Some syntactic properties of natural languages, applied to parsing of linear complexity in time

Jacques VERGNE
LAIAC
Université de Caen
F-14032 Caen cedex
France
Jacques.Vergne@univ-caen.fr

topical paper with demonstration

abstract

This paper describes some syntactic properties of natural languages, discovered by multilingual corpora observation (French, Spanish, English), by abandoning the analogy natural language - formal language.

These properties are applied to parsing, and they enable to obtain a parser of linear complexity in time.

Particular attention is given to the topology of the linear chain, seen as a one dimension space: segments hierarchy, topological relations of contiguity or inclusion between segments.

•

topic area

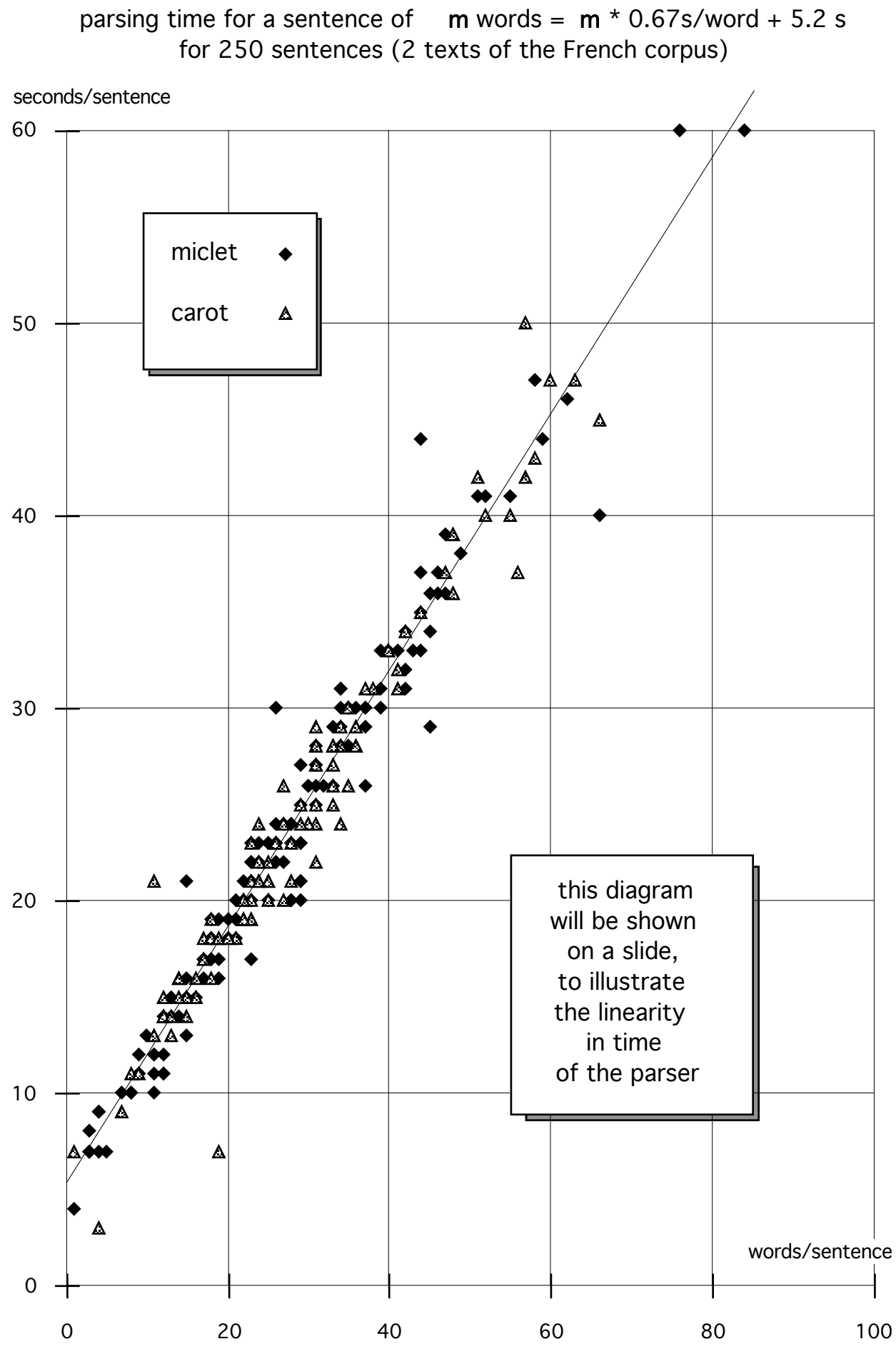
syntax and parsing

•

demonstration

computer platform required: Mac II, system 7, datashow

•


parsing time for a sentence of m words = m * 0.67s/word + 5.2 s
for 250 sentences (2 texts of the French corpus)
seconds/sentence
60
50
40
30
20
10
0
miclet
carot
this diagram
will be shown
on a slide,
to illustrate
the linearity
in time
of the parser
words/sentence
0
20
40
60
80
100

Some syntactic properties of natural languages, applied to parsing of linear complexity in time

Jacques VERGNE

Jacques.Vergne@univ-caen.fr LAIAC

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

Introduction: research domain

This work concerns the syntax of natural languages, with a dual aim:

- to discover the syntactic properties of natural languages,
- to base automatic parsing on these syntactic properties.

It explores the importance and redundancy of forms of natural languages.

The method is hypothetico-deductive: studied objects are corpora[1], with manual observations, computer aided observations (with statistics) and experiments, to find regularities and build a theory which models them.

The computer is an experimental device, a modelling tool of the theory, which lets us evaluate its validity.

In addition, the validity of linguistic concepts implies the efficiency of parsing algorithms.

The analogy **natural language - formal language** (see [Vergne 92] page 181), as a base for syntactic theories and automatic natural language processing systems, has led to important difficulties. That is why an important feature of this work is to attempt to base syntax and parsing of natural languages on features of natural languages observed in corpora, by abandoning this analogy.

This work is on French[2] in a precise way on both aspects, linguistic study and parsing; a precise linguistic study has also been done on Spanish[3] (see [López 93], chapter 4, pages 115 to 169) and English[4]; and studies of the transposability of concepts have been done on German, Polish, Basque, Japanese (with speakers of each natural language) and Latin. Otherwise proven, these concepts seem to be general properties of natural languages.

[1] Every example in this paper comes from a corpus.

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

[3] In collaboration with Eduardo López Gonzalo and Luis A. Hernández Gómez of the E.T.S.I. Telecomunicación, Universidad Politécnica Madrid, on a corpus (1700 words, 80 sentences) from an economics review (see [López 93], chapter 4, pages 115 to 169).

[4] The English corpus is made of a paper in the same review about marine biology (2200 words, 100 sentences).

I. Some syntactic properties of natural languages

I.1 A definition of the syntax of natural languages

Syntax of natural languages: to characterise it apart from the syntax of formal languages.

The study of syntax of natural languages involves:

- discovering segments of the written or spoken chain, and building a hierarchy of these segments (the "constituents"); particular attention is given to the topology of the linear chain, seen as a one dimension space: segments hierarchy, topological relations of contiguity or inclusion between segments
- formulating the problem: segmenting / linking, or: constituency / dependency
- discovering the characteristics and properties of linguistic relations between segments (dependency, co-ordination, anaphora), at different levels of the hierarchy.

This hierarchy of segments is to be considered in total generality from word to text, through the sentence and the paragraph (see [Lucas 92] on paragraph structure, and [Lucas 93] on book structure).

The present work concerns this hierarchy between word and sentence, and is to be placed in a wider research between word and text.

I.2 Two levels in the hierarchy of segments, observed between word and sentence

Properties of a segment are described in terms of lower level segments in the hierarchy:

- nominal sequences and verbal sequences are made up of contiguous words; sequences are in a relation of *contiguity*
- blocks comprise 1 to 3 contiguous sequences; blocks are in a relation of *contiguity or inclusion*.

Thus, a sentence can be seen at these two levels:

- as a chain of contiguous sequences
- as a chain of contiguous or included blocks

I.2.a Nominal sequences, verbal sequences and clips

The term "sequence" is chosen to emphasise contiguities around the noun or verb, to stress the nominal-verbal symmetry (on both linguistic and algorithmic aspects) and also to distinguish it from a phrase (complements of a sequence *are not a part* of this sequence):

a nominal sequence is made up of a noun and its *immediate* satellites: partitive, determiner, adjectives and adjective adverbs

a verbal sequence is made up of a verb (in all its forms: conjugated, infinitive, participle) and its *immediate* satellites: auxiliary, negation, clitics, verb adverb.

In both types of sequences, satellites depend on the central element: a noun or a verb.

At the sequence hierarchic level, a clip is also a segment, most often a single word: preposition, subordination or co-ordination conjunction, relative pronoun, punctuation (mainly comma, bracket and colon); but it may contain up to 4 elements: comma - co-ordination conjunction - block adverb - preposition:

example: , and also from earlier work

At the sequence hierarchic level, a **sentence** is a *tripartition* of 3 types of contiguous segments: clips, nominal sequences and verbal sequences: a sentence may be completely coloured in 3 colours according to segment type;

in the three corpora of scientific texts, the ratios of nominal sequences / verbal sequences are between 2 / 1 and 4 / 1; this ratio, peculiar to a type of corpus (rather than to a natural language), could help to characterise a corpus.

example of sentences segmented in sequences and clips:
(nominal sequences, **verbal sequences**, clips)

Many years ago such a project **might not have been undertaken** because it **was thought** that fish **emigrated** from their native stocks to perhaps a significant extent .

Dans les réseaux trophiques marins , de nombreuses molécules énergétiques **sont transférées** entre les différents niveaux d' organisation **structurant** les échanges .

Comprobaremos , a la hora de **hacer** la declaración , que **también han subido** las tarifas , y con efecto retroactivo .

I.2.b Blocks and clips

clip:

At the block hierarchic level, clips mark the beginnings of blocks; they segment the sentence in blocks, they "clip", attach every block into the sentence structure; they increase the ability to segment while listening or reading; they are few in number, and enable an unknown natural language to be easily segmented.

block structure:

A block is made of a clip (on) and a block body (adult cod): [on adult cod] [because it **was thought**]

the central block of a sentence (central from the structural point of view) is the only block without a clip:

[Introduction]
[such a project **might have been undertaken**]
[the International Council **collects** information]

The block body is the result of the combinatory of a nominal sequence and a "predicate": a "predicate" is a verbal sequence possibly with an object or attribute nominal sequence; therefore the 4 structures of a block body are:

(central block / other blocks)
nominal sequence (without a predicate): 10% / **70%**
predicate (without a subject sequence): 0.5% / 20%
subject nominal sequence + predicate: **90%** / 10%
predicate + subject nominal sequence: 0.5% /0.5%
(these statistics are given for the French corpus)

Clips typology, and their proportions:
sequences and blocks subordination clip 70%
[on adult cod] [because it **was thought**]
ending-clip (participles without an auxiliary) 10%
[**recovered**] [**excluding**]
sequences co-ordination clip [and **discussed**] 10%
bracket-clip [(Jones [, 1968])] 6%
blocks co-ordination clip: [and to **improve**] 4%

At the block hierarchic level, a **sentence** is made up of contiguous *or included* (about 1/5) blocks: some blocks include other blocks.

The central block is generally the first block, but it may be preceded by some anteposed blocks.

example of sentences segmented in contiguous blocks:
([blocks] are in square brackets)

[Many years ago] [such a project **might not have been undertaken**] [because it **was thought**] [that fish **emigrated**] [from their native stocks] [to perhaps a significant extent] .

[Dans les réseaux trophiques marins] [, de nombreuses molécules énergétiques **sont transférées**] [entre les différents niveaux] [d' organisation] [**structurant** les échanges] .

[**Comprobaremos**] [, a la hora] [de **hacer** la declaración] [, que **también han subido** las tarifas] [, y con efecto retroactivo] .

Blocks inclusion is a major syntactic phenomenon, and its precise study clarifies problems of segmentation.

definition of block inclusion and inclusion depth:

the included block (inclusion depth = 1) cuts the including block into 2 non-empty parts (inclusion depth = 0):

[including block: left part *inclusion depth 0*
[included block] *inclusion depth 1*
including block: right part] *inclusion depth 0*

[the International Council *inclusion depth 0*
[for the Exploration] *inclusion depth 1*
[of the Sea] *inclusion depth 1*
collects information] *inclusion depth 0*

This definition is only topological, and does not concern dependency: the *inclusion depth* of a block is not to be confused with its *depth in the dependency tree*.

inclusion points of a block in another block:

These inclusion points are few in number, because we observe that sequences are never cut.

We can deduce these 3 inclusion points from the block structure:

[clip [3] subject [1] **verb** [2] object]

[1] between subject nominal sequence and verb:
(the most frequent) 85%
[2] between verb and object nominal sequence: 12%
[3] between clip and block body: 3%

example of sentences segmented in contiguous blocks, with some blocks included at depth 1:
- blocks included between subject and verb:

[The exchange *inclusion depth 0*
[between the Faroe Islands] *inclusion depth 1*
[and Faroe Bank] *inclusion depth 1*
is perhaps intermediate] . *inclusion depth 0*
[La mesure *inclusion depth 0*
[de concentration] *inclusion depth 1*
[de chlorophylle] *inclusion depth 1*
est utilisée] *inclusion depth 0*
[pour **estimer** la biomasse phytoplanctonique].
inclusion depth 0

[Lo fundamental *inclusion depth 0*
[en momentos] *inclusion depth 1*
[de crisis] *inclusion depth 1*
es mantener tu estrategia] *inclusion depth 0*
[a largo plazo] . *inclusion depth 0*

- blocks included between verb and object:

[on **attribue** *inclusion depth 0*
[à le point inconnu] *inclusion depth 1*
la classe] *inclusion depth 0*
[de son plus proche voisin] *inclusion depth 0*

[Esto **ayudaría**] *inclusion depth 0*
[a **no subir** *inclusion depth 0*
[, e incluso a **reducir**] *inclusion depth 1*
, los impuestos] . *inclusion depth 0*

inclusion depths of a block in another block:

Depth 0 (non-included blocks) is the most frequent (76% to 82% in the three corpora); then comes the inclusion at depth 1, almost the only way to include a block (24% to 18%); depth 2 is very rare, almost only for prepositional blocks (<1%); depth 3 has never been observed; these statistics are very regular for different natural languages.

On the contrary, segment inclusion depths of programming languages (e.g. ALGOL, Pascal, etc.) have no limit: that is why recursiveness is needed to describe their syntax, and not needed to describe natural languages syntax.

included blocks typology:

The commonest included blocks are prepositional blocks (65%), then co-ordinated nominal sequences (16%), then past participles without an auxiliary (10%), then subordinated clauses: relative, then circumstantial clauses (for French).

We notice that the more a block is verbal, the harder it is to include it in another block.

example of a sentence segmented in blocks, with blocks included at depth 1 and 2:

[Les problèmes *inclusion depth 0*
[où le nombre *inclusion depth 1*
[de les mesures] *inclusion depth 2*
est important] *inclusion depth 1*
induisent un traitement] *inclusion depth 0*
[de ce type] . *inclusion depth 0*

[**Reducir** costes *inclusion depth 0*
[**concentrando** *inclusion depth 1*
[en una sola empresa] *inclusion depth 2*
los servicios] *inclusion depth 1*
[que **antes ofrecían** varias] *depth 1*
es la táctica] *inclusion depth 0*
[de Fernando Ocaña] *inclusion depth 0*
[, de Tapsa] . *inclusion depth 0*

hypothesis on understanding and inclusion depths:

When there is an inclusion, while listening or reading an included block, an effort of memory has to be done to remember the first (or left) part of the including block, while waiting for its second (or right) part, to link its two parts, and then to recognise the including block.

If there is an included block (at depth 2) within the included block (at depth 1), we must successively stack in memory the first (or left) part of the including block, and the first (or left) part of the included block at depth 1, then wait for the second (or right) part of the included block at depth 1, to link its two parts, and at last wait for the second (or right) part of the including block, to link its two parts.

It appears to involve a great effort of memory, which explains why no inclusion at depth 3 has been observed, and why the inclusion of a subordinated clause at depth 2 is very rare.

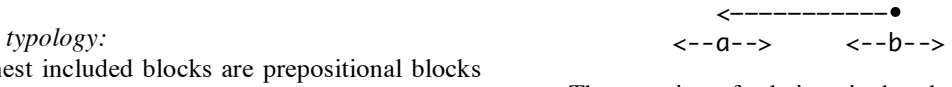
I.3 The problem: segmenting / linking, or constituency / dependency

I.3.a Definition

Every relation is based on a hierarchy of segmentation, at least a hierarchy of two levels:

a relation (<--•) links two segments a (<-a->) and b (<-b->), which are included in a same segment C (<== C ==>) of the level above;

every relation **between** a (<-a->) and b (<-b->) takes place **inside** C (<== C ==>):



The question of relations is then based on the answers to questions about segmentation.

I.3.b Application to sentence segments

While climbing the hierarchy of segments:
a relation **between** 2 words is **inside** 1 sequence
a relation **between** 2 sequences is **inside** 1 block
a relation **between** 2 blocks is **inside** 1 sentence

I.3.c Dependencies, and positions of segments in a segment of the next level up

between 2 words, **inside** 1 sequence:
dependencies on the central element of the sequence are directly function of word position in the sequence, of word categories and of gender-number agreements; it is a position - category - agreement redundancy:

the main •--> agent
•---------->

between 2 sequences, **inside** 1 block:
dependencies are also direct function of sequence positions in the block and sequence types (nominal or verbal); dependency between subject and verb is marked by person-gender-number agreement; again, this is a position - category - agreement redundancy:

[predation <--• **is considered**]
[the Council <--• **collects** <--• information]

in these two first cases, we see that agreement is both the mark of relation between 2 segments, and the mark of cohesion of the next segment above;

between 2 blocks, **inside** 1 sentence (about 1 on 4):
dependencies are not direct function of block positions, and are not marked by agreement (they must be heuristically computed; see [Vergne 90]):

[the Council **collects** information] [⌀m the gut contents] [⌀f many fishes] [im order] [to **make** estimates] [⌀f natural mortality]

I.4 Contiguity and dependency

I.4.a Determination dependency and actancial dependency

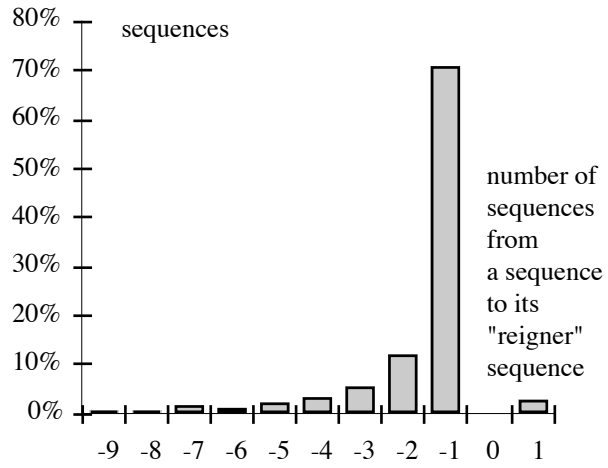
Tesnière places two types of dependency on the same level ([see Tesnière 59] pages 102 and 144):
the dependency of an "actant" on a verb, which we can call "actancial" dependency
the dependency of an adjective on a noun, which we can call determination dependency
Let us focus on determination dependencies.
In these determination dependencies: an adjective on a noun, a past participle on a nominal sequence, and a conjugated verb on its subject nominal sequence, agreement marks determination dependency (in French, German, Spanish, ...) (see [Vergne 89] pages 8 to 17):

the main •--> agent
[the number [⌀f tags] [**returned**] **is very low**]
<----------•
<----------•

I.4.b Determination dependency and contiguity at sequence level

If we observe *the same dependency type* (determination dependency) at *the same hierarchic level* (sequences), we have **homogeneous sight**:
then we observe (for French) that most sequences (70%) depend on the preceding sequence (its "reigner", Tesnière's "régissant").

In other words, the most frequent mark of determination dependency between 2 sequences is **contiguity**:



I.4.c A segment built on dependencies: the chain of contiguous dependent sequences

A sentence of a simplistic structure as "subject <-• verb <-• object" is made up of a single **chain of contiguous dependent sequences**:

[the Council **collects** information] [⌀m the gut
<------• <------• <----------•
contents] [⌀f many fishes]
<----------•

A new chain begins when a sequence does **not** depend on the preceding sequence.
The chain of contiguous dependent sequences is a new segment, and produces a new segmentation of the sentence:

[the Council [for the Exploration] [⌀f the Sea]
<----------• <------•
‖ **collects** information] [⌀m the gut contents]
<------• <------• <----------•

As regards the cut (‖) between 2 chains of contiguous dependent sequences, the more long range dependencies they are (counted in number of sequences), the more the comma is frequent at the cut in writing (checked), and the longer the pause in speech (see [López 93], chapter 4, pages 160 to 169).

II. Application to automatic parsing of linear complexity in time

II.1 A definition of automatic parsing

In the same way as with the study of the syntax (see I.1 above), automatic parsing involves:
- delimiting and identifying segments of the written or spoken chain, at different hierarchic levels: words, sequences, blocks and sentences
- linking segments: words inside sequences, sequences inside blocks, and blocks inside sentences

II.2 History: parsing with the compiling model

Bernard Vauquois (GETA), who contributed to define the programming language ALGOL-60 (1960), decided to base second generation machine translation systems on the analogy between compilers and MT systems (1961-1965).

As an automatic translation of a formal language into another formal language, compiling became the classic model of parsing a natural language: most often, it is a word-by-word process from left to right, in a single pass, segmenting and linking at the same time.

While doing this, the implicit or explicit hypothesis is made that natural languages are formal languages. This hypothesis was useful in its time for the first steps in the field, but we know now it is false; for example:

- form/meaning mapping is many-to-many in natural languages, and one-to-one in formal languages;
- redundancy of forms is very high in natural languages, and equal to zero in formal languages;
- segmentation for natural languages and for formal languages is different;
- segment inclusion depth is limited to 2 in natural languages, and unlimited in formal languages.

So, this hypothesis should now be abandoned (see introduction above, and [Vergne 92] page 181) by basing segmentation strategies on the linguistic properties of segments of natural languages.

II.3 Linguistic properties of segments => segmentation strategies

Segmentation strategies must be directly based on linguistic properties of observed segments: sequences and blocks.

II.3.a Defining word categories according to sequence tripartition

Word categories are based on sentence tripartition in: words in nominal sequences, words in verbal sequences, and words in clip segments: clip words, punctuation (processed as words) and block adverbs.

For example, an adverb may be either an adjective adverb in a nominal sequence, or a verb adverb in a verbal sequence, or a block adverb in a clip segment.

II.3.b Segment recognition while climbing the segment hierarchy

Segments are recognised while climbing the segment hierarchy, in three steps of linear complexity in time:

1) recognising words
2) recognising sequences, linking words in each sequence
3) recognising blocks, linking sequences in each block

It should be noticed that only sequence structures (in terms of word categories) and block structures (in terms of clip and sequences) are expected, but not *sentence structures* (in terms of blocks), which *are only observed*.

II.4 An algorithm for parsers of linear complexity in time

The problem of parsing a natural language is made combinatory by using formal language parsing tools; two features have made NL parsing combinatory:

- a word has more than one possible category,
- more than one rule is applicable at once, partly due to the strategy: segmenting and linking at the same time.

The first problem is solved because sequence recognition is a linear colouring of sequences; the second is solved by the parsing strategy to segment first (sequences, blocks), and link inside every segment afterwards.

That is why it is possible to build parsers which do not use any tree process and which are of linear complexity in time according to the number of words.

The text is processed sentence by sentence.

II.4.a Word recognition

1) dividing the sentence into words

amalgams are divided: *du* -> *de le* (2 words)
elided words are separated: *l'eau* -> *l' eau* (2 words)

2) building the possible categories list for every word without an exhaustive dictionary

The main lexical problem to be solved is neology and scientific and technical sub-languages.

According to neology type, we can make a typology of 3 types of words, with a specific solution for every type:

-a- grammatical words, where neology is almost nil, which belong to small and finite sets: clips, determiners, pronouns, quantifiers, adverbs not derived from adjectives;
solution: a little lexicon (400 forms in French), with codes for forms that are homographs of other categories

-b- verbs, where neology is very low, which belong to an almost finite set
solution: a lexicon of verb roots (about 7000 verbs in French ≈ 50 Kb) with codes for forms that are homographs of nouns or adjectives, and some ending rules to handle neology

-c- nouns, adjectives (and adverbs derived from adjectives), where neology is very rich, which belong to an almost infinite set
solution (for alphabetic languages): to extract ending rules from a base of forms (from BDLex); an ending rule enables possible categories, genders, numbers to be deduced from an ending (about 500 rules are enough)

at this stage, if a word does not match up with any ending rule, it is a noun or adjective with unknown gender and number.

3) shortening possible categories lists with a sieve

By automatic and systematic generation of block patterns at word level (including every case of block inclusion), from sequence and block patterns, we can get all authorised contiguities between 2 categories, and build a sieve to shorten possible categories lists of every word.

NB: as regards segmentation, this step is independent of the following steps, and could be replaced by any other morphological analysis; but the use of ending rules for nouns and adjectives shows that an exhaustive dictionary is not needed for morpho-syntactic parsing.

II.4.b Sequence recognition

Sequence recognition is a symmetrical process for nominal sequences and verbal sequences; the strategy is to make the sequence tripartition appear, by colouring the

words in 3 colours, spreading the colours inside sequences, and trying to obtain one colour for every word:

1) colouring words in 3 colours: nominal sequences, verbal sequences, and clips

-a- giving colours to every word, according to its possible categories

for every category, there is a corresponding place in the tripartition, and therefore a colour: context free rules are applied; a given word gets 1 to 3 colours: *le* may be either a determiner (nominal sequence colour), or a clitic (verbal sequence colour)

-b- giving colours to every word, according to its neighbours (categories or colours)

context sensitive rules are applied (about 50 in the French parser): for example, in French, we very often find 2 contiguous words which can either be determiner-noun or clitic-verb: *la ferme* (the farm, close it), *le bois* (the wood, drink it); if the previous word is a preposition, they are determiner-noun: *à la ferme* (at the farm), *dans le bois* (in the wood), and the colour is nominal; if the previous word is a subject pronoun, they are clitic-verb: *il la ferme* (he closes it), *je le bois* (I drink it), and the colour is verbal; these rules are found by corpus observation and/or parsing errors

-c- marking borders between colours

when the colours of two contiguous words are single and different, borders between colours (nominal and verbal sequences borders) are marked

-d- propagating colours from sequence borders

starting from left marked borders of sequences, colours are propagated from left to right for as long as possible, and also from right to left

-e- checking sequence closures

for every open sequence, we check that it is closed

-f- choosing the category of a word from its sequence type

for every word, the category is chosen from its possible categories list, according to its sequence type

2) studying every sequence: checking sequence structure and observing internal agreements

Nominal and verbal sequences structures are checked, and agreements observed and computed:

in nominal sequences, gender-number agreement between noun, determiner and adjectives

in verbal sequences with conjugated forms of "*être*", number agreement between "*être*" and attribute.

Inside sequences, dependencies on the central element (noun or verb) are computed.

3) if unexpected structure, finding another segmentation of the sequence

The most frequent case is where two contiguous sequences of the same type have been coloured as a single sequence: the structure is then not recognised, and the two sequences are separated into two parts, and their structures checked again.

Another case occurs with locutions made of a verb and a noun with no determiner: *tenir compte*, *faire partie*; these locutions have to be split into a verbal sequence and a nominal sequence: **tenir** compte **faire** partie

4) updating the lexicon of the text

At the end of this step, sequences are recognised, words tagged and lemmatised, and the lexicon of the text updated.

Usually, sentences are completely and correctly coloured, therefore with a right word tagging.

II.4.c Block recognition

1) compressing sequences into a single code

Words of a nominal sequence are coded: `N`

Verbal sequences are coded by type:

conjugated: `V` infinitive: `I`

present participle: `R` past participle: `ù`

2) cutting this pattern into blocks or block-parts

-a- locating beginnings of blocks

as blocks begin with clips, every clip marks the beginning of a block: `[p` `[P` `[,` `[c`

(`p` = preposition `P` = subordination conjunction `,` = comma `c` = co-ordination conjunction)

-b- closing complete blocks according to block patterns

blocks which do not include another block can be recognised at this stage, according to longest possible block patterns, because they are not cut:

`[p -> [pN]` `[P -> [PNV]`

relations in these blocks are computed at this time

at the end of this stage, the sentence is represented as a chain of blocks or block-parts:

`[PNV]` `[N]` `[pN]` `[V]` `[pN]` `[N]` `[pN]`

3) putting block-parts together

Blocks which include other blocks are cut in 2 to 4 parts (see inclusion points above), and these parts have to be put together:

-a- computing how every block-part can be continued

for a given block-part, and from block patterns, we compute how it can be continued:

`N:` `Ø` `V` `VN` means that a nominal sequence (`N`) can occur alone (`Ø`) or be continued with a conjugated verbal sequence alone (`V`), or with a conjugated verbal sequence with its object (or attribute) nominal sequence (`VN`)

`P:` `NV` `NVN` means that a subordination conjunction (`P`) cannot occur alone (no `Ø`) but can be continued with subject and verb (`NV`), or subject, verb and object (`NVN`)

`pN:` `Ø` means that a prepositional (`p` = preposition) block can only occur alone (`Ø`)

-b- putting block-parts together

the chain of block-parts is processed from left to right:

every block-part which can be continued is stacked;

for every block-part which can be the right continuation of a stacked left block-part (the stacked block-parts are tested backwards: the most recent first, owing to block inclusion), constraints on what can separate two block-parts are checked: subject-verb, verb-object, clip-sequence;

if the two block-parts can be put together (the right part is a possible continuation of the left part, and constraints are satisfied), they are linked, the left part is unstacked, and the depth of blocks included between the 2 parts is incremented of one level; if the two linked parts are subject and verb, number-person (and gender if verb "*être*") agreements between subject and verb are not used as constraints but merely observed: this is possible because agreements are redundant with positions and patterns.

Conclusion

This work attempts to renew the syntax of natural languages by abandoning the analogy **natural language - formal language**, and by basing linguistic concepts on observations of actual multilingual corpora.

With the computer, linguistic bases are modelled and experimented, and their validity evaluated.

In addition, the validity of linguistic concepts implies the efficiency of parsing algorithms: "man triumphs over nature only by obeying it" (Francis Bacon, 1620).

References

[**López** 93] Eduardo **López Gonzalo**: *Estudio de técnicas de procedado lingüistico y acústico para sistemas de conversión texto-voz en español basados en concatenación de unidades* tesis doctoral, E.T.S.I. de Telecomunicación, Universidad Politécnica de Madrid, julio de 1993

[**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, septembre 1992

[**Lucas** 93] Nadine **Lucas**, **Nishina** Kikuko, **Akiba** Tomoyoshi, K.G. **Suresh**: *Discourse analysis of scientific textbooks in Japanese: a tool for producing automatic summaries* Departement of Computer Science, Tokyo Institute of Technology, March 1993

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

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

[**Vergne** 90] Jacques **Vergne**: *A parser without a dictionary as a tool for research into French syntax* communication-demonstration at CoLing 90 International Conference on Computational Linguistics vol. 1 pp. 70-72, Helsinki, Finland, August 1990

[**Vergne** 92] Jacques **Vergne**: *Syntax as clipping blocks: structures, algorithms and rules* communication and demonstration at SEPLN 92 congress (Sociedad Española para el Procesamiento del Lenguaje Natural), pp. 179-197 and 467, Granada, Spain, September 1992

●

Demonstration

computer platform required:

Mac II, system 7, datashow

maquette à envoyer:
pas de trace: fichier = écran

pas de stat

pas de vrac
pas d'arrêt en fin de phrase
données compilées seulement
(pas de prep)
compléter verbes 3ième groupe
montrer résultats d'abord
algo géné, pas de détails

mode d'emploi en anglais
le plus simple possible