lattice, we have to produce a left-connected word graph of approx. 30,000 hypotheses containing a connected word graph of 3,000 word hypotheses without dead ends when processing a turn of length 20.

Having these challenges in mind, we knew that

1. a deep analysis had to be implemented for large parts of spontaneous speech turns,
2. a highly efficient search schema had to be applied to word lattices, employing more information than just acoustic and n-gram probabilities, and
3. both aspects had to be kept distinct in processing (viz., search vs. structure building).

The solution implemented in the INTARC system represents a paradigmatic change in the architecture of speech-language systems:

- Probabilistic search is no more done by decoding word sequences. The lattice parser decodes trees out of sets of word hypotheses, using probabilistic models of (acoustics), word sequences, prosodic phrasing and trees.
- The semantics parser takes decoded trees as input and builds a logical form according to the parse history, encoded in the trees. Semantics construction is constrained by the original grammar and prosodic information.
- Time synchronous processing then requires that both modules are implemented as parallel, loosely-coupled processes.

[1]E.g., the missing of a single negative polarity item will lead to a completely false result. Date/time expressions have to be translated in-depth, since one missing particle can change the meaning completely.

[2]Details and variants of this coupling are described in Hauenstein and Weber 1994, Weber 1995, and Weber 1994.

In the remainder of this paper, we will focus on details of the approach, starting with the overall architecture of the INTARC system, describing the original grammar and proceeding with the estimation of the probabilistic grammar model. After that, we review the lattice parser and the module for semantics construction, highlighting the communication protocol and the integration of prosodic information. Finally, we give a summary and indicate several improvements which we will implement next.

2 Sketch of the Architecture

In this section, we briefly describe the overall architecture of VERBMOBIL's TP-15 dialog translation system (see Figure 1). The parsing process is distributed among two parsers, running in tandem. The first parser operates on word lattices, getting its input from the speech recognition module. The second parser embodies the constraint solver, primarily interested in building up logical form. Communication between the parsers is established through the INTARC Communication Environment ICE (cf. Amtrup 1995). ICE is based on PVM, the Parallel Virtual Machine, a system for communication between many processes in a heterogeneous network. ICE itself implements an interface layer on top of PVM, abstracting from communication channels, as it is known from Occam.

The probabilistic word-lattice parser integrates both statistical and symbolic knowledge (see Section 5). It uses the full HPSG grammar offline, viz., for training. At run time, only the context-free skeleton of the grammar is employed. However, since this set of rules overgenerates w.r.t. the original grammar, certain rule applications are in fact not valid. These are ruled out by the second parser.

Semantics construction (the so-called SEM-parser) is fed with hypotheses from the word-lattice parser, using them to deterministically reconstruct the chart on the basis of the full grammar (see Section 6). This is possible by associating every lexicon entry and every rule with an index which is added at compile time and shared by both parsers. Because the search space is massively reduced by the word-lattice parser (approx. one order of magnitude less hypotheses), unification inside the SEM-parser keeps pace with the corresponding rule application inside the first parser.

This special architecture allows for efficient filtering of word hypotheses (via the word-lattice parser) without giving up *soundness* of the analysis results (guaranteed by the SEM-parser). Since lexicon entries and rules are identified by unique indexes, expensive communication via feature structures is avoided. In case that a rule application hypothesis fails under feature structure unification (or is not applicable on prosodic grounds), a message is send back to the word-lattice parser. This is important to further reduce the set of emitted hypotheses, as explained in Section 7.

The word-lattice parser as well as the SEM-parser additionally receives hypotheses from two prosodic components (see Section 8). The one simply termed "prosody" is a detector for phrase boundaries and sentence modalities (cf. Strom 1995). The other one is a detector for focus (cf. Petzold 1995).

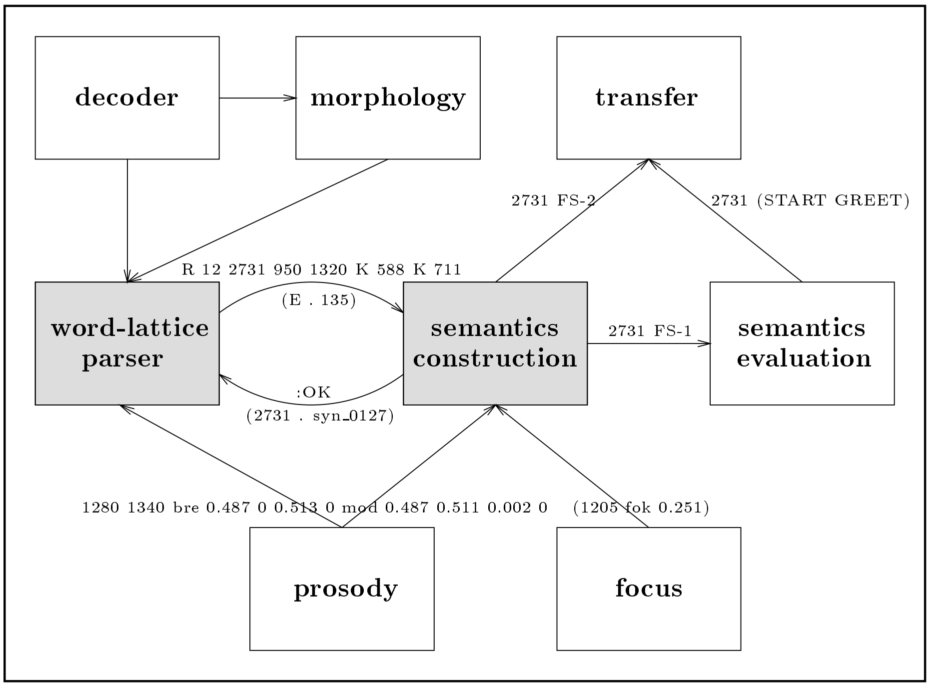

Figure 1: *The overall architecture of the* INTARC *system. The annotations at the arrows depict the different kinds of protocols between the components.*

3 HPSG Grammar for Dialog Turns

The basic units of dialogs are not sentences but *turns*, e.g., *Tut mir leid. Am neunundzwanzigsten um drei habe ich schon eine Besprechung. Dienstag den dreißigsten um drei, das ginge bei mir.* (I am sorry. On the 29th at 3 I already have a meeting. Thursday the 30th at 3, that would be fine.)

Turns usually consist of more than one of what we call a *turn segment*. In the above example, the most likely segmentation is indicated by punctuation marks. Turn segments need not be complete sentences but can be sequences of nearly any kind of phrase: *Also. Am Montag. Um wieviel Uhr denn dann?* (OK. On Monday. At what time then?)

In spoken turns the punctuation marks of course are missing, and the fact that any kind of linguistic category can also be a turn segment, that is, a "complete" utterance in itself, makes segmentation on purely linguistic grounds a highly ambiguous task. In fact, a grammar provides only weak constraints on utterances (e.g., subcategorization). On the other hand, a turn like *am Montag kommt er* (lit.: on Monday comes he) without any further clues can be understood as consisting of one declarative sentence but also as consisting of an elliptical prepositional phrase, followed by an interrogative sentence.

For the analysis of dialog turns, we use an HPSG-inspired grammar (see Pollard and Sag 1987 and Pollard and Sag 1994). The grammar consists of 65 rules, written in the typed feature formalism $\mathcal{TDL}$ (cf. Krieger and Schäfer 1994 and Krieger and Schäfer 1995). It is a codescriptive grammar specifying simultaneously syntax and semantics. In order to deal with turns consisting of several segments, the HPSG approach had to be extended, primarily to deal with the semantic composition of turn segments. Also, non-linguistic events in a dialog turn, e.g., pauses and coughs required a treatment in the grammar. The additional rules for turns do not simply concatenate turn segments but impose an intermediate structure on turns between phrasal turn segments and complete turns in order to deal, among other things, with special properties of the *uptake* phase at the beginning of a turn, with interruptions, linguistic "garbage", and echo phrases.

Semantically, turns are represented as a linear conjunction of the semantical representations of the turn segments. This conjunction is passed to semantics evaluation for further processing, such as reference resolution and dialog act identification w.r.t. the dialog model. Other extensions of the approach were required to capture information from prosody, especially information about the mood of an utterance and about focussed phrases.

The main problem for such a turn-based grammar is the problem of segmenting a turn into the correct turn segments. As indicated above, linguistic constraints are very weak and not sufficient. However, spoken language contains clues about segmentation. Taking into account such prosodic clues, turn segmentation is not only important for a correct grammatical analysis but also for the efficiency of the analysis process itself. By employing both the detector for segment boundaries as well as the turn grammar, a great deal of wrong grammatical analyses can be eliminated. How this can be achieved is described in Section 8.

4 Context-Sensitive Grammar Models

The original unification grammar would be to expensive when applied directly to word lattices. Approaches dealing with the latter (e.g., Hanrieder 1995 or Weber 1994) only assume read speech input which is an order of magnitude easier to decode than spontaneous speech. Connected word graphs used there only contain 200 or less word hypotheses (instead of 30,000).

The crucial point here is that parsing of unification grammar is basicly $\mathcal{NP}$-complete while a context-free approximation can be parsed in (less than) cubic time. Probabilistic versions of unification grammars, which help to prune derivations early, are also too expensive (e.g., PUG; cf. Weber 1995). Even if we reduce the size of the structures by distinguishing between genuine and spurious constraints, parsing is still $\mathcal{NP}$-complete (cf. Diagne et al. 1995).

A much better way is to use only the context-free backbone of the original unification grammar (which, of course, heavily overgenerates) and use a context-sensitive probabilistic model of the original grammar's derivations:

1. Parse a corpus with the original unification grammar G to produce an ambiguous tree bank B.
2. Build a stripped grammar G', such that for every rule r' in G', there is a corresponding rule r in G (and vice versa).
3. Use an unsupervised reestimation procedure to train G' on B.

The probabilities actually used were distributions on rule applications given the mother rule and its daughter number as context K. We would have wished to extend the context K to larger portions of a tree in the sense of a history-based grammar model (see Magermann 1994), but would have been running in sparse data problems then.

Since the original tree bank was produced by a HPSG-style unification grammar (where no structure sharing can be used in a tree bank), we could not use an inside-outside algorithm to estimate our distributions. Instead, we extended a PCFG reestimation procedure from Fujisaki et al. 1991 to arbitrary contexts K.

5 Lattice Parsing as Tree Decoding

The lattice parsing module is a variant of an LR-incremental active chart parser, where all empty edge introduction operations are precompiled into an LR table (cf. Weber 1995 and Weber 1994). The lattice is traversed by a beam search procedure frame-wise from left to right:

- For every new time frame, a vertex and an empty agenda of search steps are created.
- All word hypotheses ending in the actual frame are read in as edges and all pairs of edges which could be worked at are scored and pushed onto the agenda for that frame.
- Scoring is a weighted linear combination of log probability scores given by models for acoustic, bigram, grammar, and prosody models.

- As in an acoustic beam decoder (cf. Ney 1995) all steps down to a fixed offset from the maximum score are taken and all others are discarded.
- The procedure stops when the word recognition module (decoder) which supplies word hypotheses together with acoustic scores signals an end of the utterance.

Roughly speaking, we build only those trees out of billion possible ones, which span word sequences with good acoustic, bigram and prosody scores, having a good grammar score themselves.

In order to keep the search cubic in the number of frames, we represent an edge for a certain rule with given begin and end frames only once. In other words, we use structure sharing in its most radical form—even for the same analysis edge spanning different word paths in the lattice. The sharing itself is done to a fixed depth only. Every chart edge keeps a vector of length n in which additional ways leading to that edge are stored. Since we process all search steps in a strict best-first manner according to the combined scores, we can guarantee that an edge keeps the n-best word path derivation pairs for its span.

Because the algorithm globally proceeds frame-wise from left to right and because we employ structure sharing for each frame, we will find at most one passive goal edge spanning from the begin of the utterance to that frame. This goal edge will keep the n-best trees with that span. Since the structure sharing method for each edge is restricted to a fixed depth n only, unpacking a frame's n-best trees out of the chart is linear in time to the number of tree nodes used in those trees.[3]. This is due to the fact that we have to recursively traverse the edges for n trees to unpack them. Every step in that recursion takes only time $\mathcal{O}(n)$ to maximize n fields out of an $n \times n$ matrix. For unpacking an edge, the matrix consists of the vectors of active and inactive edges, keeping their local histories. This has to be done n times (for more details of the unpacking procedure, we advise the reader to a forthcoming VERBMOBIL report).

6 Semantics Construction as Constraint Solving

We have already noted that the SEM-parser gets its input (so-called *bottom-up* hypotheses) from the word-lattice parser, but also reports back failures, i.e., non-applicable rule combinations (*top-down* hypotheses). Bottom-up hypos are processed in a first-in/first-out manner, i.e., the SEM-parser deterministically reconstruct parts of the lattice parser's chart. Since the lattice parser employs information not available to the SEM-parser (e.g., bigram, acoustic scores) and

[3]Note that n is constant for a given implementation. In the INTARC 2.0 system, we have set n to 10. Otherwise worst case complexity would be cubic to n.

since the best estimated hypotheses comes in first (from left to right), we are convinced that this strategy maximizes processing efficiency.

Essentially, chart reconstruction is either achieved by copying lexical entries in case of lexical hypotheses (i.e., postulated words) or by constructing local trees, due to rule application hypotheses. Such trees are build up by means of feature structure unification. In INTARC, this task is undertaken by the sophisticated typed feature formalisms $\mathcal{TDL}$. Since the HPSG grammar for dialog turns (cf. Section 3) is strictly typed, a good deal of feature structure unifications simply reduce to type unifications which are implemented very efficiently through bit vectors and hash tables (cf. Krieger 1995).

It is worth noting that all rules of the grammar obey a certain locality requirement, meaning that they do not constrain daughters of daughters (neither through coreference requirements, nor through values). This is not only linguistically interesting (and nicely complements the "locality principle" in HPSG), but also allows us to work with local trees of depth 1 all the time, thus accelerating unification and copying massively. Since principle and rules "transport" the relevant information to the turn level, it is in fact legal to cut off the underlying derivational structure if we are only interested to feed semantics evaluation and transfer properly. Our experiments have shown that this strategy together with deterministic chart reconstruction gives us a speedup factor of 4, compared to the strict bottom-up version of the SEM-parser.

Since prosodic information is not always reliable and since it is used in the SEM-parser to rule out certain rule applications, we have extended the SEM-parser by a recovery mechanism, making it possible to reactivate exactly such excluded hypotheses and thus enforcing the reactivation of lost readings.

7 Communication Protocol

The central data structure by which synchronization and communication between the parsers is achieved is that of a completion history, containing a record on how a (sub)tree was completed. Completion histories are described by the following EBNF:

```
<compl-history> ::= {R <rule-id> <edge-id> <start> <end> {E <edge-id>}* |
                     L <lex-id> <edge-id> <start> <end>}+
```

<rule-id>, <lex-id>, <edge-id>, <start>, and <end> are integers. R <rule-id> and L <lex-id> denote rules and lexicon entries. <edge-id> uniquely identifies an edge in the chart. <start> and <end> give the start and end point of a spanning edge.

Let us focus on interesting details of the communication architecture. In a first phase, the lattice parser and the SEM-parser work in parallel. The lattice parser incrementally constructs its chart, always sending the best hypothesis having utterance status. The SEM-parser deterministically reconstructs the trees and reports back failures. These messages are ignored as long as the SEM-parser is still building up trees. Only if the SEM-parser becomes inactive, further, but lower rated hypotheses are sent. Thereby, the idle time in the SEM-parser is utilized to full advantage in order to reconstruct additional trees, perhaps becoming important during the analysis (*speculative evaluation*). That is, if the estimation of an utterance improves over time, its subtrees are in general not accessible to the SEM-parser (since they have never get a high rating). Under speculative evaluation, however, we often find that they have already been constructed, helping us to speed up parsing. Since our grammar is conceived for turns, this situation is not the exception, but in fact the normal case (recall that a turn consists of several segments, whereas each segment might be lifted to the turn level). Hence, this strategy guarantees that the utterance spanned by the trees *monotonically increases* over time.

The second phase is only entered after the probabilistic parser has reached the end of the word lattice. In case that the SEM-parser has accepted one of the previous trees as a valid reading, the lattice parser will be informed about the success.[4] Otherwise, the SEM-parser calls for further hypotheses (i.e., trees). The criteria for the selection of the next-best hypothesis are exactly those in the first phase: "long" hypotheses are prefered and in case of equal length, the one with the best internal score is chosen. I.e., in the second phase, the length of a potential utterance *decreases*. If none of the required trees are accepted, the process stops iff the lattice parser makes no further trees available. Exactly this parameter controls the duration of the second phase.

Depending on the choice which trees are sent, the lattice parser directs the behaviour of the SEM-parser. That is the essential reason why the SEM-parser must not conduct search over the set of received hypotheses. The stepwise retraction of the length of hypotheses guarantees that the longest possible valid utterance will be found. This is especially useful to analyze parts of an utterance in case that no fully spanned reading can be found.

8 Integrating Prosodic Information

Information about utterance boundaries and focus is used in many ways within the SEM-parser, helping to reduce the space of possible rule applications. First of all, prosodic information is directly mapped onto lexical items, i.e., words (actually their feature structures) integrate information about utterance

[4]Valid readings are handled in the SEM-parser through a type definition in that the feature structure of a potential reading must *unify* with the feature specification of that special type. Alternatively, subsequent modules (e.g., dialog act processing) might inform the SEM-parser whether a reading is legal or not.

boundaries and focus.[5] Later, this information is "transported" to the segment level by means of projection principles, encoded as typed feature specifications. Strong intonational phrase (or B3) boundaries are used to determine the prosodic mood of a segment, and so a turn encodes a sequence of moods, not just a single mood. Information about the foci of a turn is especially important for the transfer component in order to obtain a proper translation. Mapping prosodic information onto the proper parsing hypotheses is achieved with the help of the signal time which is used throughout the modules. This general mechanism is explained in Figure 2.

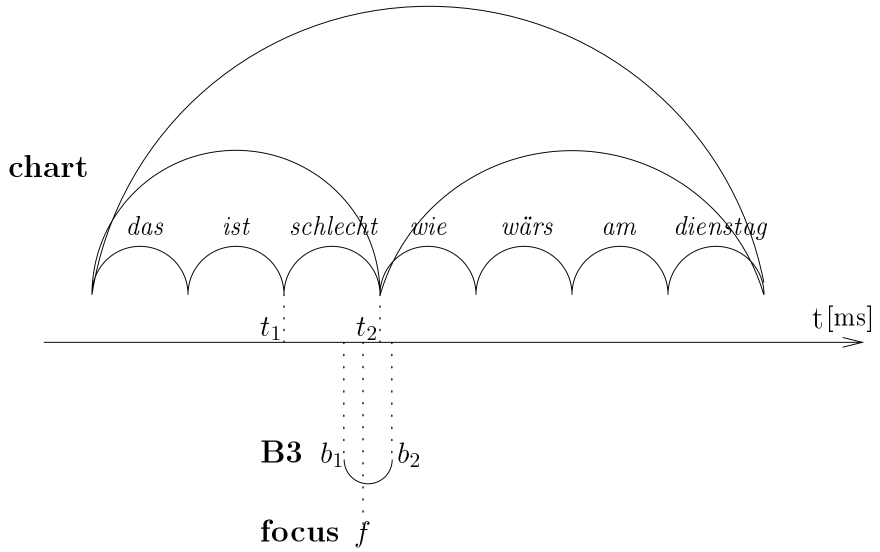

Figure 2: *Mapping B3 and focus information onto chart edges. B3 boundaries are given in terms of time intervals, so we must guarantee that* $t_1 \leq b_1 \leq t_2$. *The focus is specified as a time point, thus we make sure that* $t_1 \leq f \leq t_2$. *Both B3 and focus is associated with a confidence value which has to be above a threshold. In our case,* schlecht *both bears the segment boundary as well as the focus.*

Information about B3 boundaries helps the SEM-parser to rule out certain rule application. This is achieved by telling the parser which rules are *segment-connecting* and which are only *segment-internal*. Clearly, segment-connecting rules enforce a B3 boundary between segments—actually between the last lexical chart edge of a segment and the first lexical edge of the following segment (recall that words encode segment boundaries). The opposite case holds for the segment-internal rules: here, no B3 boundary is allowed to cross a chart edge,

[5]Procedurally speaking, this is achieved by means of unification when lexical edges are entered into the chart.

originating from such a rule application.

Obviously, such constraints heavily reduce the number of possible readings of an utterance. Our experiments have shown that a reduction of 70% is not unusual. Thus, not only does parsing efficiency profit from such constraints (smaller chart, less unification/copying), but also non-valid readings are eliminated here. Furthermore, subsequent components in INTARC (see Figure 1) welcome such a filter mechanism.

9 Summary and Outlook

In this paper, we have proposed a novel approach to time-synchronous word lattice parsing that is based on a lose coupling of two interacting parsing modules, running in parallel. Instead of simply parsing a word lattice, we rather do tree decoding with a probabilistic approximation of a given grammar, employing a beam search strategy. Logical form is build up in tandem according to the decoded derivation histories, using a codescriptive HPSG grammar for dialog turns which incorporates prosodic information to narrow the space of possible readings. Since both parsers use the same grammar, information about locally derived trees can be easily transmitted by means of few identifiers, thus avoiding expensive communication via feature structures. Our approach has been implemented in COMMON LISP and is part of VERBMOBIL's INTARC 2.0 system.

Due to space limitations, we can only indicate further improvements, waiting to be implemented. We mention only two of them. Encoding "prominent" features (say AGR) as complex non-terminals in the CF backbone of the word lattice parser should reduce the search space in both parsers, since certain rule application are then no longer possible.

Depending on the (quality of the) speech input and the settings in the lattice parser, it is often hard to find the right number of transmitted hypotheses: sometimes the SEM-parser gets too many, sometimes it is waiting for input. This problem can be solved by creating several instances of the SEM-parser, represented as parallel working processes. Such a setup should pose no problems because a derived tree is always sent as a single message, containing all depending chart edges (approx. 20–30). Furthermore, the underlying communication software ICE makes multiple write/read operations from *one* channel available, thus each instance of the SEM-parser can decide by its own when to read/write a message. We expect a speedup here nearly linear in the number of instances of SEM-parser. Clearly, if more hypotheses can be processed, we can obtain a higher recognition rate simply by increasing the beam width of the lattice parser.

References

Amtrup, J. W. 1995. ICE—INTARC Communication Environment. Users Guide and Reference Manual. Version 1.4. Technical Document 14, Verbmobil, December.

Diagne, A. K., W. Kasper, and H.-U. Krieger. 1995. Distributed Parsing With HPSG Grammars. In *Proceedings of the 4th International Workshop on Parsing Technologies, IWPT-95*, 79–86. Also available as DFKI Research Report RR-95-19.

Fujisaki, T., F. Jelinek, J. Cocke, E. Black, and T. Nishino. 1991. A Probabilistic Parsing Method for Sentence Disambiguation. In *Current Issues in Parsing Technology*, ed. M. Tomita, 139–148. Norvell, Mass. Kluver Akademic Publishers.

Hanrieder, G. 1995. *Inkrementelles Parsing gesprochener Sprache mit einer linksassoziativen Unifikationsgrammatik.* PhD thesis, Friedrich-Alexander Universität, Erlangen-Nürnberg. In German.

Hauenstein, A., and H. Weber. 1994. An Investigation of Tightly Coupled Time Synchronous Speech Language Interfaces Using a Unification Grammar. In *Proceedings of the Workshop on Integration of Natural Language and Speech Processing at AAAI 94*, ed. P. McKevitt, 42–49. Seattle, August.

Kay, M., J. M. Gawron, and P. Norvig. 1994. *Verbmobil: A Translation System for Face-to-Face Dialog.* CSLI Lecture Notes, Number 33. Stanford: Center for the Study of Language and Information.

Krieger, H.-U. 1995. Classification and Representation of Types in $\mathcal{TDL}$. In *Proceedings of the International KRUSE Symposium, Knowledge Retrieval, Use, and Storage for Efficiency*, 74–85. Also available as DFKI Research Report RR-95-17.

Krieger, H.-U., and U. Schäfer. 1994. $\mathcal{TDL}$—A Type Description Language for Constraint-Based Grammars. In *Proceedings of the 15th International Conference on Computational Linguistics, COLING-94*, 893–899. A enlarged version of this paper is available as DFKI Research Report RR-94-37.

Krieger, H.-U., and U. Schäfer. 1995. Efficient Parameterizable Type Expansion for Typed Feature Formalisms. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI-95*, 1428–1434. Also available as DFKI Research Report RR-95-18.

Magermann, D. M. 1994. *Natural Language Parsing as Statistical Pattern Recognition.* PhD thesis, Stanford University, February.

Ney, H. 1995. *Architecture and Search Strategies for Large-Vocabulary Continuous-Speech Recognition.* 59–84. NATO ASI Series. Berlin: Springer. In: *Speech Recognition and Coding*, A.J. Rubio Ayuso and J.M. Lopez Soler, editors.

Petzold, A. 1995. Strategies for Focal Accent Detection in Spontanous Speech. In *Proceedings of the 13th ICPhS*, 672–675.

Pollard, C., and I. A. Sag. 1987. *Information-Based Syntax and Semantics. Vol. I: Fundamentals.* CSLI Lecture Notes, Number 13. Stanford: Center for the Study of Language and Information.

Pollard, C., and I. A. Sag. 1994. *Head-Driven Phrase Structure Grammar.* Studies in Contemporary Linguistics. Chicago: University of Chicago Press.

Strom, V. 1995. Detection of Accents, Phrase Boundaries and Sentence Modality in German with Prosodic Features. In *Eurospeech 95*, 2039–2041.

Wahlster, W. 1993. VERBMOBIL—Translation of Face-to-Face Dialogs. Research Report RR-93-34, German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany. Also in Proc. MT Summit IV, 127–135, Kobe, Japan, July 1993.

Weber, H. 1994. Time-Synchronous Chart Parsing of Speech Integrating Unification Grammars with Statistics. *Proceedings of Twente Workshop on Speech and Language Engeneering* 107–120.

Weber, H. 1995. *LR-inkrementelles, probabilistisches Chartparsing von Worthypothesenmengen mit Unifikationsgrammatiken: Eine enge Kopplung von Suche und Analyse.* PhD thesis, Universität Hamburg, Department of Computer Science. In German.