Copernicus Project 621 - GRAMLEX
Work package 3 - Task 3D2
Deliverable R3D2 - Final report

Grammatical disambiguation of French words

using part of speech, inflectional features and lemma of words in the context

Éric Laporte, Anne Monceaux, ASSTRIL

September 23, 1997

Abstract: We describe ELAG, a new system of lexical disambiguation using grammatical information about words in the context. The disambiguation takes place after a lexical analysis of input text, but before syntactic parsing.

Introduction

Much of the ambiguity in a text can be removed with only very basic linguistic information about words: namely, parts of speech, inflectional features and lemmata of the words (what we call grammatical information), plus some punctuation information. Of course, much more elaborated information is required in many cases for a complete disambiguation; in the following sentence, for instance:

La république ne veut pas que l'école publique, son fleuron, véhicule des signes discriminatoires,

the part of speech of *véhicule* cannot be determined without syntactic information about the verbs *vouloir* and *véhiculer*, namely the fact that they are transitive verbs with a direct complement, and nor without a thorough recognition of the syntactic structure of the sentence.

However, a partial disambiguation with only grammatical information is undoubtedly worth carrying out, since this operation facilitates nearly all types of further processing, including syntactic parsing, which in turn removes some more ambiguity. The a priori limitation of the linguistic data to be used by our disambiguator is thus realistic both in terms of feasibility, and in terms of usefulness for applications. The major conceptual difficulty of the task is that when we disambiguate a word, we cannot assume that its context has been completely disambiguated beforehand.

The framework of the work related in this report is the same that has been described in literature (Silberztein, 1993, 1997; Roche, 1992) and in earlier reports (Laporte, Silberztein, 1996 ; Courtois, Laporte, 1996). Let us briefly sum up its fundamental choices.

- The only operation before disambiguation is listing all grammatical ambiguities, in other words: the lexical analysis of input text. In our experiments, lexical analysis is performed by INTEX, i.e. the tag set is rich in comparison with present standards in literature, and compound words are taken into account and assigned specific tags.
- The objective of the processing is not to eliminate all ambiguity from the input. The objective can be defined as follows: removing as much as possible of incorrect analysis (i.e. reducing noise), with the strict constraint of never eliminating a correct analysis (i.e. with zero silence).
- Due to these two choices, input and output take the form of finite acyclic automata (as in Fig. 1), which allows one

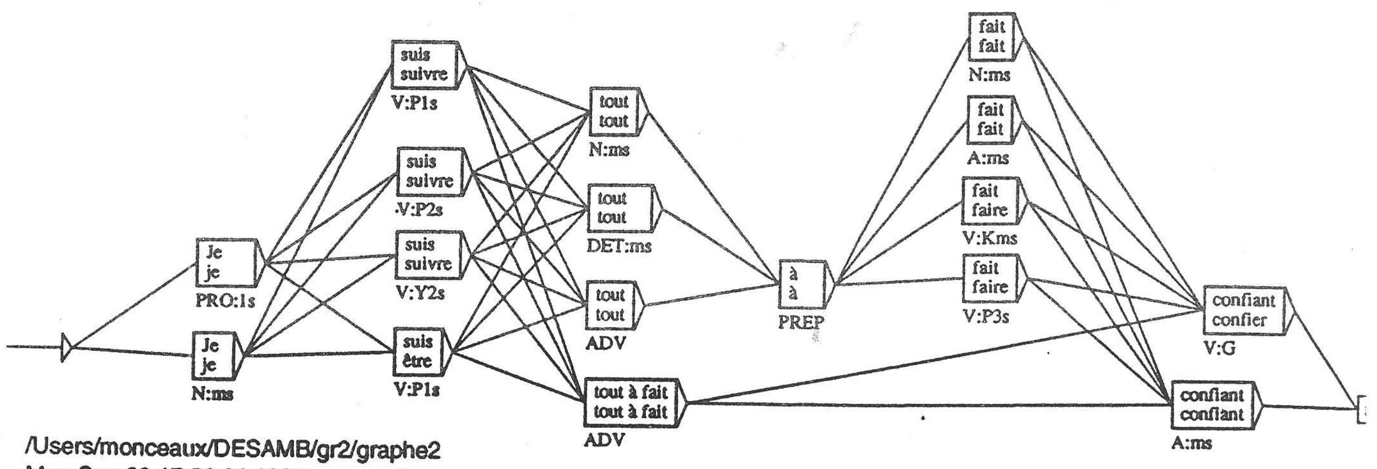
  (i) to represent concurrent analyses of the same input sentence in a compact structure where common parts are shared,

  (ii) to represent compounds although they are usually a priori ambiguous with sequences of simple words.
- Handwritten grammars are used. The linguistic data used in the grammars are all of a basic level: parts of speech, inflectional features, lemmata, grammatical words, and punctuation.

1. An original method of removing ambiguities : the system ELAG

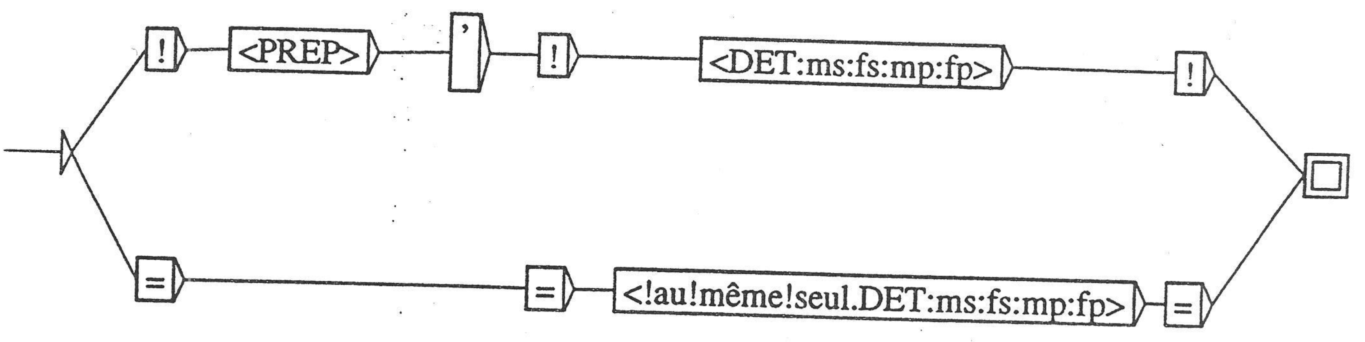
Inside this framework, we designed and built a new system: ELAG (elimination of lexical ambiguities by grammars). The general idea behind it is that a system of lexical disambiguation with good results is bound to be a large system, and that the problems of maintenance of such a system are of the same nature as those that occur during the construction and evolution of large software systems.

This is why the issue of modularity and independance between modules has been considered a priority.



/Users/monceaux/DESAMB/gr2/graphe2
Mon Sep 22 17:56:31 1997

Fig. 1



Certains déterminants ne peuvent pas être précédés d'une préposition.

s/monceaux/DESAMB/gr2/prepDet
Sep 22 18:34:02 1997

Fig. 2

The linguistic data of the disambiguator are organised in separate, compact, readable modules, that we call disambiguation grammars. Examples of disambiguation grammars are shown in Fig. 2 and 5 to 9. The way they are read is explained in section 3 of this report.

The respective effects of several disambiguation grammars on an input text are independent of each other. In other words, a given grammar has a specific effect on a given input sentence, and this effect will be the same, no matter whether this grammar is used with other grammars or not. For instance, if a correct analysis of a sentence is ruled out, a situation which we want to avoid, we can apply each of the grammars to the sentence separately, and we are sure to discover which of the grammars reject(s) the correct analysis. Then, if we modify this grammar or these grammars in order to correct them, these modifications will not change the effects of other grammars. Thus, this feature of ELAG makes it possible to detect errors and to control the evolution of the set of grammars.

This feature of the disambiguation is mathematically guaranteed by the formula used to apply grammars to sentences (cf. pp. 8-10).

The effects of disambiguation grammars are cumulative: if one writes new grammars and uses them with existing ones, the effects of the existing grammars are not modified. The new ones can only rule out further analyses. Different grammars can apply to the same sequence, or to overlapping sequences, or to sequences included in other sequences: their effects are cumulative. In consequence, disambiguation grammars can be written by several authors and used together, even without any coordination between the authors. There cannot be compatibility problems with this procedure: the only risk is to duplicate work.

The order of application of grammars is indifferent: they can be used simultaneously or sequentially and in any order, without any modification of the output. This comes from the fact that grammars are applied to sentences with the operation of intersection, a commutative operation.

In addition, the effects of a grammar on various analyses of a sentence are independent. In other words, the effect of a grammar on a given analysis of a sentence is either to reject or to accept it, and this effect is independent of other analyses of the same sentence. For instance, the disambiguation grammar shown in Fig. 2 means that a determiner occurring immediately after a preposition can be neither *au* nor *même* nor *autre*; if ones uses this grammar to disambiguate a sentence, it will rule out any analysis which comprises a preposition followed by one of the 3 forbidden determiners, e.g. *<entre.PREP></><même.DET:fs>*, where *</>* means a word limit, but it will not apply to a sequence like *<entre.V:P3s></><même.DET:fs>*, although *entre* is ambiguous between a preposition and a verb. Because of this feature of ELAG, authors of grammars can concentrate on a particular grammatical pattern, e.g. *<PREP></><DET:ms:mp:fs:fp>*, without caring about words that are ambiguous with elements of this pattern.

All the features mentioned in this section were already present in the system of Emmanuel Roche (1992), although he did not advertise much about them. They all follow mathematically from the formula which underlies the application of the grammars to the sentences. The formula used in ELAG is an elaboration of the formula in Roche (1992) and is introduced in section 3.

2. Evaluation of performances

ELAG has a procedure of evaluation of its results. Lexical disambiguation is a competitive domain at an international level, and an accurate evaluation of results is an indispensable tool. However, with current procedures of evaluation, the performances of systems cannot always be compared. More specifically, when two disambiguators use different tag sets, their performances cannot be compared without a specific work on the two tag sets (Laporte, Silberztein, 1996). For example, Fig. 3 represents the same sentence as Fig. 1 but with a poorer tag set: the quantity of ambiguity in input is not the same, and comparisons are biased. Another issue is that silence and noise (or recall and precision), which are a priori independent parameters, should be evaluated separately, but usually they are not.

We evaluate noise and silence independently. In order to evaluate noise, we compare the quantity of ambiguity before and after disambiguating the text. We define the quantity of ambiguity of a sentence as the logarithm of the number of analyses. This definition has two advantages:

- it is compatible with the fact that compounds are taken into account (if we only counted the tags of each word and computed the mean over the text, we could not take into account compounds in a natural way);
- it makes the quantity of ambiguity an additive quantity: the quantity of ambiguity of a text can be defined as the sum of the quantities of ambiguity of the sentences. This quantity will not be modified if adjacent sentences of the text are merged by erasing sentence boundaries between them. This behaviour comes from the fact that we use the logarithm of the number of analyses (if one uses the number of analyses itself, then the quantity of ambiguity increases when adjacent sentences are merged, and the rate before/after processing depends on the average length of sentences).

With this convention, the user of ELAG obtains for each sentence and for the whole text a rate that represents the proportion of ambiguity which remains after the processing.

In addition, residual ambiguities are displayed in a readable format which takes the form of a list (Fig. 4). Another readable format would be that of a graph, as in Fig. 1.

Silence is evaluated as well. Recall that we consider as a priority to keep silence to zero. In order to check that the processing does not remove correct analyses, we examine manually the readable listing of the output and we check that all correct analyses are present. This work cannot be automated. It takes a particular form in the case of a small proportion of sentences. For a few sentences, the grammars remove all input analyses, and the output finite automaton represents the empty set. For these sentences, one checks that either the correct analyses did not appear in the input finite automaton of the sentence, or that the input sentence contained a mistake and does not have any correct analysis. In the first case, the absence of the correct analyses can follow, for instance, from the absence of a compound from the dictionaries of the lexical analyser, as in

Je me suis bien amusé, au revoir, et merci !

where *au revoir* belongs to a class of compounds which cannot be reliably recognized by INTEX yet and has not been integrated into the dictionaries. An example of the second situation is

Les atteintes au paysages,

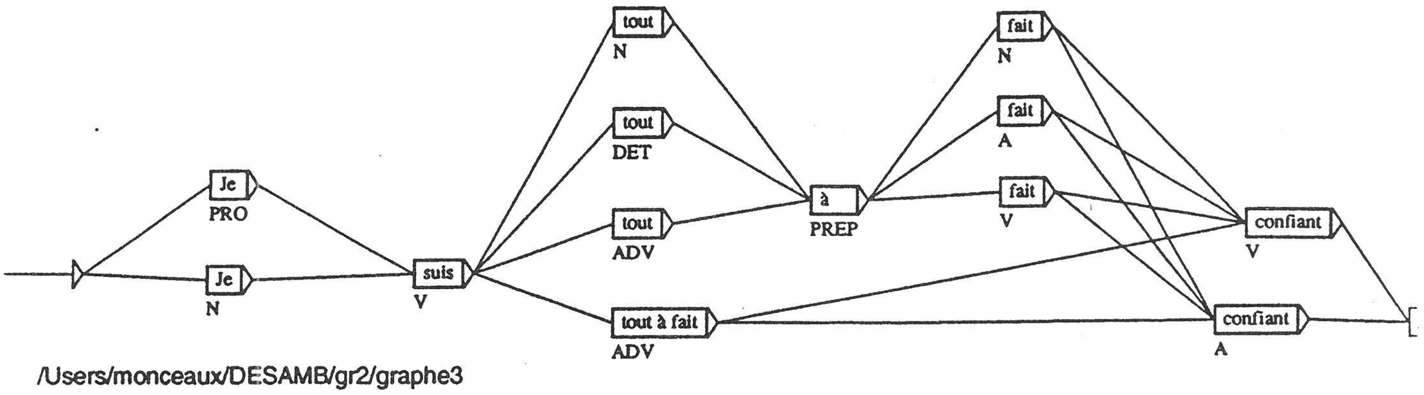
a title with an agreement mistake.

3. Disambiguation grammars

Approximately 70 disambiguation grammars have been written for ELAG by the authors of this paper, some of them in imitation of disambiguation grammars by Max Silberztein or of local grammars by Maurice Gross (1997).

Every ELAG disambiguation grammar expresses a constraint, and the effect of applying it is always to remove the analyses that do not obey the constraint.

The simplest form of an ELAG disambiguation grammar is shown in Fig. 5. This grammar means that whenever *s* is followed by an apostrophe and tagged as a form of the subordinating conjunction *si*, then it must be followed by one of the preverbal pronouns *il* and *ils*. In practice, this grammar is designed to reduce the ambiguity of the sequence *s'* in

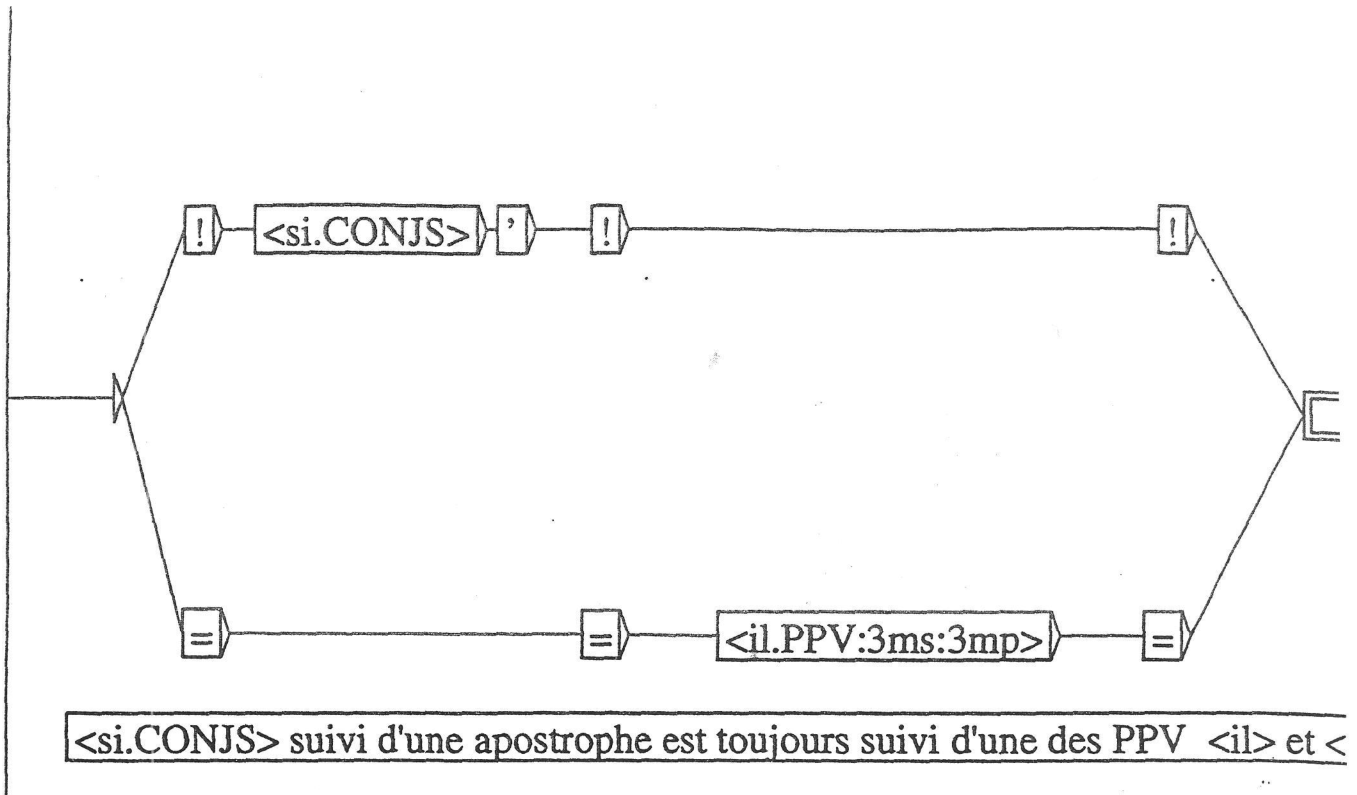


/Users/monceaux/DESAMB/gr2/graphe3
Mon Sep 22 17:57:13 1997

Fig. 3

```
#  {ou,ou.CONJC}
# Etat 53
#  {,/.PONCT}
# Etat 54
#  {que,que.CONJS}
#  {que,que.PRO}
#  {que,que.XI}
# Etat 55
#  {,/.PONCT}
# Etat 56
#  {nous,nous.N:ms} {,/.PONCT} {refusons,refuser.V:Y1p}
#  {nous,nous.N:ms} {,/.PONCT} {refusons,refuser.V:P1p}
#  {nous,nous.PPV:1p} {,/.PONCT} {refusons,refuser.V:P1p}
#  {nous,nous.PRO:1p} {,/.PONCT} {refusons,refuser.V:Y1p}
#  {nous,nous.PRO:1p} {,/.PONCT} {refusons,refuser.V:P1p}
# Etat 61
#  {,/.PONCT}
# Etat 62
#  {de,de.PREP}
# Etat 63
#  {,/.PONCT}
# Etat 64
#  {voir,voir.V:W}
# Etat 65
#  {,#.PONCT}
# Etat 66
# Taille logarithmique : 6.866933
# Ambiguïté résiduelle : 39.5 %
```

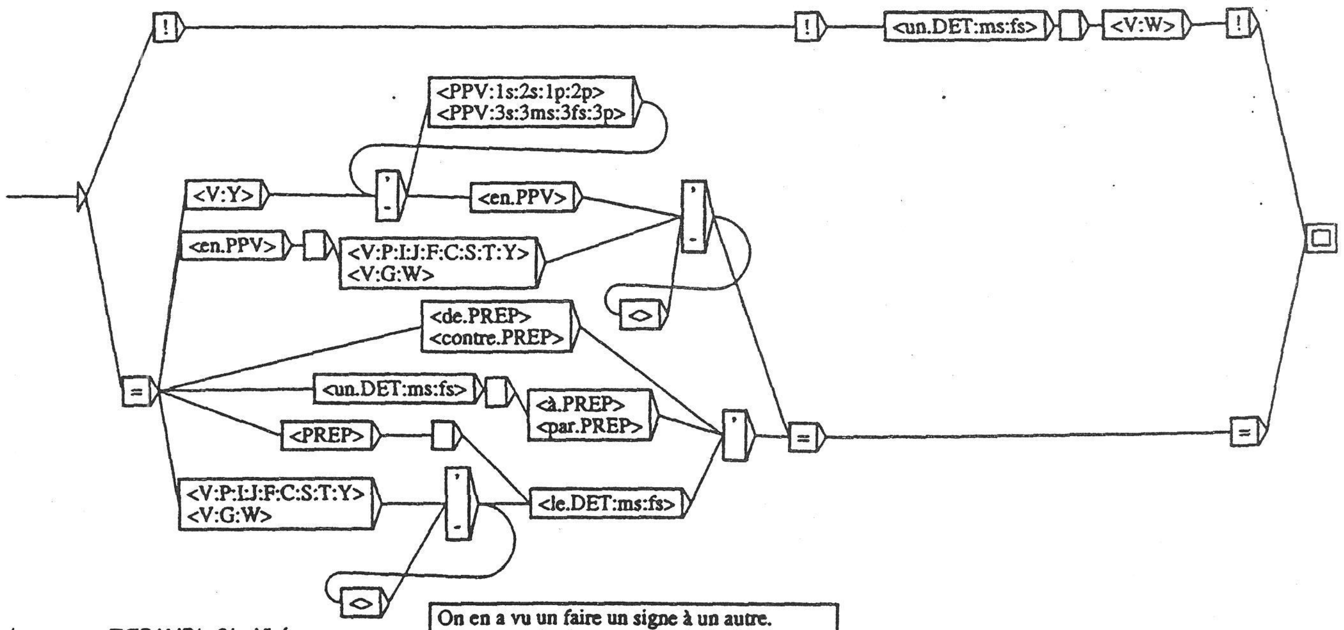
Fig. 4



<si.CONJS> suivi d'une apostrophe est toujours suivi d'une des PPV <il> et <

/Users/monceaux/DESAMB/gr/sil
Mon Sep 22 18:37:37 1997

Fig. 5



On en a vu un faire un signe à un autre.
Pour faire plaisir à l'une, mécontenter l'autre ?

/Users/monceaux/DESAMB/gr2/unVinf
Mon Sep 22 18:38:20 1997

Fig. 6

French. One of the a priori analyses of this frequent sequence is that *s* is the elided form of *si*, but this (optional) elision occurs only in the two sequences *s'il* and *s'ils*, and never before other pronouns or words. In addition, *s'* can a priori be the elided form of the preverbal pronoun *se*, but in that case the conditions of elision are different.

If we examine Fig. 5, we have an "if" part which describes <*si.CONJS*><'>, and a "then" part which contains <*il.PPV:ms:mp*>. The "if-then" structure of ELAG disambiguation grammars is borrowed from Max Silberztein, because it is in our opinion a very natural way of formulating readable disambiguation rules. In particular, many potential authors of disambiguation grammars will probably find it more comfortable than the convention of Emmanuel Roche (describing forbidden sequences of tags).

Practically, the "if" part is signalled and delimited by the three boxes with "!", and the "then" part by the three boxes with "=". This convention is used in all figures in this report. The boxes with "!" and "=" are delimitors which are used by ELAG to recognize the structure of the grammars. All other boxes contain linguistic elements that ELAG searches in input sentences when grammars are applied to text.

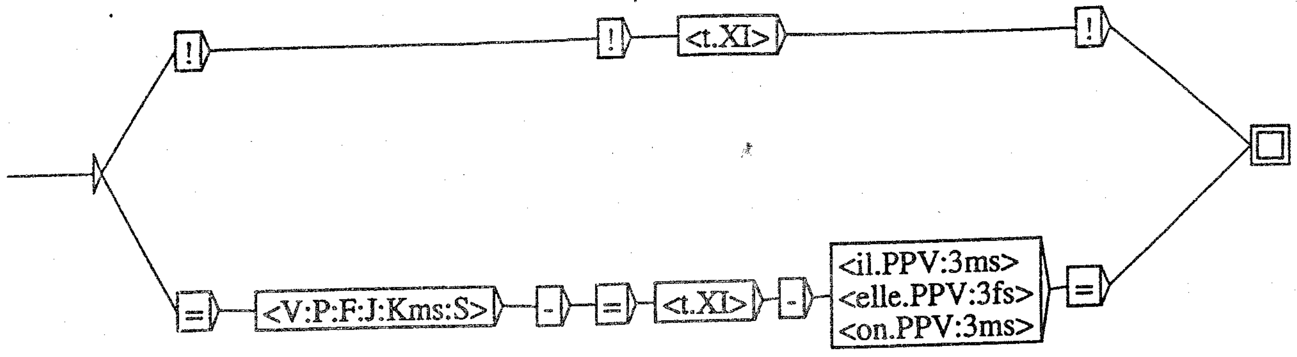
The left and right "!" and "=" are present to make the grammar more readable. The central "!" and "=" are the only element of synchronization between the "if" part and the "then" part. As a matter of fact, when the grammar of Fig. 5 is applied to a sentence, ELAG checks that the "then" pattern, <*il.PPV:ms:mp*>, occurs immediately to the right of any occurrence of the "if" pattern, <*si.CONJS*><'>, in every analysis, and it removes the analyses that do not obey this constraint.

In Fig. 6, we still have an "if" part above and a "then" part below, but the meaning is that every occurrence of <*un.DET:ms:fs*><*/*><*V:W*> (i.e. the determiner *un* followed by a verb in the infinitive, e.g. *un sortir* in *On en voit un sortir*) must immediately follow an occurrence of the complex pattern in the "then" part. The effect of applying this grammar is to remove all analyses that do not obey this constraint.

In the disambiguation grammars of Fig. 5 and 6, the "if" part and the "then" part do not overlap, but in general they can overlap without any restriction. In Fig. 7, the "if" part consists only of <*t.XI*> (which stands for the so-called euphonic *t* e.g. in *pense-t-il*), and the meaning is that every occurrence of <*t.XI*> must occur immediately after an occurrence of the pattern <*V:P:F:J:S*><*-*> (i.e. a verb at one of a few tenses and a dash), and immediately before an occurrence of the pattern <*-*>(<*il.PPV:3ms*> | <*elle.PPV:3fs*> | <*on.PPV:3ms*>), i.e. a dash and one of the preverbal pronouns *il, elle, on*.

The general form of an ELAG disambiguation grammar is shown in Fig. 8. The boxes with "!" and "=" delimit four patterns R_1, R_2, C_1, C_2, any of which can be the empty word. The meaning of a grammar is that whenever an occurrence of R_1 is immediately followed by an occurrence of R_2 in an analysis of a sentence, then the point of junction between them must be immediately preceded by an occurrence of C_1 and followed by an occurrence of C_2 in the same analysis. There is no restriction on the lengths of the sequences that belong to the patterns R_1, R_2, C_1, C_2; in fact, any of these patterns represents a set of tagged sequences that may not have all the same length. In Fig. 9, the "then" pattern imposes gender and person agreement to sequences like *lui, lui qui, lui seul, lui qui seul*, at the beginning of a sentence and before a verb.

The mathematical expression of the effect of a disambiguation grammar is defined as follows. A disambiguation grammar is defined by the four patterns R_1, R_2, C_1, C_2, each of which is a set of tagged sequences. An input sentence is represented by a pattern P, the set of a priori analyses of the sentence, which is also a set of tagged sequences. The output of the application of the grammar to the sentence is defined by the expression



Le t euphonique <t.XI> est entre deux tirets, il suit un verbe et précède un PPV sujet.

/Users/monceaux/ELAG/gr2/tinversion
Tue Sep 30 17:08:09 1997

Fig. 7

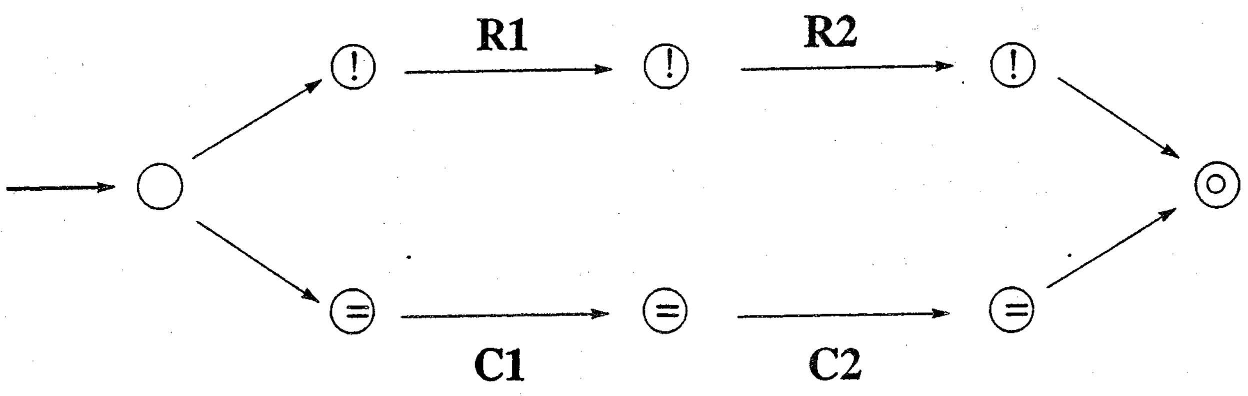


Fig. 8

$$P \setminus (A^*R_1(R_2A^* \setminus C_2A^*) \mid (A^*R_1 \setminus A^*C_1)R_2A^*)$$

where A is the tag set (including punctuation tags), $*$ is the operation of iteration, $\mid$ is the operation of set union and $\setminus$ the operation of set subtraction.

The application of grammars to text is easy to implement. Each of the patterns R_1, R_2, C_1, C_2, P is represented by a finite automaton. The output of the application of the grammar to P is obtained by computing the intersection of two automata: P and

$$A^* \setminus (A^*R_1(R_2A^* \setminus C_2A^*) \mid (A^*R_1 \setminus A^*C_1)R_2A^*).$$

The latter automaton is the compiled form of the disambiguation grammar. If several disambiguation grammars are to be used together, ELAG computes their intersection and uses it a compiled grammar to process text. ELAG has been implemented in C language by Eric Laporte and his undergraduate class of computer science at the University of Rheims-Champagne-Ardenne. It should be easily integrated into INTEX after deciding an interface format for input and output sentences.

A general method of writing an ELAG disambiguation grammar is to choose a grammatical pattern, and to describe its variants and the patterns that occur obligatorily in its immediate context. Unfortunately, this strategy cannot be simply automated.

The tag set that we used in our experiments is the tag set of INTEX, with a few modifications:

- A few new ambiguities are taken into account. For example, preverbal pronouns like *<ils,il.PPV:3mp>* are distinguished from other pronouns like *<eux,lui.PRO:3mp>*, which introduces an ambiguity between *<elle,elle.PPV:3fs>* and *<elle,elle.PRO:3s>*.
- A few errors in dictionaries were corrected.
- There are 4 punctuation tags: sentence boundary <#>, dash <->, apostrophe <'> and a tag for any other word boundary </>. In the future, it will be useful to divide this type in two: word boundaries with only spaces, and word boundaries with at least one character which is not a space. This information will allow ELAG, for example, to rule out the tag *<ce.DET:ms>* in:

Sur ce, je vous quitte

since when *ce* is a determiner, it is obligatorily followed by a word boundary with only spaces (e.g. in *Les clés étaient sur ce meuble*).

Another perspective is to develop contracted forms like *aux* into tagged sequences like *<à.PREP></><les.DET:ms:fs>* before disambiguating sentences.

Results and conclusion

The ELAG disambiguation grammars that have been implemented are finite automata which generally have 20 to 100 states. When they are compiled by batches of 5 or 6, the compilation of a batch of grammars generates a compiled grammar which generally has 100 to 150 states: the compilation does not cause an explosion of the number of states.

The grammars were tested on a text of 4.163 words (25 Kb, 175 sentences). The automata of sentences before disambiguation generally have 10 to 60 states, and the automata of the same sentences after disambiguation generally have 20 to 100 states: the application of the grammars to sentences does not cause an explosion of the number of states.

The rate of residual ambiguity is generally between 70 % and 30 %, depending on sentences.

The silence is zero.

References

Courtois, Blandine, Eric Laporte. 1996. Grammatical disambiguation of French words using part of speech of words in context, in *GRAMLEX Deliverables. June-December 1996*, E. Laporte (ed.), Laboratoire d'automatique documentaire et linguistique, University of Paris 7-Denis-Diderot, 10 p. + 68 p. annex.

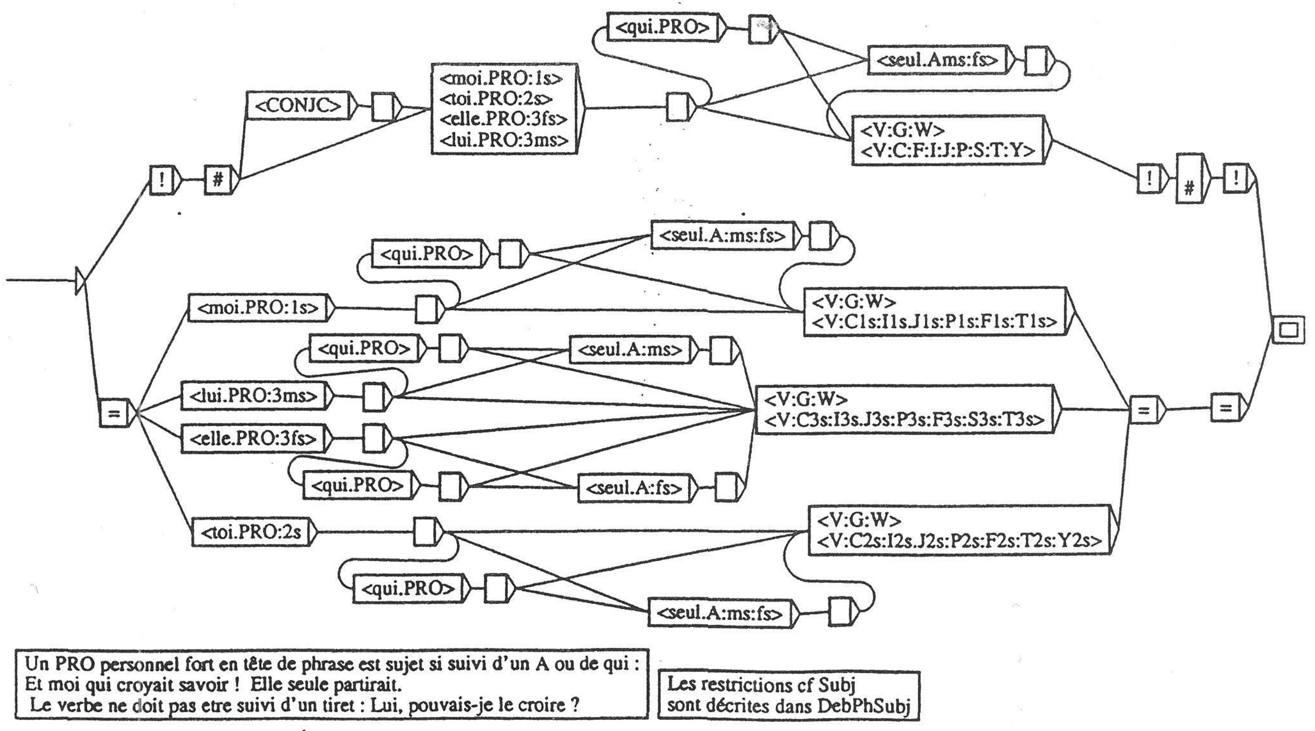
Gross, Maurice. 1997. The construction of local grammars, in *Finite-State Language Processing*, E. Roche and Y. Schabes (eds.), Cambridge, Mass./London, England: MIT Press, pp. 329-354.

Laporte, Eric, Max Silberztein. 1996. Ambiguity rates. Automatic analysis of French text corpora and computation of ambiguity rates for different tagsets, in *GRAMLEX Deliverables. October 1995-June 1996*, E. Laporte (ed.), Laboratoire d'automatique documentaire et linguistique, University of Paris 7-Denis-Diderot, 7 p.

Roche, Emmanuel. 1992. Text disambiguation by finite-state automata, an algorithm and experiments on corpora, in *Proceedings of COLING 92*, Nantes.

Silberztein, Max. 1993. *Dictionnaires électroniques et analyse automatique de textes. Le système INTEX*, Paris: Masson, 233 p.

Silberztein, Max D. 1997. The lexical analysis of natural languages, in *Finite-State Language Processing*, E. Roche and Y. Schabes (eds.), Cambridge, Mass./London, England: MIT Press, pp. 175-203.



Un PRO personnel fort en tête de phrase est sujet si suivi d'un A ou de qui :
Et moi qui croyait savoir ! Elle seule partirait.
Le verbe ne doit pas etre suivi d'un tiret : Lui, pouvais-je le croire ?

Les restrictions cf Subj
sont décrites dans DebPhSubj

/Users/monceaux/DESAMB/Annegr/debPhProFortSgV
Mon Sep 22 18:26:33 1997

Fig. 9