

Massimo Poesio**, Claudio Rullent

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

ABSTRACT

We suggest for speech understanding systems a caseframe parsing strategy which deviates from 'pure' caseframe parsing in at least two respects: parsing is not exclusively based on top-down instantiation of caseframes, and caseframes are merged before use with syntactic knowledge. Our strategy has been developed to be executed as an inferential process in which a set of Knowledge Sources cooperate through the blackboard. The Knowledge Sources are automatically defined by merging syntactic and semantic knowledge expressed declaratively: this integration of syntax rules and caseframes makes it possible to exploit simultaneously both syntactic and semantic constraints.

I. INTRODUCTION

Parsers for spoken natural language and parsers for typed natural language must meet different requirements. Parsers for speech work on a set of lexical hypotheses rather than on sequences of words. An (unpleasant) consequence is that it is usually possible to find sequences of hypotheses not corresponding to the sequence of words actually uttered but which nevertheless constitute complete and correct interpretations. A parser for speech must be able to work when some words like articles and prepositions have not been recognized, and to decide whether two slightly overlapping lexical hypotheses are competing or not. For all these reasons, parsing a spoken utterance requires inferences as complex as those of empirical, inductive enterprises as expert systems and originates similar problems.

Caseframe parsing has recently been proposed ([Hayes et al. 1986], [Brietzmann, Ehrlich 1986]) as an alternative to the methods developed during the ARPA-SUR project like semantic grammars ([Hayes-Roth 1980]) or networks ([Lowerre 1976]). Having chosen the blackboard architecture, suitable to cope with the problems of speech parsing sketched above, we have introduced two modifications to 'pure' caseframe parsing to use it effectively into such a framework: i. alternating between top-down instantiation of caseframes and bottom-up prediction of caseframes requiring a casefiller already recognized; and ii. using caseframes only after they have been merged with syntactic knowledge.

II. A REVIEW OF CURRENT RESEARCH IN CASEFRAME PARSING

The basic notion of caseframe is that of a head concept modified by a set of related concepts ([Fillmore 1968], [Tesnière

1959]). Each modifier plays a role (CASE) with respect to the head concept. Caseframe parsing has been proposed long ago for typed natural language parsers ([Hayes, Carbonell 1981]) and its applicability to parsers for speech has recently become an object of study ([Hayes et al. 1986], [Brietzmann, Ehrlich 1986]). This strategy is promising because the interpretation is anchored to most significant parts of the input, and semantic expectations generated from these more meaningful parts can be used both to discriminate between different candidate fillers and to hypothesize the meaning of troublesome fragments. But current caseframe parsers for speech are still close to parsers developed for typed text, which causes two main problems.

The first problem is that the parsing strategy they use is top-down: in a first stage all potential caseframe headers are activated, then caseframes are 'instantiated' by finding suitable fillers. This strategy would be justified for parsing speech only if the recognizer would assign the best scores to LHs corresponding to caseframe headers (verbs, common and proper nouns); in this case the search would always follow the best directions. In a previous version of SUSY we discovered that this assumption is only partially true. In Italian there are in fact words like "quale" ("which", "what") or "dove" ("where") that very often have a better score than caseframe headers and can generate useful bottom-up predictions ("dove", for instance, strongly prefers caseframes in which a LOC case is required).

Another problem of these systems is the integration of syntax and semantics. In the approach followed by ([Hayes et al. 1986]), caseframes are augmented with syntactic constraints on case fillers and separate phrase structure rules are developed to combine the resulting caseframes. Another approach ([Brietzmann, Ehrlich 1986]) is that of developing different syntactic interpretations of fragments, that are then compared to find the one that fits best. Neither of these approaches is completely satisfying. Our idea is that, both for completeness and clarity, the grammar should be developed independently (using for instance phrase structure rules) and integrated only in a second moment with semantic knowledge (as noted by [Hayes et al. 1986] their approach gets into trouble when the syntactic complexity of the sentence increases). But we also think it is necessary to reduce the size of the inferential activity, by applying both syntactic and semantic constraints simultaneously even when aggregating the smaller fragments.

III. THE SUSY SYSTEM

The Speech Understanding SYstem (SUSY) understands and answers continuous speech queries about Italian geography. SUSY receives a lattice of lexical hypotheses produced by an independent word recognition module ([Laface et al. 1987]).

* This research has been partially supported by the EEC ESPRIT Project n.26 "Advanced Algorithms and Architecture for Signal Processing".

** Current address of Massimo Poesio is Project WISBER, University of Hamburg, Postal P.O.Box 302762 D-2000 Hamburg 36 West Germany.

We have adopted for SUSY the blackboard approach, very effective in such complex problem solving tasks ([Erman et al. 1980]), and developed SUSY's parsing strategy by modifying 'pure' caseframe parsing to overcome the problems previously described and make it work in the blackboard framework. Our 'impure' methodology can be summarized as follows:

- A set of Knowledge Sources (KSs) build and validate hypotheses about utterance fragments (Deduction Instances) that are put on the blackboard. Each DI represents a sentence fragment at some stage of aggregation, and it is supported by a set of lexical hypotheses (the DIs are an elaboration of the HWIM idea of islands [Woods 1982]). DIs have a score derived from those of the associated LHs by using the score combining methods described in [Woods 1982].
- Only sentence fragments that can be characterized both syntactically and semantically are aggregated. This way we can exploit syntactic and semantic constraints simultaneously even at the lowest levels of aggregation. Syntactic and semantic knowledge are partitioned into KSs following the principle that each KS owns the syntactic and semantic competence necessary to deal with a particular class of sentence fragments.
- Top-down, expectation-based parsing activities are merged with bottom-up, predictive activities. An incomplete DI (goal DI) can originate a search for missing components, causing the activation of the KSs (and of those only) that deal with the classes of fragments that are required; vice versa, complete DIs (i.e. covering completely a time interval) when put on the blackboard activate KSs requiring a DI of their class as a component (for instance, a complete casefiller will activate KSs building caseframes requiring that particular case filler).
- A complete DI *d* carries on an interpretation of the fragment it represents; this interpretation is used by other KSs to generate interpretations of larger fragments having *d* as a component. An interpretation is not necessarily a caseframe. Also fragments whose interpretation is a casefiller or a definite description are aggregated: these DIs can be shared by several DIs to avoid repeating the same parsing steps more than once.
- Neither syntactic rules nor caseframes are used as such, but compiled to get the KSs. Syntactic and semantic knowledge are, however, expressed declaratively and independently developed. Using this approach we get more flexibility than either completely compiled systems like HARPY or systems that make use of semantic grammars, while maintaining an efficiency and a constraining ability superior to those that can be obtained by using the two forms of knowledge independently and declaratively.

IV. REPRESENTATIONS FOR SYNTAX AND SEMANTICS

For syntactic rules we have used the formalism of Dependency Grammars ([Hays 1964]). There are two advantages in using DGs: structures based on 'governors' and 'dependents' can be easily mapped onto caseframes (which is not casual, as both dependency grammar and valency theory trace back to the work by [Tesnière 1959]), and parsing is strongly anchored to input since, for a DG rule to be applied, it is necessary to find a suitable candidate for the governor.

We have adopted Conceptual Graphs (CG) [Sowa 1984] to represent caseframes. The primitives representable in CGs are individuals and relations. Each individual is characterized by

a type and an individual marker (imarker for short). Types are organized in a type hierarchy where any two types *t1* and *t2* have a maximal common subtype *t1***t2*. An individual *i* conforms to type *t* if it is an instance of a type *t1* <= *t*. A Conceptual Graph is recursively defined as either a concept, or a concept connected to other concepts by Conceptual Relations (CR). The Canonical Graph (CAGR), is a special CG associated to types to represent selectional restrictions.

Sowa's theory has the advantage of formal CAGRs unification operations that conserve the selectional restrictions. For the purposes of this paper we are interested in three of these operations: by join, two CAGRs are unified over a 'common' concept (a concept appearing in both CAGRs); by type restriction, the type of a concept can be substituted by any of its subtypes; by referent restriction, a generic referent in a concept can be substituted by an imarker.

Caseframes are represented in SUSY as CAGRs in which one concept, called the head, represents the head of the caseframe: for instance, the caseframe we use for the predicate LOCATED-IN-REGION (one of the interpretations of the verb "trovarsi", "to be located"***) is represented by the canonical graph

```
[LOCATED-IN-REGION]
->(AGNT:Obligatory)->[MOUNT+PROVINCE+LAKE];
->(LOC:Obligatory)->[REGION].
```

Fig. 1 - The caseframe for LOCATED-IN-REGION

in which cases are represented as CRs. In this framework both caseframes connection and case filling are reduced to join operations, since simple casefillers are represented as concepts (i.e. conceptual graphs without CRs) and complex casefillers are caseframes themselves.

V. COMBINING CASEFRAMES AND SYNTAX TO DEFINE KNOWLEDGE SOURCES

All KSs share a common body of procedural knowledge (routines to check temporal constraints, functions that compute the score of a DI); they all include knowledge about syntactic and semantic constraints on case fillers, and must return an interpretation, that is always built using joins and restrictions only. The behaviour of a KS is then completely defined by specifying i. the class of DIs it can aggregate, ii. its activation conditions, iii. the set of constraints and iv. the way caseframes must be instantiated to get the interpretation.

Since only fragments that can be classified both from a semantic and from a syntactic point of view are aggregated, a DI *d* could be classified twice: with a syntactic category *C* (syntactic categories are types in the type hierarchy) and with a type *T* of the domain. The type that is effectively used to classify *d* is therefore *Y* = *C***T* (for instance, the KS above will produce DIs of class VERB*LOCATED-IN-REGION). The definition of KS23.1 in Fig. 2 includes a description of the compositional structure of the fragment in terms of semantic types, that is used together with the (isomorph) compositional structures of associated syntactic rules for expectations and predictions purposes: the KS above, for instance, will aggregate sequences whose first element is of class ADV*REGION or NOUN*REGION, the second element of class PART-RIFL*JOLLY or BE*JOLLY, etc. The type JOLLY is used to classify words whose -

*** Since in our domain there are restrictions on what entities can be contained in others (an island can only be located in a sea, for instance) and representing these restrictions in CAGRs would lead to unnecessary complexities, we have introduced different subtypes of LOCATED for each 'containing' entity.

syntactic classification only is relevant (and that can eventually be skipped).

(DefKS KS23.1

```
;; Compositional Constraints
LOCATED-IN-REGION = REGION JOLLY <HEADER> MOUNT+PROVINCE+LAKE

;; Associated Syntactic Rules
(Dr23 Dr23a Dr23b Dr23c Dr23d DR23e)
;;VERB = NOUN PART-RIFL <GOVERNOR> NOUN
;;VERB = NOUN BE <GOVERNOR> NOUN
;;VERB = NOUN PART-RIFL <GOVERNOR> PROPER-NOUN
;;VERB = ADV PART-RIFL <GOVERNOR> NOUN
;;VERB = ADV BE <GOVERNOR> NOUN
;;VERB = ADV PART-RIFL <GOVERNOR> PROPER-NOUN

;; Activation Condition
G(?X) ==> ACTION (?X LOCATED)

;; Caseframes Instantiation Rule
*RIS((LOCATED ! *
      AGNT ?Z
      LOC ?Y)) <==
RIS(REGION (ACTION ! * ?X ?Z))
RIS(MOUNT+PROVINCE+LAKE (ACTION ! * ?W ?Y)))
```

Fig. 2 - KS definition

KS definitions are the result of a compilation. They are produced from dependency rules, caseframes and mapping rules. For example, the definition of KS23.1 has been obtained from the set of associated dependency rules in Fig. 2, one of which is shown in Fig. 3.a with its mapping rule (Fig. 3.b), and from the caseframe in Fig.1.

- a. (Dr23 (VERB = NOUN PART-RIFL <GOVERNOR> NOUN)
 VERB (MODO (INDIC)) (TEMPO (PRES)) (PERS (3))
 (NUM ! ?N) (TRANS (VIT))
 (COMPLEMENTO ! (STATO-LUOGO))
 (RIFL (RIFL))
 NOUN (T-COMPL ! (STATO-LUOGO))
 PART-RIFL NIL
 NOUN (NUM ! ?N))
- b. (Mr23 ((LOCATED-REGION LOCATED-PROVINCE WASH) =
 LOC (JOLLY) <HEADER> AGNT))

Fig.3 - A dependency rule with its mapping to caseframes.

The compilation is facilitated by the structural similarity between dependency rules and caseframes (both are essentially based on the idea of a 'head' modified in some way). The compositional part of the KS is derived from that of the dependency rule by substituting syntactic types with the value restrictions on case fillers specified in the caseframe; the position of cases in the fragment is specified in the mapping rule. Only those cases that can be filled using information present in the fragment are included; this pre-selection avoids searching for information that could not be found. The activation conditions are associated to types in the knowledge base.

Having partitioned the knowledge in independent chunks, when a new syntactic rule is added it is never necessary to recompile the entire set of KSs: either a previously defined KS is modified to take into account the new rule, or new KSs are added to the existing set. In this latter case, since not all syntactic/semantic combinations are admissible, the number of KSs that are generated is limited. The case in which a new type

is added to the domain has more serious consequences: with our current set of about 150 dependency rules in the average about 15 new KSs are defined when adding a new entity type. These problems are typical of semantic grammars; in our case there is the advantage that it is the system itself that generates the new KSs.

VI. CASEFRAME INSTANTIATION

The meaning of a word in the dictionary is a set of predicates able to activate the appropriate KSs: for instance, the meaning of the word "trova" includes the predicate

ACTION(TROVA LOCATED)

that activates KS23.1 of Fig.2. That KS includes a caseframe instantiation rule that produces the instantiated caseframes constituting the interpretation of the sentence fragments it analyzes. By this approach, information duplication is avoided, since a single caseframe is shared by many words, and caseframes instantiation is more efficient, since the instantiation rule specifies 'a priori' how the caseframe associated with the header must be joined with those associated with the component DIs.

Only two operations are needed to build the interpretation: referent restrictions are used to fill concepts, and joins are used to merge concepts representing cases with canonical graphs representing casefillers. Representing a canonical graph as a term and generic referents as logic variables, joins can be effectively implemented as unification operations using a PROLOG interpreter after all components indicated in the composition rule have been found. The instantiation rule of KS23.1 (reproduced in Fig.4.a) together with the CGs carried by DIs of class MOUNT+PROVINCE+LAKE (4.b) and REGION (4.c) produces the instantiated caseframe in (4.d). This unification has a very low cost, since the interpreter runs in a special context in which only the results carried by accepted components (the "ris" predicates in Fig.4) have been asserted.

- a. RIS((LOCATED ! *
 AGNT ?Z
 LOC ?Y)) <==
 RIS(MOUNT+PROVINCE+LAKE (ACTION ! * ?X ?Z))
 RIS(REGION (ACTION ! * ?W ?Y)))
- b. RIS(MOUNT+PROVINCE+LAKE
 (ACTION ! * AGNT (MOUNT ! ROSA)))
- c. RIS(REGION (ACTION ! * LOC (REGION ! ?r)))
- d. (LOCATED ! *
 AGNT (MOUNT ! ROSA)
 LOC (REGION ! ?r)))

Fig.4 - Caseframe instantiation

VII. INTEGRATING TOP-DOWN AND BOTTOM-UP CASEFRAME PARSING

Susy always moves from the best Lexical Hypothesis or Deduction Instance. A top-down, expectation based activity is started if a caseframe header is selected; a bottom-up predictive activity is initiated when a DI representing a casefiller or a LH representing a case marker are selected.

Let us consider the sentence (translated word by word):

IN QUALE REGIONE SI TROVA IL MONTE ROSA?
 (In which region is located mount Rosa?)

The LHs involved in the solution, ordered by scores (the best first) are:

- 1) QUAL (which) 1.267
- 2) IN (in) 1.301
- 3) MONTE (mount) 1.371
- 4) SI 1.485
- 5) REGIONE (region) 1.560
- 6) TROVA (is located) 1.610
- 7) ROSA (Rosa) 1.629

Fig.5 - The LHs involved in the solution

The process leading to the solution is displayed in Fig.6:

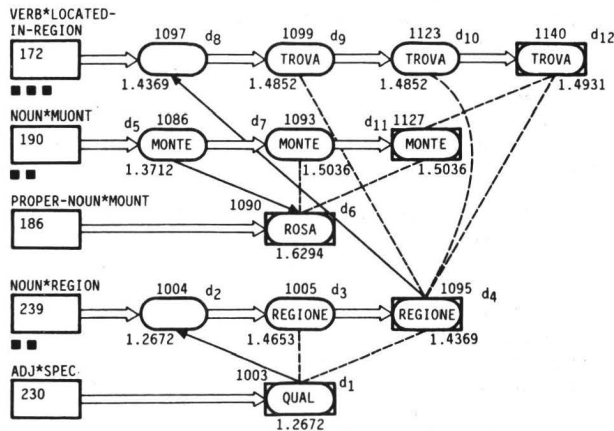


Fig.6 - The deductive process leading to the solution

- The best LH, "qual" is not a caseframe header but an interrogative adjective that acts as a casemarker together with the preposition "in". The KSs whose activation conditions are satisfied by the predicates associated with the word "qual" are triggered. One of these KSs requires no other components and produces the complete Deduction Instance d1 of class ADJ*SPEC, that is put on the blackboard.
- Since d1 has a good score the analysis proceeds along this path, causing a bottom-up activation of all the KSs having a component of class ADJ*SPEC. The activation of one of these, a KS of class NOUN*REGION, generates d2. d3 is generated when a suitable caseframe header is found (the LH "regione"); later, also the casemarker "in" is found, and the complete DI d4 is generated.
- As the score of d4 is worse than that of the LH "monte", this branch of the inferential process is momentarily abandoned. The caseframe header "monte" causes the activation of all KSs instantiating MOUNT *type caseframes. The KS of interest requires only an individual referent, that is found (the LH "Rosa"). This process ends in the DI d7 of class NOUN*MUONT.
- The score of d7 is not very good, due to the bad quality of the LH "Rosa". The deduction path that had produced d4 (with score 1.4369, better than that of d7) is resumed. The KSs requiring a LOC case to be filled by an entity of type REGION are activated; among these, KS23.1. The DI d8 is generated by KS23.1 when d4 is accepted as casefiller; d9 when the header of the caseframe for the type LOCATED-IN-REGION is found (the LH "trova").

- The caseframe for LOCATED-IN-REGION also requires the AGNT case to be filled. A suitable filler is d11 (derived from d7 after the aggregation of the article "il"). The final, complete DI d12 is obtained; note that this last step involves the confluence of two deductive processes that developed independently. The three canonical graphs of KS23.1 are joined to give the final interpretation, the instantiated caseframe of fig. 4d.

VIII. CONCLUSIONS

Two modifications have been proposed to caseframe parsing in order to make it work effectively in a blackboard framework: a more flexible strategy alternating top-down and bottom-up parsing, and the compilation into the same knowledge source of syntactic and semantic knowledge. Due to the integration of syntactic rules in the caseframe approach, the system is able to handle fairly complex sentences with a reasonable efficiency.

This approach has been implemented and tested and we are now improving the compiler and the overall blackboard architecture.

IX. REFERENCES

- 1 Brietzmann, A., Ehrlich, U., "The role of semantic processing in an automatic speech understanding system", *Proc. COLING-86*, Bonn, West Germany, 1986
- 2 Erman, L.D., Hayes-Roth, F., Lesser, V., Raj Reddy, D., "The Hearsay-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty", *Computing Surveys*, v.12, n.2, June 1980
- 3 Fillmore, C.J., "The case for case", in Bach, Harris (Eds.), *Universals in Linguistic Theory*, Holt, Rinehart, and Winston, New York 1968, 1:90
- 4 Hayes, P.J., Carbonell, J.G., "Multi-Strategy Construction-Specific Parsing for Flexible Data Base Query and Update", *Proc. IJCAI-81*, Vancouver, August 1981, 432:439
- 5 Hayes, P.J., Hauptmann, A.G., Carbonell, J.G. - Tomita, M., "Parsing Spoken Language: a Semantic Caseframe Approach", *Proc. COLING-86*, Bonn, West Germany, 1986
- 6 Hayes-Roth, F., "Syntax, semantics, and pragmatics in speech understanding systems", in W.Lea (ed.), *Trends in speech Recognition*, Prentice-Hall, Englewood Cliffs (NJ), 1980
- 7 Hays D.G., "Dependency theory: a formalism and some observations", Memorandum RM4087 P.R., The Rand Corporation, 1964
- 8 Laface, P., Micca, G., Pieraccini, R., "Experimental results on a large lexicon access task", in *Proc. ICASSP-87*, Dallas, 1987
- 9 Lowerre, B., The HARPY Speech Recognition System. Computer Science Department, Carnegie-Mellon University, April, 1976
- 10 Sowa, J.F., *Conceptual Structures*, Addison-Wesley, Reading (MA), 1984
- 11 Tesniere, L., *Elements de syntaxe structurale*, 1st edition, Librairie C.Klincksieck, Paris, 1959
- 12 Woods, W.A., "Optimal Search Strategies for Speech Understanding Control", *Artificial Intelligence*, v.18, 1982, 295:326