

A FLEXIBLE NATURAL LANGUAGE PARSER
BASED ON A TWO-LEVEL REPRESENTATION OF SYNTAX

Leonardo Lesmo and Pietro Torasso

Istituto di Scienze dell'Informazione
Università di Torino
C.so Massimo D'Azeglio 42 - 10125 TORINO - ITALY

ABSTRACT

In this paper we present a parser which allows to make explicit the interconnections between syntax and semantics, to analyze the sentences in a quasi-deterministic fashion and, in many cases, to identify the roles of the various constituents even if the sentence is ill-formed. The main feature of the approach on which the parser is based consists in a two-level representation of the syntactic knowledge: a first set of rules emits hypotheses about the constituents of the sentence and their functional role and another set of rules verifies whether a hypothesis satisfies the constraints about the well-formedness of sentences. However, the application of the second set of rules is delayed until the semantic knowledge confirms the acceptability of the hypothesis. If the semantics reject it, a new hypothesis is obtained by applying a simple and relatively unexpensive "natural" modification; a set of these modifications is predefined and only when none of them is applicable a real backup is performed: in most cases this situation corresponds to a case where people would normally garden path.

INTRODUCTION

The problem of performing an accurate syntactic analysis of Natural Language sentences is still challenging for A.I. people working in the field of N.L. interpretation (Charniak 81, Kaplan 82). The most relevant points which attracted attention recently are:

- the need of a strong connection between syntactic processing and semantic interpretation in order to reduce the space of the alternative syntactic analyses (Konolige 80, Sidner et al. 81, Milne 82)
- the convenience of a quasi-deterministic syntactic analysis, in order to reduce the computational overhead associated with a heavy use of backup (Marcus 80)
- the convenience of an approach which tolerates also (partially) incorrect sentences, at least when it is possible to obtain a meaningful inter-

pretation (Weischedel & Black 80, Kwasny & Sondheimer 81, Hayes 81).

The first two of these remarks guided the design and the implementation of a system devoted to the interpretation of N.L. (Italian) commands (Lesmo, Magnani & Torasso 81a and 81b). In that system, however, as in most N.L. interpreters, the analysis of the input sentence is mainly syntax-driven; for this reason, just in case the input sentence respects the constraints imposed by the syntactic knowledge it can be interpreted.

The problem of analyzing ill-formed sentences has received a great deal of attention recently. However, most studies (Weischedel & Black 80, Kwasny & Sondheimer 81) are based on standard syntactic analyzers (A.T.N.) which have been further augmented in order to take into account sentences lacking some required constituents (ellipsis) or where some syntactic constraints are not respected (e.g. agreement in number between the subject and the verb).

There are two problems with this approach; both of them depend on the choice of having a syntax based analysis. The first problem is the necessity of extending the grammar; of course, it is necessary, in general, to specify what is grammatical and what is not, but it would be useful that this specification does not interfere too heavily in the interpretation of the sentence. In fact, if all deviations would have to be accounted for in the grammar, an unforeseen structure would block the analysis, even if the sentence can be considered as understandable. Consider, for instance, the following sentence:

Mary drove the car and John the truck (S1)

The absence of the verb in the second clause can be considered an acceptable form of ellipsis and, consequently, the sentence can be interpreted correctly. On the other hand, it is very unlikely that an extension of the grammar would cover the following ungrammatical (see Winograd 83, pag.480) sentence:

The book that for John to read would be difficult is beautiful (S2)

However, even if some efforts are required, this sentence can be considered as understandable. As stated above, a comprehensive system must be able to detect the ungrammaticality of S2, but this detection should not prevent the construction of a structure to pass to the semantic analyzer. Moreover, it seems that a subtle grammaticality test of this kind is easier to make (and to express) on a structured representation of the sentence (e.g. a tree) than on the input sentence as such.

The second problem which must be faced when an ATN is extended to handle ill-formed sentences is the one of word ordering. ATNs are powerful formal tools able to analyze type-0 languages; in the theory of formal languages a language is defined as a set of strings; for this reason ATNs must recognize "ordered sequences" of symbols (or words). Of course also the natural languages have fixed rules which define the admissible orderings of words and constituents, but, if those constraints have to be relaxed to accept ill-formed inputs, some extensions, which are less straightforward than the ones used for handling the absence of a constituent are needed. For example, the sentence

Ate the apple John (S3)

is ungrammatical, easily understandable, but seems to require in an ATN the extension of the S net to allow to traverse the constituents in a different (even if syntactically wrong) order. Also in this case it seems that the construction of a structured representation of the sentence could be the first step of the analysis; when it is done, the ordering constraints can easily be verified and, in case they are not respected either an alternative analysis is tried or, as in the case of S3, the sentence is passed to the semantic analyzer and, possibly, the parser signals the presence of a syntactic error.

In this paper we present a parser which allows to make explicit the interconnections between syntax and semantics, to analyze the sentences in a quasi-deterministic fashion and, in many cases, to identify the roles of the various constituents even if the sentence is ill-formed.

The main feature of the approach on which the parser is based consists in the two-level representation of the syntactic knowledge: a first set of rules emits hypotheses about the constituents of the sentences and their functional role and another set of rules verifies whether a hypothesis satisfies the constraints about the well-formedness of sentences. However, the application of the second set of rules is delayed until the semantic knowledge confirms the acceptability of the hypothesis. If the semantics reject the current hypo

thesis, an alternative one is tested: this control structure guarantees that all hypotheses which satisfy the weak syntactic constraints (which govern the emission of hypotheses) and the semantic constraints are tried before considering the input sentence as uninterpretable.

The claim that the parser operates in a quasi-deterministic fashion is justified by the kind of processing that the system performs when a hypothesis is rejected: in most cases a new hypothesis is obtained by applying a simple and relatively unexpensive "natural" modification; a set of these modifications is predefined and only when none of them is applicable a real backup is performed: in most cases this situation corresponds to a case where people would normally garden path.

The decision of paying particular attention to the problem of analyzing ill-formed sentences is motivated by the intended application of the parser. In fact it is included in a larger system, which allows the user to interact in natural language with a relational data base (Siklossy, Lesmo & Torasso 83, Lesmo, Siklossy & Torasso 83). Various systems have been developed in the last years, which act as N.L. interfaces to data bases (Harris 77, Waltz 78, Konolige 80) and all of them pointed out the necessity of having at disposal mechanisms for handling ill-formed inputs (mainly ellipsis).

In the following some example sentences will be discussed; they refer both to the implemented system and to more general sentences. This is justified, because the linguistic coverage of the parser is wider than the one required by a data base interface, even if the data base, the semantic knowledge and the lexicon are restricted to a particular domain.

AN EXAMPLE OF THE PARSER'S RESULT

Before describing the parser control structure, it is worth having a look at the final representation of the input sentence which is produced by the parser. It consists in a tree which represents the relationships existing among the constituents of the input sentence according to the "head and modifier" approach (Winograd 83, pag.73)^o. An example of such a tree is reported in fig.1.

It may be noticed that the tree is a case representation of the sentence: in the verbal nodes

^o This structure might be related to the "syntactic/semantic shape representation of RUS (Sidner et al. 81), but we are not sure.

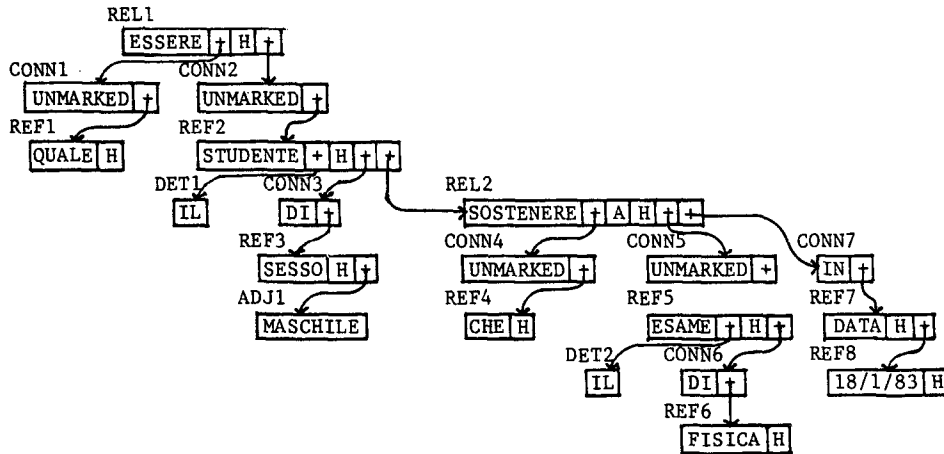


Fig.1 - Result of the analysis of the sentence:

"Quali sono gli studenti di sesso maschile che hanno sostenuto l'esame di Fisica in data 18/1/83?"
 (Who are the students of male sex who passed the test of Physics on 18/1/83?).

HEAD	Root of the verb
TENSE	Present, Past, Future
MODE	Indicative, Participle ...
FORM	Active, Passive
GENDER	M, F
NUMBER	Singular, Plural
PERSON	1, 2, 3
AUX	Yes, No
MOOD	Declarative, Interrogative ...
DEPEND	Main, Relative ...
TYPE	REL
LINKUP	a pointer
ROLE 1	
ROLE 2	
...	...
ROLE n	
TRANSL	a translation

(a)

ROLETYPE	POINTER	SPECIAL	SYNTFUN
----------	---------	---------	---------

(b)

Fig.2 - Prototypical structure of the REL nodes.

All the slots appearing in fig.2a are atomic and their possible contents are exemplified in the slot (LINKUP is the upward pointer which enables to traverse the tree bottom-up; this link is not depicted in fig.1); the only exception are the ROLES, which correspond to the links shown in fig.1 and whose structure is shown in fig.2b. For the meaning of the different fields refer to the example of fig.3. The TRANSL slot refers to the interpretation (in terms of data base operations) of the constituent headed by the node (see explanations in the text).

HEAD	SOSTENERE		
TENSE	Present Past		
MODE	Indicative		
FORM	Active		
GENDER	Any		
NUMBER	Plural		
PERSON	3		
AUX	No		
MOOD	Declarative		
DEPEND	Relative		
TYPE	REL		
LINKUP	REF2		
ROLES	CASE	CONN4	RELPRON
	AUX	/	/
	H	/	/
	CASE	CONN5	NIL
	CASE	CONN7	NIL
TRANSL	(select &pass		
	((\$course eq Fisica) (\$date eq 18/1/83))		

Fig.3 - Actual contents of the node REL2 (SOSTENERE)

of fig.1. Five ROLES appear in this instance of REL. In the first, fourth and fifth ROLE the ROLETYPE is "CASE", because they refer to actual cases of the verb; the syntactic function of each case is reported in the fourth field (SYNTFUN). The second and third ROLE have the only function of marking the position in the sentence of the auxiliary (hanno - have) and of the verbal head (sostenuto - passed). The SPECIAL field is used to mark cases filled by interrogatives, reflexive pronouns, etc. (RELPRON means RELATIVE PRONoun). Notice that the AUX slot is used to signal the fact that the head of the verb is (or is not) an auxiliary.

REL	Relation	Verbs, copulas
REF	Referent	Nouns, pronouns
CONN	Connector	Prepositions, conjunctions
DET	Determiner	Articles, demonstrative adjectives, adjectival question words
MOD	Adverbial Modifier	Adverbs
ADJ	Adjectival Modifier	Adjectives

Table 1 - The node types: the first column contains the name (actual and extended); the second one contains the classical syntactic categories associated with the node type.

(RELation) each pointer corresponds to a syntactic case associated with the verb; in the REF nodes, which roughly correspond to nouns and pronouns, the dependent structures represent the specifications of the node. The H field indicates the position of the constituent's head (i.e. the verb or noun) in the surface sentence and the A fields are used in the REL nodes to indicate the position of the possible auxiliaries. The actual structure of the nodes appearing in the figure is much more complex; for example, the prototype description of the REL nodes is reported in fig.2. In fig.3 the actual structure of the node REL2 (SOSTENERE) is reported. A number of remarks are required:

- when a REL node is instantiated it does not contain any ROLE slot. Whereas the other slots are "filled" when the needed piece of information is available (normally this happens when the head of the verb is scanned), the ROLE slots are dynamically created when a given constituent is attached to the REL node (with the exception of AUX and H);
- some slots are redundant, since their contents can be deduced by traversing the tree. For example, the contents of the slot DEPEND and of the field SPECIAL of the ROLE slot can be obtained on the basis of the LINKUP node and of the first case of the clause respectively. They have been included for the sake of efficiency;
- the sole input word of the example sentence which does not appear in a node of fig.1 is the auxiliary "hanno". Auxiliaries have been considered as components of the verb, so that their presence is signalled only by means of an AUX role. The actual auxiliary, its tense, its number, etc. are deducible from the contents of the other slots of the REL node.

The different types of nodes which have been defined are listed in Table 1.

As stated in the introduction, the system should act as a natural language front-end for a

relational data base. The structure reported in fig.1 is the basis for performing the semantic checks and for translating the sentence in a relational algebra expression (Date 81) which corresponds to the input query. As will be described in the following sections, neither the semantic checks nor the actual translation of the query are done at the end of the syntactic analysis; in fact the semantic checks are performed when a node is filled with a content word and the translation is obtained in an incremental way from the constituents occurring in the tree. For instance, the semantic check procedures will be triggered when the word "sesso" (sex) is encountered and the corresponding REF node is created, linked and filled to verify that the students have a sex (or, more precisely, that the sequence "studente di sesso" is acceptable).

As regards the translation, it is worth noticing that it does not represent the interpretation of the given node, but the data base interpretation of the whole constituent headed by that node; for this reason it is obtained by combining the translations of all depending constituents. Let us consider, for example, the node REF2. The translation associated with CONN3 is

```
(join %student
  (select &sex (($sex eq m))
    (%student eq %person))
```

The translation associated with REL2 is

```
(select &pass (($course eq Fisica)
  (%date eq 18/1/83)))
```

The resulting translation associated with REF2 is

```
(join (join %student
  (select &sex (($sex eq m))
    (%student eq %person))
  (select &pass (($course eq Fisica)
    (%date eq 18/1/83)))
  (%student eq %student))
```

A detailed description of the way this translation is obtained is reported in (Lesmo, Siklossy, Toraso 83). However, for the sake of clarity it is important to say that %student is the unary relation whose unique attribute is \$student and which contains the names of all the students whose data are stored in the data base; &sex is a binary relation (attributes \$person and \$sex) containing the sex of all the persons known to the system; finally &pass is the relation (attributes \$student, \$course, \$grade, \$date) where are stored the results of the tests passed by the students. The translation which have been shown are stored in the TRANSL slot of the associated nodes.

THE CONSTRUCTION PROCESS

The tree described in the previous section is built by means of a set of rules of the form condition-action. With each syntactic category a subset of these rules is associated: when an input word of the given category is encountered in the input sentence, then the subset of rules associated with that category is activated and the conditions are evaluated. The conditions involve tests on the current structure of the tree (i.e. the "status" of the analysis) and may request a one-word lookahead. If just one rule is selected (i.e. all other conditions evaluate to false), its action part is executed. An action consists in the construction of new nodes, in their filling up with particular values (normally depending on the input word) and in their attachment to the already existing tree. In table 2 are reported as an example some of the rules of the packet associated with the category ADJECTIVE. The rules which are not reported handle the cases of predicative adjectives and adjectives preceded by adverbs. In some of the rules a one-word lookahead is used; it allows the parser to build the right structure in virtually all simple cases. In fact, even if the semantic knowledge source does not affect the choice of the rule, it can trigger the natural changes, which modify the tree; these changes substitute the backup in many of the cases where the hypothesized syntactic structure does not satisfy the semantic constraints.

An example of a sentence portion which can be disambiguated only by inspecting the semantic constraints is the following:

... - Determiner - Noun - Adjective - Noun - ...

In this case the adjective may modify either the preceding or the following noun. Consider the sentences S4 and S5^o:

Per le persone anziane bevande ghiacciate (S4)
sono dannose
(For old people icy-cold drinks are harmful)

Si arrampicano sulle montagne agili (S5)
scalatori
(Agile cragsmen scramble up the mountains)

The strategy adopted by the parser is to attach the node representing the adjective to a newly created REF node which will be filled when the second noun is analyzed (see the action part of Rule 4 in tab. 2). In case the semantics reject this choice (sentence S4) a natural change is triggered; it disconnects the adjectival node and moves it back to the REF node which represents the first noun.

^o The sequence of categories given in the text corresponds to "... le persone anziane bevande ..." in S4 and to "... le montagne agili scalatori ..." in S5.

RULE 1	COND: CURRENT CONN ACTION: CRLINK REF CONN CRLINK ADJ REF FILL ADJ *
RULE 2	COND: UNFILLED REF <u>or</u> (CURFILL ADJ <u>and</u> NEXT ≠ NOUN) ACTION: CRLINK ADJ REF FILL ADJ *
RULE 4	COND: (CURFILL REF <u>or</u> CURRENT NIL <u>or</u> CURRENT REL) <u>and</u> NEXT = NOUN ACTION: CRLINK CONN REL FILL CONN 'UNMARKED CRLINK REF CONN CRLINK ADJ REF FILL ADJ *

Table 2 - Some of the rules associated with the syntactic category ADJECTIVE.

The predicates used in the conditions are CURRENT X: TRUE if the current node is of type X.

UNFILLED X: TRUE if the current node or the node above is of type X and it is not filled yet.

CURFILL X: TRUE if the current node is of type X and is filled.

NEXT CAT: is a lookahead function which returns TRUE if the category of the next word in the input string is CAT.

The structure-building functions used in the actions are

CRLINK X1 X2: creates a new node of type X1 and links it to a node of type X2.

The node which must be used is located by moving up on the rightmost branch of the tree.

FILL X VAL: a node of type X (located as in CRLINK) is filled with the value VAL (* denotes the normalized form of the current word).

In general, however, it is not possible to avoid the use of backup. The backup mechanism is needed when more than one of the conditions of the rules associated with a particular category is matched, but this case is actually restricted to very complex (and unusual) relative clauses. More often, the backup is required when the input word is ambiguous, i.e. it belongs to more than one syntactic categories. In this case all conditions associated with the different categories are evaluated and in some cases more than one of them is matched. In all these cases the status of the analysis is saved (i.e. the current tree) together with the identifiers of the matched rules and a pointer to the input sentence.

As an example of sentences in which the back

up mechanism is used consider the sentences S6-S8; in them there is a lexical ambiguity for the word "che" (it acts as a relative pronoun in S6, as a conjunction in S7 and as an adjectival modifier in S8); moreover in S6 and S7 "pesca" is a form of the verb "pescare" (to fish) whereas in S8 it is a noun (the fishing).

- Dì a quel ragazzo che pesca di andarsene (S6)
 (Tell that boy who is fishing to go away)
- Dì a quel ragazzo che pesca male (S7)
 (Tell that boy that he is fishing badly)
- Dì a quel ragazzo che pesca fantastica
 hai fatto (Tell that boy what a marvelous
 fishing you have done). (S8)

THE VERIFICATION PROCESS

When a node is filled, it is supposed to be already attached to the tree. The filling operation triggers some procedures associated with the type of the node which is being filled. Among them, the AGREEMENT procedures have the task of checking person, number and gender agreement between a node and its dependants. Particularly important is the agreement procedure associated with the REL node type, because it selects the REF node which can act as syntactic subject of the sentence (this suggestion may be overcome later by virtue of semantic considerations). If the agreement constraints are violated, then the natural changes are attempted; if no restructuring of the tree is successful, then the initial status is maintained without changes and a warning message is issued.

Perhaps, among the procedures triggered by the filling of a node, the one which have the most dramatic effects on the subsequent behavior of the system is the semantic check procedure. In fact, if the outcome of the semantic check procedure reports the non-admissibility of an attachment, the parser is forced to find another alternative. This is done by first applying the natural changes and then, if all of them fail, by performing a backup. A semantic procedure refers to the semantic knowledge of the domain under consideration, which is stored in form of a two-level network (Lesmo, Siklossy & Torasso 83); the external level allows to perform the checks, whereas the internal level carries the information necessary to perform the translation.

Different checks are done depending on the type of the node. When an ADJ node is attached to a REF node, the system has to verify that the adjective is an acceptable linguistic description of the noun stored in the REF node. In case two REF nodes are attached (this case occurs in Italian only when the lower REF contains a proper noun) the system has to verify that the lower REF con-

tains a possible identifier of the class represented by the noun stored in the upper REF. When two REFs are attached via a CONN node, the constituent headed by the lower REF has the purpose either of specifying a subset of the class identified by the noun stored in the upper REF or to refer to a property of a given object. An example of the first kind is "the professors of the department X" and an example of the second kind is "the sex of the professors ...". In this case the semantic procedure accesses the net to reject incorrect specifications of the form "the sex of the department X". A quite different behavior characterizes the attachment of a role to a verb (a REF node to a REL node via a CONN node); of course, the attachment of a new case cannot trigger a simple case check, but must take into account also all the cases attached before. A side effect of this process is the binding of the actual cases to the cases predicted in the net; this can be useful when there are two cases which have the same marker (or which are both unmarked) to determine, by using the selectional restrictions stored in the net, the actual role of the filler of each case (e.g. syntactic subject or syntactic object).

The completion of a constituent triggers the last set of syntactic rules; they verify whether the constituent (that is the node itself and its descendants) respects the ordering constraints. In case those constraints are violated (e.g. "belli i bambini sono" - nice the babies are) a warning message is issued but the sentence is considered as interpretable.

A word is due to explain the meaning of the term "complete". The constituent headed by the node n_i is considered as complete when a new node n_j is attached to a node n_k which is an ancestor of n_i ; all constituents headed by the nodes belonging to the rightmost path of the tree are considered as complete when the system encounters the end of the sentence. The concept of "completion" of a constituent is particularly important because only when the constituent headed by the node n_i is complete the system translates the constituent by using different pieces of information gathered by the semantic procedures and stores the translation in the TRANSL slot of the node n_i .

NATURAL CHANGES VERSUS BACKUP

The natural changes have the purpose of restructuring the tree by moving around constituents without requiring backup. They are represented as pattern-action rules, where the pattern part is used to select the rules which can be applied, whereas the action part implements the transformation of the tree. The natural changes currently implemented are of two main types:

- MOVE UP (the easiest and most common): it at-

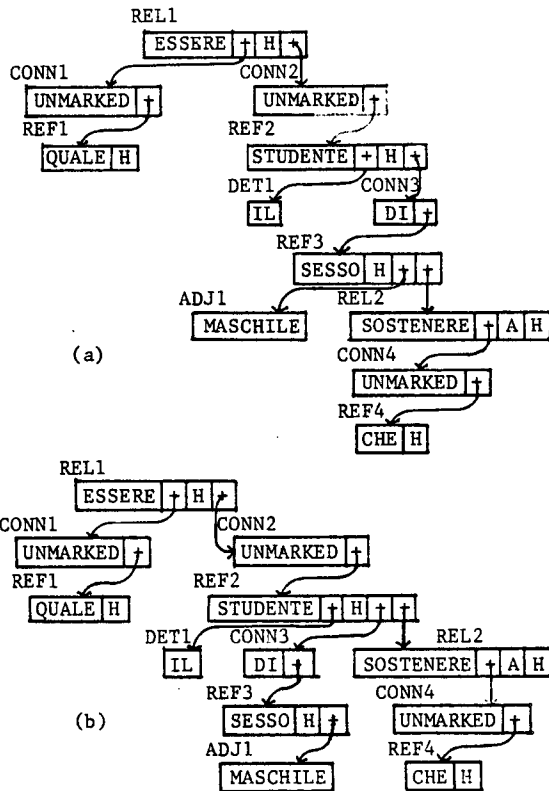


Fig.4 - Example of the use of a MOVE UP natural change. The semantic procedure associated with the REL node type detects that "sesso" cannot fill any of the cases of "sostenere" (a), so that the constituent headed by "sostenere" is MOVED UP to "studente" (b).

taches a constituent (i.e. a subtree) to a higher node (whose type is specified in the rule) of the current branch of the tree.

- MOVE BACK: it attaches a constituent to the right most leaf of the preceding branch of the tree. For example; a MOVE UP rule is used to build the tree shown in fig.1: the relative clause "che hanno sostenuto ..." is firstly attached to the nearest REF node ("sesso"); when the verb is found the node REL2 is filled (fig.4a), the agreement and semantic check procedures are triggered and this latter returns that "sesso" cannot fill an unmarked case of "sostenere", so that the partially built relative clause is moved up to REF2 ("studente" - fig.4b); this new hypothesis is validated by the agreement and semantic procedures. An example of the application of a MOVE BACK rule has been given in the third section, in connection with the problem of attaching the adjectival nodes (see fig.5).

As stated in the previous section, the natural changes do not substitute in all cases the backup mechanism; the backup is strictly connected with the concept of "garden path". PARSIFAL (Marcus 80)

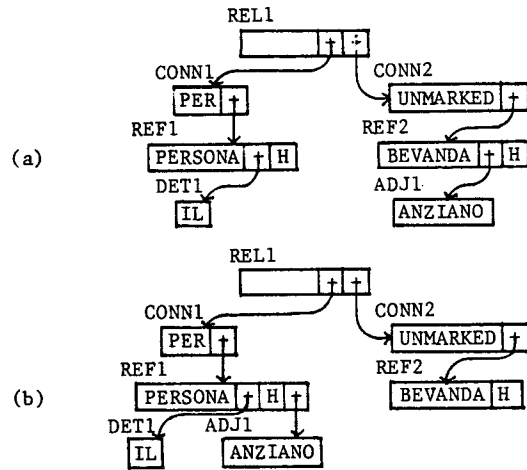


Fig.5 - Example of MOVE BACK natural change. When the word "bevande" (drinks) is scanned the node ADJ1 is MOVED BACK from REF2 (a) to the last REF node of the previous branch of the tree, i.e. REF1 (b).

is able to parse sentences in a deterministic way when they are not garden paths. However it has been shown (Milne 82) that:

- For a pair of potential garden path sentences, it is not possible to uniquely determine which is a garden path and which is not (different people may choose in different ways).
- The choice of having a n-constituent lookahead (as in PARSIFAL) does not allow to decide whether a sentence is a potential garden path in a psychologically plausible way.
- The semantic knowledge plays a fundamental role in choosing a particular analysis.

Milne argues that a one-word lookahead, with the substantial help of semantic information is what is needed to provide a model of N.L. which is psychologically sound (one-word lookahead plus semantics is also advocated in RUS - Brachman et al. - 79).

We think that the approach adopted in our parser basically agrees with this position. In a rather vague sense, the non-complete nodes of our tree correspond with the Active Node Stack, i.e. with the not yet completed constituents of the sentence. The natural changes allow to operate on these nodes on the basis of semantic information. However there is a fundamental difference: our parser has at disposal the whole structure built previously. An example of the possibility of using non-active constituents is given by the MOVE BACK natural changes where a previous constituent (already completed) is used to attach a node (see REF1 in fig.5). This greater flexibility has the disadvantage of not giving any cue for deciding a-priori what is a valid natural change and what is not (it is possible to devise natural changes for all possible kinds of restructuring of the tree); however, it allows to

choose heuristics which are in agreement with the actual behavior of humans and which fit in a simple way in the proposed model.

As regards the use of backup, the cited works do not give an account of what happens in the parser when an analysis fails due to a garden path (see, however, Marcus 80, pp.202-220). Our provisional solution is to use the backup, a computational tool heavier than the natural changes: it should correspond to the situation when "the user must consciously undo this previous choice after detecting an inconsistency" (woods 73, pag.133). We acknowledge the problems associated with this choice, e.g. the need of saving at some times the status of the analysis, the possibility of interference with the natural changes, etc., but the backup is used parsimoniously (due to the condition part of the syntactic rules) and, anyway, we do not believe it is the final solution to this problem.

CONCLUDING REMARKS

The paper describes a parser for a large subset of Italian. The novel control structure involves the use of natural changes which restructure the tree representing the status of the analysis without the intervention of the backup mechanism. This allows the system to operate in a pseudo-deterministic way, in that the use of backup is limited to sentences which could make people garden path.

Another major feature of the parser is its ability to cope with some kinds of ill-formedness of the input sentences. This is obtained by a decomposition of the syntactic knowledge into two levels: the first level contains structure building rules, whereas the second level contains rules of agreement and rules related with the ordering of constituents. This structuring of the syntactic knowledge allows the parser to be data driven: the scanning of a new input word produces its insertion into the analysis tree; this may be seen as an hypothesis of interpretation, which can be accepted or rejected later on the basis of other independent knowledge sources. This allows the system to avoid the use of classical rewriting rules or transition networks which represent in an integrated way all syntactic constraints.

As stated in the introduction, the authors are developing a N.L. interface to a relational data base. The lexical analyzer and the access procedures to the network representing the semantic constraints are running, the construction rules and the natural changes are being debugged, whereas the ordering rules are under development. The translation into the actual data base query is running. The system is written in FRANZ LISP and runs on a VAX 11/780 under the UNIX operating system.

REFERENCES

- Brachman R.J. et al.: Research in Natural Language Understanding. BBN Report no.4274 (1979).
- Charniak E.: Six Topics in Search of a Parser: An Overview of AI Language Research. Proc.7th IJCAI, Vancouver, B.C. (1981), 1079-1087.
- Date C.J.: An Introduction to Database Systems (3rd edition), Addison Wesley (1981).
- Harris L.R.: User-Oriented Data Base Query in the ROBOT Natural Language Query System. Int. J. of Man-Machine Studies 9 (1977), 697-713.
- Hayes P.J.: Multiple Strategies Construction Specific Parsing for a Flexible Database Access and Update. Proc.7th IJCAI, Vancouver(1981), 432-439.
- Kaplan S.J. (ed.): Special Issue on Natural Language Processing, SIGART Newsletter 79 (1982).
- Konolige K.G.: A Framework for a Portable Natural Language Interface to Databases. In D.Sagalowicz (ed.): Mechanical Intelligence: Research and Applications, Final Tech. Rep., SRI Int. (1980).
- Kwasny S.C., Sondheimer N.K.: Relaxation Techniques for Parsing Grammatically Ill-Formed Input in Natural Language Understanding Systems. AJCL 7 (1981), 99-108.
- Lesmo L., Magnani D., Torasso P.: A Deterministic Analyzer for the Interpretation of Natural Language Commands. Proc.7th IJCAI, Vancouver B.C. (1981a), 440-442.
- Lesmo L., Magnani D., Torasso P.: Lexical and Pragmatic Knowledge for Natural Language Analysis. Proc.IEEE Int.Conf. on Cybernetics and Society, Atlanta GA (1981b), 301-305.
- Lesmo L., Siklossy L., Torasso P.: A Two-Level Net for Integrating Selectional Restrictions and Semantic Knowledge. IEEE Int. Conf. on Cybernetics and Society, Bombay and New Delhi (Dec 1983).
- Marcus M.P.: A Theory of Syntactic Recognition for Natural Language. MIT Press, Cambridge MA (1980)
- Milne R.W.: Predicting Garden Path Sentences. Cognitive Science 6 (1982), 349-373.
- Sidner C.L. et al.: Research in Knowledge Representation for Natural Language Understanding. BBN Report no.4735 (1981).
- Siklossy L., Lesmo L., Torasso P.: Flexible Pragmatics for Database Oriented Query Answering. ISI Internal Report (1983).
- Waltz D.L.: An English Language Question-Answering System for a Large Relational Data Base. Comm. ACM 21 (1978), 526-539.
- Weischedel R.M., Black J.E.: Responding Intelligently to Unparsable Inputs. AJCL 6 (1980), 97-109.
- Winograd T.: Language as a Cognitive Process. Vol.1: Syntax. Addison Wesley (1983).
- Woods W.A.: An Experimental Parsing System for Transition Network Grammars. In R.Rustin (ed.): Natural Language Processing. Algorithmics Press, New York (1973), 111-154.