Efficient Representation of Linguistic Knowledge for Continuous Speech Understanding

P. Baggia, E. Gerbino, E. Giachin and C. Rullent

CSELT - Centre Studi e Laboratori Telecomunicazioni
Via G. Reiss Romoli 274 - 10148 Torino, Italy

Abstract

This paper describes a linguistic knowledge representation technique suitable for reducing analysis time and memory requirements in a parser for continuous speech. Parsing speech, having to process a lattice of word hypotheses instead of a string of words, involves a tremendous amount of search and the generation of a high number of phrase hypotheses. The aim is, while using powerful and flexible formalisms for syntax and semantics, to generate "compact" phrase hypotheses, each one accounting for many syntactic rules simultaneously. The proposed method is able to cope with, and to take advantage from, the fact that short words are often missing from the lattice. A detailed example is given to clarify this method. Finally experimental data arc presented and discussed, showing the effectiveness of the proposed technique.

1 Introduction

The linguistic processor of a speech understanding system has to deal with a lattice of lexical hypotheses, that is with a set of hundreds of overlapping words hypothesized by a recognition subsystem, each with a particular score denoting its acoustical likelihood. To parse a lattice means to extract from it the best-scored word sequence that is compatible with the system linguistic knowledge. This activity induces a tremendous amount of search for any non-toy application. The uncertainty of input, in fact, combined with the large size of the language model, involves the generation of a high number of partial phrase hypotheses during parsing, especially for languages in which the constituents order is relatively free, like Italian. This difficulty may be coped with in two ways. One is to devise an intrinsically efficient parsing control strategy, as discussed in [Giachin and Rullent, 1989], where a fast algorithm for parsing word lattices, suitable for parallel implementation, was proposed. The other is a way for parsimoniously representing and efficiently using linguistic knowledge; this is the problem tackled here.

2 Linguistic Knowledge

It is generally acknowledged that different pieces of knowledge, like syntax and semantics, have to be used joindy during parsing [Hayes *et at.,* 1986; Niemann *et al.,* 1986; Poesio and Rullent, 1987]. For these reasons, two types of language representations are used in this research. One is a high-level, human-oriented representation; the other, automatically obtained from the first one through a *compiler,* is used by the parser and is able to cope with the problems mentioned above. The high level formalisms are a Dependency Grammar for syntax and Caseframes for semantics. The dependency grammar is augmented with information expressing morphological constraints between constituents, in a way similar to the approach followed in unification grammars. Dependency rules and caseframes are compiled into structures called Knowledge Sources (KSs), which retain the basic structure of the dependency rules and add semantic constraints to their constituents. The compiler partitions the head words into *word classes.* Each word class is associated to one or more KSs. Each KS defines all the possible connections between the words of its word class and other words.

The process of compilation *is* described in more detail in [Poesio and Rullent, 1987]. The focus of this paper is on the output of the compiler, not on the compiler itself; the formalism used at the high level is not relevant for this discussion.

Parsing may be viewed as the process of linking word hypotheses together into phrase hypotheses according to a priority defined by their scores. But what kind of structures can we build for representing phrase hypotheses? Suppose a "classical" grammar, like a context free grammar, *is* used and that we are trying to connect two words into a grammatical structure. In general, this can be done in several ways according to different grammar rules. Since structures built with different rules may connect with different word

This research has been partially supported by EEC ESPRIT project no. 2218 SUNDIAL.

hypotheses, we are compelled to record a new memory object for every structure. This makes no serious problem if we are processing written language, but in the case of speech there are two undesirable consequences. First, a very large memory size is required, owing to the high number of word combinations allowed by word lattices. Second, each of the structures will be separately selected and expanded, possibly with the same words, during the score-guided analysis, thus introducing redundant work.

We can avoid these two drawbacks by approaching the extreme situation in which only one single "compact" structure is generated for every group of different words, condensing all the information necessary to account for the different ways this word group may be connected to other words. Therefore, the compiler should generate a small number of "compact" KSs, still keeping the maximum discrimination power.

This is accomplished by the *fusion* technique. Two types of fusion have been experimented and both are described in the following. The second type of fusion is a more powerful generalization of the first type, hence it will be described in greater detail.

This approach would be useless if the constraints in the "compact" structures were hard to check and propagate, so efficient representation methods have been studied.

3 The Parsing Approach

3.1 Parsing as linking words

The parsing of a lattice may be seen as the activity of linking word hypotheses together on grounds of linguistic knowledge. Each link established between two word hypotheses is a hypothesis itself and is called *link hypothesis* (LH). It is represented in Fig.l.a.

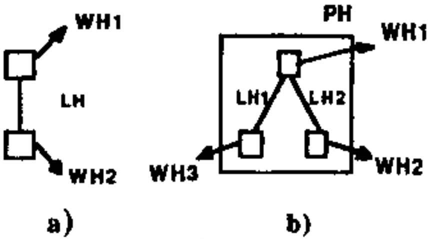

Fig 1 - a) A link hypothesis from WH1 to WH2.
b) A phrase hypothesis PH of two LHs.

A set of word hypotheses connected by link hypotheses is called *phrase group* (PG) and the final parsing result is a PG whose word hypotheses cover the whole speech time interval[1]. A PG is obtained step by step: at each step a new link hypothesis between two words is established; each of the two words may be already connected to other words.

A word in the dictionary is called *head* word if it can be, according to the system linguistic knowledge, head of a syntactic constituent, otherwise it is called *terminal* word.

Each PG is characterized by an *active head:* a head word that has been selected as the one in charge to be connected to other words; if the PG corresponds to a complete (sub)constituent, then the active head should be the lexical head of the (sub)constituenL

3.2 Representing phrase groups

At each parsing step the control strategy must select the best PG or word hypothesis to generate new PGs. A PG is based on the simultaneous presence of a certain number of LHs that, when departing from the same head word, are grouped together to form a compound set of links called *phrase hypothesis* (PH) (see Fig. l.b).

Fig 2 shows the creation of a new LH between WH1 and a certain PG having active word WH2.

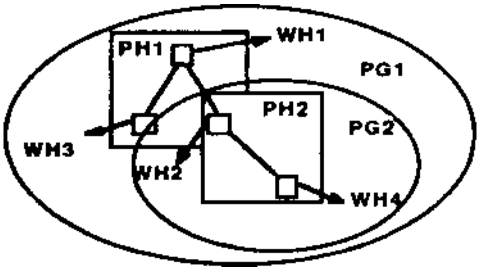

Fig 2 - The phrase group PG1 includes the four WHs.

For a more efficient representation, the control strategy always selects, as the active head of a PG, the deepest head word not yet saturated, i.e. that has still the possibility of being connected to other words. For instance let us suppose it is WH2 (see Fig. 2); then we can represent the Phrase Group PG1 simply through PH2 augmented with a pointer to its *context* PHI (se Fig. 3). In this way each time a new LH is added, only a new Phrase Hypothesis is created, having all the information needed by the control strategy.

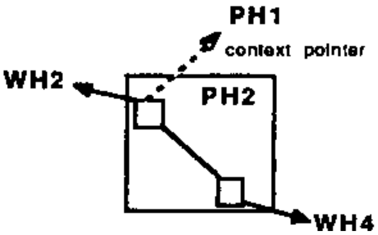

Fig 3 - PH2 in the context of PHel.

3.3 The control strategy

Each WH has a score assigned by the recognition stage; each PG has a score that is derived directly from the scores

[1] The word hypothesis refers to the non-determinism of data, the link hypothesis to that of linguistic knowledge.

and time intervals of the involved WHs using a density method [Woods, 1982] that guarantees that WH scores and PG scores are comparable^2. At each parsing step the best item (either a WH or a PG) is selected and the KSs are used to try to create LHs that, starting from that element, connect it to other (linguistically and temporally acceptable) items: words in the lattice and already produced PGs.

4 Rule Fusion

The production of KSs according to restricted rule fusion aims at compacting rules having the same constituent number and order.

Consider, as an example, two grammars rules, each with one head and one dependent. Suppose the head is the same, but the dependent is different in terms of syntactic and/or semantic constraints. Fusion produces one single KS, grouping the information of both rules. The KS accounts for the same general structure as the rules, i.e. it has a head and one dependent. The head is of the same type as in the original rules. The dependent, instead, has a generalized type. The KS contains a set of possible constraints (called *conds)* between the head and its dependent, derived from the original rules. When a head word for this KS connects itself to another word, the resulting PH will have all the conds active, except those that are incompatible with the features of the head word and of the dependent. When the PH is selected by the scheduler for expansion and new structures, containing more words, are created, these structures will contain all the remaining active conds, except those incompatible with the newly added words, and so on.

This technique provides the advantage of sensibly reducing the number of structures generated during parsing.

4.1 Generalized rule fusion

The generalized rule fusion aims at compacting together rules with constituents in different order or even with different number of constituents.

Table 1 contains four head words together with their classes, while Table 2 contains eight sentence fragments that can be composed together to generate a large number of Italian sentences.

M =	messaggio	(message)
A =	Roberto	
B =	ieri	(yesterday)
C =	spedito	(sent)

Table 1

M1 =	"messaggio	(message)
A1 =	"di Roberto"	(of Roberto)
A2 =	"da Roberto"	(by Roberto)
A3 =	"Roberto"	(Roberto)
A4 =	"a Roberto"	(to Roberto)
B1 =	"di ieri"	(of yesterday)
B2 =	"ieri"	(yesterday)
C1 =	"spedito"	(sent)
C2 =	"che ha spedito"	(that has been sent)

Table 2

Of course the admissible sequences do not depend only on the class of the fragments but also on their specific characteristics, like morphological features, grammatical relations, semantic contents, etc. Here, to simplify the description, different sets of constraints for a sentence fragment are represented just by appending a number to the fragment class.

row	position	
	-1	0
a1	di	A1
a2	da	A2
a3		A3
a4	a	A4

Table 3

row	position	
	-1	0
b1	di	B1
b2		B2

Table 4

Tables 3, 4, 5, 6 contain, for each head class (A, B, C, M), a sketchy representation of the rules involved. The positions of the constituents are also shown. The zero position indicates the head.

row	position				
	-2	-1	0	1	2
c1			C1	A2	
c2			C1	B2	
c3			C1	A2	B2
c4			C1	B2	A2
c5	che	ha	C2	A3	
c6	che	ha	C2	A3	B2
c7			C1	A4	
c8			C1	A4	B2
c9			C1	B2	A4

Table 5

row	position		
	0	1	2
m1	M1	A1	
m2	M1	B1	
m3	M1	A1	B1
m4	M1	B1	A1
m5	M1	C1	
m6	M1	A1	C1
m7	M1	B1	C1
m8	M1	C2	
m9	M1	B1	C1

Table 6

4.2 KSs as rules

Let us suppose we have a WH of class C "spedito" (sent) and we want to connect it to other words that can depend on it and that are adjacent to the head on the right; from Table 5 constituents of both classes A and B arc involved. Let us focus on the class A case.

Without any kind of fusion we would have a KS for each row of Table 5. As we want to find class A constituents, on the right of the header, six KSs are involved, corresponding to rows cl,c3,c5,c6,c7,c8; the first two KSs propagate constraints (summarized by A2) that

all WHs of a given PG have the same time interval, then the score of the PG would be the mean value of these scores.

will be considered by the second KS of class A (Table 3, row a2).

If such KS can find a WH like "Roberto", it tries to link it to the preposition "da" (by) that is supposed to be missing from the lattice; a place holder (a virtual WH) is created and linked. Two couples of PHs are generated, one for each KS (rows cl and c3 of Table 5), to represent the two phrase groups that are supported by the same set of WHs ("spedito", "Roberto" and the place holder)

Other two couples of PHs are generated in a completely similar way by the KSs of row c7 and c8, the only difference being that the place holder is generated by the fourth KS of class A (row a4) and will be for a WH like "a" (to).

The KSs of rows c5 and c6 require a sentence fragment of kind A3, i.e. a preposition is not required; two new PHs will be generated.

4.3 KSs with generalized fusion

In this case there is just one KS for the nine different rows of Table 5. The C KS propagates the constraints for the A KS: it propagates A2+A3+A4 (only Al is excluded) and the time constraint that the constituent must be adjacent (on the right) to the header with a threshold that includes the possibility of a missing preposition.

Only one search into the lattice is performed by the A KS, obtaining the "Roberto" WH. Only one PH is created for rows c5 and c6 (see Fig. 4.a), marked with active rows c5 and c6 and with constraints C2. A new couple of PHs is created for the the rows cl,c3,c7,c8 (sec Fig. 4.b). Just one place holder is created, standing either for a word hypothesis "da" (by) or "a" (to).

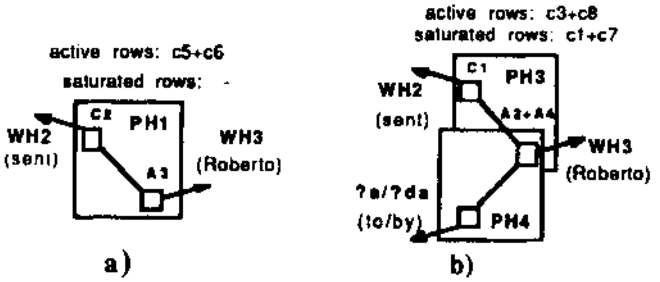

Fig 4 - a) The PH for constraints A3.
b) The PG3 for constraints A2 and A4.

PH4 is characterized by the union of constraints A2 and A4. In its turn PH3 is characterized by constraints Cl, active rows c3 and c8 and saturated rows cl and c7.

Active rows are used to decide if and how new links can be established; for instance, the link starting from the head WH2 of PH3 (Fig. 4.b) has to be of class B and constraints B2 (position 2 for rows c3 and c8 in Table 5).

The fusion techniques greatly reduce the number of intermediate items (PHs) that have to be generated. However, this would be of no use if it were balanced by an increased activity for checking and propagating constraints. To avoid this problem each KS position (excluding the head position) is characterized by a list of admissible classes (A and B for position 1 of the KS of Table 5) and for each of them by a bit position of the rows involving that class in that position. The active rows of a PH are represented by a bit position too, so both the activity of deciding if a PH is saturated and the decision if the word can be linked to a word of a given class, can be performed through quite simple AND operations. Bit positions are also used to represent morphological information, admissible grammatical relations and semantic information (i.e. what has been indicated here just as a subscript).

5 Parsing a Lattice: An Example

The parsing input is a lattice of WHs, each one characterized by the word label, the beginning frame, the ending frame and the acoustical score. Fig. 5 is an example of a lattice[3] for the utterance "mcssaggio spedito da Roberto".

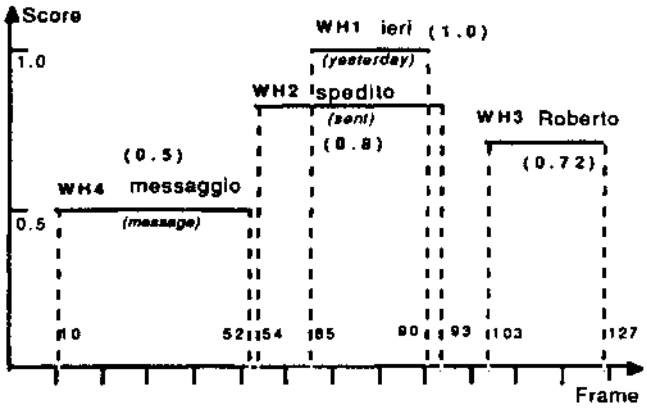

Fig 5 Example lattice for the utterance: "messaggio spedito da Roberto (message sent by Roberto).

The best scored WH in the lattice, WH1 "ieri" of class B and score 1.0, is selected. The KSs could be used to try to connect WH1 either with WHs depending on it *{top-down* strategy) or with head WHs from which WH1 may depend from *(bottom-up* strategy). Both directions could be tried, but it is better to find first a complete constituent before starting a bottom-up activity.

The KS of class B (see Table 4) has two admissible sequences of constituents; row bl requires the presence of a terminal word: the preposition "di" (of), on the left of the head; for b2, the head alone is already a complete constituent Therefore both the strategies are tried.

Following the top-down action a link towards a preposition is tried. As short words may be missing from

[3] Not a realistic one: real lattices contain hundreds of WHs.

the lattice, it is possible to hypothesize the correct word without its presence in the lattice [Giachin and Rullent, 1988]. The result is the creation of a link from WH1 to the missing WH, labelled ?di, creating an instance of a phrase hypothesis (PHI) as in Fig. 6.a. The Phrase Group PG1 ("?di ieri") is represented by PHI, having the same score of WH1 (i.e. 1.0) and a time boundary that takes into account a default time interval for ?di.

In the bottom-up action all the KSs that may connect head words to the class B word "ieri" are triggered; in the example they are of class C and M (see Tables 5 and 6). Exploiting the time constraints from WH1, the lattice is searched to find proper head WHs for each of the two KSs. Only class M KS is successful by finding the WH4 "messaggio" that is linked to WH1. A check is made to see if the link has determined a complete constituent; this is the case as the m2 row of the M KS (see Table 6) requires just a class B dependent on the right of the head. The result of the successful connection is PG2 ("messaggio ?di ieri") as in Fig. 6.b.

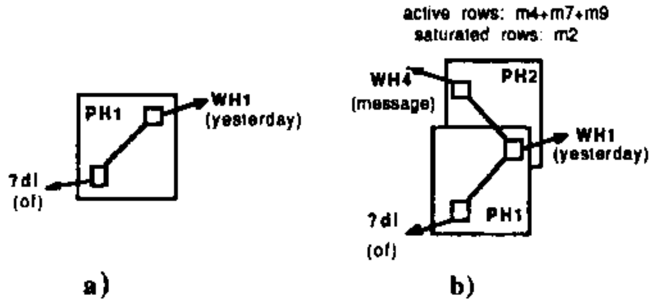

Fig 6 - a) PG1 "?di ieri", score 1.0.
b) PG2 "messaggio ?di ieri", score 0.69.

PH2 contains all the run-time informations pertaining to the links, i.e. the possible rows of the KS have been restricted for PH2 only to m2, m4, m7 and m9, and a new score (0.69) has been computed.

Since WH2 "spedito" of class C has a beuer score (0.8) than PG2 (0.69), it is now selected. For the KS of class C (see Table 5) only a top-down activity is possible. Trying to exploit time adjacency, position 1 dependents are taken into account; they must be of class A or B, in four different characterizations (A2, A3, A4 and B2).

Only WH3 "Roberto" of class A, satisfies the time constrains given by WH2. A direct link between WH2 and WH3 (justified by row c5 and c6) is not created because A3 imposes a strict time adjacency between WH2 and WH3, while the gap between the two WHs is supposed to be larger than the *fixed* threshold between adjacent words.

So class A KS must connect WH3 to the preposition "da" (by) or "a" (to) to generate constituents that satisfy either the A2 or A4 characterization (see Table 3). This activity is performed in the context determined by WH2 and the result is as presented in the previous section (see Fig. 4.b).

When PH4 is created, PH3 rows are restricted to cl +c3 (due to A2) and c7+c8 (due to A4). Rows cl and c7 corresponds to complete constituents (saturated rows in Fig. 4.b), while for rows c3 and c8 a dependent of class B is still required, but it is not present in the lattice.

The resulting complete PG, PG3 ("spedito ?da?/?a? Roberto" with score 0.78), can start a bottom-up activity. The only KS that is able to connect WH2 is the class M KS, that at last can create PG5 (see Fig. 7.a) of score 0.66. This is a solution (S2) but has a score worse than that of WH3; so it is not accepted and has to wait until no other better items remain active.

Now the best score pertains to WH3 of class A (0.72) and the KS of class A (see Table 3) is considered; there are four possible characterizations (AI, A2, A3 and A4). Characterization AI, A2 and A4 requires the presence of a short word on the left of the head, but only one PH, (PH6 in Fig. 7.b) is generated.

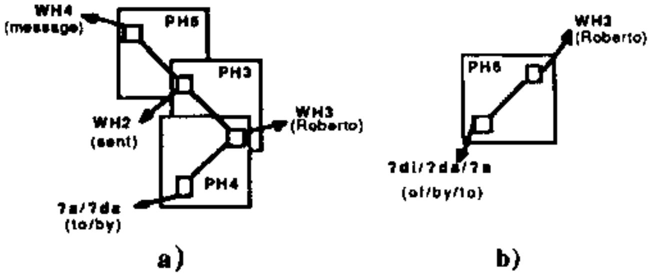

Fig 7 - a) PG5 "mess. spedito ?a/?da Roberto", score 0.66
b) PG6 "?di/?da/?a Roberto, score 0.72.

From row a3 it is not possible to obtain any link with other WHs in the lattice (as we have seen before), while from PH6 a link can be tried towards WH4 in PH2 (see Fig. 6.b), so that a merge of two PGs can be attempted. One of the active rows still alive in PH2 (m4) allows the link with a dependent of class A (with characterization AI) in position 2 and a cross restriction is performed on PH2, with the creation of a new PH7, which maintains only the admissible row m4. PH6 is copied to create PH8 and the ambiguity about the preposition is solved thanks to constraint AI, see Fig. 8.

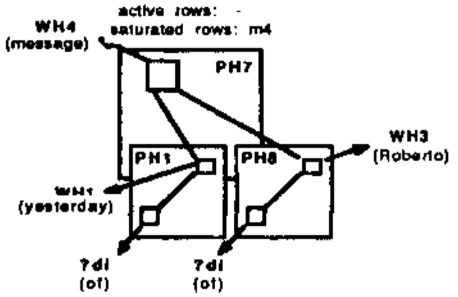

Fig 8 - PG7 "mess. ?di ieri ?di Roberto", score 0.69.

The compound PG7, "messaggio ?di ieri ?di Roberto", represents a solution (SI) with score 0.69, which covers the whole time interval.

At this point, being the parser almost optimal and PG7 a solution, the parsing process could stop. In rare occasions, as in the example, the solution may be wrong. For this reason, the *verification phase* has been introduced. The parser does not stop at the first solution, but continues for a fixed time until a few solutions have been collected.

In the example the solution S2 ("messaggio spedito ?da/?a Roberto") previously generated will be accepted by the control strategy (see Fig. 7.a).

At the end of the parsing activity a phase of acoustical verification of the solutions eliminates the ambiguity between multiple candidates for each short word and reorders the set of solutions. In the example the acoustical verification will identify the new best scored solution S2 as the correct one and will eliminate the ambiguity between the two prepositions ?a and ?da in S2 by selecting the correct one ("da").

6. Experimental Results

The validity of the approach has been assessed by three series of experiments, using the no fusion, the restricted rule fusion, and the generalized rule fusion approach respectively. In all cases the test data consist of 600 word lattices produced by a speaker independent recognition system from sentences uttered in a continuous fashion (i.e., with no pauses between words) and with large linguistic freedom. The utterances, referring to a 787-word electronic mail management task, were recorded from a PABX and include 10 different speakers.

The recognition system, described in more detail in [Fissore *et al.,* 1989], is based on Hidden Markov Model technology. 310 discrete models are used to describe subword speech units. The lattices are produced using a Viterbi-likc decoding algorithm that finds the best match for each word in every possible position of the utterance.

	No Fusion	Restricted Fusion	Generalized Fusion
No.PHs generated	806	383	91
Parsing time (s)	1.56	0.86	0.38

Table 8

Table 8 reports the average number of PHs generated and the CPU time (in seconds) for each sentence. The parser, written in C, runs on a Sun SparkStation 1. The most dramatic improvement refers to the reduction of PHs. This is not surprising: "compact" KSs permit to spare the generation of redundant structures, but each structure requires more work to be build. The high reduction of generated structures, anyway, compensates this fact and still allows a more than fourfold cut of total timings with respect to the no fusion case.

Table 9 shows the effect of the verification phase on the percentage of correctly understood sentences.

No verification	With verification
66.0%	72.2%

Table 9

7. Conclusions

One of the problems in parsing word lattices is the generation of a large number of partial phrases. This number may be reduced provided a compact representation of similar language constructs is adopted. The proposed approach consists in fusing several rules of a high-level, human oriented language representation into a small number of knowledge sources. Each knowledge source is provided with a concise definition of syntactic and semantic constraints necessary to specify how words may be connected together. Alongside, efficient methods of checking and propagating these constraints are provided. The approach has been validated through an extensive experimentation with real data on a non-toy application obtaining a great decrease of the number of generated phrases and of total analysis times.

References

[Fissore *et al.,* 1989] L. Fissore, P. Laface, G. Micca, and R. Pieraccini, "A Word Hypothesizer for a Large Vocabulary Continuous Speech Understanding System", *Proc. ICASSP 89,* Glasgow, May 1989.

[Giachin and Rullent, 1988J E.P.Giachin and C.Rullent. "Robust Parsing of Severely Corrupted Spoken Utterances", *Proc. COUNG-88,* Budapest, August 1988.

[Giachin and Rullent, 1989] E.P.Giachin and C.Rullent, "A Parallel Parser for Spoken Natural Language", *Proc. 1JCAI 89,* Detroit, August 1989.

[Hayes *et al,* 1986] P.J.Hayes, A.G.Hauptmann, J.G. Carbonell and M.Tomita, "Parsing Spoken Language: a Semantic Caseframe Approach", *Proc. COLING 86,* Bonn, WG, August 1986.

[Niemann *et al.,* 1986] H.Niemann, A.Brictzmann, U.Ehriich and G. Sagerer, "Representation of a Continuous Speech Understanding and Dialog System in a Homogeneous Semantic Net Architecture", *Proc. ICASSP 86,* Tokyo, April 1986.

[Poesio and Rullent, 1987] M.Poesio and C.Rullent, "Modified Caseframe Parsing for Speech Understanding Systems", *Proc. UCAJ 87,* Milano, August 1987.

[Woods, 1982] W.A.Woods, "Optimal Search Strategics for Speech Understanding Control", *Artificial Intelligence,* vol. 18, 1982.