

Incremental Speech Translation: A Layered Chart Approach

Jan W. Amtrup

Computing Research Lab
New Mexico State University, Las Cruces, NM
jamtrup@crl.nmsu.edu
<http://crl.nmsu.edu/Lab/Personnel/jamtrup.html>

Abstract. Human speech understanding works incrementally. We begin to process acoustic input before the speaker's utterance has ended. A system capable of performing sophisticated communication in a natural dialogue or simultaneous interpreting, has to work incrementally, too. The architecture of such a system should be modular, uniform and integrated. We present an architectural framework that suits these three requirements by implementing layered charts, a multi-purpose data structure intended to represent several competing hypotheses about linguistic content of utterance intervals based on hypergraphs. We demonstrate the feasibility by presenting results from an actual interpreting system.

1 Introduction

Human natural language comprehension and production is inherently incremental in nature. Incrementality means to begin the processing of parts of the input (or even to generate output) before the input is complete. We do this by understanding spoken words while or even before they are being uttered. This mode of operation enables us to follow an almost continuous stream of speech signals. Simultaneous conference interpreters take a step further and even generate the content of what they understood incrementally in another language [12, 15]. Psycholinguistic research isolated many processes and features that demonstrate the incremental operation. The applicability of the concept ranges from the early stages of speech recognition, e.g. described by the cohort model [25, 24], to context influences on word recognition [31] and syntactic analysis [27].

The application of incremental principles within systems designated to process natural human speech seems to be appropriate in this light. Only if speech understanding is performed incrementally, one can expect performance similar to human speech comprehension, like dialog systems which interrupt the user, or simultaneous translators. But even if this approach is not taken to mimic the human model, incrementality offers significant advantages compared to non-incremental operation. First of all, it enables the introduction of inter-modular parallelism into a speech understanding application without the need of two independent components operating on the same data. Second, modules may influence the operation of other modules working on the same interval of the input

by exploiting top-down interactions. Third, a system may already start to analyze input even if the speaker still continues to utter words.

Thus, incrementality is a natural and useful paradigm for natural language processing systems which has been mostly explored punctually so far [14, 17]. It is highly convenient to constitute an architecture for NLP systems which reflects the properties of incremental processing and which minimizes redundancy to reduce the negative effects of incrementality. The sources of those negative effects are twofold: First, the amount of data to be processed increases. This is due to the fact that a system never knows if a partial hypothesis can be extended into the future because it does only know the left context. Second, the structure of search spaces is much less strict compared to the non-incremental case. Any ranking must be done locally and thus is suboptimal from a global point of view.

We are going to present an approach to architecture that is modular, uniform and integrates information of all modules in a convenient way. *Layered Charts* are used to represent partial hypotheses throughout an application. They are centered around the assumption that every partial result describes some interval of the input to the system. The content of the description may vary from hypotheses about what words were spoken during a specific interval in time to hypotheses about what should be the translation of a part of the input utterance. The representation schema captures the differences of several types of linguistic knowledge by allowing any kind of feature structure description while simultaneously retaining the common ground of results, namely time.

Two successful existing incremental systems are TDMT [22], which takes an example-based approach to translation on large scale parallel machines and INTARC [16], which at least partly uses a chart-based method for the analysis phase. The transfer and generation modules, however, deviate from this schema by being oriented at dialogue act transfer and schema-based generation [19]. One recent attempt to design an architectural framework for NLP systems, white-boards [8], is able to emulate incremental operation. But, since the control schema for white-boards is centralized, a parallel, distributed system can not easily be built.

2 Layered Charts based on Hypergraphs

Every hypothesis being processed in a speech understanding system is strongly connected with the underlying input: It describes some property of an interval of the speech signal. The range of different types of descriptions is broad. There are hypotheses about which word was actually spoken during some time, what kind of syntactic structure has to be assigned to a sequence of words presumably spoken, what semantic content is included, etc. We assume, however, that the temporal extension within the input speech signal represents the common ground for all information.

The lowest level of representation we take into account are *word graphs* [7]. These graphs are able to represent a huge number of utterance hypotheses in a very compact manner. For example, the graph in Fig. 1 is built out of only 461 edges, but contains $1.2 \cdot 10^{23}$ paths. This compactness and the number of potential candidates are highly advantageous and yet problematic. On one hand, the probability of the correct utterance hypothesis being part of the graph rises with the number of paths, but on the other hand, the amount of input data to linguistic processing reduces performance drastically.

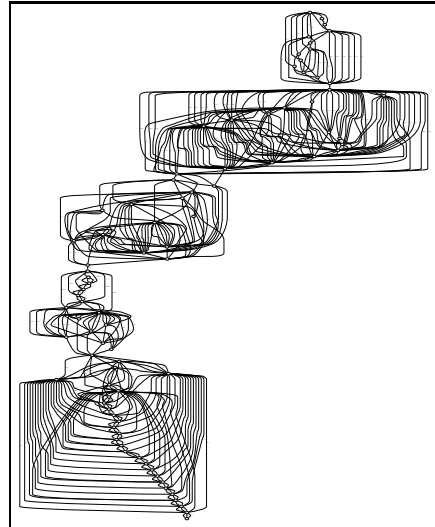


Fig. 1. A word graph

One simple, yet efficient, approach to increase efficiency is to make use of the fact that word graphs usually contain bundles of similar word hypotheses, bearing the same label, but having slightly different start and end times. The processing burden can be greatly reduced if one does not treat these word edges independently, but groups *families of edges* together as hyperedges in a hypergraph [6]. Now, edges do not connect two distinct vertices (points in time), but rather two sets of vertices.

Word graphs and their generalizations to hypergraphs are instances of a chart-like structure [20] or a generalized chart with hyperedges. Charts are directed, acyclic graphs that are used to store partial and completed results of some linguistic processing. The origins can be found in the domain of parsing [20], where charts are used extensively, and in many systems. Additionally, charts have been proposed as central data structure for generation [21] and transfer in machine translation [5]. Usually, edges of a chart carry data related to the specific task at hand, be that structural information used for parsing, semantic content for transfer or generation information.

Layered charts offer a method to separate information of different origin. Starting from the hyperedges representing word hypotheses, each component in a distributed system may add knowledge to the current state of processing by adding edges containing information relevant for that component (cf. Fig. 2). Depending on criteria defined individually for each module, edges are considered useful for other components. In that case, they are transmitted to components which can utilize them. Thus, using a layered chart, a distributed system can be constructed in an integrated fashion. The amount of data each component has to store individually is minimized, yet at the same time every bit of data a component may need is presented to it.

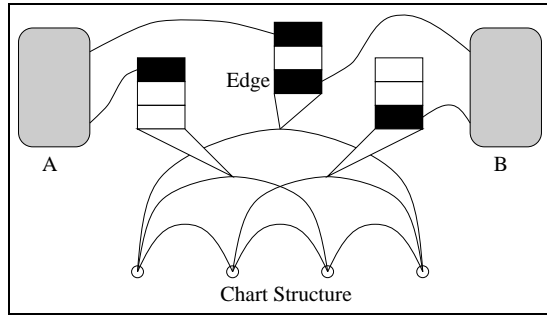


Fig. 2. The principal layout of layered charts

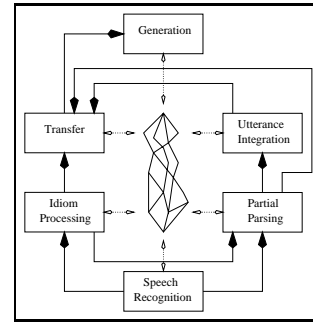


Fig. 3. The architecture of the prototypical interpreting system

The uniformity of a layered chart system is guaranteed by using a unique formalism throughout. We developed a linguistic description language capable of representing well-typed feature structures with appropriateness [10]. Since we assume that a large proportion of the feature structures in a system has to be transmitted to some other component during its lifetime, we implemented feature structures using an automaton-like approach [29]. All references are local to a feature structure, leading to a memory-position independent representation. That way, the transmission does not require linearization in the source component and reconstruction in the target component, but a feature structure can be directly sent as stream of bytes by retaining its semantics [4].

Using an integrated, uniform representation enables easy exchange of data between components. But layered charts are more than a measure for information reuse within natural language processing systems. They provide a direct way to view the union of all edges as the current state of processing. Naturally, there is no global state of a distributed system, but given the edges present in all components one can always get a notion of the progress of each component. This progress can be visualized with an additional component performing no linguistic task, but only user interface functions. Figure 4 shows a screen-dump of some results using the system described here. Since each hypothesis covers a certain interval in time, the relation between edges is always evident. This orientation at a common scale is an important advantage for incremental architectures. It allows the easy introduction of feedback loops which can be used to let a component influence the behavior of another one. For example, the search space of a component may be restricted due to work done by the component receiving the results; it is even possible to influence the order in which search spaces are explored. This concerns the crucial speech-language interface [17] as well as interactions which possibly result from higher-level knowledge [13].

The micro structure of layered charts is given by the individual word hypotheses, as already mentioned. Those are assigned a score which measures the acoustic correspondence between the model for a word and the incoming speech

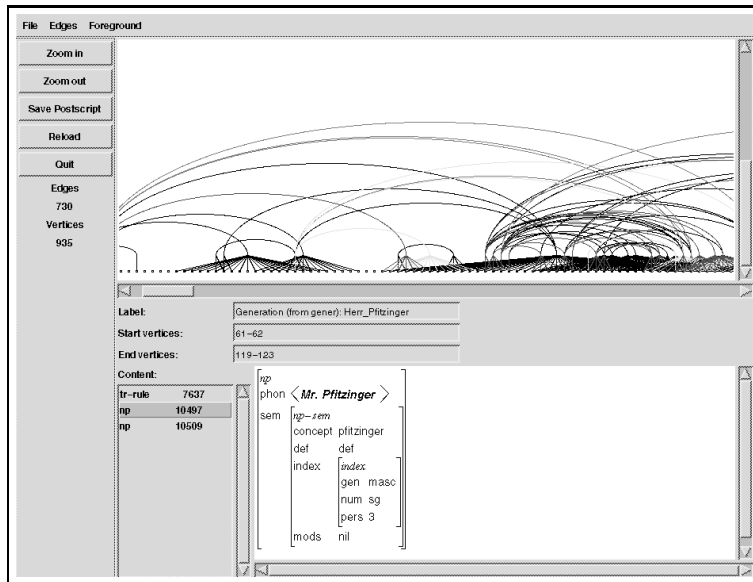


Fig. 4. A layered chart with some hyperedges

signal. The design of layered charts opens the possibility to introduce additional scores into the application. This begins with language model scores and can be possibly extended to the probability of syntactic rule applications, the preferences for specific translations, or the specificity of generation rules.

Furthermore, work done by different modules on neighboring paths within the application can be combined. This feature can be used to establish a selection function to choose between several results computed by modules using different approaches, e.g. to choose between the results of a deep analysis and some shallow understanding [9]. But even more complicated schemata can be implemented. It is possible to modify scores of certain edges based on the evidence available in one component. At present, this is used to prevent a parser from searching for compositional interpretations of idiomatic expressions (see below).

From a software-engineering point of view, layered charts form an architectural framework for natural language processing applications. Distributed systems comprising of several specialized components can easily be built, since they all share the same data structures. Communication between modules takes place, if needed, using a message-passing paradigm [3] which lets the system yield overall system performance in a cooperative way.

3 Architecture

In this section, we will present the architecture and components of an incremental system designed to translate spontaneous conversational speech. It is

centered around the notion of layered charts and was used to demonstrate the feasibility of our approach and to evaluate it. The global architecture of the system is shown in Fig. 3 in section 2. In the center of the figure, the graph-like data structure of the layered chart symbolizes its use in all components. Boxes indicate the individual modules performing linguistic analysis, while arrows represent the directed data flow within the application. The MILC system (*Machine Interpreting with Layered Charts*) translates spontaneously spoken utterances from dialogs in the Verbmobil domain of appointment scheduling from German into English.

The first component is a HMM-based speech recognizer [18], which produces word graphs incrementally, i.e., they contain dead ends where no further word hypothesis was found with a sufficiently high probability. In fact, we use preproduced graphs with a word recognition quality of about 76%. These word graphs are converted online into hypergraphs in order to model incremental distribution of word edges to the system. Hyperedges are updated if new word hypotheses arise that fit into an already existing context. These hyperedges are delivered to two components responsible for idiom detection and partial parsing according to the 10ms resolution of the HMM-recognizer.

The Idiom processor searches for lexically defined, fixed expressions such as greetings (*guten Tag*) or utterance parts that are used to continue the dialog flow (*einen Moment bitte*). Currently, we do not model inflected variants of idioms (like, e.g., support-verb constructions). After detection of an idiom it is sent to transfer and is treated as one atomic construction. Thus, the non-compositional character of idioms is taken into account. Additionally, information about the idiom is delivered to partial parsing, which renders two effects: First, the idiom can be integrated into larger constituents. Second, and evenly important, the word hypotheses the idiom is made of receive a penalty score. This reflects the assumption that in general it is fruitless to try to compositionally analyze an idiom. Because idiom recognition is much faster than parsing, this should also add to the performance of the system.

Syntactic and semantic interpretation are divided into two stages, partial parsing and utterance integration. This is due to the fact that spoken language shows a wide range of phenomena not usually covered by a standard grammar designed with written language in mind. Second, the construction of complement complexes where verbs are yet unknown (e.g. in German subordinate clauses) leads to complexity problems [4]. Consequently, we introduced two modules: The partial parser builds relatively small constituents (noun phrases, prepositional phrases, date expressions etc.), while the utterance integrator selects verbs and tries to construct verb expressions based on the relevant subcategorization information. Furthermore, PP attachment is handled here. The integrator is able to perform island analyses.

The next component in the application is incremental transfer. The transfer stage in a machine interpreting system has to obey incremental operation if the system as a whole is to meet the criteria set out in the introduction [5]. Transfer in MILC is based on chart processing algorithms, too. This enables the reuse

of already constructed target language constructions. The mapping algorithm is based on semantic knowledge, functor-argument structures are transferred from German into English.

Transfer starts with the smallest semantic object available from the integrator, the partial parser or the idioms processor. Typically, these are small NP constituents, mostly stemming from pronouns which tend to be short and are often recognized spuriously. As soon as richer semantic content is present in transfer, recursive equations in transfer rules are explored. Then, reuse of already constructed parts comes into play. The result of transfer is in any case a semantic description of the source language utterance parts in terms of the target language semantics. This selection is passed to generation.

The generation is chart-based like all components of the MILC system. The behavior of the generator is a mixture between [28], who binds the generator tightly into the domain of time present in analysis, and [21], who uses a chart to represent which part of the semantic content has already been taken care of. Our approach retains the temporal structure of the source language utterance. This entails that the extension in time is recorded for all edges that are received from transfer. But this does not extend to subsequent smaller parts that have to be generated according to generation rules. Here, the relative position of edges can be neglected, since they are only used to be integrated into larger chunks, and will never reach the system surface.

Nowhere in the system a requirement exists that one single edge has to cover the whole input. This is in contrast to most existing systems; only recently the incorporation of units smaller than a sentence or utterance has begun [1, 23, 30]. One consequence of this procedure is the presence of multiple solutions of the translation task within the generation component. There is a whole graph of possible partial surface forms that could be given to the user by synthesis. The approach we are taking here is to incrementally present growing optimal sub-paths of the solution graph. For the time being, the search criterion is the acoustic score of the source language words, combined with a penalty for skipping vertices of the solution graph (which results in a preference for one single long edge over several small ones). This selection schema means that we search for generation results of well recognized word sequences that can be translated.

4 Experiments

We have carried out preliminary experiments using the system described in the previous section. We used dialogs taken from the Verbmobil corpus of spontaneous speech. The results presented stem from an experiment covering one dialog (m123n) of eleven utterances, which was also used to construct the grammars of the system. The average utterance length was 16.4 words and 5.25 seconds speaking time. We used pregenerated incremental word graphs with an average number of 4157 edges, which corresponds to a hypothesis density of 253 edges per reference word. The utterances have not been previously used for training of the word recognizer. The overall acoustic recognition rate was approximately

76% based on the best matching word sequence compared to a reference. The linguistic knowledge sources consisted of a type hierarchy of 453 types, grammars with 99 rules and lexicons with 720 word forms (ca. 80% analysis, 20% generation, a hint to the sometimes schematic type of generation). Processing time was 15.25 seconds of CPU time per utterance on the average, system elapsed time was 12.55 s on a 2-processor SUN Ultra-4.

To give an impression of the kind of operation, consider the utterance `guten Tag Herr <NIB> Klitscher hier ist wieder Fringes ich möchte gerne diesmal einen Termin <NIB> für das Arbeitstreffen in der Filiale <SPELL> in Potsdam mit Ihnen vereinbaren <NIB> (m123n000)`. The output presented by the generator starts with several small constituents as shown in Tab. 1. The vertical lines (|) denote edge boundaries and demonstrate the incremental search for a best path through the solution graph. Finally, the best path for the completed generation graph is `Hello |Mr. Pfitzinger |it |from you |it |my |it |appointment for the work meeting in the branch |I |in the Potsdam |up to tuesday |I`.

Table 1. The first lines generated by MILC

Hello	Hello Mr. Quell
Hello it	Hello Mr. Quell it
Hello me	Hello it Mr. Pfitzinger
Hello me we	Hello Mr. Pfitzinger
Hello me Mr. Kopp	Hello Mr. Pfitzinger it
Hello Mr. Kopp	Hello Mr. Pfitzinger it you
Hello in Mr. Kopp	Hello Mr. Pfitzinger it to you
Hello me Mr. Quell	Hello Mr. Pfitzinger it from you

We evaluated the translations to be approximately correct to 64%. This is a preliminary evaluation as the utterances were used to construct the grammars of the system, but experiments with unseen data are underway. Moreover, a strict evaluation should cover a larger amount of test data. What we did was simply to judge if the central intention of the source language speaker and certain central propositional content like dates could be transported successfully into the target language. In the future, we will carry out a more thorough evaluation using a methodology similar to that used in [11]. But even now, it is obvious that the translation accuracy is too low for practical purposes and that the style of the translation needs to be improved. Measures to take into account are for example the utilization of prosody and dialog management, which have been deliberately left out in our experimental system, but which nevertheless have a great impact on the performance of a system [26, 2]. Moreover, the word recognition rate was only 76%, which should be increased. And finally, we need to model the selection process from the sets of generation candidates in a more suitable way. The reduction to acoustic evidence from the source language is not enough to guarantee a smooth output. At the moment, the quality of the combination of generation edges is neglected. A better selection schema could, for example, try to reanalyze the generation output to grade the legibility in a “hearing while speaking” model.

5 Conclusion

We have presented layered charts, a architectural framework for distributed, incremental systems for natural language processing, especially in the area of speech. They enable the construction of large, parallel applications that allow the exploration of complex interactions between speech processing components. We described an experimental interpreting system based on layered charts which demonstrated the feasibility of the approach. Further improvement is necessary by integrating prosodic interpretation and dialogue management, but the performance of less than threefold real time seems promising.

References

1. Steven Abney. Partial Parsing via Finite-State Cascades. In *Proceedings of the ESSLLI '96 Robust Parsing Workshop*, 1996.
2. Jan Alexandersson, Norbert Reithinger, and Elisabeth Maier. Insights into the Dialogue Processing of Verbmobil. In *Proc. of the 5th Conference on Applied Natural Language Processing*, Washington, D.C., 1997.
3. Jan W. Amtrup. ICE: A Communication Environment for Natural Language Processing. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA97)*, Las Vegas, NV, July 1997.
4. Jan W. Amtrup. Layered Charts for Speech Translation. In *Proceedings of the Seventh International Conference on Theoretical and Methodological Issues in Machine Translation, TMI '97*, Santa Fe, NM, July 1997.
5. Jan W. Amtrup. Perspectives for Incremental MT with Charts. In Christa Hauen-schild and Susanne Heizmann, editors, *Machine Translation and Translation Theory. Perspectives of Co-operation*, Text, Translation, Computational Processing (TTCP), number 1. Mouton de Gruyter, 1997.
6. Jan W. Amtrup and Volker Weber. Time Mapping with Hypergraphs. In *Proc. of the 17th COLING*, Montreal, Canada, 1998.
7. Xavier Aubert and Hermann Ney. Large Vocabulary Continuous Speech Recognition Using Word Graphs. In *ICASSP 95*, 1995.
8. Christian Boitet and Mark Seligman. The “Whiteboard” Architecture: A Way to Integrate Heterogeneous Components of NLP systems. In *COLING-94: The 15th International Conference on Computational Linguistics*, Kyoto, Japan, 1994.
9. Thomas Bub, Wolfgang Wahlster, and Alex Waibel. Verbmobil: The Combination of Deep and Shallow Processing for Spontaneous Speech Translation. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pages 1/71–1/74, Munich, Germany, 1997.
10. Bob Carpenter. *The Logic of Typed Feature Structures*. Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge, 1992.
11. David Carter et al. Translation Methodology in the Spoken Language Translator: An Evaluation. In *ACL Workshop on Spoken Language Translation*, 1997.
12. G. V. Chernov. Message redundancy and message anticipation in simultaneous interpretation. In Lambert and Moser-Mercer, editors, *Bridging the Gap: Empirical Research in Simultaneous Interpretation*, pages 139–153. John Benjamins, 1994.
13. Michael Finke, Maria Lapata, Alon Lavie, Lori Levin, Laura Mayfield Tomokiyo, Thomas Polzin, Klaus Ries, Alex Waibel, and Klaus Zechner. CLARITY: Inferring

- Discourse Structure from Speech. In *Proceedings of the AAAI 98 Spring Symposium: Applying Machine Learning to Discourse Processing*, pages 23–32, Stanford, CA, 1998.
14. Wolfgang Finkler and Anne Schauder. Effects of Incremental Output on Incremental Natural Language Generation. In *Proc. of the 10th ECAI*, pages 505–507, Vienna, Austria, August 1992.
 15. D. Gerver. Empirical studies of simultaneous interpretation: A review and a model. In R.W. Brislin, editor, *Translation: Applications and Research*, pages 165–207. Gardner Press, New York, 1997.
 16. Günther Görz, Marcus Kessler, Jörg Spilker, and Hans Weber. Research on Architectures for Integrated Speech/Language Systems in Verbmobil. In *Proc. of the 16th COLING*, pages 484–489, Copenhagen, Denmark, August 1996.
 17. Andreas Hauenstein and Hans Weber. An Investigation of Tightly Coupled Speech Language Interfaces Using an Unification Grammar. In *Proceedings of the Workshop on Integration of Natural Language and Speech Processing at AAAI '94*, pages 42–50, Seattle, WA, 1994.
 18. Kai Huebener, Uwe Jost, and Henrik Heine. Speech Recognition for Spontaneously Spoken German Dialogs. In *ICSLP96*, Philadelphia, 1996.
 19. Susanne J. Jekat. Automatic Interpretation of Dialogue Acts. In Christa Hauen-schild and Susanne Heizmann, editors, *Machine Translation and Translation Theory. Perspectives of Co-operation*, Text, Translation, Computational Processing (TTCP), number 1. Mouton de Gruyter, 1997.
 20. Martin Kay. Algorithmic Schemata and Data Structures in Syntactic Processing. Technical Report CSL-80-12, Xerox Palo Alto Research Center, Palo Alto, 1980.
 21. Martin Kay. Chart generation. In *Proc. of the 34nd ACL*, pages 200–204, Santa Cruz, CA, June 1996.
 22. Hiroaki Kitano. *Speech-to-Speech Translation: A Massively Parallel Memory-Based Approach*. Kluwer Academic Publishers, Boston, 1994.
 23. Marc Light. CHUMP: Partial Parsing and Underspecified Representations. In *Proceedings of the ECAI-96 Workshop: Corpus-Oriented Semantic Analysis*, 1996.
 24. W.D. Marslen-Wilson. Functional Parallelism in Spoken Word Recognition. *Cognition*, 25:71–102, 1987.
 25. W.D Marslen-Wilson and A. Welsh. Processing Interactions during Word Recognition in Continuous Speech. *Cognitive Psychology*, 10:29–63, 1978.
 26. Heinrich Niemann, Elmar Nöth, Andreas Kiessling, Ralf Kompe, and Anton Batliner. Prosodic Processing and its Use in Verbmobil. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 1997.
 27. Michael Niv. *A Computational Model of Syntactic Processing: Ambiguity Resolution from Interpretation*. PhD thesis, Univ. of Pennsylvania, 1993.
 28. Manny Rayner and David Carter. Hybrid Language Processing in the Spoken Language Translator. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, Munich, Germany, 1997. <http://www.cam.sri.com/tr/crc064/paper.ps.Z>.
 29. Shuly Wintner and Nissim Francez. Parsing with Typed Feature Structures. In *Proceedings of the 4th International Workshop on Parsing Technologies (IWPT95)*, pages 273–287, Prague, September 1995. Charles University.
 30. Klaus Zechner and Alex Waibel. Using Chunk Based Partial Parsing of Spontaneous Speech in Unrestricted Domains for Reducing Word Error Rate in Speech Recognition. In *COLING98P, COLING98L*, 1998.
 31. P. Zwitserlood. The Locus of Effects of Sentential-Semantic Context in Spoken-Word Processing. *Cognition*, 32:25–64, 1989.