3 ACL European Chapter Conference
Copenhagen, April 1987

FRAGMENTATION AND PART OF SPEECH DISAMBIGUATION

Jean-Louis Binot
IBM Thomas J. Watson Research Center,
P.O. Box 704
Yorktown Heights, N.Y. 10598

ABSTRACT

That at least some syntax is necessary to support semantic processing is fairly obvious. To know exactly how much syntax is needed, however, and how and when to apply it, is still an open and crucial, albeit old, question. This paper discusses the solutions used in a semantic analyser of French called SABA, developed at the University of Liege, Belgium. Specifically, we shall argue in favor of the usefulness of two syntactic processes: fragmentation, which can be interleaved with semantic processing, and part-of-speech disambiguation, which can be performed as a preprocessing step.

FRAGMENTATION AND PART OF SPEECH DISAMBIGUATION

Jean-Louis Binot
IBM Thomas J. Watson Research Center,
P.O. Box 218
Yorktown Heights, N.Y. 10598

ABSTRACT

A crucial question for "integrated" NL processing systems is to know how much syntax we need to support semantic processing, and how and when to use it. This paper offers a partial answer to this question. Specifically, we shall argue in favor of two syntactic processes: fragmentation and part-of-speech disambiguation.

1. THE ROLE OF SYNTAX.

(Lytinen 86) distinguishes two approaches to NL processing. Followers of the "modular" approach believe usually in the autonomy of syntax and in the usefulness and cost-effectiveness of a purely syntactic stage of processing. Results of this approach include the development of new grammatical formalisms (Shieber 85) (Weir et al. 86) (Ristad 86), and of large syntactic grammars (Jensen et al. 86). Followers of the "integrated" approach, on the contrary, believe that semantics should be used as soon as possible in the parsing process. Exactly how the integration between syntax and semantics should be done, however, is still an open question. Some integrated systems, such as IPP (Schank et al. 80) and Wilks' system (Wilks 75), were trying to reduce the role of syntax as much

as possible. Lytinen proposes a more moderate option in which separate syntactic and semantic rules are dynamically combined.

In this paper, we shall present some additional arguments in defense of integration, and offer our partial answer to the questions it creates. Specifically, we shall argue in favor of two syntactic processes: fragmentation, which can be interleaved with semantic processing, and part-of-speech disambiguation, which is usefully performed in a preprocessing step. These processes have been implemented in a **robust** and **portable** semantic analyser of French called SABA (Binot 85), which was developed in LISP at the university of Liege, Belgium, and tested successfully on a corpus of 125 French sentences. SABA is not based on a French grammar, but on semantic procedures which directly build a semantic dependency graph from the natural language input (Binot and Ribbens 86). In the present paper, we discuss the kind of syntactic support needed for this type of semantic processing.

2. FRAGMENTATION

Consider sentence (1) below and suppose that a purely semantic system were to understand it by establishing semantic dependencies between words:

(1) **Le pont que le convoi a passe quand il a quitte New York etait tres long.** (The bridge that the convoy crossed when it left New York was very long.)

There would be no reason for such a system to refrain from attempting to connect "long" to "convoy", for example. More important, if the attempt is made, then no amount of semantic or pragmatic knowledge will be able

to prevent the connection, which is perfectly valid as such. Note also that a simple proximity principle would not work in this case.

Thus, any natural language processing system must take into account, in some way, the **structure** of a sentence. However, we don't necessarily need to build an intermediate syntactic structure, such as a parse tree, showing the detailed "phrase structure" of the input. The most crucial structural information needed for an accurate semantic processing concerns "boundaries" across which semantic processing should not be allowed to relate words. These boundaries can be identified by a fragmentation process which will cut a sentence into useful fragments by looking for specific types of words.

Except maybe in Wilks' system fragmentation has not received the attention it deserves as a faster alternative to full syntactic parsing. The basic problems of fragmentation are to determine the most useful fragment size(s) and to decide how and when fragmentation should be performed. We have implemented a hierarchic fragmentation algorithm for French where sentences are fragmented into clauses and clauses into nominal groups. This algorithm is is repetitively interleaved with semantic processing:

Fragmentation algorithm: repeat these steps until success or dead end[1]

1. Fragment the sentence into clauses;
2. Select the innermost clause;

---

[1] Dead ends (if fragmentation can find no new clause or if the selected clause cannot be processed) are handled by a backtracking mechanism which modifies fragmentation or selection choices.

3. Process the selected clause, which includes:

   a. The fragmentation of the clause into groups;

   b. The establishment of semantic dependencies inside each group;

   c. The establishment of semantic dependencies at the clause level;

4. If the processing is successful, erase the text of the clause from the input and replace it by a special non-terminal symbol.

Fragmentation of a sentence into clauses proceeds by extending to the left and to the right of each verb[2] and checking each encountered word looking for clause delimiters. The checks are performed by heuristic rules based on the part-of-speech of each word. Other rules will then look at the delimiters to find the innermost clause. In this summary, we shall simply illustrate the effect of the rules on example (1). Figure (2) below shows the successive states of the input text, with the last fragmentation result indicated by underlining.

(2) Le pont que le convoi a passe quand il a quitte New-York etait tres long.

Le pont que le convoi a passe PC etait tres long.

Le pont PR etait tres long.

PP[3]

A single fragmentation pass yields imperfect results. There can be **holes** (sentence fragments which are not included in any clause, like "Le pont" in the first two steps) and **overlappings** (like "New-York" in the first

---

[2] Except auxiliaries that are part of a compound verbal form.

[3] The non-terminal symbols PC, PR and PP represent respectively a conjunctive clause, a relative clause and a main clause.

step). This is why fragmentation is repetitive. Successive erasing of the innermost clauses from the input text, once they have been processed by the semantic module, will gradually make the holes disappear, and thus reveal the content of the main clause(s). Overlapping fragments will be kept in the first clause in which they are semantically acceptable.

3. PART OF SPEECH DISAMBIGUATION

Many lexically ambiguous words can have different parts of speech (hereafter POS). The following table enumerates the main POS ambiguities for example (1).

Le (occurs twice): article or personal pronoun (the, him, it)
que: subordinate conjunction, relative or interrogative pronoun, particle (that, which, what, than)
quand: subordinate conjunction or adverb (when)

The ambiguity problem is further compounded by an accentuation problem. "Passe", third person of the present of the indicative of the verb "passer", is quite different in French from "passé", past participle of the same verb.[4] Similarly, "a", indicative of avoir ("to have"), has nothing to do with the preposition "à". However, forgetting an accent is one of the most common spelling mistakes. A robust system such as SABA must consider words such as "a" or "passe" as ambiguous. This would give at least 128 possible POS combinations for example (1)! The size of the

---

[4] Verb mood ambiguities can usefully be considered at the same level as POS ambiguities.

problem shows the interest of having a POS disambiguation preprocessor, which could avoid or greatly reduce combinatorial explosion.

We have developed a POS disambiguation preprocessor for French which is used as the first stage of the SABA system. This preprocessor, applying the expert system methodology, is more flexible than an earlier attempt of the same kind for English (Klein and Simmons 63). It consists of heuristic rules which are applied to each word in order to assign to every possible POS a certainty factor. Possible POS combinations are then tried in decreasing order of likeliness.

These heuristics use the fact that it is not necessary to scan the entire sentence to choose the appropriate POS for most words. The "local context" (i.e. the few surrounding words) proves often enough to provide an accurate indication. If a word like "passe" is closely preceeded by an auxiliary, it is almost certainly a participle. "Le", if closely followed by a noun, is more likely to be a pronoun than an determiner. As an illustration, the exact formulation of this last rule in our system is given below:

Rule 8: If the current word can be a determiner or a pronoun, then:

1. If it is followed by a word that could be a noun or a pronoun, and is only separated from it by words that could be adjectives, adverbs, determiners or conjunctions, then:

   determiner CF = 0.9; other possible POS CF = 0.1;

2. If it is followed by a word that could be a verb, and is only separated from it by words that could be pronouns, then:

   pronoun CF = 0.9; other possible POS CF = 0.1;

3. else: pronoun CF = 0.6; other possible POS CF = 0.4.

Each rule can be seen as a production rule with a condition and an action. The condition is the clause starting with the first "if" of the rule; it checks whether this particular rule should be applied to the current word. The action is often itself a conditional statement, each branch of which must include assigments of certainty factors. Certainty factors range from 0 (absolute uncertainty) to 1 (absolute certainty).

The POS disambiguation preprocessor processes successively all the words of the input, trying to apply all rules to each word. If several rules can be applied to the same word, certainty factors are combined by the following formula: CF = 1 - ((1 - CF1)*(1 - CF2)) where CF1 and CF2 are the certainty factors to be combined. When this is done, possible POS combinations are ordered by decreasing order of likelihood. The likelihood of a combination is simply defined as the product of the certainty factors of the parts-of-speech included in that combination.

4. RESULTS AND CONCLUSIONS

We have presented two syntactic processes which offer sufficient support for semantic processing and which are faster than a classical syntactic parser. Both are based on simple heuristic rules assisted by a backtracking mechanism. The rules that we designed for French make the right choice in more than eighty percent of the cases. We

believe that such an approach could be applied to other languages as well.

REFERENCES

1. Binot, J-L. SABA: vers un systeme portable d'analyse du francais ecrit. Ph.D. dissertation, University of Liege, Belgium, 1985.
2. Binot J-L. and Ribbens D. Dual frames: a new tool for semantic parsing. In Proc. AAAI86, Philadelphia, 1986.
3. Jensen K., Heidorn G.E., Richardson S. and Haas N., PLNLP, PEG and CRITIQUE: three contributions to computing in the Humanities. In Proc. of the conf. on Computers and Humanities, Toronto, 1986.
4. Klein S. and Simmons R.F. A computational approach to grammatical coding of English words. Journal of the ACM. 10, March 1963.
5. Lytinen S.L. Dynamically combining syntax and semantics in natural language processing. In Proc. of AAAI86, Philadelphia, 1986.
6. Ristad E.. Defining natural language grammars in GPSG. In Proc. of the 24th meeting of the ACL, New-York, 1986.
7. Schank R.C., Leibowitz M. and Birnbaum L. An integrated understander. In Journal of the ACL, 6:1., 1980.
8. Shieber S. An introduction to unification-based approaches to grammar, tutorial session, 23rd meeting of the ACL, 1985.
9. Weir D.J., Vijay-Shanker K. and Joshi A.K. The relationship between Tree adjoining grammars and head grammars. In Proc. of the 24th meeting of the ACL, New-York, 1986.
10. Wilks Y. An intelligent analyser and understander of English. CACM 18:5, 1975.