AUTOMATIC PROCESSING OF WRITTEN FRENCH LANGUAGE

J.L.Binot, M.Graitson, Ph.Lemaire, D.Ribbens
Computer Sciences Department
University of Liège
Belgium

Abstract

An automatic processor of written French language is described. This processor uses syntactic and semantic informations about words in order to construct a semantic net representing the meaning of the sentences. The structure of the network and the principles of the parser are explained. An application to the processing of the medical records is then discussed.

1. Introduction

SABA ("Semantic Analyser , Backward Approach") is an automatic parser of French language currently developped at Liège University, Belgium[1]. It is now aimed at the processing of medical records[2]. However, the principles of this system were conceived independently of any specific application. SABA has been fundamentally conceived as a general, flexible French language parser, which could be used as a component of a natural language interface between a human user and a given computer process[8]. This parser is not limited to the processing of correct, academic French. It is aimed also at processing the casual language of an average user.

Though our system is uniquely concerned with French, we have translated our examples in English everytime that it was possible. In this way, we hope that the non French-speaking reader might be able to get the flavour of our work.

2. General description of the system

SABA, as a parsing system, is essentially semantically oriented. Its goal is not to identify the complete syntactic structure of the input sentences, but rather to determine the possible semantic relationships between the terms of these sentences. More specifically, the system tries to characterize the semantic dependencies that appear in a sentence between the complements and the terms which are completed by them (from now on, a term of this last kind will be called a "completee"). We will insist immediately upon the fact that both concepts of"complement" and of "completee" are to be taken in a general way. The syntactic subject of a verb is thus treated as a complement of this verb.

To characterize these semantic dependencies, the system uses a small set of relationships like AGENT, OBJECT, INSTRUMENT, LOCUS, and so on. In this way, our system is related to the family of "case systems", using the now well known principles of case grammars[3 14]. However, in contrast to some authors[3 15 17 18], we don't try to find a complete and minimal set of universal relationships. The only criterion for the choice of our relationships is their practical usefulness. For the time being, about twenty different relationships are used by the system.

All the relationships which are identified in an input sentence are summarized in a semantic network, which represents the semantic structure of this sentence. The (simplified) representation of a complete sentence may be illustrated by the figure 1. The fundamental principles of the network will be described in the next section.

The grammar used by the system has two components, syntactic and semantic, which are used interactively. The syntactic component has two main tasks. First, it segments the sentence into syntactic units. Second, it defines an order of processing for all these units. This syntactic component, which is implemented in a procedural way, will be described in section 5.

The semantic component defines which semantic relationships are acceptable between terms. As we shall see later, its scope is not only the relationships between verbs and nominal groups, but also the dependencies between nouns, between nouns and adjectives, and, in fact, all possible dependencies expressible in the French language. The semantic component will be described in section 4.

3. A semantic net

Since a few years, semantic nets are well known as tools for expressing knowledge and meaning[13 16 17]. Let us recall briefly the principle of such networks:
a semantic net is a set of nodes, which may represent the different significant terms of a text or of a domain of knowledge. These nodes are interconnected by labelled arcs, which represent the semantic relationships established between them.

A complete semantic network, which must be able to express a great variety of semantic informations, is generally a very complex structure[7 9 17]. The structure that we use may be somewhat simpler, because it is aimed only at the representation of sentences, and not of general knowledge (at least at this state of our work). However it is still fairly complex, as it can be seen in figure 1.

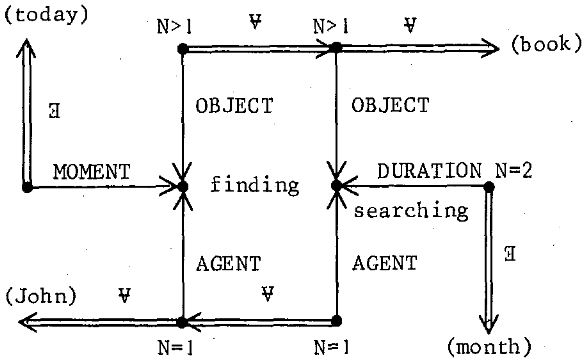


Figure 1. Representation of : "John has found today the books that he was searching for since two months.

We will not try here to discuss all the subtleties of our net structure. Rather, we will restrict ourselves to the statement of a few basic principles. All these principles can be explained with the help of the very simple example of the figure 2.

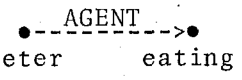


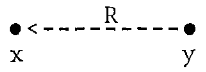
Figure 2. Representation of : "Peter eats".

First of all, in our terminology, verbs are not treated as predicates, (the arguments of which being the different nouns of the sentence), but rather as arguments themselves. We have abandoned the dominant point of view that the verbs express mainly relationships, while the others terms express objects or properties. Instead, we admit that a sentence is composed of content words, which we call "semantic entities", related by semantic relationships. The semantic entities include not only the nouns, but also the verbs, adjectives, adverbs, and some prepositions.

Secondly, the semantic relationships are oriented (the positive orientation being denoted in the network by an arrow). By definition, the positive orientation is such that :

- the origin of the arc is the node corresponding to the term which appears in the sentence as the complement, and
- the extremity of the arc is the node corresponding to the term which appears in the sentence as the completee.

Third, a logical interpretation corresponds to the semantic net. We will admit that to the graphic configuration



corresponds the logical proposition :

$$R(x,y) = \text{True}$$

We will remark that the relation R is not symmetrical with respect to its arguments : the first argument corresponds to the destination node of the network representation.

4. A semantic grammar

The task of the semantic component of the grammar is to define which semantic relationships are acceptable between the semantic entities. In order to do that, we shall use semantic knowledge about words. To each content word are assigned two different "property lists" : an "E-list" and a "T-list".

E-lists.

The E-list which is associated with one term lists all the relationships where this term may appear as a "completee". As an alternate definition, we may say that an E-list lists all possible kinds of complements for the associated term. For example, the E-list of the verb "to eat"would be something like that :

(eat(E-list(AGENT OBJECT INSTRUMENT LOCUS TIME)))

E-lists appear to be very similar to the traditional case frames used by the case grammar theory. There exists however a distinction : case frames were meaned to indicate possible ARGUMENTS for verbs, considered as predicates. E-lists are used to indicate possible RELATIONSHIPS for the associated terms, which are considered as arguments.

The E-list associated to a term is a characteristic of this term itself, and cannot be deduced from the context. It must be given by a dictionary.

T-lists.

The T-list which is associated to a term lists the possible relationships where this term may appear as a complement. We may also understand a T-list as the list of the possible kinds of "completee" of a term. In contrast to the E-list, the T-list of a term is, at least partially, bound by the context of this term in a sentence. The T-list of a noun, for example, is provided by the preposition which begins the nominal group. Each preposition introduces thus a given, fixed T-list. And, to preserve the generality of this rule, the lack of preposition is assimilated to the presence of an "empty" preposition, called PHI.

For example, the T-list introduced by the French preposition "par" is something like this :

(par(T-list(AGENT,PATH,ORIGIN,DESTINATION,QUANTITY)))

Of course, we do not consider that the lists given as examples are complete. They are only an illustration of the real configuration of the system.

Some properties of T-lists and E-lists.

a) From a logical point of view, the occurrence of a relationship, say AGENT, in the E-list associated with a given term X is equivalent to the following proposition :

$$(\exists y)\mathrm{AGENT}(X,y)$$

The occurrence of the same relationship in the T-list associated to X is equivalent to :

$$(\exists y)\mathrm{AGENT}(y,X)$$

Consequently, the only difference between T-lists and E-lists lies in the orientation given to the relationships described by them.

b) For any relationship, such as AGENT, we may define the "inverse relationship" AGENT^{-1}, such that :

$$\mathrm{AGENT}(x,y) \equiv \mathrm{AGENT}^{-1}(y,x)$$

Given these inverse relationships, we have the following property of E-lists and T-lists :

"The occurrence of a given relationship in the E-list associated with a term X is equivalent to the occurrence of the inverse relationship in the T-list associated to the same term, and reciprocally".

This property is used in some complex situations where a term which appears in the input sentence as a complement must be represented in the network as a completee. This is the case, for example, of past and present participles used as adjectives.

c) The same relationship may not occur twice in a given list of properties (E-list or T-list). Concerning E-lists, this restriction may be translated as : "two different terms cannot play in the same sentence the same role with respect to a given term",
which is a typical restriction in some case systems[3 15].

d) Only one of the relationships listed in the T-list of a term may be used in a given sentence. This means that each term in a sentence has a single role to play. This condition is not true for E-lists : all the relationships given by the E-list of a term may be used in a sentence where this term occurs.

The properties c) and d) are called the two "exclusivity principles" of the system.

Compatibility condition and selectional restrictions

We will now show how we will use these property lists in our system. First we will state a compatibility condition : "given two terms, one of which is a possible complement of the second, a necessary condition to establish a given relationship between them is that this relationship be present both in the E-list of the possible completee and in the T-list of the possible complement".

This condition is a necessary but not a sufficient one. The reason of this can be shown in the following example :

Let us admit that we want to establish the AGENT relationship between "eating" and "Peter" in "Peter eats". We must of course know that

$$(\exists y)\mathrm{AGENT}(\mathrm{EATING},y) \text{ and } (\exists x)\mathrm{AGENT}(x,\mathrm{PETER}),$$

i.e. that the act of eating takes an AGENT, and that Peter may be the AGENT of some activity. But, in order to be allowed to state

$$\mathrm{AGENT}(\mathrm{EATING},\mathrm{PETER}),$$

we must also know whether the two assignments

$$x=\mathrm{EATING}$$
$$y=\mathrm{PETER}$$

are correct.

These assignments will be submitted to a set of restrictions. These restrictions are associated with the property lists of the terms. Restrictions concerning the complements of a term are associated with the E-list of this term. Restrictions concerning the completee of a term are associated with its T-list.

The system uses different kinds of restrictions, in order to solve different kinds of ambiguities. The main one, which concerns nouns (and adjectives), uses a classification of these terms into a hierarchized set of semantic classes. With the help of this classification, we can for example express that "the AGENT of the action of eating must be an Animate being", which is denoted in the E-list of "eating" as :

(eating(E-list(AGENT(Animate),...)))

Here is a more complete example. Given the classification

Peter ∈ Human being
Human being ⊂ Animate being
Apple ∈ Fruit
Fruit ⊂ Comestible object
Knife ∈ Instrument

and the property lists

```
(PHI(T-list(AGENT,OBJECT,INSTRUMENT,.)))
(WITH(T-list(INSTRUMENT,...)))
(EATING(E-list(AGENT(Animate),OBJECT
 (Comestible),INSTRUMENT(Instrument))))
```

the system can easily parse the sentence

"Peter eats an apple with a knife"

and produce the three relationships

AGENT(eating,Peter)
OBJECT(eating,apple)
INSTRUMENT(eating,knive)

Other kinds of restrictions are based on syntactic classes or on modal properties (of verbs). We shall not discuss them further here.

5. The parser

The grammar, and thus the parser, are not completely free of syntactic considerations. The syntactic part of the parser has two main tasks :
- the segmentation of the input text into "syntactic units"
- the determination of a parsing strategy.

Syntactic units.

Four kinds of syntactic units are defined : words, groups, clauses and sentences.

- Words, or atomic symbols are strings of characters which match dictionary entries (this definition takes into account not only single words, but also locutions, as "at least", that will be treated by the system as atomic symbols). To each Word are associated syntactic and semantic properties. Words are divided into two main classes :
  - the semantic entities, or content words, which can be arguments of semantic relationships : nouns, pronouns, verbs, adverbs, adjectives, and some prepositions.
  - the function words, which cannot be arguments of semantic relationships : coordinate and subordinate conjunctions, articles, some prepositions, negation words, and so on.
- Groups are sequences of Words. Each Group has a central term, which may be a noun (nominal Groups), a pronoun, an adjective or an adverb.
- A Clause consists of one and only one verb, and of a set of Groups. A Clause has also a central term, which is its verb.
- A Sentence is a sequence of Words delimited by a "terminator" (punctuation mark like a period, a question-mark,...) A Sentence contains a set of Clauses.

The parsing strategy.

The parsing strategy is fundamentally a bottom-up strategy, which may be defined recursively as follows :

For each syntactic unit (except Words), execute the following steps :
- the segmentation of this unit into its own internal syntactic units,
- the parsing of these internal units according to a definite order,
- the determination of the semantic relationships between the internal units,
- the substitution of the given unit, at the next higher level, by a special symbol, which represents the analyzed unit.

The semantic relationships are determined according to the semantic grammar defined above. We want now to insist on the two other crucial points of this algorithm : the segmentation of a given unit, and the order for parsing the internal units.

The segmentation procedures has two tasks : breaking down sentences into clauses, and clauses into groups.

The segmentation of sentences into clauses is based on the following technics : starting at a verb, and moving one word at a time to the left AND to the right until a delimiter is recognized. For groups, the same technics applies, except that a group is never extended to the right of its main term. Lists of clause-delimiters and of group-delimiters are known by the system. Coordinate conjunctions, which can or cannot be delimiters depending on the context, do receive a special treatment.

An important point concerning the segmentation of the sentence into clauses must be stressed. It is performed each time that a clause must be selected to be analyzed by the system. This strategy gives to the system the possibility to use informations collected in a prior state of the analysis. In this way, the structure of very complex sentences can be successfully analyzed.

All segmentation procedures are already

implemented and function satisfyingly. An example of segmentation of a French sentence is shown in the figure 3. The sentence appears as a list of words, delimited on the left and on the right by the special symbol SB (Sentence Boundary). A syntactic category is assigned to each word. The results shown in the figure are a simplification of the output of the system.

a) Les chiens auxquels vous vous attachez et qui vous rendent de l'affection deviennent d'inestimables compagnons.

b) The dogs that you love and who love you in return become precious comrades

c) ((SB)(LES ART)(CHIENS NOM)(AUXQUELS PR-REL)(VOUS PR-PERS NIL(OD OI PR S)) (VOUS PR-PERS NIL(OD OI S))(ATTACHEZ VERBE IND)(ET CC)(QUI PR-REL)(VOUS PR-PERS NIL(OD OI PR S))(RENDENT VERBE IND)(DE PREP)(L ART)(AFFECTION NOM) (DEVIENNENT VERBE IND)(D ART)(INESTIMABLES ADJ)(COMPAGNONS NOM)(SB))

d) ((SB)(LES ART)(CHIENS NOM)(PR)(ET CC) (QUI PR-REL)(VOUS PR-PERS NIL(OD OI PR S))(RENDENT VERBE IND)(DE PREP)(L ART) (AFFECTION NOM)(DEVIENNENT VERBE IN) (D ART)(INESTIMABLES ADJ)(COMPAGNONS NOM)(SB))

e) ((SB)(LES ART)(CHIENS NOM)(PR)(ET CC) (PR)(DEVIENNENT VERBE IND)(D PREP) (INESTIMABLES ADJ)(COMPAGNONS NOM)(SB))

f) ((SB)(PP)(SB))

Figure 3 : segmentation of a sentence

a) the original French sentence
b) the English translation
c) the input of the segmentation procedure
d) the state of the sentence after the analysis of the relative clause "auxquels vous vous attachez", which is replaced by the special symbol PR
e) the state of the sentence after the analysis of the relative clause "qui vous rendent de l'affection", which is replaced by PR
f) final state of the sentence : the main clause was found and replaced by PP

Concerning the order for parsing the internal units at a given level, two strategies are applied, one for clauses, and one for groups.
For clauses, we simply follow the bottom-up strategy, with the following rule : all subordinate clauses (relative, conjunctive, infinitive,...) are processed before the clauses on which they depend. If two clauses are on the same level, a left to right priority is applied.
For groups, a backward strategy is applied : the system always starts from the end of the clause, and moves towards the beginning. At each step, the internal structure of a group is parsed, AND THEN THE POSSIBLE RELATIONSHIPS BETWEEN THIS GROUP AND THE FOLLOWING GROUPS (ALREADY PARSED) ARE INVESTIGATED. This particular order (after which the system is named) has a crucial importance. It is based on two facts :
- the first is related to the structure of the language. In French, complements are nearly always at the right of the terms on which they depend;
- the second is related to the system : we know that the T-lists of the semantic entities are, at least partially, deduced from the context. Consequently, at the moment when the system investigates the potential relationships between a term and some possible complement, the group in which this complement appears must have already been parsed !

6. Conclusions

We have presented a parsing system for French language sentences. This system is characterized by
- its generality,
- a good amount of flexibility, due to the fact that the system is semantically oriented,
- its capability to cope with complex structures, including subordinate clauses, conjunctions and references (these last two features were not discussed in the paper).

The system was designed independently of any specific application and was tested on a limited corpus (approximatively one hundred) of common french sentences. We believe that this system is applicable in all domains, provided a structured semantic dictionary. An application of our system to the medical domain is currently under development[2 6]. In this domain, existing automatic processing systems do function successfully for English and French pathology data[4 5 10 11 12]. We took up the challenge to design a system for processing patient discharge summaries in internal medicine.

It is quite true that the natural language of internal medicine data does not always contain well formed sentences. This is not however a real problem. One of the main advantages of the system stems from the fact that it was designed to handle French free text, no matter how academically correct or incorrect it might be. Consequently, we expect that the system can handle all kinds of medical diagnoses. Moreover, since it has been conceived to process questions as well as complete sentences, the system is not limited to the processing of medical data. Ultimately, it will be used to implement a complete natural language interface for a data base management system.

References

1. Binot,J.L.,Lemaire,Ph., & Ribbens,D., *Description d'un système automatique d'analyse du français.* Rapport interne, Université de Liège, 1979.

2. Binot,J.L., Lemaire,Ph., & Ribbens,D. *Analyse automatique de textes médicaux et codification des diagnostics.* Rapport interne, Université de Liège, 1980.

3. Fillmore,C.J., *The case for case,* in Bach,E. & Harms,R. (Eds.), Universals in Linguistic Theory, Holt, Rinehart & Winston,Inc., N.Y., 1968.

4. Graitson,M., *Identification et transformation automatique des morphèmes terminaux dans le lexique médical français,* Cahiers de Lexicologie, XXVI, 1975.

5. Graitson,M., *Traitement automatique du français médical,* Cahiers de Lexicologie, XXX, 1977.

6. Graitson,M., *Système d'indexation automatique des dossiers médicaux de l'Université de Liège,* Rapport interne, Université de Liège, 1980.

7. Hendrix,G., *Encoding knowledge in partitioned networks*, in N.V.Findler (Ed.),Associative Networks - The representation and use of knowledge in computers, Academic Press, New-York.

8. Laubsch,J.H., *Natural language interface engineering,* Séminaire international sur les systèmes intelligents de questions-réponses et de grandes banques de données, Rapport de l'IRIA.

9. Martin,W.A., *Roles, co-descriptors, and the representation of quantified English expressions,* Laboratory for Computer Sciences, M.I.T. Press,1979.

10. Pratt,A.W., *Automatic processing of pathologic data,* International Conference on Computational Linguistics, Stockholm, 1969.

11. Pratt,A.W., *Progress towards a medical information system for the research environment,* in Fuchs,G. & Wagner,G. (Eds), Krankenhaus-Informationssysteme : Erstrebtes und erreichtes, F.K.Schattauer Verlag, Stuttgart, 1972.

12. Dunham,G.S.,Pacak,M.G.,& Pratt,A.W., *Automatic indexing of pathology data,* Journal of the American Society for Information Sciences, March 1978.

13. Quillian,M.R., *Semantic memory,* in Minsky (Ed.), Semantic Information Processing, M.I.T. Press,Cambridge, Mass., 1968.

14. Samlowsky,W., *Case grammar,* in Charniak,E., & Wilks,Y. (Eds), Computational Semantics, North-Holland, 1976.

15. Schank,R., *Identification of conceptualizations underlying natural language,* in Schank & Colby (Eds), Computer Models of Thought and Language, Freeman, San-Francisco, 1973.

16. Scragg,G., *Semantic nets as memory models,* in Charniak,E., & Wilks,Y. (Eds), Computational Semantics, North-Holland, 1976.

17. Simmons,R.F., *Semantic networks : computation and use for understanding English sentences,* in Schank,R.C. & Colby,K.M., Computer Models of Thought and Language, Freeman and Company, 1973.

18. Wilks,Y., *Preference semantics,* in Keenan,E. (Ed.), Formal Semantics of Natural Language, Cambridge U.P., 1975.