CoLing 90 Helsinki

project note with software demonstration

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

Jacques VERGNE

Université de Caen JVergne@ucaen.univ-caen.fr

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

Aim and methodology

The aim is to better understand the mechanisms of syntax. The method (in fact, the scientific method) consists in working on large corpora, making fine **observations**, then building linguistic **hypotheses**, and testing them on the corpora, using the parser as **an experimental device**.

Linguistic observations

A natural language is not a formal language

Here are some important differences between natural and formal languages:

criteria	natural language	formal language
origin	natural	artificial
form / meaning mapping	many-to-many	one-to-one
explicitation	partial	total
lexicon	infinite evolutive	finite fixed
syntax	not well-known evolutive	well-known fixed
formal redundancy	very high	null or artificial

A natural language has a very high formal redundancy

Redundancy is a common property of information circulating inside living organisms. Its most important consequence is that information is more safely transmitted.

In a natural language, morpho-syntactic clues are numerous enough to make deductions upon categories and relations in many converging ways.

Research into 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 nets, formal and quantitative constraints upon these nets.

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 syntax.

A sentence may be considered as a stack of NPs

An initial NP is laid (it is often the topic of the sentence), and the following phrases precise and determine it. Determination here 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 verb syntax is unified in such a way 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 (and not the word) 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 NP syntax and a sentence syntax:

• the internal NP syntax is centered on a nominalized word (noun, nominalized adjective or verb), like the inside structure of the cell, centered on its nucleus. The nominalized word reigns over its determiner and adjectives at the root of a determination-dependency tree. This syntax has been presented for instance in [Vergne 86] and developed in [Vergne 89];

• the sentence syntax, outside the NP, 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.

This "**NP stack syntax**" is used to:

• validate the sentence structure as a stack of NPs,

• confirm the function of words external to NPs,

• compute some tight dependencies external to NPs, such as verb-object (type -**2**-, see below).

Three relations nets

• determination dependencies

• co-ordinations

• references between an anaphoric and its referent.

Some relations of one net are computed from relations of another net: a dependency from a co-ordination (of NPs or PPs), a dependency from a reference (as in relative clauses), for instance.

Determination dependencies typology

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

-**2**- dependencies external to the NP, but *internal to a leaf* (see beelow) *pattern* (for instance, subject ← verb and verb ← object in a SVO sentence). They are computed at the leaf 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**- dependencies external to the NP, and *external to the leaf patterns*, as for instance, the relations "reigner" ← prepositional phrase (PP): PPs are recognized as a leaf pattern of the form: preposition-NP, which does not contain the reigner; the reigner is either a verbal "link", or a determined NP. These relations are computed by **valuation functions**. The computation then is **heuristic**.

These determination dependencies proceed from the tighter ones, inside the NP (-**1**-), to the looser ones, outside the NP, and outside the leaf patterns (-**3**-):

type	/ NP	/ leaf pattern	computation
-**1**-	internal	-	algorithmic
-**2**-	external	internal	algorithmic
-**3**-	external	external	heuristic

The NP stack syntax

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 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 its leaves progressively (it is a process from the outside of the sentence to its inside, and not left-to-right):

- leaves are replaced or removed in a precise order, until the heart is reached:
  - a replaced leaf may be: verb-adverb → verb, so negations, adverbs, auxiliaries are erased;
  - a removed leaf may be a PP, a subordinated clause, a co-ordinated NP, PP or clause;
- simultaneously, each time a leaf is replaced or removed, relations internal to this leaf are computed (dependencies of type -**2**-).

The final state of the pattern once plucked off must be a NP (the only possible heart), which may be: alone, or determined by another NP through a conjugated verb, or 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 (type -**2**) 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 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

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.

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

When to use valuation functions

- 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 (**dependencies** of type -**3**-).
- 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.
- Search for a **referent** which agrees with an anaphoric.

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 modelize 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. This hesitation between two candidates often meets human hesitation.

final output:

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 determiners of the two potential co-ordinated NPs: both *definite determiners*, both *indefinite determiners* or both *without determiner*, 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: Pascal
- machine: Macintosh
- source size: ≈ 16 000 lines, 860 Ko
- code size: ≈ 400 Ko
- research and development: ≈ 3 man-years

Parsing quality on a corpus of about 10 000 words

- category errors: ≈ 1% / words
- dependency errors: ≈ 3% / dependencies

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, with statistics upon categories,
  - relations of the text, classed by syntactic type,
  - problems met during the parsing,
  - statistics about text size, processing speed, tested patterns, categories, deduction modes, relations and patterns of the text and parsing quality.

Conclusions

The NP stack syntax 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 dependencies clearly gives the distinction between dependencies the computation of which can be algorithmic and dependencies 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, and with a quite low error rate; thus it confirms 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

●