From Part of Speech Tagging to Memory-based Deep Syntactic Analysis

Emmanuel Giguet and **Jacques Vergne**
GREYC — CNRS UPRESA 6072 — Université de Caen
14032 Caen cedex — France

{Emmanuel.Giguet,Jacques.Vergne}@info.unicaen.fr

Abstract

This paper presents a robust system for deep syntactic parsing of unrestricted French. This system uses techniques from Part-of-Speech tagging in order to build a constituent structure and uses other techniques from dependency grammar in an original framework of memories in order to build a functional structure. The two structures are build simultaneously by two interacting processes. The processes share the same aim, that is, to recover efficiently and reliably syntactic information with no explicit expectation on text structure.

1 Introduction

This paper describes a robust system for deep syntactic parsing of unrestricted French. In this system, deep syntactic analysis means identifying constituents and linking them together. A brief presentation of the problematics will clarify the choice we have made.

In western-european natural languages such as French, both constraints and liberties exist in respectively two distinct levels. The stylistic liberties of the French author (Molière, 1670) illustrate this phenomenon in the famous following lines:

"[Belle marquise], [vos beaux yeux] [me font] [mourir] [d'amour]"
"[D'amour] [mourir] [me font], [belle marquise], [vos beaux yeux]"
"[Vos beaux yeux] [d'amour] [me font], [belle marquise], [mourir]"

The order of small chunks of words (here bracketed) is fairly free. However, we point out a roughly fixed word-order within these chunks which strong constraints are applied to. These two kinds of opposite and independent behaviors in two distinct levels led us to consider that these levels could not be managed properly with a unique process and a unique representation. One will note that chunks are traditional syntactic groups as defined in (Le Goffic, 1993) *without* their dependents[1]. These chunks are Non-Recursive phrases (*nr*-phrases).

The first level, called word-level, is achieved within the framework of partial parsing also called shallow parsing (Voutilainen and Järvinen, 1995; Abney, 1996; Chanod and Tapanainen, 1996): it uses Part-Of-Speech Tagging and Chunking techniques in order to reliably and efficiently build up *nr*-phrases on unrestricted texts.

The second level, called *nr*-phrase-level, works with no explicit expectation on the input structure in order to deal with unrestricted text. We have developped our own approach to relations computation by revising the definition of specifying them. This approach allows a flexible management of both short and long dependencies. Moreover, it intrinsically enables the interaction with higher-level knowledge sources required in a framework of deep syntactic analysis.

Each of the two levels has its own process and its own representation; both representations are build simultaneously by the two interacting input-driven processes. The interaction is a requirement since many ambiguities arising at word-level can not be solved reliably there; the *nr*-phrase-level helps solving them.

Hereafter, we first describe the architecture of the parser. Then, we introduce the reader to the building of *nr*-phrases. Then, we emphasize on the way of linking *nr*-phrases together. After presenting how the two levels interacts, a concrete analysis is detailed. Then, we show the adequacy of the model, studying the resolution of some major linguistic problems. Finally, a precise evaluation is carried out, empirically demonstrating the adequacy of our novel concepts.

[1] Dependents are excluded in order to unify the representation of contiguous and discontiguous dependents.

2 The Architecture

The architecture of the process combines two techniques: (1) partial parsing, or more precisely Part-Of-Speech tagging techniques at word-level that build a constituent structure (each constituent is an *nr*-phrase); (2) dependency rules at *nr*-phrase-level that link *nr*-phrases to build a functional structure. In our approach, both constituent and functional structures are build simultaneously by two interacting processes. The analysis is carried out as shown in figure 1.

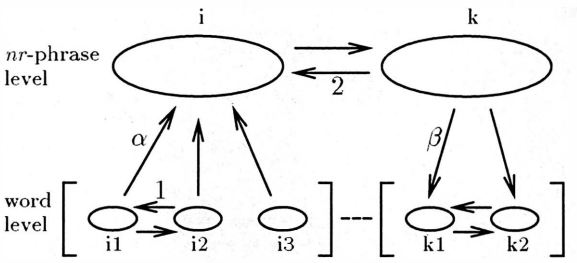


Figure 1: Process of analysis

Figure 1 shows two processes, labelled 1 and 2, managing respectively the word-level and the *nr*-phrase-level. The first process assigns tags to each part-of-speech and defines *nr*-phrase boundaries, shown as square brackets (section 3). The second process defines relations between *nr*-phrases (section 4). The two labels α and β show the interactions between word-level and *nr*-phrase-level (section 5). The execution of an entire basic cycle of deductions is successively: $1, \alpha, 2, \beta$.

The parse is done from left to right with a possible change of past deductions (and their consequences) when new knowledge invalidates them. Let us now describe precisely the two processes and their interaction.

3 Building Non-Recursive phrases

3.1 Framework

Word-level builds up *nr*-phrases with the help of an extended Part-Of-Speech (POS) tagger using hand-coded *affirmative* ϵ rules.

In rule-based POS tagging, two approaches exist. In the negative approach, a set of tags is first assigned to each token then contextual rules try to discard tags in order to avoid impossible contiguities. In the affirmative approach, contextual rules assign tags which are certain in the local context.

While tagging, two special tags are assigned as *nr*-phrase boundaries (beginning/end of *nr*-phrase). The *nr*-phrases correspond to "chunks" as defined in (Abney, 1996).

In this section, the problematics of tagging is assumed to be known and will not be completely detailed. Specific features required for the general understanding of the whole system are described and several tagging improvements are suggested.

3.2 Tag Assignment

The tagging is achieved in two steps: an initial step which can be seen as a bootstrap for a second step, called the deduction step.

The initial step is first to assign grammatical words with their most likely tag. One will note that during the deduction process, contextual rules may change these default tags. For instance, in French the ambiguity determiner/clitic[2] is solved by assigning the default tag *determiner*. Subsequently, contextual rules may change this tag into *clitic* if a personal pronoun, an other clitic or a verb appears in its local context. This process has a close relative in (Chanod and Tapanainen, 1995b; Constant, 1991). In (Brill, 1995), most likely tags are not only assigned to grammatical words (or most frequent words) but also to every other word.

[2] Clitic: French adverbial pronouns (*en,y*) and object personal pronouns (*le,la,les,lui,leur,l'*) always next to the verb.

In the initial step, we also use a lexicon containing all inflexional information on verbs (and their explicit polycategory) and a guesser that studies endings in order to give default assignments to other lexical words. Explicit polycategory of lexical words is pointed out by assigning sets of tags. During the deduction process, contextual rules may change the default tags to handle the incompleteness of knowledge(see section 3.4).

In the deduction step, all the rules have the following template: "the tag of this word is in this set of tags in this local context". Obviously, the set of tags contains only one item if the context is not ambiguous. If a set of tags is already assigned to a word, the intersection of the two sets is assigned to the word. If the intersection of the two sets is empty, the old set of tags is replaced by the new one.

While tagging, two special tags are assigned as *nr*-phrase boundaries (beginning/end of *nr*-phrase). The *nr*-phrases correspond to "chunks" as defined in (Abney, 1996). This is a generalization of "baseNP" used by (Collins, 1996) or of minimal NP to every other kind of phrases (e.g., verbal, adjectival). The syntactic features of *nr*-phrases are defined by the interaction α with the *nr*-phrase-level (see section 5.1).

3.3 Access to Knowledge Sources

In the earlier version of our system, access to the lexicon was always done first, then to morphological information and finally to contextual information. This strategy is also chosen in most systems but is this strategy the best one ?

Tests show that about 85% of the default tags that are assigned in the initial step are not changed by the contextual rules. However, contextual rules that change default tags always replace them ignoring their default value. Moreover, the changes done by the contextual rules are always right. Thus, in order to express that contextual information is always stronger than lexical and morphological information, the latter should only be accessed when no more information can be found through contextual deductions rather than be accessed solely in the initial step. Here is an analysis of the tagging process for *je le bois* (I drink it).
Notation:

$\xrightarrow{la}$: lexical access,
$\xrightarrow{ci}$: contextual information access.

Tagging with lexical access first:

$$je\ le\ bois \xrightarrow{la} je_{pp}\ le\ bois \xrightarrow{la} je_{pp}\ le_{det}\ bois \xrightarrow{la} je_{pp}\ le_{det}\ bois_{vb|nm} \xrightarrow{ci} je_{pp}\ le_{po}\ bois_{vb|nm} \xrightarrow{ci} je_{pp}\ le_{po}\ bois_{vb}.$$

Tagging with lexical access when required:

$$je\ le\ bois \xrightarrow{la} je_{pp}\ le\ bois \xrightarrow{ci} je_{pp}\ le_{po}\ bois \xrightarrow{ci} je_{pp}\ le_{po}\ bois_{vb}.$$

A partial implementation of this principle has been carried out and already shows a better interaction between the knowledge sources.

3.4 Handling Double Incompleteness of Lexicon

While processing unrestricted text, the problem of incompleteness of lexical knowledge is often pointed out. The incompleteness of lexicon is partly due to unknown words, such as domain-specific words, borrowings from one language to another, neologisms, spelling mistakes and so on (Chanod and Tapanainen, 1995a). It is also due to an interesting effect of missing categories for particular words in the lexicon.

The use of guessers is a traditional way to deal with unknown words but its sole advantage is to partially overcome one cause of the incompleteness. Furthermore, guessers can introduce irrelevant tags on unknown words and are efficient only on highly inflected languages. For these reasons, we have to find out complementary ways to handle incompleteness of lexical and morphological knowledge if we want to be able to deal with the common problems which usually arise while processing unrestricted text.

In this system, using the fact that contextual information is stronger than lexical and morphological information, we have added many affirmative contextual rules which can partially handle the two causes of incompleteness. For instance, a rule such as "after a personal pronoun, there is a verb" can handle sentences such as "*je positive.*" where *positive* is first person singular present of the verb *positiver*, a neologism that probably can not be found in a lexicon. This kind of rules can also deal with missing categories. For instance, if *bois* only occured as a verb in the lexicon (I drink), contextual rules could deduce it can also be a noun (wood) as in "*dans le bois*" (in the wood).

It is interesting to notice that negative constraints can not be used in this case since they discard tags assigned by a previous access to lexical or morphological information.

3.5 Reliability of Local Deductions

We saw in the introduction that *nr*-phrase order is fairly free, therefore, contiguous words of different *nr*-phrases are not supposed to have any relations.

With respect to this principle, when the local context defined by an *nr*-phrase contains too little information, some words should remain ambiguous since they can not be disambiguated reliably enough at word-level by POS tagging techniques. In other words, reliable deductions are those whose context has a scope restricted to one *nr*-phrase, using its inside strong syntactic constraints. Involving information outside this scope is not reliable since deductions that have a scope of several *nr*-phrases can not capture linguistic generalities.

Thus at word-level, we try to write rules propagating contextual deductions on words, and this solely within *nr*-phrases. Usually, these rules involve a grammatical word which helps categorizing the contiguous lexical words of their chunk. Remaining ambiguities are solved by the interaction with higher-level knowledge involved at *nr*-phrase-level.

4 Linking Non-Recursive phrases

4.1 Framework

The linguistic background of our work is based on the work of Lucien Tesnière (Tesnière, 1959). From his first approach to dependency definition "*Between a word and its neighbours, the mind foresees some connections.*", we have derived our own concepts for specifying *nr*-phrases relations. We have pointed out that the traditional static descriptive relation definition is not precise enough in order to be used efficiently in a parsing process. Thus, we introduce a dynamic analysis-oriented relation definition, that takes into account the linking constraints of the other relations. In our system, we have implemented this definition so that it handles all major dependency relations, the coordination relation and the antecedence relation.

The approach revises the notion of dependency as a relation between *nr*-phrases, and not between words, as opposed to (Covington, 1990; Covington, 1994; Tapanainen and Järvinen, 1997). As said in (Abney, 1996), "*By reducing the sentence to chunks* [i.e., *nr*-phrases], *there are fewer units whose associations must be considered, and we can have more confidence that the pairs being considered actually stand in the syntactic relation of interest, rather than being random pairs of words that happen to appear near each other*".

4.2 A new approach to relation specification

Dependency grammar-based formalisms allow for the specification of general relations by defining (1) the two structures being considered in the relation of interest and (2) static constraints existing between these two structures. This way of specifying relations leads to a failure since either the constraints on the structure of the two items are too relaxed and the silence is high, or they are too strict and the noise is too high.

Such static constraints on structures are unavoidable. Introducing constraints of possible or impossible occurences of structure between the two items can only lead to a failure since any such rule can be proved wrong within any unrestricted corpus.

Specifying a relation only with constraints on structure does not seem to be the right solution. Thus, we make the hypothesis that relations have to be defined with the help of three kinds of linguistic knowledge:

1. the two structures being considered in the relation of interest,
2. the constraints existing between these two structures,
3. the *other linking constraints* which, within the sentence, constrain these two items to be linked.

The two structures have to be computed *dynamically* in order to be robust. The constraints between the two structures are made of *static* knowledge. The interdependency of relations, as defined in point 3, has to be handled *dynamically*.

4.3 Dynamic handling of relation interdependencies

In our research, we have emphasized the handling of relation interdependencies which has become the predominant feature of our architecture. In other words, we have studied how the instanciation of a relation reduces the complexity of further decisions by discarding potential choices.

An example illustrates this general concept which happens to be the process definition.

Considering the sentence: "*The flight from Paris is cancelled because of a strike*"

By instanciating a subject-verb relation between *The flight* and *is cancelled*, the process discards *from Paris* as expecting any other relation. Thus, when *because of a strike* occurs it can only be attached to *The flight* or to *is cancelled*.

4.4 Implementation of the linking process

Our linking process is an implementation of the linguistic process described above.

The process is both data-driven and declarative: condition-action rules does not describe syntactic structures but the linking process. They manage both relations instanciation and linking constraints between relations. Relation instanciations are achieved in two distinct steps by two distinct kinds of rule actions:

1. *store* an *nr*-phrase as a candidate for some particular relations of interest in the relevant memories,
2. *attach* one *nr*-phrase to another one and *discard* some particular items as possible candidates for some particular relations from the relevant memories.

Thus, building up the functional structure is constrained by the interactions of the rules through the memories. In fact, when a rule discards an item in a memory, this corresponds to the death of a potential relation.

For instance, here are the two independent rules written to manage a subject-verb relation.

if the current *nr*-phrase
  is a nominal *nr*-phrase and
  is not object and
  is not already subject and
  is not attached to a preposition
then
  it is stored as possible subject
    into the subject-verb memory.

if the current *nr*-phrase
  is a verbal *nr*-phrase and
  there are possible subjects
    in the subject-verb memory
then
  retrieve the best fit subject from the memory
  attach the verb to this subject,
  discard this subject from the memory,
  discard items located between the subject and
    the verb from every memory.

The method which selects the best fit candidate is described in section 4.5.

The conditions within the rules allow the manipulation of: (1) relations in the dependency tree (defined by the functional structure); (2) heads of *nr*-phrases; (3) features of *nr*-phrases; (4) and status of the memories.

There are two kinds of actions within the rules: (1) actions on a memory (storing an *nr*-phrase and linking two *nr*-phrases, erasing the content of a memory, discarding an item in a memory), (2) actions on an *nr*-phrase (to change/add a feature).

The analysis is carried out from left to right. When an action updates *nr*-phrase features, coherence with word-level is achieved via interactions (see section 5.2).

Such an implementation requires the system to store candidates for possible expectations (e.g, nominal *nr*-phrase possibly expecting a verb for a subject-verb relation) and to retrieve the best-fit candidate for a particular relation. This ability is provided by the memory-based framework.

4.5 Memory-Based Framework

Memories as favoured places to perform relations

The process is based on a set of memories. Each memory is dedicated to the management of one specific relation (e.g, subject-verb, verb-object, coordination, PP attachment). A memory contains *nr*-phrases whose

association with a future *nr*-phrase must be considered. For instance, the memory that manages the subject-verb dependency relation contains nominal *nr*-phrases which can be involved in a future relation with a verbal *nr*-phrase.

The power of such an approach is that all relevant candidates are together in a single location and at the time when the relation has to be computed (a memory is a limited search-space): for a specific relation, the required knowledge sources can choose a successful candidate in the best conditions (see section 4.5).

Moreover, when the selection has to be performed, the process does not have to consider the past of the analysis but the current state of the memories. Therefore, far discontiguous relations are handled the same way as contiguous relations (if necessary, ways to distinguish them exist).

An other interesting point is that memories contain candidates for an association with a future *nr*-phrase. No requirement is made on the presence of the future *nr*-phrase. If such an *nr*-phrase never occurs then at the end of the sentence, the memory is erased: the candidates are forgotten. This means that when a new *nr*-phrase is added to a memory, no explicit expectation on structure is done, only implicit expectations are described by the rules. This kind of behaviour is to be related to tagging techniques and is fundamental to deal with unrestricted text.

Selection in a memory

Each memory is dedicated to the management of a specific relation. It is obvious that the knowledge required for selecting a candidate in the different memories is not always the same. In this system, every memory has its own specific method for choosing the successful candidate.

For instance, in our system, syntactic knowledge is involved for constraining the search space (i.e., the memory) depending on number, person and gender in a subject-verb dependency relation; similarity of structures is considered for coordination relation; psycholinguistic knowledge constrain the distance between the future associated *nr*-phrases.

It is interesting to point out that the above-mentioned knowledge sources are not enough to deal with complex phenomena. In memories, semantic and pragmatic knowledge sources can also interact with other knowledge sources to constrain the search space.

Focusing on the Subject-Verb memory

It is interesting to show in a concrete way how modularity of memories leads to flexibility, and to clarify how it helps us mastering the triggering of adequate knowledge sources and which items the triggered sources will act on.

The subject-verb memory is an example of such a memory where several kinds of knowledge are combined in order to handle the corresponding relation in a reliable and robust way. We will see that all the knowledge which deals with subject selection is clearly located in a single place:

- Syntactic constraints on agreement: these constraints are based on coordination relations, person and number of *nr*-phrases.

- Structural constraints on *nr*-phrase: they are involved in specific configurations in order to favour subject with determiner rather than subject without determiner.

- Basic semantic constraints are used to avoid some particular temporal NP to be taken as subject.

- This memory selects the leftmost possible subject close to the first barrier (e.g, a relative pronoun, a subordination conjunction) located on the left-hand side of the verb. This models the linking process of a subject with its verb, taking into account embedded clauses.

The latter shows the tight links between memories and the dynamic linking process which feeds them. Selection in memories is usually achieved with the help of a standard constraints relaxation mechanism.

5 Interaction between levels

As seen in section 2, the two levels are build up simultaneously, so that information has to be sent from one level to the other for the sake of coherence. The two processes parse the input from left to right. We choose to make deductions as soon as possible in the two structures even if, sometimes, this implies the ability to change some decisions when new information appears.

To every *nr*-phrase defined at *nr*-phrase-level corresponds a chunk of words delimited at word-level by two *nr*-phrase boundaries. Deductions from word-level to *nr*-phrase-level have a restricted scope: from words within a chunk to their corresponding *nr*-phrase, and vice-versa.

5.1 Interaction from word to *nr*-phrase

The interaction from word-level to *nr*-phrase-level (labelled α in figure 1) allows the assignment of features to *nr*-phrases. These informations activate the *nr*-phrases in the building process of the functional structure.

Un fichier d'incidents complète le dispositif. (1)

(A file of incidents completes the device.)

For instance, in sentence 1, when *masculine singular determiner* is assigned to "*Un*" at word-level, a nominal *nr*-phrase is created and the feature *masculine singular* is added to this *nr*-phrase.

5.2 Interaction from *nr*-phrase to word

The interaction from *nr*-phrase-level to word-level (labelled β in figure 1) is needed to constrain word features. In fact, we explained in section 3.5 that the local context of a word can contain too little information to be fully disambiguated.

For instance, in sentence 1, as soon as we know at *nr*-phrase-level that "*complète*" is a verbal *nr*-phrase (because it has been linked to its subject "*Un fichier*"), the tag *verb* is assigned to "*complète*" at word-level, thus removing the *adjective/verb* ambiguity.

6 Detailed Analysis

Here is a detailed analysis of the sentence "*Un fichier d'incidents complète le dispositif.*" (a file of incidents completes the device) as solved by our system. The word *complète*$_{adj|vb}$ has to be disambiguated. (1) and (2) are the two processes; (α) and (β) are the two interactions defined in section 2.

• 1st execution cycle :
(1) builds the chunk "*un fichier*" (square bracketed),
(α) creates the corresponding NP (oval in the Figure),
(2) stores this NP in two memories: as possible subject, and as possibly expecting a PP
(β) does nothing.

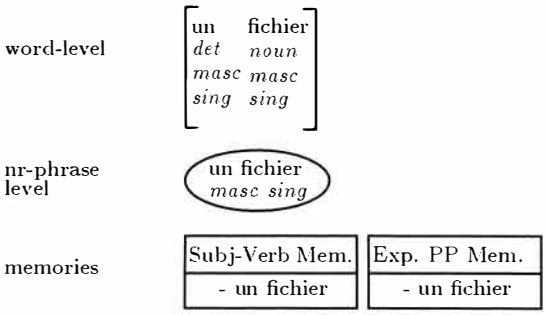


Figure 2: Cycle 1

• 2nd execution cycle:
(1) builds the chunk "*d'incidents*",
(α) creates the corresponding PP: PP_1,
(2) selects in the "Exp. PP memory" the NP "*un fichier*" as regent of PP_1, stores PP_1 as possible regent of an other PP.
(β) does nothing.

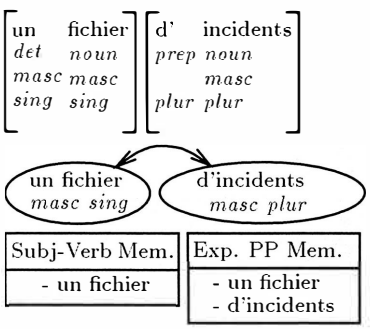


Figure 3: Cycle 2

• 3rd execution cycle:
(1) builds the chunk "*complète*" (the word "*complète*" is ambiguous),
(α) creates the ambiguous corresponding VP|AP (verbal or adjectival *nr*-phrase),
(2) supposes it is a VP, finds a matching NP as possible subject ("*un fichier*"), tags the ambiguous *nr*-phrase as VP, then it links this VP to the NP, and updates the memories,
(β) tags the word "*complète*" as *verb* for coherence.
This figure displays the chunk structure after (1) and the structure in progress at the end of the third cycle.

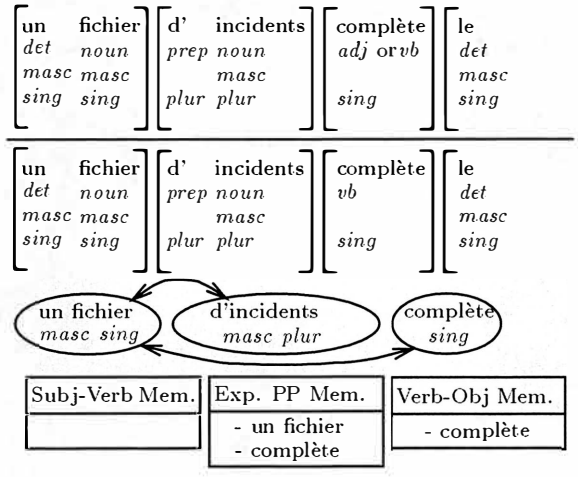


Figure 4: Cycle 3

• 4th execution cycle:
(1) builds the chunk "*le dispositif*",
(α) creates the corresponding NP,
(2) finds the matching VP "*complète*" to attach this NP as object, stores the NP as possibly expecting a PP.
(β) does nothing.

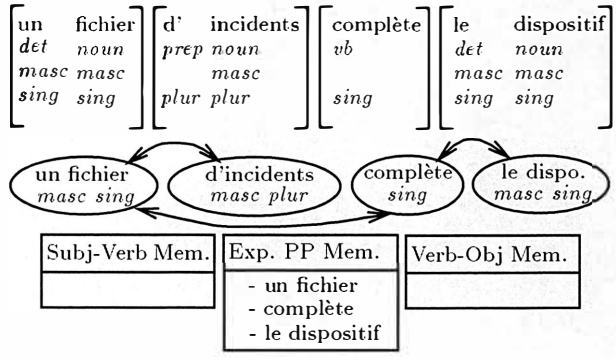


Figure 5: Cycle 4

At the end of the sentence, some memories contains items: they are erased.

7 Some linguistic studies

Some well-known difficult problems find a solution in this framework.

• *soit... soit* (either... or)
The first "*soit*" is either the subjonctive form of "*être*" (to be) or a conjunction (either).
"*C'est à l'inspection académique désormais de proposer aux familles, (...), soit un autre établissement - ce qui reviendrait sans doute à déplacer le problème - soit une formule de cours par correspondance (...).*"
In our framework, the previous example is processed as follows: the first "*soit*" is not the subjonctive form of "*être*" because no subject is available; it is stored in a memory, expecting an eventual second "*soit*"; when this second "*soit*" arrives, the first "*soit*" is found in the memory, and a coordination between the two nominal *nr*-phrases is set up.
The structure *ni... ni* (neither... nor) is solved in the same way.

• *plus... que* (more... than)
"*Les appels à la recomposition (...) me paraissent relever plus de la mise en valeur d'un parti socialiste syndical (...) que de la construction d'une réelle organisation syndicale indépendante.*"
"*plus*" is stored in a memory, expecting an eventual "*que*"; then, the following PP is linked to "*relever*"; when "*que*" occurs, its following PP is also linked to "*relever*".

• "*de*": *preposition* or *partitive* is a recurrent problem in French. It finds a natural solution in our system.
"*De nombreuses molécules sont transférées dans les réseaux trophiques marins.*"
"*De nombreuses molécules*" is an ambiguous *nr*-phrase: NP if the *nr*-phrase is either subject or object, PP otherwise. If it is NP then "*De*" is a *determiner*, otherwise it is a *preposition*: the solution is impossible to find

with POS tagging. In our system, when the verb "*sont transférées*" occurs, the process call the subject-verb memory to look for a subject, "*De nombreuses molécules*" is the only possible candidate: it becomes NP and an interaction tags "*De*" as *partitive*.

• Verbal ellipsis resolution:

Ellipsis resolution is a major difficulty to handle in traditional formalisms. In our approach, it can be handled as relation computation, defining it as a process.

"*Les lapins ne se contentent plus de luzerne, les cochons de pommes de terre et les poulets de grains de maïs*"
(Rabbits do not contend themselves with grass, pigs with potatoes and chicken with corn seeds.)

This kind of ellipsis can be handled in our model as an enumeration of possible clauses whose heads are subjects: *Rabbits*, *pigs* and *chicken*. This enumeration is managed as every other relation in a dedicated memory. When *with potatoes* occurs, its similarity with the verb dependent *with grass*, the absence of Subject-Verb relation involving *pigs* and the presence of a possible elliptic verb in the ellipsis memory allows *with potatoes* to be linked to *pigs* via a Subject-Verb-Object relation. Then, the Subject-Verb relation *pigs-contend* and Verb-PP *contend-with potatoes* can be computed.

8 Evaluation

The evaluation has been carried out on both structures build by the parser: a tagging evaluation and a relation computation evaluation.

The precise evaluation we offer on relations is restricted to subject-verb relations since no french treebank is available yet. However, it is possible to use our syntactic parse viewer on internet at http://www.info.unicaen.fr/~giguet (for Java-enabled browsers) to have an idea of the parser reliability for other relations.

8.1 Corpus Metrics

The evaluation of the parser has been carried out on a set of articles from the newspaper "Le Monde". This corpus has not been used to build up the parsing rules. This set is made of 24 articles (dealing with politics, economics, fashion, high-technology, every day life, ...) representing 474 sentences (max. length: 82 words, avg. length: 24.43 words). The definition of sentence is standard but includes two additional boundaries ";" and ":".

8.2 Tagging Evaluation

The corpus has been manually annotated by a linguist from the GRACE[3] tagging evaluation project, with the standard tagset proposed in MULTEXT[4] and EAGLES[5] projects. This tagset is made of 11 main categories. Each category can be completed with a maximum of 6 attributes which can take their value from a set containing up to 8 distinct values.

For the needs of the evaluation, a translation function has been written to convert our tagset into the tagset of the annotated corpus.

The fine grain tokenization (e.g., apostrophies are tokens) makes 12691 tokens appear.

In the protocol, an assigment is (1) *correct* if the parser assigns one tag with the correct value to every field, (2) *ambiguous* if the parser assigns more than one tag or more than one field value but the correct tag exists, (3) *incorrect* if the correct tag can not be found.

Two evaluations are carried out. In the first one, the tag has only one field: the main category. In the second one, the tag is composed of the main category and the relative attributes.

Figure 6 presents the results. These results are not the official results of the GRACE evaluation project since the evaluation contest is not started yet.

These results are computed from the output of the parser, that is, not only with process 1 at word-level but also with interactions β generated at *nr*-phrase-level when relations are computed by process 2 (see figure 1). About 1% of the correct tags are computed thanks to a relation computation, using interactions β, the others are computed at word-level.

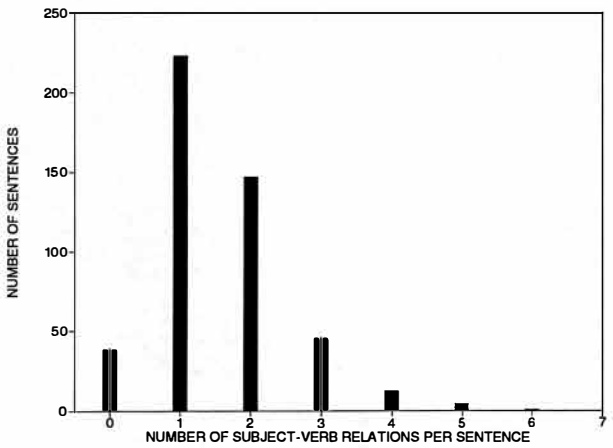
[3] http://www.ciril.fr/~pap/grace.html
[4] http://www.lpl.univ-aix.fr/projects/multext
[5] http://www.ilc.pi.cnr.it/EAGLES/home.html

Evaluation	tokens	correct	ambiguous	incorrect	% correct
Complete Tag	12691	11502	516	673	**90.63%**
Main Category	12691	12524	0	167	**98.68%**

Figure 6: Tagging Evaluation

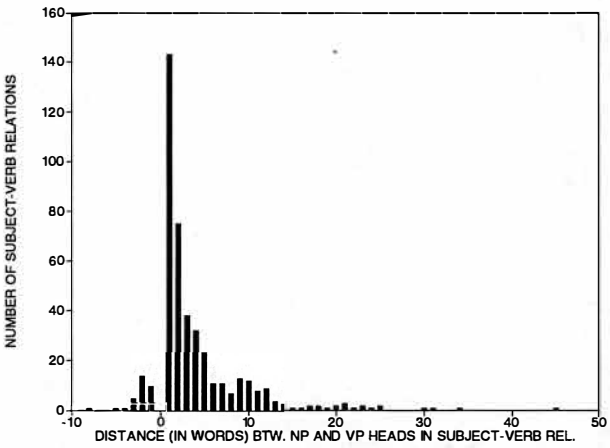
8.3 Relation computation evaluation

Subject-Verb relations in the corpus



nature of subject	number
NP	458
Infinitive VP	2
Relative Pronoun	85
Personal Pronoun	193
Total	738

Figure 7: Subject-Verb Relations



Max. dist. btw. NP-subject and VP heads	
in standard relations	45 words
in inverted relations	8 words

Figure 8: NP-subject in Subject-Verb Relations

In this corpus, there are 738 Subject-Verb relations. Figure 7 shows the span of subject-verb relations in the sentences: 39 sentences do not have subject verb relations and the maximum number of subject-verb relation per sentence is 6. According to the nature of the subject, we distinguish 4 kinds of SV relations: relations involving (1) an NP subject, (2) an Infinitive VP subject, (3) a Relative Pronoun subject and (4) a Personal Pronoun subject.

An other interesting metric is the distance between the verb and its subject. Figure 8 illustrates this phenomenon only for NP-subject. This metric is less relevant for other relations, even if several sentences contain personal pronouns and relative pronouns that are far subjects, for instance in cases of verb enumeration or prepositional phrase insertion. The figure shows that the distance between an NP-subject and its verb can reach up to 45 words.

Evaluation on Subject-Verb Relations

The evaluation function is based on the following principle: every verb has to be attached to no more than one subject. From this starting point, 3 cases exist: it is a *correct* relation if the verb is attached to the expected subject, *incorrect* if not and a *silence* if no subject is provided but one was expected.

In cases of subjects coordination, each verb depending on the coordination has to be attached to the head of this coordination, that is, to the head of the first item. In cases of verbs coordination, one correct relation counts for each verb attached to the expected subject and one incorrect relation for each verb not attached to the expected subject.

The results are listed in Figure 9. *Precision* is the ratio of correct links over the number of computed links. *Recall* is the ratio of correct links over the number of expected links. The very high rate that we report (96.39% precision and 94.04% recall) empirically validate the approach of defining relations as a linguistic process.

Our results can still be improved since this evaluation was the first on large corpora. The 42 silences and incorrect relations can be classified in 5 categories: (1) incorrect implementation of agreement check, (2) illformed *nr*-phrases, (3) coordination not found, (4) inverted subject in reported speech, (5) incorrect *nr*-phrase tags. We have pointed out better ways of solving the three first classes. The fourth and fifth classes requires further studies to be carried out in a general way.

Nature of subject	number	correct	incorrect	silence	precision	recall
NP	458	418	26	14	94.14%	91.27%
Infinitive VP	2	2	0	0	100.00%	100.00%
Relative Pronoun	85	85	0	0	100.00%	100.00%
Personal Pronoun	193	191	0	2	100.00%	98.96%
Total	738	694	26	16	**96.39%**	**94.04%**

Figure 9: Evaluation on Subject-Verb Relations

Comparison with other systems

It is still difficult to compare these results with other french systems since no strictly comparable experiment has been carried out under such difficult evaluation conditions. However, Xerox has evaluated its last incremental finite-state parser(Aït-Mokhtar and Chanod, 1997) for *subject recognition*. This task is less complex than ours since subject-verb relation computation includes the resolution of both subject and verb coordinations. On a half-sized evaluation corpus from "Le Monde" (5872 words, 249 sentences), their parser achieves 92.6% precision and 82.6% recall.

9 Conclusion and Future Work

We have described a system for syntactic parsing of unrestricted French. The contribution of this work can be summarized in two points. First, we have shown that a restriction of POS tagging to deductions within *nr*-phrase could lead to better global results since an interaction with the linking process is more powerful for deductions between *nr*-phrases. Second, we have provided a flexible memory-based framework for unrestricted relations and a process which has no explicit expectation on structures.

The success of our system is due to the new approach to computing relations: dynamically taking into account all the relevant relations which, within the sentence, constrain the two items to be linked.

The result is a flexible architecture which has the ability to handle in a natural way all the major syntactic relations in a unique framework: standard relation such as subject-verb, verb-object, PP attachment, but also complex relations such as coordination, enumeration, apposition, antecedence, and ellipsis.

Running on a collection of newspaper articles from "Le Monde" (11583 words, 474 sentences and 739 subject verb relations) where very complex structures appear, we get 96.39% precision and 94.04% recall for suject-verb relations. These first results empirically validate the approach and we can say the parser is very reliable for this relation. Moreover, it is robust since one parse is always provided (sometimes a partial parse). The present version of the linking process is very efficient: it is deterministic and it has a linear complexity in time. Today, we are working on a slightly modified version of the parsing process in order to enable new knowledge to change past deductions. In this case, these deductions and their consequences are discarded.

We now have to continue precise evaluation of our parser for all the other kinds of relations (a hard work since no treebank is available at the moment) and generally to continue improving the parser. A demo is available at http://www.info.unicaen.fr/~giguet.

References

Steven Abney. 1996. Part-of-speech tagging and partial parsing. In Ken Church, Steve Young, and Gerrit Bloothooft, editors, *An Elsnet Book*, Corpus-Based Methods in Language and Speech. Kluwer Academic, Dordrecht.

Salah Aït-Mokhtar and Jean-Pierre Chanod. 1997. Incremental finite-state parsing. In *Proceedings of the fifth Conference on Applied Natural Language Processing (ANLP'97)*, pages 72–79, Washington, DC USA, April. Association for Computational Linguistics.

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, December.

Jean-Pierre Chanod and Pasi Tapanainen. 1995a. Creating a tagset, lexicon and guesser for a french tagger. In *Proceedings of the European Chapter of the ACL SIGDAT Workshop "From text to tags : Issues in Multilingual Language Analysis"*, pages 51–57, Dublin, Ireland, March.

Jean-Pierre Chanod and Pasi Tapanainen. 1995b. Tagging french — comparing a statistical and a constraint based method. In *Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics (EACL'95)*, Dublin, March. Association for Computational Linguistics.

Jean-Pierre Chanod and Pasi Tapanainen. 1996. A robust finite-state parser for french. In *ESSLLI'96 workshop on robust parsing*, Prague, Czech, August.

Michael John Collins. 1996. A new statistical parser based on bigram lexical dependencies. In *Proceedings of 34th Annual Meeting of the Association for Computational Linguistics (ACL'96)*, University of California, Santa Cruz, June. Association for Computational Linguistics.

Patrick Constant. 1991. *Analyse Syntaxique Par Couches.* Ph.D. thesis, École Nationale Supérieure des Télécommunications, April.

Michael A. Covington. 1990. A dependency parser for variable-word-order languages. Technical Report AI-1990-01, Artificial Intelligence Programs, The University of Georgia Athens, Georgia 30602 USA, January.

Michael A. Covington. 1994. Discontinuous dependency parsing of free and fixed word order: Work in progress. Technical Report AI-1994-02, Artificial Intelligence Programs, The University of Georgia Athens, Georgia 30602 USA.

Pierre Le Goffic, 1993. *Grammaire de la Phrase Française*, chapter 1-2, pages 7–51. Hachette Éducation.

Molière, 1670. *Le Bourgeois gentilhomme*, pages 39–48. Presse Pocket. Acte II Scène 4.

Pasi Tapanainen and Timo Järvinen. 1997. A non-projective dependency parser. In *Proceedings of the fifth Conference on Applied Natural Language Processing (ANLP'97)*, pages 64–71, Washington, DC USA, April. Association for Computational Linguistics.

Lucien Tesnière. 1959. *Éléments de syntaxe structurale.* Klincksieck, Paris.

Atro Voutilainen and Timo Järvinen. 1995. Specifying a shallow grammatical representation for parsing purposes. In *Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics (EACL'95)*, Dublin, Ireland, March. Association for Computational Linguistics.