paper with demonstration

Syntax as clipping blocks: structures, algorithms and rules

Jacques VERGNE

Vergne@univ-caen.fr Laboratoire d'Informatique

Université de Caen F-14032 Caen cedex **France**

abstract

Research topic

My research topic is into **syntax of natural languages**, with, as a guideline, the very high formal redundancy of natural languages. This redundancy allows to make a morpho-syntactic parser working without an exhaustive dictionary.

Methodology

The methodology comes from the vision of linguistics **as a science**: corpus observation, experiments upon this corpus, using **a parser as an experimental device** to validate the theory.

Main observation: a natural language is not a formal language

Most NLP systems process natural languages as if they were formal languages. Because of the important differences between natural and formal languages, theories and tools for formal languages are not appropriate to natural languages.

Brought out segmentations

Between sentence and word levels, 3 other partition levels are observed: the partition of the sentence in one **center** and some **blocks**, the clip-subject-predicate tripartition, and the clip-noun-verb tripartition in **clips**, **nominal sequences and verbal sequences**. These segments are defined and explained.

Main algorithm

The validation of the sentence pattern is described: it is a sequential process **from the outside to the inside** of the sentence structure by removing the block patterns. Then it is explained how to transpose relations internal to a block by simulated reclothing of the sentence pattern.

Rules

The inventory of rules for simplifying verbal sequences, removing blocks, and for verifying centers is given for French language, in the order of the sequential algorithm. The general structures of centers and blocks are presented. An example on a sentence of the corpus is commented. Then some statistics about patterns in the corpus are set out.

Enclosure: a topological constraint on dependencies

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

Conclusions

Syntax as clipping blocks is an **explicative theory**, and an **operative theory**.

Its model applies this main principle for NLP: fidelity to structures of the processed object: **a natural language**.

Multilingual tests are under way on English, German, Polish, Basque, and Japanese with speakers of each language. First observations are that the main concepts seem to be transposable.

paper with demonstration

Syntax as clipping blocks: structures, algorithms and rules

Jacques VERGNE

Vergne@univ-caen.fr Laboratoire d'Informatique

Université de Caen F-14032 Caen cedex **France**

Research topic

My research topic is into **syntax of natural languages**, 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 should be the foundation of NLP.

This redundancy is high enough to make a morpho-syntactic parser working without an exhaustive dictionary (only with a lexicon of 400 grammatical words, and 300 ending rules), and which outputs the lexicon of the text.

Methodology

If we consider **linguistics as a science**, it is a science of reality, like physics, geophysics, astrophysics, not as a science of imagination like mathematics, logics; it is a science of life, like biology, entomology.

Features of natural languages are not to be invented, but discovered by observation and experimentation.

I think 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 for research into syntax comes from this vision of linguistics: corpus observation, experiments upon this corpus, using a morpho-syntactic **parser as an experimental device**, as a modeling tool to validate the theory; in this way, forms are processed apart from meaning; these experiments are automatically observed with a statistic tool which allows to collect finely the syntactic forms and measure 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 (7000 words, 250 sentences).

Main observation: a natural language is not a formal language

Most NLP systems process natural languages as if they were formal languages. But there are some important differences between natural and formal languages:

criteria	natural languages	formal languages
origin	societies and mothers	some persons
form/meaning mapping	many-to-many	one-to-one
explicitation	incomplete	total
lexicon	infinite, evolutive	finite, fixed
syntax	not well-known, evolutive	known, fixed
formal redundancy	very high, unknown	null or artificial, known
meta-language	a natural language	a natural language too

Because of these differences, theories and tools for formal languages are not appropriate to natural languages.

Brought out segmentations

Segmentation levels

The main object of research into syntax is to **discover segments** in the written or spoken chain (we must remember that the written chain is a representation of the spoken chain).

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

The hierarchy of these segmentations goes from words to entire texts (see [Lucas 92] about the paragraph structure).

The present paper is about the study of the sentence structure: 5 levels have been observed in the hierarchy of the sentence segmentations; we can describe the structure of each segment with the segments of the level below (the hierarchy is to be read bottom-up):

- the sentence level
- the partition of the sentence in one **center** and some **blocks**
- the clip-subject-predicate tripartition of the sentence, in 3 types of segments:
  clip, nominal sequence (subject or not actant), predicate (= a verbal sequence + eventual object NS only)
- the clip-noun-verb tripartition of the sentence, in 3 types of segments:
  clip, **Nominal Sequence** (NS = noun + its immediate satellites),
  Verbal Sequence (VS = verb + its immediate satellites)
- the word level

A **sentence** is made of one **center** to which **blocks** are added; this addition is done by subordination or co-ordination (usually by adjunction or sometimes by inclusion). A sentence has in average 7 blocks (6.9 and 7.1 in the two texts); the most frequent are prepositional phrases (PPs).

The **center** is the restricted main clause: only the subject nominal sequence (NS) and the conjugated predicate.

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

A **block** begins with a "**clip**": preposition, subordination or co-ordination conjunction, relative pronoun. Then comes the block body: a NS and/or a predicate (one of them or both). The center is actually a block without a clip.

In speech, block borders are marked by pauses and prosody, better than words and sentences, which are avatars of writing; transcription of this feature in written language: blocks borders are often marked by commas and brackets.

Agreement in gender, number and person between a subject NS and its predicate is a mark of block cohesion.

A **nominal sequence** is made of one **noun** and its immediate satellites which all depends on (determines) this noun: partitive, determiner, adjective; adjectives or nouns may be co-ordinated, adjectives may be preceded by an adverb.

Agreement in gender and number inside nominal sequences is a mark of their cohesion.

A **verbal sequence** is made of one **verb** and its immediate satellites which all depends on this verb: auxiliary, negation, verb adverb and preverbal pronouns (clitics).

Inside blocks, words categories are in 3 disjoined sets that make **the clip-noun-verb tripartition**:

- clips: preposition, subordination or co-ordination conjunction, relative pronoun; about one word out of 5 is a block clip (22% in the preface, 22% in the paper);
- categories inside nominal sequences: non clitic pronoun or noun and its immediate satellites: partitive, determiner, adjective, adjective adverb; about one word out of 2 is inside a NS (51%, 57%), and about one word out of 3 is a noun or an adjective (31%, 34%);
- categories inside verbal sequences: verb and its immediate satellites: auxiliary, negation, verb adverb, preverbal pronoun; about one word out of 6 is inside a verbal sequence (17%, 15%), and about one word out of 10 is a verb (10%, 7%).

Outside blocks, block adverbs and block separators (commas and brackets) make about 6%.

Here is an example of the segmentation of a sentence of the corpus (climbing bottom-up in the hierarchy):

On peut définir la reconnaissance des formes comme l'ensemble des techniques informatiques de représentation et de décision permettant aux machines de simuler un comportement "sensible".

clip-noun-verb tripartition: underlined clips, [nominal sequences in square brackets], **verbal sequences in bold**

[On] ***peut définir*** *[la reconnaissance] de [les formes] comme [l'ensemble]*
de [les techniques informatiques] de [représentation] et de [décision]
permettant *à [les machines] de* ***simuler*** *[un comportement "sensible"].*

clip-subject-predicate tripartition: predicate are double underlined

[On] ***peut définir*** *[la reconnaissance] de [les formes] comme [l'ensemble]*
de [les techniques informatiques] de [représentation] et de [décision]
permettant *à [les machines] de* ***simuler*** *[un comportement "sensible"] .*

center-blocks partition:

[On] ***peut définir*** *[la reconnaissance] de [les formes] comme*
[l'ensemble]
––––––––––––––––––––––––––––––––– ––––––––––––– ––––––––––––––––

de [les techniques informatiques] de [représentation] et de [décision]
–––––––––––––––––––––––––– ––––––––––––––– ––––––––––––––

permettant *à [les machines] de* ***simuler*** *[un comportement "sensible"] .*
–––––––––– ––––––––––––– ––––––––––––––––––––––––––––––––––

Dependencies inside and between segments

Dependencies have various tightnesses according to the segment where are the two linked elements:

- *very tight* dependencies inside nominal sequences (26% of all dependencies) and inside verbal sequences (10%)
- *tight* dependencies inside the predicate between a verbal sequence and its object nominal sequence (4%)
- *tight* dependencies inside the block between a clip and a block body (22%)
- *medium* dependencies inside the block between a subject nominal sequence and its predicate (6%)
- *weak* dependencies inside the sentence between 2 blocks (25%)

According to these tightnesses, a **block insertion** is impossible inside nominal or verbal sequences; it is possible and rare between a verbal sequence and its object nominal sequence, or between clip and block body; it is possible and frequent between a subject nominal sequence and its predicate; and it is easy and common between 2 blocks.

Only these weak dependencies between 2 blocks (one dependency out of 4) have to be **heuristically computed** (by valuation function): for instance the expected/observed behavior gap in attaching PPs is about 10% (10.2% and 11.3%); all the others are **algorithmically computed**: the gap in dependencies inside blocks is about 2% (1.8% and 2.0%), and the gap in dependencies inside nominal sequences is less than 1% (0.3% and 0.9%).

Main algorithm

A sentence is parsed progressively block by block to limit the combinatory of words possible categories in a segment as small as possible. That is why the global complexity of the parser is linear in time for 95% of sentences.

Validating the nominal sequence pattern of the sentence

At the beginning of this step of parsing, the sentence is represented by its NS pattern: it is a sequence of letters, in which each letter represents either a Nominal Sequence (N), or a clip, or a category in the verbal sequence, or a block adverb or a block separator.

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 to the inside** of the sentence structure, and not left-to-right).

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

A question arises here: is there a unique order in pattern recognition? Or: are the blocks patterns inclusions always in the same order?

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 nominal sequence in brackets is inside a subordinated clause, and elsewhere a subordinated clause is in brackets with a nominal sequence; otherwise, the array is completely empty under the diagonal: it means that **every pattern is tested only once, always in the same order**: the validation algorithm is not recursive, nor repetitive, but sequential, therefore deterministic.

Before removing blocks, verbal sequences are simplified: non object preverbal pronouns, adverbs, auxiliaries and negations are erased. About one verbal sequence out of 5 has to be simplified (4.22 and 5.02 in the two texts).

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

- simplifying verbal sequences (first stage):
  . erasing non object preverbal pronouns
  . erasing adverbs of attributes, verbs, blocks, nominal sequences (NSs)
  . erasing auxiliaries negations,
- removing PPs (before erasing auxiliaries, because PPs are sometimes between the auxiliary and the auxilied)
- simplifying verbal sequences (second stage):
  . 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 dependencies internal to a block or a center by simulated reclothing

Each time a verbal sequence is simplified, or a block is removed, each time a center is recognized, simultaneously, dependencies **internal** to this verbal sequence, this block or this center are **algorithmically** computed.

A dependency is an oriented link between 2 elements of a block pattern. Dependencies internal to a block pattern are to be transposed into the entire sentence pattern by computing the positions of these 2 elements in the sentence pattern.

From the relative position of a block pattern element in its block pattern (**this position marks its function in the block**), and from the position of the block pattern in the sentence pattern at this moment of the unclothing, we can compute the absolute position of this element in the entire sentence pattern.

To find this absolute position, we only have to simulate reclothing, by using the historical account of the unclothing, just before removing the block. After simulated reclothing (by applying rules in reverse order), we obtain the absolute position of a block pattern element in the entire sentence pattern.

In a later step of the parsing, after the internal analysis of nominal sequences (NSs), 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 NS level pattern, then finally transposed into the sentence word level pattern.

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

Rules

Here is now the inventory of rules for simplifying verbal sequences, removing blocks, and for verifying centers for French language, in the order of the sequential algorithm.

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. While a rule pattern is found in the sentence pattern, the rule is applied.

simplifying verbal sequences (first stage)

Dependency relations in simplifying verbal sequences: the erased word (a satellite) always depends on the remaining word (the verb).

erasing non object preverbal pronouns (code: m)

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

	transitive	intransitive
conjugated verb	$\underline{\text{mV→V}}$	$\underline{\text{mU→U}}$
infinitive	$\underline{\text{mI→I}}$	mi→i
present participle	mR→R	$\underline{\text{mr→r}}$

dependency: m-->V

It is important to distinguish object (M) from non object (m) preverbal pronouns to know if the object valency of a verb is saturated or not; verbs are set transitive by default, and are set intransitive if not any object is found.

erasing attribute adverbs (code: j for all predicate adverbs)

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 nominal sequence (NS)

$\underline{\text{jj→j}}$ $\underline{\text{jT→T}}$ $\underline{\text{jz→z}}$ $\underline{\text{ju→u}}$ $\underline{\text{jù→ù}}$ $\underline{\text{jF→F}}$
$\underline{\text{Tj→T}}$ $\underline{\text{zj→z}}$ $\underline{\text{uj→u}}$ $\underline{\text{ùj→ù}}$ Fj→F

dependency: j-->T

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

erasing verbs adverbs (code: j)

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

$\underline{\text{Vj→V}}$ $\underline{\text{Uj→U}}$ $\underline{\text{Ij→I}}$ $\underline{\text{ij→i}}$ Rj→R $\underline{\text{rj→r}}$
$\underline{\text{jV→V}}$ jU→U $\underline{\text{jI→I}}$ $\underline{\text{ji→i}}$ jR→R jr→r

dependency: V<--j

Postposed adverbs are preferred: the rule Vj→V precedes the rule: jV→V .

erasing auxiliaries adverbs (code: j)

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

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

Wj→W $\underline{\text{Xj→X}}$ $\underline{\text{Yj→Y}}$

dependency: X<--j

No anteposed adverb for auxiliaries.

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

YjI Y j-->I

erasing blocks adverbs (code: w)

For instance: *enfin, aussi, comme, uniquement, que* with *ne* in *ne que*

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 nominal sequence (NS)
	q	preposition before an infinitive
	s	*en* before a present participle
	P	subordination conjunction

	‘w‘	w‘	w
p	$\underline{\text{‘w‘p→p}}$	$\underline{\text{w‘p→p}}$	$\underline{\text{wp→p}}$
q	$\underline{\text{‘w‘q→q}}$	w‘q→q	$\underline{\text{wq→q}}$
s	$\underline{\text{‘w‘s→s}}$	w‘s→s	ws→s
P	‘w‘P→P	$\underline{\text{w‘P→P}}$	wP→P

dependency: w-->p

Commas and brackets are processed as words.

Verb adverb is preferred to block adverb: Uw pN U<--w pN

erasing adverbs (code: w) *of nominal sequences* (code: N)

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

$\underline{\text{‘w‘N→N}}$ $\underline{\text{w‘N→N}}$ $\underline{\text{wN→N}}$ $\underline{\text{Nw→N}}$

dependency: w-->N

Anteposition is preferred.

Adjective adverb is not concerned because inside nominal sequence (NS).

erasing auxiliaries negations (codes: k l)

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

$\underline{\text{kWl→W}}$ $\underline{\text{kXl→X}}$ $\underline{\text{kYl→Y}}$

dependencies: k-->W<--l

It is done after erasing adverbs: k Xj l→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, ‘ is the code for commas.

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

{CpN}‘ {CpN} ‘CpN‘ CpN‘ ‘CpN CpN
{pN}‘ {pN} ‘pN‘ pN‘ ‘pN pN

dependencies: C-->N p-->N

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: N pN V→NV V pN N→VN

It must be done before erasing auxiliaries, because PPs are sometimes between the auxiliary and the auxilied:
W pN z→Wz→V Y pN I→YI→V

simplifying verbal sequences (second stage)

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 : kWl z→Wz→V

codes: M object preverbal pronoun

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

YMI→MV
YI→V Yi→U Yî→Y Yâ→W Yê→X
Wz→V Wu→U Wy→Y Wà→W Wx→X
Xx→X
Xz→V Xu→U
îI→I âz→I êz→I
îi→i âu→i êu→i
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 is done after erasing adverbs: k Vj l→kVl→V

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:

N pN ù V→NV V pN ù N→VN

removing (co-ordinated) infinitive PPs

CqIN CqMI Cqi CqêT CqêN Cqê
qIN qMI qi qêT qêN qê
cIN cMI ci cêT cêN cê

If we call $\prod_I$ (like infinitive predicate paradigm) the following paradigm:

IN MI i êT êN ê = $\prod_I$

we can summarize:

Cq$\prod_I$ q$\prod_I$ c$\prod_I$

removing (co-ordinated) present participles

CsRN CsMR Csr CsëT CsëN Csë
sRN sMR sr sëT sëN së
cRN cMR cr cëT cëN cë
RN MR r ëT ëN ë

If we call $\prod_R$ (like present participle predicate paradigm) the following paradigm:

RN MR r ëT ëN ë = $\prod_R$

we can summarize:

Cs$\prod_R$ s$\prod_R$ c$\prod_R$ $\prod_R$

removing (co-ordinated) relative clauses

codes of their clips (they are clip and pronoun simultaneously):

Q	subject relative pronoun: *qui*
O	object relative pronoun: *que*
pn	prepositional relative pronoun: *par lequel*

$\mathrm{CQ}\prod_{\mathrm{V}}$ $\underline{\mathrm{CONV}}$ COVN $\underline{\mathrm{CpnN}\prod_{\mathrm{V}}}$ $\mathrm{Cpn}\prod_{\mathrm{V}}\mathrm{N}$
$\underline{\mathrm{Q}\prod_{\mathrm{V}}}$ $\underline{\mathrm{ONV}}$ $\underline{\mathrm{OVN}}$ $\underline{\mathrm{pnN}\prod_{\mathrm{V}}}$ $\underline{\mathrm{pn}\prod_{\mathrm{V}}\mathrm{N}}$

in: OVN $\mathrm{pn}\prod_{\mathrm{V}}\mathrm{N}$ the predicate is anteposed

It must be done before any subject recognition, because relative clauses may be between the subject and the verb:
$\mathrm{N}\ \mathrm{Q}\prod_{\mathrm{V}}\ \mathrm{V} \rightarrow \mathrm{NV}$

removing co-ordinated centers

$\underline{\mathrm{cNVN}}$ cNMV cNU $\underline{\mathrm{cNXT}}$ cNXN cNX $\underline{\mathrm{cN}}$

If we call $\prod_{\mathrm{V}}$ (like conjugated predicate paradigm) the following paradigm:
VN MV U XT XN X $= \prod_{\mathrm{V}}$
we can summarize: $\mathrm{cN}\prod_{\mathrm{V}}$ cN

removing nominal sequences (NSs) in brackets

$\{\mathrm{N}\}$

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):

$\underline{\mathrm{CPN}\prod_{\mathrm{V}}}$ $\mathrm{CP}\prod_{\mathrm{V}}\mathrm{N}$
$\underline{\mathrm{PN}\prod_{\mathrm{V}}}$ $\underline{\mathrm{P}\prod_{\mathrm{V}}\mathrm{N}}$

in: $\mathrm{P}\prod_{\mathrm{V}}\mathrm{N}$ the predicate is anteposed

removing centers in brackets

$\underline{\{\mathrm{N}\prod_{\mathrm{V}}\}}$ $\underline{\{\mathrm{N}\}}$

removing centers preceded by a colon (code: {)

$\underline{\{\mathrm{N}\prod_{\mathrm{V}}}$ $\underline{\{\mathrm{N}}$

removing co-ordinated conjugated predicates

$\underline{\mathrm{c}\prod_{\mathrm{V}}}$

verifying the final state of the pattern once unclothed

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

$\underline{\mathrm{N}\prod_{\mathrm{V}}}$ $\underline{\prod_{\mathrm{V}}\mathrm{N}}$ $\underline{\mathrm{N}}$ $\underline{\prod_{\mathrm{V}}}$ (imperatives)

General center and block structure

We can summarize the center and block structures with a two dimensions array:

- vertically, **center structures = block bodies**: *nominal sequence* (N), *predicate* (conjugated in centers, all forms in blocks), *subject nominal sequence* + *postposed* (as usual) *conjugated predicate*, or *anteposed conjugated predicate* + *subject nominal sequence*;
- horizontally, **clip types**: *subordination clip preceded by the block co-ordination conjunction* (C), *subordination clip* (p q s Q O n P), *co-ordination clip* (c), *ending clip* for ù (*-é*) R (*-ant*) or *empty clip* for F, *bracket clip* (centers in brackets):

centers		**blocks**					by clip type	
		co-ord. subord.	*subord. clip*	*co-ord. clip*	*ending clip*	*bracket clip*		
		4%	70%	12%	8%	6%	by body type	
10%	N	ℂ p N	p N	c N	/	{N}	72%	*nominal sequence*
2%	∏V	ℂ q ∏ I s ∏ R	q ∏ I s ∏ R	c ∏ V ∏ I ∏ R z u ù T F	∏R ù F	{∏V}	18%	*predicate*
88%	N ∏V	ℂ Q ∏ V O N V p n N ∏ V P N ∏ V	Q ∏ V O N V p n N ∏ V P N ∏ V	c N ∏V N ∏R	N ∏R	{N ∏V}	10%	*nominal sequence + predicate*
0,5%	∏V N	ℂ O V N p n ∏ V N P ∏ V N	O V N p n ∏ V N P ∏ V N	c ∏ V N	/	{∏V N}	0,3%	*predicate + nominal sequence*

Dependencies inside blocks and centers

Dependencies inside blocks and centers are **algorithmically** computed when simplifying verbal sequences, removing blocks or verifying centers (see above).

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

Inside the block body:

- if the block body is made of a subject and a predicate, the predicate depends on the subject (subject <–– predicate), in a qualified <–– qualifier or determined <–– determiner dependency, marked by agreement as every determination dependency;

this determination dependency is not to be confused with the actantial dependency (see [Tesnière 59] pages 102 and 144) which is usually seen in the reverse way (subject ––> verb <–– object) (see [Vergne 89] pages 8 to 17 for discussion)

- inside predicates:

any object (N M O) depends on the verb (V I R)

any *être* (X ê ë x) depends on the attribute (T N or a block attribute)

An example of unclothing a pattern

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

[On] ***peut définir*** *[la reconnaissance]* *de* *[les formes]* *comme* *[l'ensemble]*
N Y I N p N p N

de *[les techniques informatiques]* *de* *[représentation]* *et de* *[décision]*
p N p N C p N

permettant *à* *[les machines]* *de* ***simuler*** *[un comportement "sensible"]* .
r p N q I N

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

```
          111111111122
123456789012345678901 axis of the elements of the NS level pattern
NYINpNpNpNpNCpNrpNqIN
NYINpNpNpNpN__rpNqIN by removing the block: CpN
   co-ordination relation in: CpN      13 C --> 15 N
   clip relation in: CpN               14 p --> 15 N
NYIN_pNpNpN   rpNqIN    by removing the block: pN
NYIN  _pNpN   rpNqIN    by removing the block: pN
NYIN    _pN   rpNqIN    by removing the block: pN
NYIN      _   rpNqIN    by removing the block: pN
NYIN           r_qIN    by removing the block: pN
N_VN           r qIN    by erasing the auxiliary: YI→V
   auxiliary-->auxilied in: YI 2 Y --> 3 I
   between auxiliary and auxilied:  nothing =>authorized
N VN           r qIN
N VN           r __     by removing the block: qIN
   verb<--objet in: qIN        20 I<-- 21 N
   between verb and object:  nothing =>authorized
   clip relation in: qIN               19 q --> 20 I
N VN           r
N VN           _        by removing the block: r

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

The sentence pattern NYINpNpNpNpNCpNrpNqIN is validated at its NS level.

In a later step of the parsing, after the internal analysis of nominal sequences (NSs), the sentence will be validated at the word level and these relations will be transposed into the word level pattern:

```
         11 1111 11 112 2 222 22222
12345 678 901 2345 67 890 1 234 56789  axis of the elements of the word level pattern
NYIdS pdS pdS pdSE pS CpS r pdS qIdSE     (d=determiner, S=substantive, E=epithet)
```

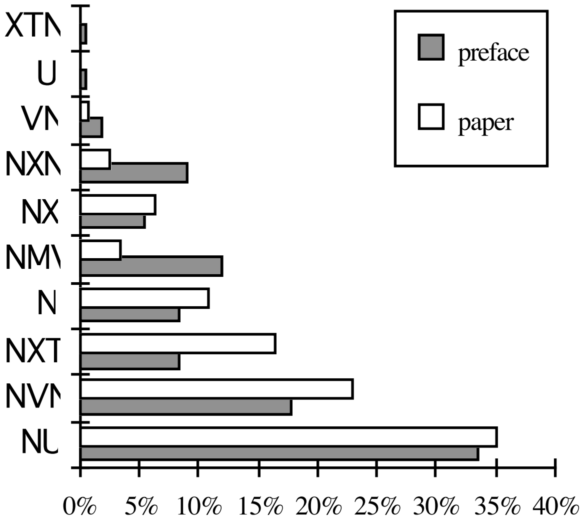
Statistics about patterns in the corpus

Globally, we can notice the low disparity between the two texts of the corpus: it is a clue of the stability of forms distribution.

statistics about corpus:

	preface	paper	total	possible	maximum
words	3880	3102	6982		
centers and blocks	1098	879	1977		
sentences	139	108	247		
words/center or block	3.53	3.54	3.53		
center and blocks/sentence	7.90	8.14	8.00		29
words/sentence	27.91	28.72	28.27		96
different centers	10	8	10	13	
different blocks	94	65	112	708	

statistics about centers:



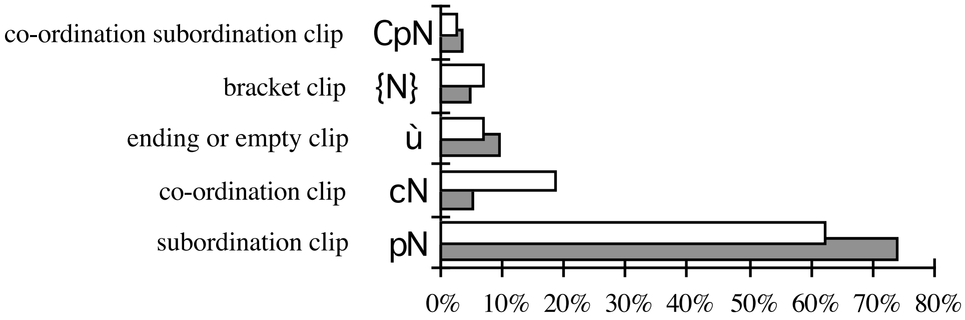
A predicate is present in 90% of centers, a nominal sequence (NS) is alone in 10%.

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, according to their clip type, and if we call each group by its most frequent pattern, we obtain:



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 block patterns are the most frequently observed: the 8 most frequent block patterns out of 708 possible block patterns (pN cN ù {N} qIN CpN qi RN) represent 80% of observed block patterns.

More precisely, if we study each group (according to their clip type), 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:

pN	81.9%	*à certaines machines*
qIN	4.7%	*de simuler un comportement*

qi PNXT PNU QVN PNVN are under 2%

in blocks with a co-ordination clip:

cN	77.7%	*ou un algorithme rapide*
cNU	4.7%	*et leurs propriétés changent*

cT cù cF cNVN cNXT cNMV cU cX are under 2%

in blocks without any clip:

ù	68.2%	*les sensations reçues*
RN	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:

{N}	74.0%	*capacité d'écoute (audition)*
{N	9.8%	*un paramètre : l'exactitude*
{NVN}	4.4%	*(on calcule l'intégrale)*
{NVN	2.6%	*:on attribue la classe*

{NU {NXN {NXT {NU} are under 2%

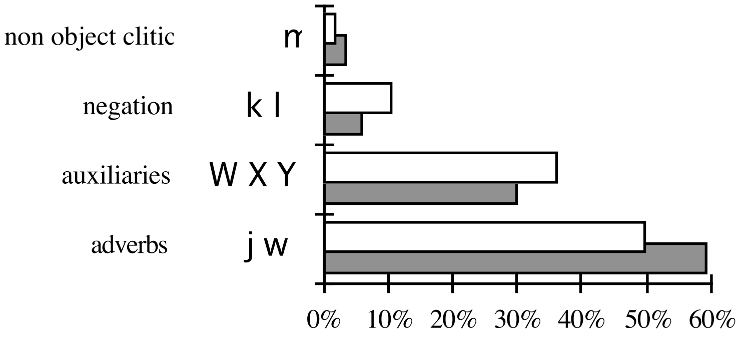
in co-ordinated blocks with subordination clip:

CpN	70.4%	*de la taille et de la position*
CqIN	11.5%	*et de prendre une décision*
Cqi	6.9%	*ou d'aider à l'apprentissage*
CqMI	2.8%	*et de le programmer*

CONV CpnNVN CpnNXT CPNVN CPNX CPNXT are under 2%

statistics about simplifying verbal sequences:

If we make four groups in verbal sequences patterns, according to the erased satellite 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 topological constraint on determination dependencies

Enclosures are the main formal constraint upon the dependency tree.

Every dependency:

- creates an **enclosure**, the borders of which are the two linked elements
- 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 ...

```
              ——————————————————————————————————>
     N         Y   p    N     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 one of these two borders: either *pourra* or *calculer* .

Another example:

PPs (pN) may be inserted in four different places in a NVN center:

```
           <————————————————  <——————————————————
pN1        N   pN2   V   pN3    N   pN4
           <————————  ———————><———————  ————————>
                                                 <——————
                              <—————————————————————————
           <————————————————————————————————————————————
```

The anteposed PP (pN1) disobeys the constraint (see below).

The PP between the subject and its verb (pN2) is enclosed inside the enclosure which is created by the dependency between them, and may depend only on one of these two borders.

It is the same for the PP between the verb and its object (pN3) .

The PP after the object (pN4) is out of every already created enclosure, and may depend only on one of the two borders of the two enclosures NV and VN, thus either 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 tree, which drastically reduces the number of candidates when (heuristically) computing dependencies between blocks.

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

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

By disobeying the constraint, anteposition is a kind of "macro-clip" that clips the sentence in the paragraph structure (see [Lucas 92]).

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 as clipping blocks has the following features:

- it is an **explicative theory**, built upon corpus observation with a scientific methodology, and which matches attested facts;
- when parsing, it is an **operative theory**, which allows to expect quite low expected/observed behavior gap (12.4%) on heuristically computed dependencies (one out of 4), between blocks, and gaps nearly equal to zero (1.2%) on algorithmically computed dependencies, inside nominal and verbal sequences and inside blocks;
- its **formalism** is adequate to express patterns, rules, syntactic facts and properties, and has an easy computability.

Syntax as clipping blocks is modeled with a parser which applies this main principle for NLP: fidelity to structures of the processed object: **a natural language**

- keep close to the partitions of the sentence: the clip-noun-verb tripartition, the clip-subject-predicate tripartition and the center-blocks partition;
- validate structures: • at one level of the partition hierarchy at once,
  • from periphery to center;
- leave use of tools of formal language processing: • word by word processing,
  • left → right processing in an unique pass,
  • categorize and link at once.

Multilingual tests are under way on English, German, Polish, Basque, and Japanese with speakers of each language.

First observations are that the main concepts seem to be transposable:

- the clip-noun-verb tripartition: clips, nominal sequences, verbal sequences; clips may be anteposed and/or postposed, separate or case endings: for instance, prepositions, conjunctions are anteposed and separate in French, English, German; particles are postposed and separate in Japanese; there are case endings in German, Polish, and Basque;
- the partition of the sentence in one center and some blocks; the center may be at the beginning (French, English, German, Polish, Basque) or at the end of the sentence (Japanese);
- the sequential order determined<––determiner may be different at different levels in the same language: for example in English, we have the order determiner––>determined inside nominal sequences (the opposite of French), and the order determined<––determiner between nominal and verbal sequences inside blocks, and between blocks (the same as in French).

References

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

[**Lucas** 92] Nadine **Lucas**: *Syntaxe du paragraphe dans les textes scientifiques en Japonais et en Français* Colloque international: Parcours linguistiques de discours spécialisés, Université Paris III, September 1992

[**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* (project note with demonstration) CoLing 90 International Conference on Computational Linguistics pp. 70-72, Helsinki, August 1990

●