project note with software demonstration

A parser without a dictionary as a tool for research into French syntax

Jacques VERGNE

LIMSI 29 rue Titon
F-75011 Paris **France**

A natural language is not a formal language

That is why the syntax of a natural language may not be described by the rules of the syntax of a formal language. And that is why a natural language cannot be parsed the same way as a formal language. To parse a natural language, it would be necessary to know its syntax.

Thus, major difficulties in parsing a natural language are not algorithmic but linguistic.

Here, I assume that natural language has **a very high formal redundancy**. In other words, morpho-syntactic clues are numerous enough to make deductions upon categories and relations in many converging ways.

Research into French syntax then consists in discovering these clues: morphology of words, agreements, relative positions of elements, segmentation in NPs, or in larger segments, typology of relations, structure of the relations net, formal and quantitative constraints upon this net.

The parser presented here is **an experimental device** used to test and confirm the linguistic hypotheses upon large text corpora. The formal redundancy is high enough to parse without a dictionary with very few data (300 ending rules and a lexicon of grammatical words of 4 kB) and a NP stack grammar.

Linguistic hypotheses

A sentence is considered as a stack of NPs

In our Western grammar traditions, action is the main interest and we place verbs in a central position.

But, in scientific texts, NPs are the main carrier of meaning, as they name concepts. They are also used as terms for indexation. From the statistical point of view, too, nouns are by far the most numerous category.

That is the reason why the basis for this parser is that the sentence is considered as **a stack of NPs**, with relations of determination between them. An initial NP is laid (it is often the focus of the sentence), and the following phrases precise and determine it. Here determination is considered as adding more data. These different NPs are connected in a more or less narrow way by "links":

- "non verbal links": prepositions, co-ordinations, subordinating conjunctions, relative pronouns,
- verbs in all their forms: conjugated, present and past participles, infinitives.

Thus the processing of the verb is unified in such a manner that every verb, conjugated or not, of a main or subordinated clause, acts as an adjective (is a "translaté" to adjective, according to Tesnière's concept, in [Tesnière 59]), determines a NP and is a link in the "chain" of NPs.

Therefore, the NP is considered as the basic and constitutive element of the sentence (as the cell is the basic and constitutive element of the living tissue). The inside and the outside of the NP are processed separately and differently, so we have:

- a grammar of the NP, centered on a nominalized word (like the inside structure of the cell, centered on its nucleus),
- and a grammar of the sentence, outside the NP, which is expressed in terms of NPs, each of them considered as a closed entity (like the structure of the tissue expressed as an architecture of cells). Let us name this grammar the "**NP stack grammar**" or "**NPSG**".

Inside the noun phrase

The internal grammar of the NP is centered on a nominalized word, a noun, a nominalized adjective or a verb. It reigns over its determinant and adjectives at the root of a dependency-determination tree.

The branches of the tree are made of partitive, determinant, indefinite adjective, anteposed adjective (very rare in scientific texts) before the nominalized word, and contiguous or co-ordinated postposed adjectives, after the nominalized word. The nominalized word is determined and qualified by the other words of the NP. This grammar has been presented for instance in [Vergne 86] and developed in [Vergne 89].

Outside the noun phrase

The **NP stack grammar** or **NPSG**, is used to:

- validate the sentence structure as a stack of NPs,
- confirm the function of words external to NPs,
- compute some tight relations external to NPs, such as verb-object.

Valuation functions are used to compute other looser relations external to NPs, such as prepositional phrase attachment or co-ordinations.

Relations typology

I propose to distinguish three types of relations:

-**1**- relations internal to the NP (mainly the determination noun ← adjective). These relations are computed during the internal analysis of a NP.

-**2**- relations external to the NP, but *internal to a recognition pattern*, as for instance, the relations subject ← verb and verb ← object in a SVO sentence. These relations are computed at the recognition time during the validation of the sentence at the NP level. This computation is **algorithmic**.

Nota bene: to avoid the Chomsky's term governer, I propose the term **reigner** for Tesnière's concept "**régissant**" (in [Tesnière 59]). They have the same etymology, and both are verbal derivatives:

the **reigner** reigns over its **dependents**.

-**3**- relations external to the NP, and *external to the recognition patterns*, as for instance, the relations "reigner" ← prepositional phrase (PP): the PPs are recognized by a pattern of the form: preposition-NP, a pattern which does not contain the reigner, which is either a verbal "link", or a determined NP. These relations are computed by **valuation functions**. The computation then is heuristic.

The three types of relations are determination relations. They proceed from the tighter ones, inside the NP (-**1**-), to the looser ones, outside the NP, and outside the recognition patterns (-**3**-).

The NP stack grammar

Validating the NP stack pattern

At the beginning of this step of the parsing, the sentence is represented by a pattern made of a sequence of letters, in which each letter represents either a NP, or a word external to the NP (preposition, verb, for instance).

It is possible to imagine this pattern as an **artichoke**, made of **leaves** around the **heart**.

Validating the pattern then consists in plucking off parts of it progressively:

- the leaves are replaced or removed in a precise order, until the heart is reached:
  - leaves are replaced by context sensitive rules which erase negations, adverbs, auxiliaries;
  - leaves are removed by context free rules which remove everything else but the heart;
- simultaneously, each time a leaf is replaced or removed, the relations internal to this leaf are computed (relations of type -**2**- in the typology exposed above).

That is the reason why I name this parser: "**plucking off parser**" or "**POP**".

The final state of the pattern once plucked off must be one of the different possible hearts. They are:

- a NP alone,
- a NP determined by other NPs through a conjugated verb,
- or a NP determined by an attribute, a past participle or a NP through an auxiliary "être".

Transposing relations internal to a leaf by simulated reclothing

Relations which are internal to a leaf pattern must be transposed into the entire NP level pattern. From the positions in a leaf pattern, we are able to compute the positions in the entire NP level pattern.

To retrieve these absolute positions, we have only to simulate the reclothing of the heart, by using the historical account of the plucking off. After simulated reclothing (by applying the rules in the reverse order), we obtain the absolute positions in the entire NP level pattern.

In a later step of the parsing, after the internal analysis of NPs, these relations will be transposed into the word level pattern.

In such a way, all relations internal to a leaf pattern are computed inside the leaf pattern, then transposed by simulated reclothing into the entire NP level pattern, then at last transposed into the word level pattern.

These two transpositions may be seen as **reference point changes**, from a relative position in the leaf pattern (NP level), to an absolute position in the entire pattern (word level).

Valuation functions: an heuristic way to choose

Valuation functions have been described in detail in [Vergne 89].

Principle

A valuation function is a clear and fine way to express an heuristic, when criteria are too fuzzy to make a choice with an algorithm (a binary tree of "if then else" for instance).

The objective is to make an automatic choice without an algorithm. The principle is the following:

- Determine the objects to valuate: the candidates distinguished from the non candidates.
- Quantify valuation of the candidates, using criteria to discriminate them. The criteria represents the knowledge we have about the phenomenon.
- The candidate who obtained the higher valuation is choosen.

When to use valuation functions

Valuation functions are used to compute the relations of type -**3**- :

- to search for the "reigner" of a nominal or infinitive PP, of a present participle or of a gerundive introduced by "en", of a past participle, of a subordinated clause;
- to search for the left co-ordinated of a NP, of a nominal or infinitive PP, of an infinitive, of an attribute, of a main or subordinated clause;
- to search for a referent which agrees with an anaphoric.

Using valuation functions with formal criteria is based on the hypothesis of the **high formal redundancy of natural language**.

Attaching prepositional phrases

principle:

At the beginning of the computation, a "**power to reign**" is affected to each word according to its category and its eventual **verbal** derivational origin.

This computation is thought of as the simulation of the conflicts that words have between them to reign over other words. During this computation, powers are variable: cancelled, reduced, augmented or transmitted.

the valuation formula:

The function used to valuate a reigner-candidate is a linear function of its power and of its distance to the PP. The valuation is the apparent power of the reigner-candidate, when seen from the dependent.

The apparent or relative dimension of an object depends on its absolute dimension, and on the distance between the object and the observer. It is possible to think too of the physical analogy of the field, a concept which gives an explanation of the influence of an object on the objects in the space around: the universal attraction (gravitation field), for instance, is governed by a rule in 1 / distance ^{2} . It is the purpose here to modelise the influence of a reigner over its dependents.

The candidate is valuated according to the following formula:

relative power = 2 * absolute power - 10 * distance

error detection:

If two valuations are very close, the higher is choosen, as usual, but marked as uncertain.

final output:

Then the reigner is lemmatized, and the relation is output, with its type.

If the dependent is co-ordinated, the dependency of the right co-ordinated noun is output too. It has the same reigner as its left co-ordinate.

Computing co-ordinations of noun phrases

principle:

The valuation function used to compute the co-ordinations is a linear function consisting of criteria that finely measure the isomorphism of the two co-ordinated phrases. It depends neither on the powers nor on the distances. Note that this isomorphism is not the fundamental feature of co-ordinated phrases, but that, more globally, it is the most common mark of an "isofunctionality", perhaps also for some aesthetic reasons: euphony, taste for balance, symmetry, repetition (of structure).

the valuation criteria:

- a first group of criteria is used to measure the isomorphism of the determinants of the two potential co-ordinated NPs: both *definite determinants*, both *indefinite determinants* or both *without determinant*, both *possessives* or both *demonstratives*;
- a second group of criteria is used to measure the isomorphism of nouns: identical nouns, both nominalized verbs, or same number;
- a criterion concerns the "isoqualification" of nouns: both qualified or both not qualified.

Other features of the parser

Technical realization

- programming language: Turbo Pascal
- machine: Macintosh
- source size: ≈ 16 000 lines
- code size: ≈ 340 Ko
- research and development: ≈ 3 years-man

Performances on a corpus of about 10 000 words

- processing speed: ≈ 3 words per second on Mac II
- category errors: < 1% / words
- relation errors: < 3% / relations

Output

The parser outputs the results into following files:

- results, sentence by sentence: relations, features of each word, and the determination tree;
- other results, grouped and classed on the whole text:
  - the lexicon of the text, lemmatized, classed by category, and reversed,
  - relations of the text, classed by syntactic type of relation,
  - problems met during the parsing, classed by type of problem,
  - statistics about text size, processing speed, number of tested patterns, categories, deduction modes, relations and patterns of the text.

Conclusions

The NP stack grammar is based on **the central place of the noun phrase.**

It allows, before the internal analysis of NPs, to verify a sentence pattern, by a fast and non recursive algorithm.

The typology of relations gives clearly the separation between relations the computation of which can be algorithmic and relations the computation of which must be heuristic.

Using valuation functions to compute relations is an efficient mean to exploit as best as possible purely formal criteria, without using semantic information, thus confirming the **high formal redundancy of natural language.**

Quoted references

[**Tesnière** 59] Lucien **Tesnière**: *Eléments de syntaxe structurale* Klincksieck (Paris) 1982

[**Vergne** 86] Jacques **Vergne**, Pascale **Pagès**: *Synergy of syntax and morphology in automatic parsing of French language with a minimum of data* Coling 86 International Conference on Computational Linguistics pp. 269-271, Bonn, august 1986

[**Vergne** 89] Jacques **Vergne**: *Analyse morpho syntaxique automatique sans dictionnaire* thèse de doctorat de l'Université Paris 6, june 1989