Bente Maegaard - Ebbe Spang-Hanssen

SEGMENTATION OF FRENCH SENTENCES

1. This paper describes a programme which, by means of a very limited number of criteria, analyses French sentences into principal clauses and subordinate clauses. We prefer, however, to speak of segmentation rather than of analysis, because the analysis in question is very incomplete; there is, so to speak, no analysis at all; what we are doing is simply to try to find the extent to which the computer, using only a few kinds of landmarks in the text, is able to divide the sentence into certain segments, viz. the clauses. We raise the question whether French sentences can be segmented correctly by the computer recognizing in the input-text as linguistic signs having a special value, only words of the following three types:

1) finite verb-forms,
2) conjunctions and relative pronouns,
3) punctuation marks.

All other words are considered as belonging to a single class, the unmarked words. In practice we shall make some subdivisions within the three main groups so that we finally get eight classes of signs in the input-text between which the computer should be able to distinguish:

1) words introducing subordinate clauses (subordinating conjunctions, relative pronouns);
2) coordinating conjunctions (*et, mais, ou*);
3) comma;
4) unmarked words, i.e. every word not belonging to any of the other classes;
5) finite verb without clitic subject pronoun + clitic pronouns and particles;
6) full stop, semicolon, colon, exclamation-mark, question-mark.
7) " *car* ";
8) finite verb with a clitic subject pronoun + clitic pronouns and particles.

As will be seen, the computer will not, as a rule, try to identify the subject of the clause. Preliminary experience has shown us, however, that we could diminish considerably the number of errors by conserving the information given by the existence, or non-existence, of a clitic subject pronoun before the finite verb. We find that information of this kind is of the same type as the other " landmarks ", i.e. word-forms that one can identify fairly easily without going very far in the analysis of the context.

2. *Survey of the different stages of the project.*

The first part of the programme converts the input-text into a string of symbols (the numbers of the eight classes just defined). Example:

```
c'est l'heure dangereuse où le froid, si on n'est pas couvert, vous rend malade.
 8 4 4       4        1 4   4 3 1  8    4     4 3    5       4   6
```

The conversion programme, which should be the first to operate on the text, is nevertheless the last to be elaborated and, at present, it is still not operative in its entirety. At the first stage of our work, we made the conversion manually, wanting to see the efficiency of our parser before undertaking the rather tiresome work of writing the conversion-programme. Therefore, the segmentations presented in this paper are all segmentations of manually written strings of numbers which correspond to some French sentences; but, indeed, when writing these input-strings, we were careful to do only what our conversion programme should be able to make the computer do.

Although the conversion programme does not pretend to any degree of originality, it might be useful to give some idea of the amount of analysis contained in it; this point might be essential to the discussion of the interest of the entire project. The conversion programme consists mainly of the following subsections:

a) segmentation of the input-text into sentences and words. This programme especially examines the puntuation marks (in order, for instance, to find out whether a point is a full stop or something else), the hyphens, and the apostrophes; both hyphens and apostrophes may, or may not, separate words.

b) Conversion of the punctuation marks into one of the two symbols, 3 or 6.

c) Conversion of highly-frequent grammatical words (82 different words are listed) into the symbols 2 or 7 or into the symbols that represent two auxiliary classes. Apart from the eight classes taken over by the parser, we have set up, for the purpose of the conversion only, two auxiliary classes: determiners (for the purpose of disambiguating homographs that may be either nouns or finite verb-forms) and prepositions (on account of the possibility that they form part of a relative-phrase introducing a subordinate clause).

d) Conversion of finite verbs into one of the two symbols, 5 or 8. This routine is undoubtedly the most important part of the conversion programme. It examines every word from right to left in order te see whether the ending of the word matches with an entry in the list of verb-flexives. If the answer is in the affirmative, the routine will try to match the remainder of the token with an entry in the list of stems. This list comprises about 3200 stems which correspond to the verbs selected as those belonging to ordinary French in J. CAPUT, *Dictionnaire des verbes Français*, Larousse 1970.

e) Conversion into the symbol 1 of the words introducing subordinate clauses.

f) Conversion into the symbol 4 of every word for which the previous tests have failed.

The second part of the programme, the segmentation programme proper, analyses the string of numbers and gives as an output a string segmented in the following way (we take as an example the sentence quoted above):

```
input:  84441443184435 46
output: 8444/
              1443/       546
                   18443/
```

The oblique strokes indicate the boundaries of the clauses. We have introduced, however, the convention that a single stroke is not a boundary if followed at the same level by a space and a string which is not preceded by a stroke. In our example, this is the case at the second level where, as a matter of fact, there is only one clause:

c'est l'heure dangereuse/

où le froid/ *vous rend m.*

si on n'est pas couvert/

The output does not only give information about the boundaries of the clauses; it also indicates the levels of the clauses and the ways in which they are embedded in one another. In our example it is easy to see that the clause at the lowest level is embedded in the clause at the second level which, in turn, is subordinated to the clause at the top level.

It is evident that it is only a simple matter of programming to convert the output-string consisting of numbers into an image of the segmented sentence. This conversion, too, will be included in the final system.

3. *Examples.*

```
INPUT STRING    NR   60
443443445418323434454246
OUTPUT
4434434454/     /23434454246
           183/
```

Son dos, ses bras, son cou lui font mal quand il les remue, et, surtout, sa tête est lourde et douloureuse.

```
INPUT STRING    NR   61
4454442444441544454444446
OUTPUT
445444/244444/      544444446
              15444/
```

Son sang bat derrière son front et la dure lumière du jour qui traverse ses lunettes noires lui pique les yeux de ses centaines d'aiguilles.

```
INPUT STRING    NR   62
454444344444444325446
OUTPUT
454444344444443/25446
```

Jean voudrait pouvoir se reposer vraiment, dormir toute une nuit sans se réveiller, mais cela n'est pas possible.

```
INPUT STRING    NR   63
443844444244446
OUTPUT
443844444244446
```

Chaque soir, il retrouve l'air encore plus chaud et lourd de la moustiquaire.

```
INPUT STRING      NR   64
4434438444441445444441844446
OUTPUT
443443844444/
             144544444/
                       1844446
```

Dix fois, vingt fois, il se retourne sur son lit de toile pendant que les moustiques cherchent sans fin le trou par où ils pourront entrer pour le piquer.

```
INPUT STRING     NR   68
443844446
OUTPUT
443844446
```

Chaque jour, il mange un peu plus mal;

```
INPUT STRING  NR   69
484442184444444418344445434444344443846
OUTPUT
48444/2/                  44445434444344443/846
        1844444444/      /
                   183/
```

Bientôt il n'a plus de pain et comme il ne sait pas assez bien chasser pour tuer les animaux qu'il rencontre, presque tous ses repas sont maigres: un peu de riz, quelques pommes de terre, c'est tout.

4. *The interest of the project.*

If the programme turns out to be reasonably successful – that is to say if the automatic conversion of the text into symbols will not increase too much the number of errors – one might think of two possible applications:

The programme might be useful for obtaining more interesting concordances. The computer has often been thought of as a means for the linguist of obtaining better documentation than that of the old-fashioned card-index made manually by the linguist himself; but if the computer is to replace the old card-index, the linguist must have the opportunity not only to give to the machine a list of words in which he is interested, but also to define for the machine the kinds of context in which the key-words should occur. A programme that recognizes the boundaries of the clauses should offer a good many possibilities for defining the contexts that the linguist is looking for.

Another possible application of the segmentation programme might be as a first stage of a complete analysis. It might be profitable to start analysis with a tentative segmentation of the sentences into clauses.

For us, however, the linguistic interest proper of the project is more important than the rather hypothetical applications, even though it seems quite clear that our grammar does not give a better understanding of the structure of the French sentence than that which could be found in a traditional grammar. Our grammar does not give any insight into the deep structure of the sentence, but it gives a description of the profile of the surface structure. One might speak of it as an investigation of some surface-constraints in French. It is not without interest to examine the extent to which it is true that French sentences are cast in a certain mould so that only a few marks are sufficient for a correct segmentation. Or to put it more simply: the segmentation programme can be seen as a new approach to the study of word-order in French.

5. *The linguistic facts.*

In this section we will discuss the linguistic facts that make possible a segmentation solely on the basis of the finite verb-forms, the conjunctions, and the punctuation marks.

We have found it convenient to set up the rule that every finite verb constitutes a clause, that is to say there can be only one finite verb in a clause. This view is linguistically justifiable, for instance within the framework of a dependency grammar. Consequently, we shall say that there are two clauses in each of the following sentences:

La voiture démarre et part en vitesse
La voiture démarre, part en vitesse.

In general, it is fairly easy to identify the beginning of a subordinate clause. It is a striking peculiarity of the French language that every (or nearly every) subordinate clause is introduced by some kind of introductory word, a conjunction or the like. Or to put it otherwise: the first word in a subordinate clause is always a word that belongs to a very limited class and one which it is rather easy to recognize (subordinating conjunctions, relative pronouns). Some of these words can play other roles as well (for instance *si*, *combien*), but the ambiguities are relatively few or unimportant.

It is much more difficult to find out where the subordinate clauses end. It is perhaps one of the merits of our model that it puts that question – until now, as far as we know, grammarians have almost exlusively treated the beginning of the subordinate clauses. There is no problem, of course, if the end of the subordinate clause coincides with the end of the sentence; the real problem arises with the embedded clauses.

Normally, the information contained in our strings of numbers is sufficient for the identification of the end of the embedded clauses on account, above all, of the fact that the subordinate clauses are most often embedded before the verb of the dominating clause (i.e. a clause that contains a subordinate clause – a subordinate as well as a principal clause may be a dominating clause):

le vin qu'il boit est très fort
le vin qu'il boit le soir est très fort.

If the clause is embedded before the finite verb of the dominating clause, and if there is no comma or coordinating conjunction between the verb of the subordinate clause and the verb of the dominating clause, the end of the embedded clause is to be found just before the verb of the dominating clause (as in the two examples just quoted). In a string such as

4 4 1 4 5 4 4 4 5 6

where " 1 " indicates the beginning of an embedded clause, one can safely put the final boundary of this clause just before the second " 5 " (finite verb). This observation, which does not seem to be a matter of course, can be stated in the following way in terms of traditional grammar: if you insert a clause just after the subject, you cannot insert anything more before the verb-phrase.

The unmarked words (denoted here by the number 4) standing between two finite verbs are connected with the preceding verb. This is true also of a string of " 4 "s surrounded by commas:

> *le vin qu'il boit le soir, quelques minutes avant de se coucher, est très fort.*

There is only one exception to these rules concerning the final boundary of clauses embedded before the verb of the dominating clause. If the clause is inserted after an adverbial phrase at the head of the dominating clause, the subject of the dominating clause will be placed after the embedded clause just before the verb:

> *le jour où le livre parut Jacques était absent de Paris.*

Our grammar does not solve this problem if the embedded clause is not terminated by a comma, or if the subject of the dominating clause is not a clitic pronoun. But normally there is a comma after an embedded clause which is part of an adverbial phrase at the head of the sentence:

> *Dans une société qui fonctionne comme un organisme robuste, l'élimination se fait toute seule.*

The problems are more complicated when between the two finite verbs there are one or several coordinating conjunctions or a number of commas different from two. Our grammar analyses correctly most of these combinations such as, for instance, the following case:

> *son dos, ses bras, son cou lui font mal quand il les remue, et, surtout, sa tête est lourde et douloureuse.*

In this example, the stroke that marks the end of the embedded clause will be placed just before *et.*

The weakest point in the model is the treatment of clauses embedded after the verb of the dominating clause; in this case, we have no definite means of identifying the end of the subordinate clause. Example:

> *il avait donné à tous ceux qui s'étaient présentés des cadeaux magnifiques.*

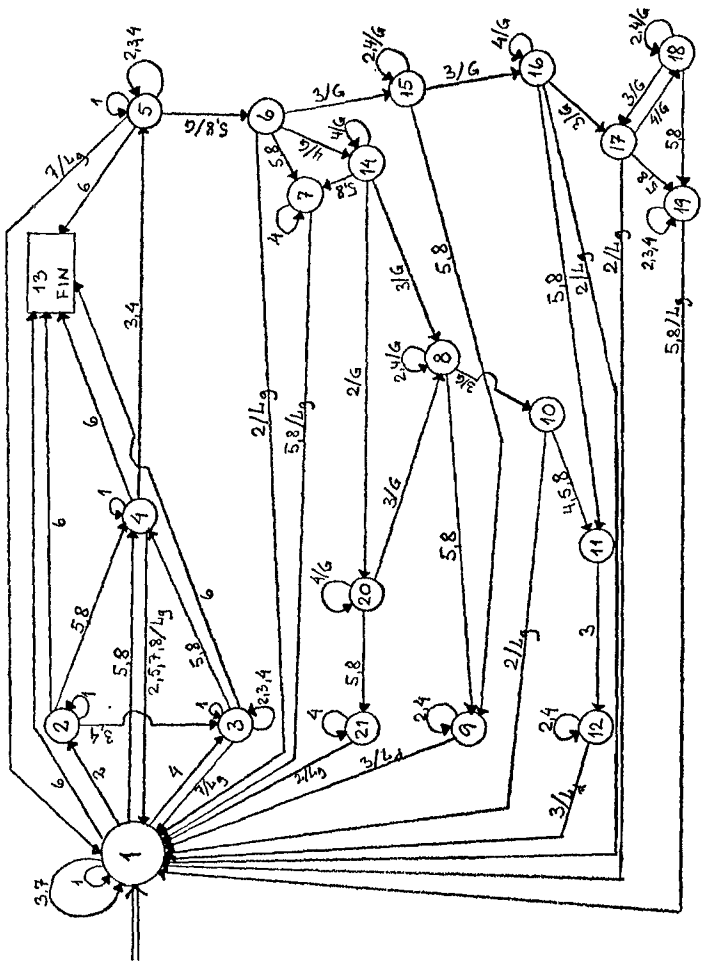
This case is, fortunately, rather infrequent; it is much more common to find the subordinate clause at the end of the dominating clause, at the head of the dominating clause, or before the verb of the dominating clause. In the texts analysed until now by the computer, more than 90 % of the sentences have been correctly segmented.

6. *The grammar.*

Our grammar is intended to analyse grammatical French sentences. Ideally, it should be able to analyse correctly every grammatical French sentence. On the other hand, it is not designed for discarding ungrammatical sentences.

The grammar has the form of two state diagrams, one of the main clauses, and another of the subordinate clauses. In many states, in both of the diagrams, the parsing procedure can leave the level where it is and enter the diagram of the subordinate clauses at the next, lower, level. In principle, there is no limit to the number of times the procedure can make new entries, at still lower levels, into the diagram of the subordinate clauses. Consequently, the grammar can be considered as an extended finite state grammar, rather similar to the *Transition network grammar* described by WOOD (in « Communications of ACM », XIII (1970) no. 10.

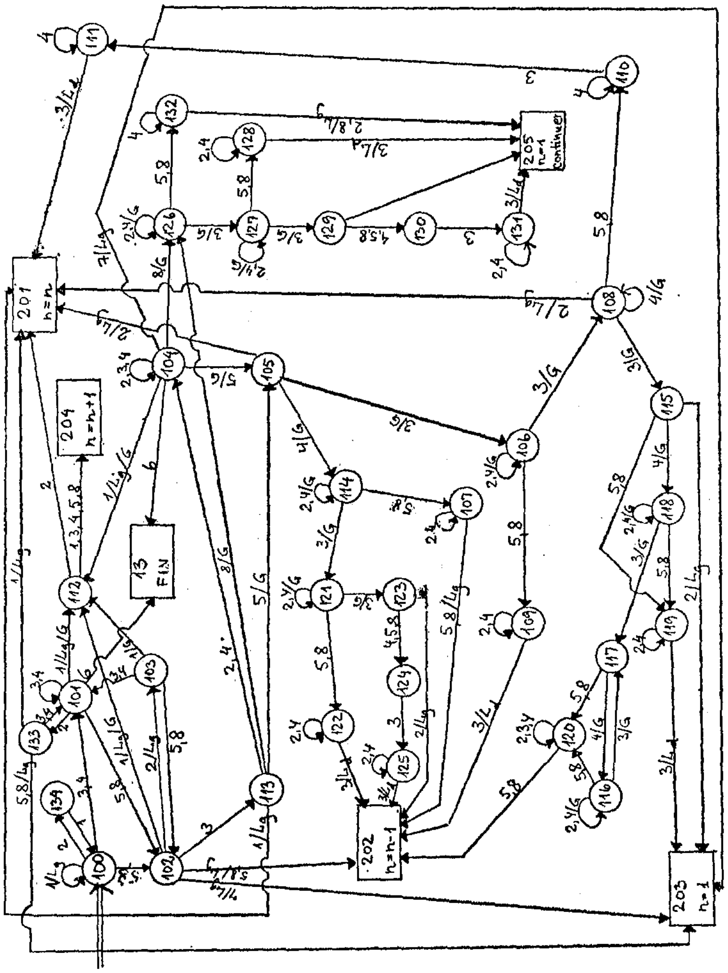
It is the loops provided with the number 1 (the symbol for the class of words introducing subordinate clauses) that have this particular effect of making a new entry into the diagram of the subordinate clauses.



MAIN CLAUSE.

Thus, a loop carrying the number 1 does not have exactly the ordinary meaning of a loop, viz. that the machine shall stay in the state where it is. The machine will pursue the analysis at a lower level, but when it has found the end of the subordinate clause, it will come up again into the state it left, and in this way, from the view-point of the upper level, the loop can be said to keep its ordinary meaning.

Apart from the numbers symbolizing the different word-classes, the arrows of the diagram may carry some other symbols. *Ld* and *Lg* signify that the machine shall print, at this moment, a stroke marking



SUBORDINATE CLAUSE.

the boundary between two clauses. *Ld* means that the stroke shall be printed on the right side of the symbol which has just been read, whereas *Lg* will put the stroke on the left side. G means that the machine, when arrived at the new state, shall read the symbol to the left of the symbol just read, i.e. the machine shall read from right to left.

The backward reading, from right to left, plays a rather important role in this grammar, because, among the 8 symbols, some are more informative than others, in particular class 1 (words introducing subordinate clauses) and the classes 5 and 8 (finite verbs): each of these three numbers indicates by itself the existence of a new clause, whereas the commas and the coordinating conjunctions have only a secondary importance. For this reason, the machine will most often, during a first reading from left to right, remain in the same state after having read a comma or a conjunction. On the contrary, when the machine has read a new finite verb (the state *q*6 in the diagram of the main clauses, the state *q*105 in the diagram of the subordinate clauses), it will know that it must look for a place to put a boundary somewhere to the left of the verb it has met; at that moment the commas and the coordinating conjunctions become interesting as the words that are most likely to indicate the boundary.

The states that mark the end of a subordinate clause are indicated by rectangles; these states control the level shifts.

The computer programme had to solve the rather puzzling problem of combining the backward reading with the shifting of levels. From a linguistic point of view, the computer programme adds to the grammar only a device for the treatment of the " holes " which appear in the string when a part of the string has been put at a lower level. In the programme, such a hole is considered as being framed by two commas, even if the subordinate clause was not framed by commas; this means that, at the level of the dominating clause, the embedded clause has the same status as a string of unmarked words framed by two commas.[*]

[*] We suppose that the programme in its entirety will be operative in the course of autumn 1973. It is our intention then to analyse a certain number of French texts – at present we have at our disposal 12 novels on paper tape which we have borrowed from « Le Centre pour un Trésor de la Langue Française » (Nancy) and which we have transcoded.