

RECENT ADVANCES IN JANUS: A SPEECH TRANSLATION SYSTEM

*M.Woszczyna, N.Coccaro, A.Eisele, A.Lavie, A.McNair, T.Polzin, I.Rogina,
C.P.Rose, T.Sloboda, M.Tomita, J.Tsutsumi, N.Aoki-Waibel, A.Waibel, W. Ward*

Carnegie Mellon University
University of Karlsruhe

ABSTRACT

We present recent advances from our efforts in increasing coverage, robustness, generality and speed of JANUS, CMU's speech-to-speech translation system. JANUS is a speaker-independent system translating spoken utterances in English and also in German into one of German, English or Japanese. The system has been designed around the task of conference registration (CR). It has initially been built based on a speech database of 12 read dialogs, encompassing a vocabulary of around 500 words. We have since been expanding the system along several dimensions to improve speed, robustness and coverage and to move toward spontaneous input.

1. INTRODUCTION

In this paper we describe recent improvements of JANUS, a speech to speech translation system. Improvements have been made mainly along the following dimensions: 1.) better context-dependent modeling improves performance in the speech recognition module, 2.) improved language models, smoothing, and word equivalence classes improve coverage and robustness of the sentence that the system accepts, 3.) an improved N-best search reduces run-time from several minutes to now real time, 4.) trigram and parser rescoring improves selection of suitable hypotheses from the N-best list for subsequent translation. On the machine translation side, 5.) a cleaner interlingua was designed and syntactic and domain-specific analysis were separated for greater reusability of components and greater quality of translation, 6.) a semantic parser was developed to achieve semantic analysis, should more careful analysis fail.

The JANUS [1, 2] framework as it is presented here also allows us to experiment with components of a speech translation system, in an effort to achieve both robustness and high-quality translation. In the following we describe these efforts and system components that have been developed to date. At present, JANUS consists conceptually out of three major components: speech recognition, machine translation and speech synthesis. Since we have not made any significant attempts at improving performance on the synthesis end (DEC-talk and synthesizers produced by NEC and AEG-Daimler are used for English, Japanese and German output, respectively), our discussion will focus on the recognition and translation parts.

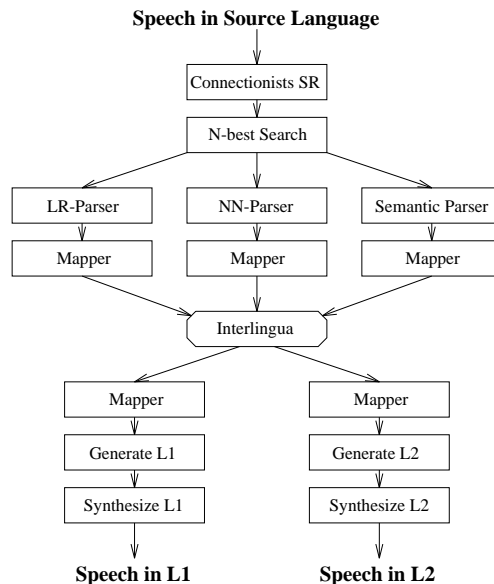


Figure 1: Overview of the System

2. RECOGNITION ENGINE

Our recognition engine uses several techniques to optimize the overall system performance. Speech input is preprocessed into time frames of spectral coefficients. Acoustic models are trained to give a score for each phoneme, representing the phoneme probability at the given frame. These scores are used by an N-best search algorithm to produce a list of sentence hypotheses. Based on this list, more computationally expensive language models are then applied to achieve further improvement of recognition accuracy.

2.1. Acoustic modeling

For acoustic modeling, several alternative algorithms are being evaluated including TDNN, MS-TDNN, MLP and LVQ [6, 5]. In the main JANUS system, an LVQ algorithm with context-dependent phonemes is now used for speaker independent recognition. For each phoneme, there is a context independent set of prototypical vectors. The output scores for each phoneme segment are computed from the euclidian distance using context dependent segment weights.

Error rates using context dependent phonemes are lower by a factor 2 to 3 for English (1.5 to 2 for German) than using context independent phonemes. Results are shown in table 1.

language model	English		German	
	PP	WA	PP	WA
none	400.0	58.2	425.0	63.0
word-pairs	28.9	83.4	20.8	89.1
bigrams	16.2	92.6	18.3	93.7
smoothed bigrams	18.1	91.5	28.90	84.7
after resorting	—	98.8		

Table 1: Word Accuracy for First Hypothesis

The performance on the RM-task at comparable perplexities is significantly better than for the CR-task, suggesting that the CR-task is somewhat more difficult.

2.2. Search

The search module of the recognizer builds a sorted list of sentence hypotheses. Speed and memory requirements could be dramatically improved: Though the amount of hypotheses computed for each utterance was increased from 6 to 100 hypotheses, the time required for their computation could be reduced from typically 3 minutes to 3 seconds.

This was achieved by implementing the word dependent N-best algorithm[3] as backward pass in the forward backward algorithm[4]: First a fast firstbest only search is performed, saving the scores at each possible word ending. In a second pass, this information is used for aggressive pruning to reduce the search effort for the N-best search. Further speedup was achieved by dynamically adapting the beam width to keep number of active states constant, and by carefully avoiding the evaluation of states in large inactive regions of words. Important for total system performance is the fact that the firstbest hypothesis can already be analyzed by the MT modules while the N-best list is computed.

All language models (word-pairs, bigrams or smoothed bigrams, and trigrams for resorting) are now trained on more than 1000 CR-sentences, using word class specific equivalence classes (digits, names, towns, languages etc.)

2.3. Resorting

The resulting N-best list is resorted using trigrams to further improve results. Resorting improves the word accuracy for the best scoring hypothesis (created using smoothed bigrams) from 91.5% to 98%, and the average rank of the correct hypothesis within the list from 5.7 to 1.1;

Much longer N-best lists have been used for experiments (500-1000). However it is very unlikely that a rescoring algorithm

moves a hypothesis from the very bottom of such a long list to the 1st position. For practical application, a number of 100 hypotheses was found to be best.

3. THE MACHINE TRANSLATION (MT) ENGINE

The MT-component that we have previously used has now been replaced by a new module that can run several alternate processing strategies in parallel. To translate spoken language from one language to another, the analysis of spoken sentences, that suffer from ill-formed input and recognition errors is most certainly the hardest part. Based on the list of N-best hypotheses delivered by the recognition engine, we can now attempt to select and analyze the most plausible sentence hypothesis in view of producing an accurate and meaningful translation. Two goals are central in this attempt: *high fidelity* and *accurate translation* wherever possible, and *robustness* or *graceful degradation*, should attempts for high fidelity translation fail in face of ill-formed or misrecognized input. At present, three parallel modules attempt to address these goals: 1) an LR-parser based syntactic approach, 2) a semantic pattern based approach and 3) a connectionist approach. The most useful analysis from these modules is mapped onto a common Interlingua, a language independent, but domain-specific representation of meaning. The analysis stage attempts to derive a high precision analysis first, using a strict syntax and domain specific semantics. Connectionist and/or semantic parsers are currently applied as back-up, if the higher precision analysis fails. The Interlingua ensures that alternate modules can be applied in a modular fashion and that different output languages can be added without redesign of the analysis stage.

3.1. Generalized LR Parser

The first step of the translation process is syntactic parsing with the Generalized LR Parser/Compiler [16]. The Generalized LR parsing algorithm is an extension of LR parsing with the special device called "Graph-Structured Stack" [14], and it can handle arbitrary context-free grammars while most of the LR efficiency is preserved. A grammar with about 455 rules for general colloquial English is written in a Pseudo Unification formalism [15], that is similar to Unification Grammar and LFG formalisms. Figure 2 shows the result of syntactic parsing of the sentence "Hello is this the conference office".

Robust GLR Parsing: Modifications have been made to make the Generalized LR Parser more robust against ill-formed input sentences [18]. In case the standard parsing procedure fails to parse an input sentence, the parser non-deterministically skips some word(s) in the sentence, and returns the parse with fewest skipped words. In this mode, the parser will return some parse(s) with any input sentence, unless no part of the sentence can be recognized at all.

```

(HELLO IS THIS THE CONFERENCE OFFICE $)
;+++ GLR Parser running to produce English structure +++++
(1) ambiguities found and took 1.164878 seconds of real time
(((PREV-SENTENCES ((COUNTER 1) (MOOD *OPENING)
                    (ROOT *HELLO)))
  (COUNTER 2)
  (MOOD *INTERROGATIVE)
  (SUBJECT ((AGR *3-SING) (ROOT *THIS)
            (CASE (*OR* *NOM *OBL))))
  (FORM *FINITE)
  (PREDICATE
   ((DET ((ROOT *THE) (DEF *DEF))) (AGR *3-SING)
         (ANIM *-)
         (A-AN *A)
         (ROOT *CONFERENCE-OFFICE)))
  (AGR *3-SING)
  (SUBCAT *SUBJ-PRED)
  (ROOT *COPULA)
  (TENSE *PRESENT)))

```

Figure 2: Example F-Structure

In the example in figure 3, the input sentence "Hello is this is this the office for the AI conference which will be held soon" is parsed as "Hello is this the office for the conference" by skipping 8 words. Because the analysis grammar or the interlingua does not handle the relative clause "which will be held soon", 8 is the fewest possible words to skip to obtain a grammatical sentence which can be represented in the interlingua. In the Generalized LR parsing, an extra procedure is applied every time a word is shifted onto the Graph Structured Stack. A heuristic similar to beam search makes the algorithm computationally tractable.

When the standard GLR parser fails on all of the 20 best sentence candidates, this robust GLR parser is applied to the best sentence candidate.

3.2. The Interlingua

This result, called "syntactic f-structure", is then fed into a mapper to produce an Interlingua representation. For the mapper, we use a software tool called Transformation Kit [17]. A mapping grammar with about 300 rules is written for the Conference Registration domain of English.

Figure 4 is an example of Interlingua representation produced from the sentence "Hello is this the conference office". In the example, "Hello" is represented as speech-act *ACKNOWLEDGEMENT, and the rest as speech-act *IDENTIFY-OTHER.

```

Input sentence :
(hello is this is this the AI conference office which will be held soon $)

Parse of input sentence :
(HELLO IS THIS THE CONFERENCE OFFICE $)

Words skipped : ((IS 2) (THIS 3) (AI 7) (WHICH 10) (WILL 11)
                (BE 12) (HELD 13) (SOON 14))

```

Figure 3: Example for robust parsing

```

((PREV-UTTERANCES ((SPEECH-ACT *ACKNOWLEDGEMENT) (VALUE *HELLO)))
 (TIME *PRESENT)
 (PARTY
  ((DEFINITE +) (NUMBER *SG)
   (ANIM -)
   (TYPE *CONFERENCE)
   (CONCEPT *OFFICE)))
 (SPEECH-ACT *IDENTIFY-OTHER))

```

Figure 4: Example: Interlingua Output

The JANUS interlingua is tailored to dialog translation. Each utterance is represented as one or more speech acts. A speech act can be thought of as what effect the speaker is intending a particular utterance to have on the listener. Our interlingua currently has eleven speech acts such as request direction, inform, and command. For purposes of this task, each sentence utterance corresponds to exactly one speech act. So the first task in the mapping process is to match each sentence with its corresponding speech act. In the current system, this is done on a sentence by sentence basis. Rules in the mapping grammar look for cues in the syntactic f-structure such as mood, combinations of auxiliary verbs, and person of the subject and object where it applies. In the future we plan to use more information from context in determining which speech act to assign to each sentence.

Once the speech act is determined, the rule for a particular speech act is fired. Each speech act has a top level semantic slot where the semantic representation for a particular instance of the speech act is stored during translation. This semantic structure is represented as a hierarchical concept list which resembles the argument structure of the sentence. Each speech act rule contains information about where in the syntactic structure to look for constituents to fill thematic roles such as agent, recipient, and patient in the semantic structure. Specific lexical rules map nouns and verbs onto concepts. In addition to the top level semantic slot, there are slots where information about tone and mood are stored. Each speech act rule contains information about what to look for in the syntactic structure in order to know how to fill this slot. For instance the auxiliary verb which is used in a command determines how imperative the command is. For example, 'You must register for the conference within a week' is much more imperative than 'You should register for the conference within a week'. The second example leaves some room for negotiation where the first does not.

3.3. The Generator

The generation of target language from an Interlingua representation involves two steps. Figure 5 shows sample traces of German and Japanese, from the Interlingua in figure 4.

First, with the same Transformation Kit used in the analysis phase, Interlingua representation is mapped into syntactic f-

structure of the target language.

```

;+ TransKit rules being applied to produce G structure ++
((PREV-SENTENCES ((VALUE HALLO) (ROOT LITERAL))))
  (ROOT SEIN) (CAT V) (PERSON 3)
  (SUBJECT
    ((CAT N) (CAS N) (DIST +) (LOC +) (PERSON 3)
     (NUMBER SG) (ROOT D-PRONOUN)))
  (NUMBER SG) (FORM FIN) (MOD IND) (TENSE PRES)
  (MOOD INTERROG)
  (PRED
    ((DET ((CAS N) (GENDER NEU)
           (NUMBER SG)
           (CAT DET)
           (ROOT DER)))
     (CLASS SW) (NUMBER SG) (PERSON 3) (CAT N)
     (COMPOUND
      ((CAT N) (PL-CLASS PL3)
       (SG-CLASS SG0)
       (GENDER FEM)
       (ROOT KONFERENZ))))
     (ROOT SEKRETARIAT) (PL-CLASS PL5) (SG-CLASS SG3)
     (GENDER NEU) (CAS N) (ANIM -))))

;+ GenKit rules being applied to produce German text ++
"HALLO , IST DORT DAS KONFERENZSEKRETARIAT ?"

;+ TransKit rules being applied to produce J structure ++
((PREV-UTTERANCES
  ((FOR-REMOVE-DESU *IDENTIFY-OTHER) (VALUE MOSHIMOSHI)
   (ROOT *LITERAL)))
 (VTYPE MEISHI)
 (SUFF (*MULTIPLE* KA DESU))
 (PRED ((ROOT GAKKAIJIMUKYOKU) (CAT N)
       (DEFINITE +)
       (NUMBER *SG)
       (ANIM -)))
 (ROOT COPULA))

;+ GenKit rules being applied to produce Japanese text ++
"MOSHIMOSHI GAKKAI JIMUKYOKU DESUKA"

```

Figure 5: Output language F-structure

There are about 300 rules in the generation mapping grammar for German, and 230 rules for Japanese. The f-structure is then fed into sentence generation software called "GENKIT" [17] to produce a sentence in the target language. A grammar for GENKIT is written in the same formalism as the Generalized LR Parser: phrase structure rules augmented with pseudo unification equations. The GENKIT grammar for general colloquial German has about 90 rules, and Japanese about 60 rules. Software called MORPHE is also used for morphological generation for German.

3.4. Semantic Pattern Based Parsing

A human-human translation task is even harder than human-machine communication, in that the dialog structure in human-human communication is more complicated and the range of topics is usually less restricted. These factors point to the requirement for robust strategies in speech translation systems.

Our robust semantic parser combines frame based semantics with semantic phrase grammars. We use a frame based parser similar to the DYPAR parser used by Carbonell, et al. to process ill-formed text,[9] and the MINDS system previously developed at CMU.[10] Semantic information is represented in a set of frames. Each frame contains a set of slots representing pieces of information. In order to fill the slots in the frames,

we use semantic fragment grammars. Each slot type is represented by a separate Recursive Transition Network, which specifies all ways of saying the meaning represented by the slot. The grammar is a semantic grammar, non-terminals are semantic concepts instead of parts of speech. The grammar is also written so that information carrying fragments (semantic fragments) can stand alone (be recognized by a net) as well as being embedded in a sentence. Fragments which do not form a grammatical English sentence are still parsed by the system. Here there is not one large network representing all sentence level patterns, but many small nets representing information carrying chunks. Networks can "call" other networks, thereby significantly reducing the overall size of the system. These networks are used to perform pattern matches against input word strings. This general approach has been described in earlier papers. [7, 8]

The operation of the parser can be viewed as "phrase spotting". A beam of possible interpretations are pursued simultaneously. An interpretation is a frame with some of its slots filled. The RTNs perform pattern matches against the input string. When a phrase is recognized, it attempts to extend all current interpretations. That is, it is assigned to slots in active interpretations that it can fill. Phrases assigned to slots in the same interpretation are not allowed to overlap. In case of overlap, multiple interpretations are produced. When two interpretations for the same frame end with the same phrase, the lower scoring one is pruned. This amounts to dynamic programming on series of phrases. The score for an interpretation is the number of input words that it accounts for. At the end of the utterance, the best scoring interpretation is picked.

Our strategy is to apply grammatical constraints at the phrase level and to associate phrases in frames. Phrases represent word strings that can fill slots in frames. The slots represent information which, taken together, the frame is able to act on. We also use semantic rather than lexical grammars. Semantics provide more constraint than parts of speech and must ultimately be dealt with in order to take actions. We believe that this approach offers a good compromise of constraint and robustness for the phenomena of spontaneous speech. Restarts and repeats are most often between phrases, so individual phrases can still be recognized correctly. Poorly constructed grammar often consists of well-formed phrases, and is often semantically well-formed. It is only syntactically incorrect.

The parsing grammar was designed so that each frame has exactly one corresponding speech act. Each top level slot corresponds to some thematic role or other major semantic concept such as action. Subnets correspond to more specific semantic classes of constituents. In this way, the interpretation returned by the parser can be easily mapped onto the interlingua and missing information can be filled by meaningful default values with minimal effort.

Once an utterance is parsed in this way, it must then be mapped onto the interlingua discussed earlier in this paper. The mapping grammar contains rules for each slot and subnet in the parsing grammar which correspond to either concepts or speech acts in the interlingua. These rules specify the relationship between a subnet and the subnets it calls which will be represented in the interlingua structure it will produce. Each rule potentially contains four parts. It need not contain all of them. The first part contains a default interlingua structure for the concept represented by a particular rule. If all else fails, this default representation will be returned. The next part contains a skeletal interlingua representation for that rule. This is used in cases where a net calls multiple subnets which fill particular slots within the structure corresponding to the rule. A third part is used if the slot is filled by a terminal string of words. This part of the rule contains a context which can be placed around that string of words so that it can be attempted to be parsed and mapped by the LR system. It also contains information about where in the structure returned from the LR system to find the constituent corresponding to this rule. The final part contains rules for where in the skeletal structure to place interlingua structures returned from the subnets called by this net.

3.5. Connectionist Parsing

The connectionist parsing system PARSEC [12] is used as a fall-back module if the symbolic high precision one fails to analyze the input. The important aspect of the PARSEC system is that it learns to parse sentences from a corpus of training examples. A connectionist approach to parse spontaneous speech offers the following advantages:

1. Because PARSEC learns and generalizes from the examples given in the training set no explicit grammar rules have to be specified by hand. In particular, this is of importance when the system has to cope with spontaneous utterances which frequently are “corrupted” with disfluencies, restarts, repairs or ungrammatical constructions. To specify symbolic grammars capturing these phenomena has been proven to be very difficult. On the other side there is a “build-in” robustness against these phenomena in a connectionist system.
2. The connectionist parsing process is able to combine symbolic information (e.g. syntactic features of words) with non-symbolic information (e.g. statistical likelihood of sentence types). Moreover, the system can easily integrate different knowledge sources. For example, instead of just training on the symbolic input string we trained PARSEC on both the symbolic input string and the pitch contour. After training was completed the system was able to use the additional information to determine the sentence mood in cases where syntactic clues were not sufficient. We think of extending the idea of

integrating prosodic information into the parsing process in order to increase the performance of the system when it is confronted with corrupted input. We hope that prosodic information will help to indicate restarts and repairs.

The current PARSEC system comprises six hierarchically ordered (back-propagation) connectionist modules. Each module is responsible for a specific task. For example, there are two modules which determine phrase and clause boundaries. Other modules are responsible for assigning to phrases or clauses labels which indicate their function and/or relationship to other constituents. The top module determines the mood of the sentence.

Recent Extensions: We applied a slightly modified PARSEC system to the domain of air travel information (ATIS). We could show that the system was able to analyze utterance like “show me flights from boston to denver on us air” and that the system’s output representation could be mapped to a Semantic Query Language (SQL). In order to do this we included semantic information (represented as binary features) in the lexicon. By doing the same for the CR-task we hope to increase the overall parsing performance.

We have also changed PARSEC to handle syntactic structures of arbitrary depth (both left and right branching) [13].

the main idea of the modified PARSEC system is to make it auto recursive, i.e. in a recursion step n it will take its output of the previous step $n-1$ as its input. This offers the following advantages:

1. **Increased Expressive Power:** The enhanced expressive power allows a much more natural mapping of linguistic intuitions to the specification of the training set.
2. **Ease of learning:** Learning difficulties can be reduced. Because PARSEC is now allowed to make more abstraction steps each individual step can be smaller and, hence, is easier to learn.
3. **Compatibility:** Because PARSEC is now capable of producing arbitrary tree structures as its output it can be more easily used as a submodule in NLP-systems (e.g. the JANUS system). For example, it is conceivable to produce as the parsing output f-structures which then can be mapped directly to the generation component [11].

4. SYSTEM INTEGRATION

The system accepts continuous speech speaker-independently in either input language, and produces synthetic speech output in near real-time. Our system can be linked to different language versions of the system or corresponding partner systems

via ethernet or via telephone modem lines. This possibility has recently been tested between sites in the US, Japan and Germany to illustrate the possibility of international telephone speech translation.

The minimal equipment for this system is a Gradient Desklab 14 A/D-converter, an HP 9000/730 (64 Meg RAM) workstation for each input language, and a DECTalk speech synthesizer.

Included in the processing are A/D conversion, signal processing, continuous speech recognition, language analysis and parsing (both syntactic and semantic) into a language independent interlingua, text generation from that interlingua, and speech synthesis.

The amount of time needed for the processing of an utterance, depends on its length and acoustic quality, but also on the perplexity of the language model, on whether or not the first hypothesis is parsable and on the grammatical complexity and ambiguity of the sentence. While it can take the parser several seconds to process a long list of hypotheses for a complex utterance with many relative clauses (extremely rare in spoken language), the time consumed for parsing is usually negligible (0.1 second).

For our current system, we have eliminated considerable amounts of communication delays by introducing socket communication between pipelined parts of the system. Thus the search can start before the preprocessing program is done, and the parser starts working on the first hypothesis while the N-best list is computed.

5. CONCLUSION

In this paper, we have discussed recent extensions to the JANUS system a speaker independent multi-lingual speech-to-speech translation system under development at Carnegie Mellon and Karlsruhe University. The components include an speech recognition using an N-best sentence search, to derive alternate hypotheses for later processing during the translation. The MT component attempts to produce a high-accuracy translation using precise syntactic and semantic analysis. Should this analysis fail due to ill-formed input or misrecognitions, a connectionist parser, PARSEC, and a semantic parser produce alternative minimalist analyses, to at least establish the basic meaning of an input utterance. Human-to-human dialogs appear to generate a larger and more varied breadth of expression than human-machine dialogs. Further research is in progress to quantify this observation and to increase robustness and coverage of the system in this environment.

References

1. A. Waibel, A. Jain, A. McNair, H. Saito, A. Hauptmann, and J. Tebelskis, *JANUS: A Speech-to-Speech Translation System Using Connectionist and Symbolic Processing Strategies*, volume 2, pp 793–796. ICASSP 1991.
2. L. Osterholtz, A. McNair, I. Rogina, H. Saito, T. Sloboda, J. Tebelskis, A. Waibel, and M. Woszczyna. *Testing Generality in JANUS: A Multi-Lingual Speech to Speech Translation System*, volume 1, pp 209–212. ICASSP 1992.
3. Austin S., Schwartz R. *A Comparison of Several Approximate Algorithms for Finding N-best Hypotheses*, ICASSP 1991, volume 1, pp 701–704.
4. Schwartz R., Austin S. *The Forward-Backward Search Algorithm*, ICASSP 1990, volume 1, pp 81–84.
5. O. Schmidbauer and J. Tebelskis. *An LVQ based Reference Model for Speaker-Adaptive Speech Recognition*. ICASSP 1992, volume 1, pages 441–444.
6. J. Tebelskis and A. Waibel. *Performance through consistency: MS-TDNNs for large vocabulary continuous speech recognition*, Advances in Neural Information Processing Systems, Morgan Kaufmann.
7. W. Ward, *Understanding Spontaneous Speech*, DARPA Speech and Natural Language Workshop 1989, pp 137–141.
8. W. Ward, *The CMU Air Travel Information Service: Understanding Spontaneous Speech*, DARPA Speech and Natural Language Workshop 1990.
9. J.G. Carbonell and P.J. Hayes, *Recovery Strategies for Parsing Extragrammatical Language*, Carnegie-Mellon University Computer Science Technical Report 1984, (CMU-CS-84-107)
10. S.R. Young, A.G. Hauptmann, W.H. Ward, E.T. Smith, and P. Werner, *High Level Knowledge Sources in Usable Speech Recognition Systems*, in Communications of the ACM 1989, Volume 32, Number 2, pp 183–194
11. F. D. Buø, *A learnable connectionist parser that outputs f-structures (working title)*, PhD-Thesis proposal, University of Karlsruhe, in preparation.
12. A.J. Jain, A. Waibel, D. Touretzky, *PARSEC: A Structured Connectionist Parsing System for Spoken Language*, ICASSP 1992, volume 1, pp 205–208.
13. T.S. Polzin, *Pronoun Resolution. Interaction of Syntactic and Semantic Information in Connectionist Parsing*, Master Thesis, Carnegie Mellon University, Department of Philosophy, Computational Linguistics, in preparation.
14. Tomita, M. (ed.), *Generalized LR Parsing*, Kluwer Academic Publishers, Boston MA, 1991.
15. Tomita, M., *The Generalized LR Parser/Compiler* in 13th International Conference on Computational Linguistics (COLING90), Helsinki, 1990
16. Tomita, M., Mitamura, T., Musha, H. and Kee, M.; *The Generalized LR Parser/Compiler Version 8.1: User's Guide*, Technical Memo, Center for Machine Translation, Carnegie Mellon University, CMU-CMT-88-MEMO, 1988.
17. Tomita, M. and Nyberg, E.; *The Generation Kit and The Transformation Kit: User's Guide* Technical Memo, Center for Machine Translation, Carnegie Mellon University, CMU-CMT-88-MEMO, 1988
18. Lavie, A and Tomita, M.; *An Efficient Word-Skipping Parsing Algorithm for Context-Free Grammars* submitted to 3rd International Workshop on Parsing Technologies (IWPT93) Belgium, 1993.