topical paper with software demonstration

Syntax as clipping blocks: structures, algorithms and rules

Jacques VERGNE

Jacques.Vergne@univ-caen.fr LIUC Université de Caen

esplanade de la Paix F-14000 Caen **France**

Research topic

My research topic is into **syntax**, with, as a guideline, the very high formal redundancy of natural languages.

Formal redundancy implies safer transmission and memorization of information; redundancy is used in data processing, but is particularly present inside living organisms at every level: genotype transmission at cell level, for instance, and language at society level.

Formal redundancy of natural languages is the basement of NLP, but it is still not studied nor used very much: most NLP process natural languages as formal languages, which are **not** redundant.

Methodology

I consider **linguistics as a science**, a science of reality, like physics, geophysics, astrophysics, not as a science of imagination like mathematics, logics. I also consider linguistics as a science of life, like biology, entomology. Features of natural languages are not to be invented, but discovered by observation and experimentation.

I consider that the field of linguistics does not include thought, cognition, nor the way language is produced or understood; its object is not to separate right sentences from wrong sentences; its object is to study, observe, understand, explain attested facts; these facts are to be **external to the observer**, otherwise it becomes introspection (like working on artificial sentences), which is incompatible with a scientific approach.

The methodology comes from this vision of linguistics: corpus observation (written French), experiments upon this corpus, using automatic processes as experimental devices, as modelization tools; these experiments are observed with a statistic tool which allows to collect finely the syntactic forms and study the gap between the expected and the observed behavior of the model.

The corpus is made of two informative (scientific) texts in French language: the preface of a book about pattern recognition, and a paper in a review about marine biology; it contains 7000 words, 250 sentences.

Brought out structures

The sentence is observed as a multiple net made of objects which are connected in a more or less narrow way; this net may be projected on the written or spoken chain.

These objects are to be delimited, categorized, organized into a hierarchy, and connected.

These objects are of 4 types, each at a level of the hierarchy:

- sentences
- centers and blocks
- noun phrases (NPs) and words external to NPs
- words internal to NPs.

Sentences are made of one **center** (most often topical and nominal) to which small sequences of words and NPs are added; this addition is done by subordination, co-ordination or juxtaposition (each by inclusion or adjunction); let us call these sequences **blocks** and let us define them by instances and properties. A sentence has in average 7 blocks (6.9 and 7.1 in the two texts); the most frequent are prepositional phrases (PPs).

Blocks are for instance: PPs, subordinated clauses, co-ordinated NPs, co-ordinated PPs or co-ordinated clauses.

Centers and blocks are made of NPs (two maximum) and words external to NPs which have between them dependency relations that can be transposed in the whole sentence at the word level by algorithmic computation.

A block begins by a "clip": preposition, subordination or co-ordination conjunction, relative pronoun.

Blocks are marked by prosody; transcription of this property in written language: blocks borders often are marked by commas and brackets.

NPs are defined in a restricted way: a sequence of contiguous agreed words:

(partitive) (determiner) (adjective) noun (adjective)
adjectives and nouns may be co-ordinated
adjectives may be preceded by an adverb.

Words categories are in two disjoined sets:

- categories inside NPs: partitive, determiner, adjective, noun
- categories outside NPs: block clips, verbs, for instance.

About one word out of 2 is inside a NP: 48.5% in the preface, 52.6% in the paper. About one word out of 3 is a noun or an adjective (30.5%, 33.8%); about one word out of 5 is a block clip (22.0%, 21.6%), and about one word out of 10 is a verb (10.1%, 7.3%).

These objects have various internal cohesions, based on relations of three types: subordination, co-ordination and reference; these relations build three nets which have deduction relations between them, and are subject to formal and quantitative constraints.

The three cohesion levels are:

- between two different blocks, weak cohesion,
- inside a block, medium cohesion,
- inside a NP, high cohesion.

According to the objects linked by a relation, its computability is:

- **heuristic** computability (by valuation function) between two different blocks (about 24% of all dependencies): for instance the error rate in attaching PPs is about 10% (10.2% and 11.3%)
- **algorithmic** computability inside a block (about 44% of all dependencies): the error rate in dependencies inside blocks is about 2% (1.8% and 2.0%)
- **algorithmic** computability inside a NP (about 27% of all dependencies): the error rate in dependencies inside NPs is less than 1% (0.3% and 0.9%).

Thus, only one dependency out of 4 must be heuristically computed, all the others being algorithmically computed.

Algorithms

Validating the NP pattern of the sentence

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

Validating the pattern of the sentence then consists in progressively removing the block patterns from the sentence pattern (it is a process from the outside of the sentence structure to its inside, and not left-to-right).

A sentence is usually made of one center and contiguous blocks, but sometimes, there is a block inside another: for instance a PP inside a subordinated clause after its subject NP; this implies to recognize PPs **before** subordinated clauses: it is an instance of chronology constraints upon blocks recognition.

A question arises here: is there a unique order, or: are the blocks patterns inclusions always in the same way?

Answer upon the corpus:

by study of the pattern recognition precedence array (precedence = a pattern is recognized before another), on 40 000 possible precedences, 2 500 observed precedences (7%), 3 cases of reciprocal precedences are observed: a NP in brackets is inside a subordinated clause, and elsewhere a subordinated clause is inside a NP in brackets; otherwise, the array is completely empty under the diagonal: it means that **every pattern is tested only once, always in the same order**: the algorithm is not recursive, nor repetitive, but sequential.

Before removing blocks, these are cleaned off: non object preverbal pronouns, adverbs, auxiliaries and negations are erased. About one block or center out of 5 has to be cleaned (4.22 and 5.02 in the two texts).

This "unclothing" of the sentence pattern is done in the following order:

- cleaning blocks and centers:
  . erasing non object preverbal pronouns
  . erasing adverbs of attributes, verbs, blocks, NPs
  . erasing auxiliaries negations,
- removing PPs
- cleaning blocks and centers:
  . erasing auxiliaries
  . erasing verbs negations,
- removing all other blocks
- verifying the final state of the pattern once unclothed: it must be one of the possible centers.

Transposing relations internal to a block or a center by simulated reclothing

Each time a block is cleaned or removed, each time a center is recognized, dependency relations internal to this block or this center simultaneously are computed.

Relations internal to a block pattern are to be transposed into the entire sentence pattern. From the positions of objects in a block pattern, and from the position of the block pattern in the sentence pattern during the unclothing, we can compute the absolute positions of objects in the entire sentence pattern.

To find these absolute positions, we only have to simulate the reclothing, by using the historical account of the unclothing, just before removing the block. After simulated reclothing (by applying rules in the reverse order), we obtain the absolute positions in the entire sentence 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 dependencies internal to a block pattern are defined inside the block pattern, then transposed by simulated reclothing into the entire sentence NP level pattern, then finally transposed into the sentence word level pattern.

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

Rules

Here is now the inventory of the rules for cleaning and removing blocks, and for verifying centers in French language.

Underlined rules have been applied when parsing the corpus.

If a pattern includes another, it must be the first, otherwise it would not be recognized. If a rule pattern is found in the sentence pattern, the rule is applied.

Of course, a word may have more than one category.

cleaning blocks and centers

Dependency relations in cleaning blocks: the erased word depends on the remaining word.

erasing non object preverbal pronouns (code: m)

For instance: *lui, leur, y, en*

	transitive	intransitive
conjugated verb	mV→V	mU→U
infinitive	mI→I	mi→i
present participle	mR→R	mr→r

dependency: m-->V

It is important to distinguish object (M) from non object (m) preverbal pronouns to know if a verb can have an object or not; that is also why transitive and intransitive verbs are distinguished too.

erasing attribute adverbs (code: j)

For instance: *surtout, souvent, assez, extrêmement*

codes:

T	attribute (always with *être*)
z	transitive past participle with *être* (to be)
u	intransitive past participle with *être*
ù	past participle without *être*
F	epithet adjective, disjoined from its noun, thus outside NP

jj→i jT→T jz→z ju→u jù→ù jF→F
Tj→T zj→z uj→u ùj→ù Fj→F

dependency: j-->T

Anteposed adverbs are preferred: the anteposed adverbs rules are tested before the postposed adverbs rules.

erasing verbs adverbs (code: w)

For instance: *donc, aussi, maintenant, également*

ww→w
Vw→V Uw→U Iw→I iw→i Rw→R rw→r
wV→V wU→U wI→I wi→i wR→R wr→r

dependency: V<--w

Postposed adverbs are preferred: the rule Vw→V precedes the rule: wV→V .

erasing auxiliaries adverbs (code: w)

For instance: *ici, donc, évidemment, souvent*

codes:

W	conjugated *avoir* (have)
X	conjugated *être* (be)
Y	conjugated *pouvoir* (can)

Ww→W Xw→X Yw→Y

dependency: X<--w

No anteposed adverb for auxiliaries.

Anteposed adverb of infinitive is preferred to postposed adverb of auxiliary in:

YwI Y w-->I

erasing blocks adverbs (code: w)

For instance: *enfin, aussi, comme, uniquement*

This adverb marks the beginning of some blocks, before their "clip".

It may be preceded and followed by a comma.

codes:

‘	comma
p	preposition before a NP
q	preposition before an infinitive
s	*en* before a present participle
P	subordination conjunction

	‘w‘	w‘	w
p	‘w‘p→p	w‘p→p	wp→p
q	‘w‘q→q	w‘q→q	wq→q
s	‘w‘s→s	w‘s→s	ws→s
P	‘w‘P→P	w‘P→P	wP→P

dependency: w-->p

Commas and brackets are processed as words.

Verb adverb is preferred to block adverb: UwpG

U<--w pG

erasing NPs (code: G) *adverbs* (code: w)

For instance: *surtout, toujours, beaucoup, seulement*

‘w‘G→G w‘G→G wG→G Gw→G

dependency: w-->G

Anteposition is preferred.

Adjective adverb is not concerned because inside NP.

erasing auxiliaries negations (codes: k l)

For instance: *ne pas, ne plus, ne que, ne*

kWl→W kXl→X kYl→Y

dependencies: k-->W<--l

It is done after erasing adverbs: kXwl→kXl→X

removing (co-ordinated) PPs

codes: C is the block co-ordination conjunction before subordination clips (p q s P); { } are the codes for brackets.

As commas and brackets mark blocks borders, every block pattern is tested with brackets and commas:

{CpG}‘ {CpG} ‘CpG‘ CpG‘ ‘CpG CpG
{pG}‘ {pG} ‘pG‘ pG‘ ‘pG pG

dependencies: C−−>G p−−>G

It must be done before any recognition of subject or object, because PPs are sometimes between the subject and the verb or between the verb and the object: GpGV→GV VpGG→VG

cleaning blocks and centers

erasing auxiliaries

It consists in replacing both auxiliary and auxilied by the "unauxilied" form, for instance:

categories level: W z → V
words level: *a décidé* → *décide*
(but rules are applied at category level)

It must be done after erasing auxiliaries negations : kWlz→Wz→V

It must be done after removing PPs, because PPs are sometimes between the auxiliary and the auxilied: WpGz→Wz→V YpGI→YI→V

codes: M object preverbal pronoun

conjugated	*a*	W Y	*est*	X	*peut*	
infinitive	*avoir*	â	*être*	ê	*pouvoir*	î
present participle	*ayant*	ä	*étant*	ë	*pouvant*	ï
past participle			*été*	x	*pu*	y

YMI→MV
YI→V Yi→U Yî→Y Yâ→W Yê→X
Wz→V Wu→U Wy→Y Wx→X
Xx→X
Xz→V Xu→U
qîI→qI qâz→qI qêz→qI
qîi→qi qâu→qi qêu→qi
ïI→R äz→R ëz→R
ïi→r äu→r ëu→r

dependency: the auxiliary depends on its auxilied: W−−>z

erasing verbs negations

It must be done after removing PPs, because PPs are sometimes between the verb and *que* in *ne que*: kVpGl→kVl→V

It must be done after erasing auxiliaries, because YI or Xz may be between *ne* and *que* in *ne que*: kYIl→kVl→V kXzl→kVl→V

It is done after erasing adverbs: kVwl→kVl→V

kXTl→XT only for *ne que*
kMVl→MV kVl→V kUl→U klI→I
kMV→MV kV→V kU→U
ne alone is erased after *ne pas*

erasing auxiliaries negations (k alone)

kY→Y kX→X kW→W

removing all other blocks

removing co-ordinated past participles and attributes

code:
c co-ordination conjunction in clip position

cz cu cT

removing (co-ordinated) disjoined qualifications

cù cF ù F

It must be done before any recognition of subject or object, because ù and F are sometimes between the subject and the verb or between the verb and the object, after a PP: GpGùV→GV VpGùG→VG

removing (co-ordinated) infinitive PPs

CqIG CqMI Cqi CqêT CqêG
qIG qMI qi qêT qêG
cIG cMI ci cêT cêG

If we call Π_I (like infinitive predicate paradigm) the following paradigm:
IG MI i êT êG
we can summarize:
CqΠ_I qΠ_I cΠ_I

removing (co-ordinated) present participles

CsRG CsMR Csr CsëT CsëG
sRG sMR sr sëT sëG
cRG cMR cr cëT cëG
RG MR r ëT ëG

If we call Π_R the following paradigm:
RG MR r ëT ëG
we can summarize:
CsΠ_R sΠ_R cΠ_R Π_R

removing co-ordinated centers

cGVG cGMV cGU cGXT cGXG cGX cG

If we call Π_V the following paradigm:
VG MV U XT XG X
we can summarize:
cGΠ_V cG

removing (co-ordinated) relative clauses

codes of their clips:
Q subject relative pronoun: *qui*
O object relative pronoun: *que*
pg prepositional relative pronoun: *par lequel*

CQΠ_V COGV COVG CpgGΠ_V CpgVG CpgXTG
QΠ_V OGV OVG pgGΠ_V pgVG pgXTG

in: OVG pgVG pgXTG the predicate is anteposed

removing NPs in brackets

{G}

It is the only pattern tested twice because of some reciprocal block inclusions (see above).

removing (co-ordinated) subordinated clauses

Their clip is a subordination conjunction (P):

CPG$\prod_V$
PG$\prod_V$ PTXG PMVG

in: PTXG PMVG the predicate is anteposed

removing centers in brackets

{G$\prod_V$} {G}

removing centers preceded by a colon (code: {)

{G$\prod_V$ {G {

removing co-ordinated predicates

c$\prod_V$

General center structure

The final state of the pattern once unclothed must be one of the possible centers:

G$\prod_V$	$\prod_V$G	G	$\prod_V$
88%	0,5%	10%	2%

A **center** may be either a subject NP before its postposed (as usual) predicate: G$\prod_V$ (the most frequent) or an anteposed predicate before its subject NP: $\prod_V$G or a NP alone: G or a predicate alone: $\prod_V$ (imperatives).

General block structure

Blocks are made as centers in 2 ways:

- blocks made as centers clipped by subordination or co-ordination, made of 3 parts (94%):
  - the **block co-ordination conjunction** (facultative, only before subordination clips): C
  - the **clip** (empty clip for ù F R) that may be:
    . either a subordination clip: p q s P Q O g
    . or a co-ordination clip: c
  - the **body** is like a center, with other frequencies:

G$\prod_V$	$\prod_V$G	G	$\prod_V$
9%	0,3%	73%	18%

it may be either a subject NP with its predicate: G$\prod_V$ or $\prod_V$G or a NP alone: G (the most frequent) or a predicate alone: $\prod_V$ $\prod_I$ $\prod_R$ z u T ù F

- blocks made as centers (always a NP, with an eventual predicate) in brackets or preceded by a colon.

Dependencies inside blocks and centers

Dependencies inside blocks and centers are computed when removing blocks or verifying centers.

The block co-ordination conjunction and the clip depend on the body, on the subject if the body is made of a subject NP and a predicate.

Inside the body:

- if the body is made of a subject and a predicate, the predicate depends on the subject, in a qualified <-- qualifier dependency (see [Vergne 89] for discussion)
- inside predicates:
  any object (G M O) depends on the verb (V I R)
  any *être* (ê X) depends on the attribute (T G)

An example of unclothing a pattern

Here is a sentence to be parsed, and its NP pattern:

```
On  peut  définir  la reconnaissance  des formes
 G   Y    I      G          p G

comme l'ensemble     des techniques informatiques
 p    G       p      G

de  représentation  et  de  décision  permettant  aux

machines
p   G    C p G   r  p  G

de simuler       un comportement "sensible".
q  I         G
```

Here is now the evolution of the NP pattern during unclothing, and computing dependencies inside blocks and center:

```
123456789012345678901
GYIGpGpGpGpGCpGrpGqIG
GYIGpGpGpGpG_rpGqIG              by removing: CpG
   co-ordination relation in: CpG      13  C --> 15  G
   clip relation in: CpG               14  p --> 15  G

GYIG_pGpGpG  rpGqIG              by removing: pG
GYIG  _pGpG  rpGqIG              by removing: pG
GYIG   _pG  rpGqIG               by removing: pG
GYIG    _  rpGqIG                by removing: pG
GYIG       r_qIG                 by removing: pG

G_VG       r  qIG                by: YI→V
   auxiliary-->auxilied in: YI   2  Y --> 3  I
   between auxiliary and auxilied:  nothing =>authorized

G VG       r  qIG
G VG       r  _                  by removing: qIG
   verb<--objet in: qIG          20  I<-- 21  G
   between verb and object:  nothing =>authorized
   clip relation in: qIG               19  q --> 20  I

G VG       r
G VG       _                     by removing: r

   GVG is a possible center
   subject<--verb in: GVG              1  G <-- 3  I
   verb<--object in: GVG               3  I <-- 4  G
   between verb and object:  nothing =>authorized
```

The pattern is validated at its NP level.

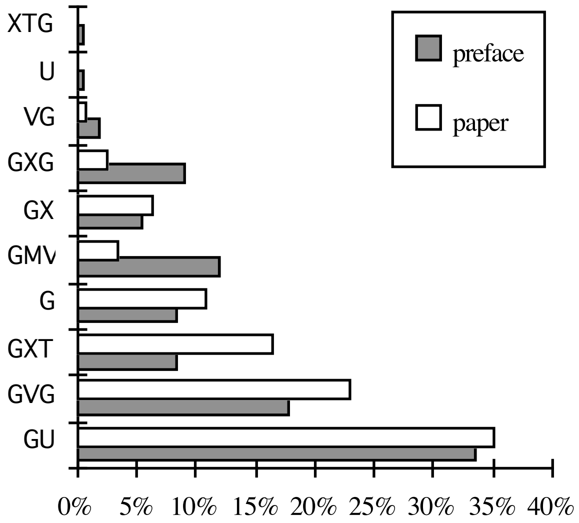
Statistics about patterns in the corpus

about corpus:

	preface	paper	total	maximum
words	3880	3102	6982	
sentences	139	108	247	
different centers	10	8	10	13
different blocks	94	65	112	708

16% (112/708) of the possible block patterns are observed.

statistics about centers:



A predicate is present in 90% of the centers, a NP is alone in 10%.

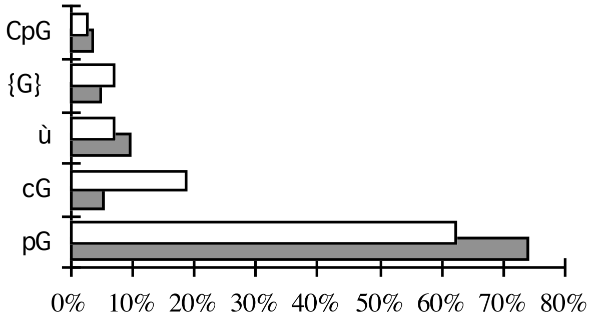
Predicates are nearly in three thirds:

- intransitives: 35%
- transitives: 30%
- attributes: 25%

statistics about removed blocks:

If we make five groups in the removed blocks patterns, and if we call each group by its most frequent pattern, we obtain:

CpG for co-ordinated blocks with subordination clip
{G} for centers in brackets or preceded by a colon
ù for blocks without any clip
cG for blocks with a co-ordination clip
pG for blocks with a subordination clip



We can notice that subordination is the most frequent way to clip blocks (about 70%), then come co-ordination and juxtaposition (about 10% each), and at last centers in brackets and co-ordinated subordinated blocks (about 5% each).

We notice also that very few blocks are very frequent: the 8 most frequent blocks represent 80% out of 112 block patterns.

More precisely, if we study each group, we observe that the one or two most frequent patterns in a group make the major part of the group (% of patterns in each group, by decreasing frequency):

in blocks with a subordination clip:
pG 81.9%*à certaines machines*
qIG 4.7% *de simuler un comportement*
qi PGXT PGU QVG PGVG are under 2%

in blocks with a co-ordination clip:
cG 77.7%*ou un algorithme rapide*
cGU 4.7% *et leurs propriétés changent*
cT cù cF cGVG cGXT cGMV cU cX are under 2%

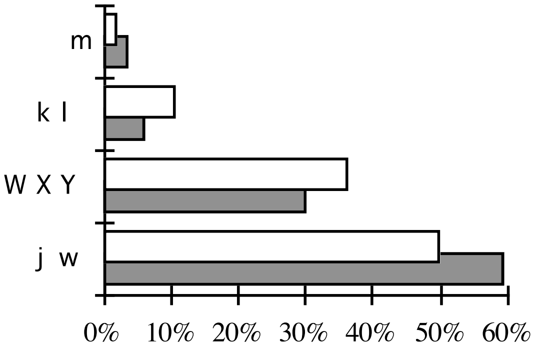
in blocks without any clip:
ù 68.2% *les sensations reçues*
RG 14.3%*propriétés facilitant la décision*
F 11.2% *le point de vue nutritionnel*
r 5.8% *techniques permettant de ...*
mr is under 2%

in centers in brackets or preceded by a colon:
{G} 74.0% *capacité d'écoute (audition)*
{G 9.8%*un paramètre : l'exactitude*
{GVG} 4.4%
{GVG 2.6%
{GU {GXG {GXT {GU} are under 2%

in co-ordinated blocks with subordination clip:
CpG 70.4% *de la taille et de la position*
CqIG 11.5% *et de prendre une décision*
Cqi6.9%*ou d'aider à l'apprentissage*
CqMI 2.8% *et de le programmer*
COGV CpgGVG CpgGXT CPGVG CPGX CPGXT are under 2%

statistics about cleaning blocks:

If we make four groups in the cleaning block patterns, according to the erased categories, we obtain:



We can notice that adverbs (about 55%) and auxiliaries (about 33%) are the most frequently erased categories, then come negations (8%), and non object preverbal pronouns (3%).

Enclosure: a constraint on dependencies

Enclosures are the main formal constraint upon the dependency net.

Every dependency:

- creates an **enclosure**, the borders of which are the two related objects
- defines the inside, the borders and the outside of the enclosure
- stops definitively any dependencies chain inside the enclosure.

Example:

PPs may be inserted between an auxiliary and its auxilied:

Une distance pourra, dans un tel espace, se calculer ...

```
                ------------------------------>
        G       Y  p    G   M   I
                <------------- -------------->
```

The PP between the auxiliary and its auxilied is enclosed inside the enclosure which is created by the dependency between them, and may depend only on these two borders: *pourra* or *calculer* .

Another example:

PPs (pG) may be inserted in four different places in a GVG center:

```
              <--------------- <---------------
 pG1   G   pG2   V   pG3    G  pG4
              <-------- ------><------- ------>
                                               <------
                               <----------------------
              <---------------------------------------
```

The anteposed PP (pG$_1$) disobeys the constraint (see below).

The PP between the subject and its verb (pG$_2$) is enclosed inside the enclosure which is created by the dependency between them, and may depend only on these two borders.

It is the same for the PP between the verb and its object (pG$_3$) .

The PP after the object (pG$_4$) is out of every already created enclosure, and may depend only on the borders of the two enclosures GV and VG, thus on the subject, its verb or its object.

Every dependency implies that any ulterior block (after its enclosure) cannot depend on something inside its enclosure.

This is the main formal constraint upon the dependency net, which drastically reduces the number of candidates when (heuristically) computing dependencies between blocks.

This property of the dependency net is always verified, except for anteposition:

- an anteposed block, before the subject of a clause, depends on the predicate
- in relative clauses, O and pg are anteposed object and PP, before the subject of the clause.

Anteposition is a way to stress a block by disobeying the constraint.

This constraint comes from a topological fact:

two enclosures (as any sequence) are either in a **inclusion** relation or in a **disjunction** relation (contiguous or not): so a new enclosure made by a new dependency may be only either inside or outside already existing enclosures.

Conclusions

Syntax considered as clipping blocks has some interests:

- it is an **explicative theory**, built upon corpus observation with a scientific methodology, and which matches with attested facts;

- when parsing, it is an **operative theory**, which allows to expect quite low error rates on heuristically computed relations, between blocks (one dependency out of 4), and error rates nearly equal to zero on algorithmically computed relations, inside a block and inside a NP;

- its **formalism** is adequate to express patterns, rules, syntactic facts and properties, and has an easy computability.

References

[**Combettes** 88] Bernard **Combettes** and Roberte **Tomassone**: *Le texte informatif, aspects linguistiques* De Boeck Université (Bruxelles) 1988

[**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

[**Vergne** 90] Jacques **Vergne**: *A parser without a dictionary as a tool for research into French syntax* CoLing 90 International Conference on Computational Linguistics pp. 70-72, Helsinki, August 1990