Between dependency tree and linear order, two transforming processes

Jacques VERGNE
GREYC - CNRS UPRESA 6072 - Université de Caen
F-14032 Caen cedex France
Jacques.Vergne@info.unicaen.fr

Abstract

In this paper, we present together on one hand two structural forms a sentence can be represented in: its linear order and its dependency tree, and on the other hand two processes which enable a human being to transform one structure into the other. We hypothesise that this simultaneous presentation of structures and processes allows a better theorization of the whole matter.

Introduction

In our research, we propose to stop trying to exhaustively enumerate syntactic structures, and we rather try to make parsing processes explicit, as contextual deductions in tagging. And structures will be results of such parsing processes.

In section 1, we present both structures, linear order and dependency tree and both processes which allow a human being to transform one structure into the other. We present in section 2 the transformation process of the dependency tree into a linear sentence, that is an aspect of the production process. In section 3, we present the transformation process of a linear sentence into a dependency tree, that is an aspect of the reception process. In both processes, memory constraints have a central position.

1. Two structures and two processes

1.1 Two structures

We first present the elements of structures, then the structure definitions using these elements.

1.1.1 Elements of structures

We have to define (1) a type of sentence segment, and (2) a type of relation between these segments.

• The sentence segment is not the word, which is conventional and unstable throughout different natural languages, but a group of words: the Non Recursive Phrase (NRP in this paper), phrase without its phrase-complements, "core phrase", or "chunk" (see Abney 1996). This segment is stable throughout different natural languages, and is approximately equivalent to the accentual group in oral language.

So we have a three level hierarchy of segments: words, NRPs, sentences. In this hierarchy, a segment of a level is made of segments of the lower level: a whole has a different type from the type of its parts; on the contrary, a recursive phrase is made of words and recursive phrases.

Words in an NRP, and NRPs in a sentence have very different behaviours: words of an NRP make a very constrained aggregate around a noun or a verb, but NRPs in a sentence are under more relaxed constraints.

To illustrate this, in his well-known verses from "Le Bourgeois Gentilhomme", Molière swaps NRPs in the sentence, but not words in NRPs:

[Belle marquise] , *[vos beaux yeux]*
[me font mourir] *[d'amour]* . [1]
[Vos beaux yeux] , *[belle marquise]* ,
[d'amour] *[me font mourir]* .

• The relation linking two NRPs is the dependency that we call "memorial dependency", defined and set by a reception process in two steps, by the mean of a memory. For instance, the relation between a nominal subject NRP and a verbal NRP is computed in two distinct and successive steps:

- step 1: a nominal NRP (heard or read) is memorized as an eventual subject, expecting an eventual verbal NRP,
- step 2: when a verbal NRP occurs (heard or read), it is linked to the expecting nominal NRP, which then does not expect a verbal NRP anymore, and therefore is "forgotten" from the memory of subjects expecting a verb.

This "memorial dependency" is defined with more details below in 3.1.

1.1.2 Definitions of structures

From the elements defined above, we consider that a sentence may have two structural forms: its linear order and its dependency tree.

In its linear order, a sentence is made of linked segments: depending NRPs, placed sequentially on a one dimensional axis (time in oral form, a

line in written form). A metrics is defined in the linear order: the unit is one NRP. The linear order is oriented in the direction of the chronology of production - reception processes.
In the dependency tree, nodes are segments, that is NRPs, and branches are relations between NRPs, that is memorial dependencies (also see figure 2 below). Let us notice that, in our work, constituency and dependency are neither declared equivalent, nor opposed, but used together at two different levels: constituency for words inside an NRP, and dependency for NRPs inside a sentence.

Lucien Tesnière presents two similar concepts in (Tesnière 59), page 18, § 8:
8.- Nous appellerons ordre linéaire celui d'après lequel les mots viennent se ranger sur la chaîne parlée. L'ordre linéaire est, comme la chaîne parlée, à une dimension. [2]
and page 16, § 1:
1.- L'ordre structural des mots est celui selon lequel s'établissent les connexions. [3]
Let us notice that, for Tesnière, the segment is not the NRP, but the word, and the relation is not the memorial dependency, but the "*connexion*" (see discussion in 3.1).

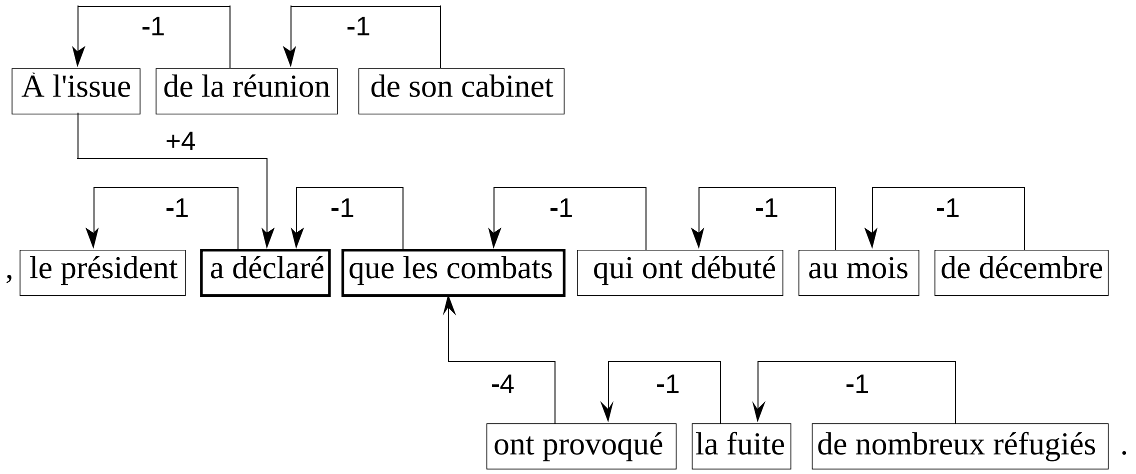

Figure 1: A linearized dependency tree : linked NRPs, in the linear order [4]

Figure 1 presents a sentence in the form of its linearized dependency tree, that is in the linear order, segmented in NRPs linked by memorial dependency relations. These relations have an algebraic length, computed in a number of NRPs, oriented in the direction of the chronology of writing - reading processes.

1.2 Two processes

1.2.1 Definition of these processes

These two processes enable a human being to transform one structure into the other:

<u>emitted</u> dependency tree	→	producing, writing a sentence generating a sentence linearizing the tree	→	<u>linear sentence</u>
Tesnière: structural order	→	***speaking** a natural language*	→	*linear order*
<u>linear sentence</u>	→	receiving, reading the sentence parsing the sentence rebuilding the tree	→	**<u>received</u> dependency tree**
Tesnière: linear order	→	***understanding** a natural language*	→	*structural order*

During these two successive processes, the dependency tree is thus the object, the information to be transmitted between two persons who communicate. This dependency tree is temporally linearized (and coded, compacted) in the linear sentence.
Lucien Tesnière presents an analogous concept in (Tesnière 59), page 19, § 4:
[...] nous pouvons dire que [...] ***parler*** *une langue, c'est en transformer l'ordre structural en ordre linéaire, et inversement que* ***comprendre*** *une langue, c'est en transformer l'ordre linéaire en ordre structural.* [5]

1.2.2 Execution of these processes

Obviously, a human being can execute these processes, but it is possible to simulate them on a computer, with two possible aims: either an operatory aim to generate or parse sentences inside a larger software (machine translation, information retrieval, grammar checking), or a research aim into natural language syntax. In the second case (which is also ours, see below in 3.2), a computer becomes a tool to observe, experiment, model, and confront concepts to corpora.

1.3 Processes and structures:

1.3.1 Constraints on processes and structures

Processes and structures are shaped by many constraints:
- the constraint of geometrical properties of structures, mainly because the linear sentence has one dimension;
- the constraint that the information contained in the tree is not lost during the linearization: this information is differently coded in the linear sentence;
- the constraint of the chronology of processes and deductions;
- the constraint of the minimization of the memory effort, which implies the minimization of distances between nodes in the linear order.

1.3.2 Theorization of processes and structures

It is hard to build a static theory of both structures, without these processes, as the attempts of linguists and NLP searchers show it.
Linguists mainly tried to model trees and linear sentences in a static way. Chomsky's generation is only the deduction process in the hypothetico-deductive method, (Mel'cuk 88) tells page 129 his difficulty to define the "*syntactic dependency*". And Tesnière's "*connexion*" is defined as a perception process (see below in 3.1) but the concept of dependency tree (the "*stemma*") remains static, and the connection between linear order and structural order, presented as two processes, is not developed anymore, but only sketched as a projected picture ("*image projetée*"), only in the following paragraph of (Tesnière 59), page 20, § 10:
[...] syntaxiquement, la vraie phrase, c'est la phrase ***structurale*** *dont la phrase linéaire n'est que l'image projetée tant bien que mal, et avec tous les inconvénients d'aplatissement que comporte cette projection sur la chaîne parlée.* [6]
In classical automatic syntactic parsing, a combinatory process tests all possible choices until matching explicitly expected structures.
We propose here to build together a theory of both structures and both processes, an explicit theory of processes, without making syntactic structures explicit.

2. The process: dependency tree → linear order

We shall now study how, during production, the linear order is produced from the dependency tree, what the linear order regent - dependants is ("regent" is used for Tesnière's term "*régissant*"), what the linear order of dependants is, and at last what the criteria of the linearization computation are.

2.1 Linear order = linearized tree, while minimizing distances between nodes

Producing a sentence is aligning linked segments on the syntagmatic axis (a one dimensional space), that is enumerating nodes of the dependency tree in a certain order. Theoretical computer science proposes clear and adequate concepts about trees and possible orders to enumerate the nodes of a tree: the process of searching a tree while picking up nodes. Figure 2 reminds us the elements of a tree, and the definition of nodes and branches of a dependency tree:

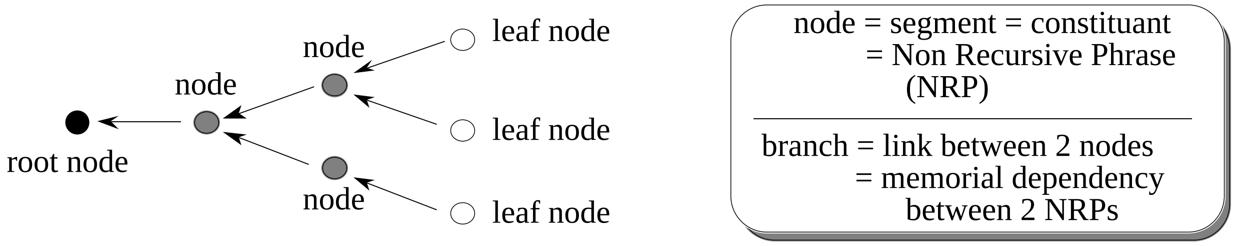

Figure 2: Nodes and branches of a dependency tree

Searching a tree consists in passing through every node while following a given path, starting at the root, and coming back to the root; "picking up" nodes is taking every node only once during the tree search, that is exactly enumerating nodes in a certain order.

> We can consider that linearizing a tree consists in searching it while picking up its nodes.

We can observe that most often this linearization is the one that places linked nodes as close as possible (contiguous if possible) in the linear

order, and we bring the hypothesis that it is to **minimize** memory efforts in production and reception.
Between width-first search and depth-first search, the latter is the one that minimizes distances between nodes: going from the root to the leaves, and from the leaves back to the root, while following every branch. For a two branches node, a first branch is covered in its whole, before the second branch is covered in its whole too.

In such a depth-first search, one passes three times through a regent node which has two depending branches (see figure 3):
going forward: root → regent node → leaf of depending branch 1,
going backward: leaf of depending branch 1 → regent node,
going forward: regent node → leaf of depending branch 2,
going backward: leaf of depending branch 2 → regent node → root.

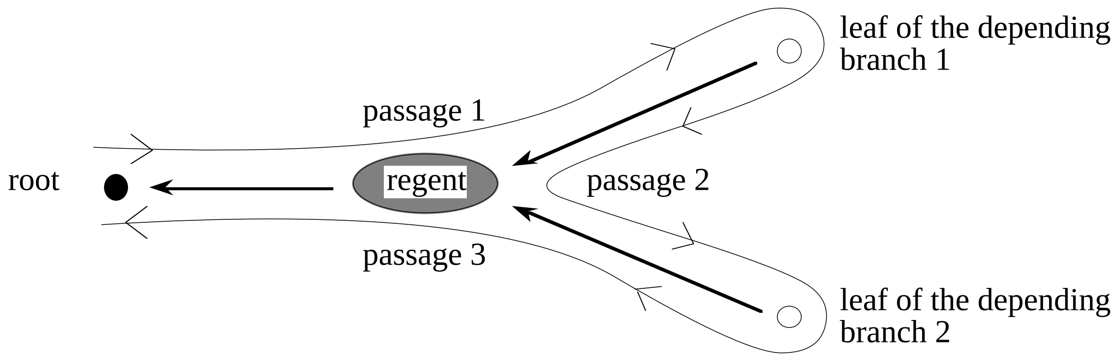

Figure 3: The depth-first search of a dependency tree

The picking up mode of the regent node specifies at which passage it is picked up:
- at passage 1, the regent is picked up **before** its dependants (**prefixed** picking up)
- at passage 2, the regent is picked up **between** its dependants (**infixed** picking up)
- at passage 3, the regent is picked up **after** its dependants (**postfixed** picking up).
While doing a depth-first search, every node has its proper picking up mode. In French, the usual picking up mode is the prefixed one: the linear order regent - dependants.
These concepts allow us to categorize different possible linear orders between a regent and its dependants (without the projection concept).

2.2 Linear order regent node - depending nodes

When a regent node has a single depending node (the most frequent case), the latter follows it in contiguity, but when a regent node has two depending nodes, they cannot follow it both in contiguity. It is a constraint of a one dimensional space, or one of these "flattening disadvantages" ("*inconvénients d'aplatissement*") as Tesnière says. Then three ways of picking up a regent node are possible, according to the three picking up modes of a node (see examples of figures 4, 5 and 6):

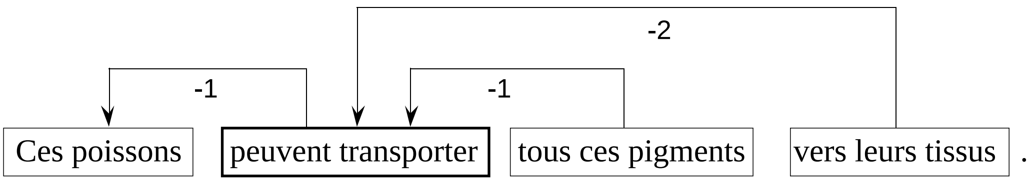

Figure 4: **Prefixed** regent node **before** its dependants [7]

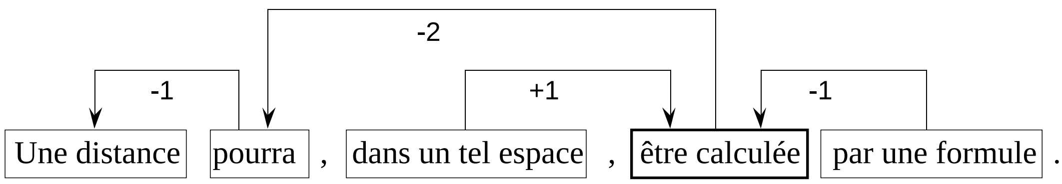

Figure 5: **Infixed** regent node **between** its dependants [8]

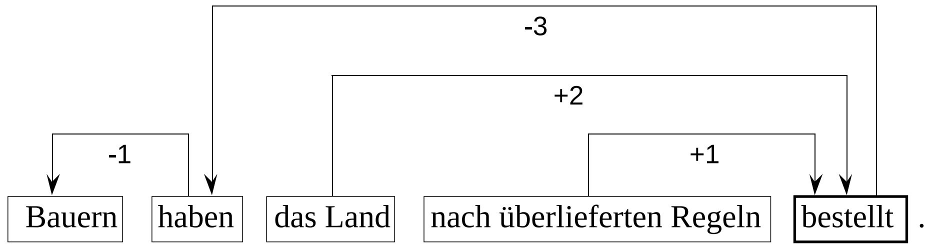

Figure 6: **Postfixed** regent node **after** its dependants [9]

2.3 Linear order of depending nodes

Now let us study what is the linear order of depending nodes after a prefixed regent node. For instance, let us take the case where a verb (prefixed regent node) has a direct object and an indirect complement (two depending nodes). The two possible orders are either verb - object branch - complement branch, or verb - complement branch - object branch, because every branch is covered in its whole (figures 7 and 8):

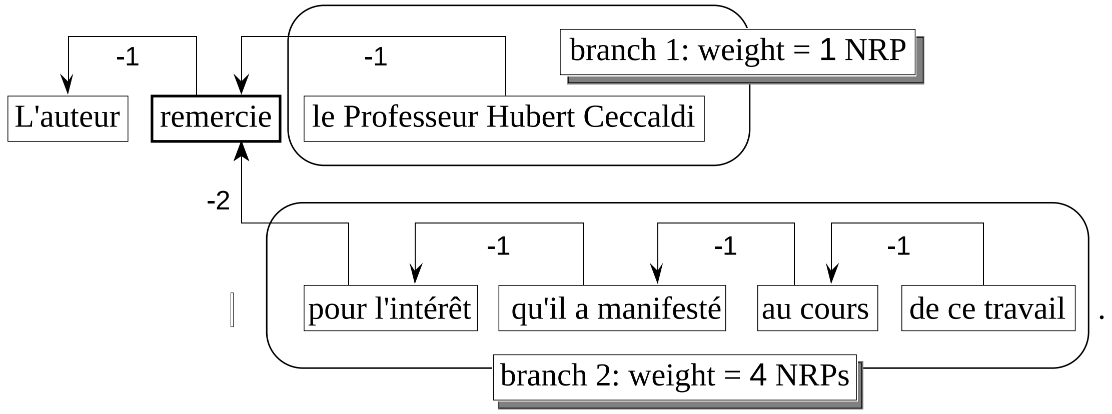

Figure 7: Linear order: verb - object branch - complement branch [10]

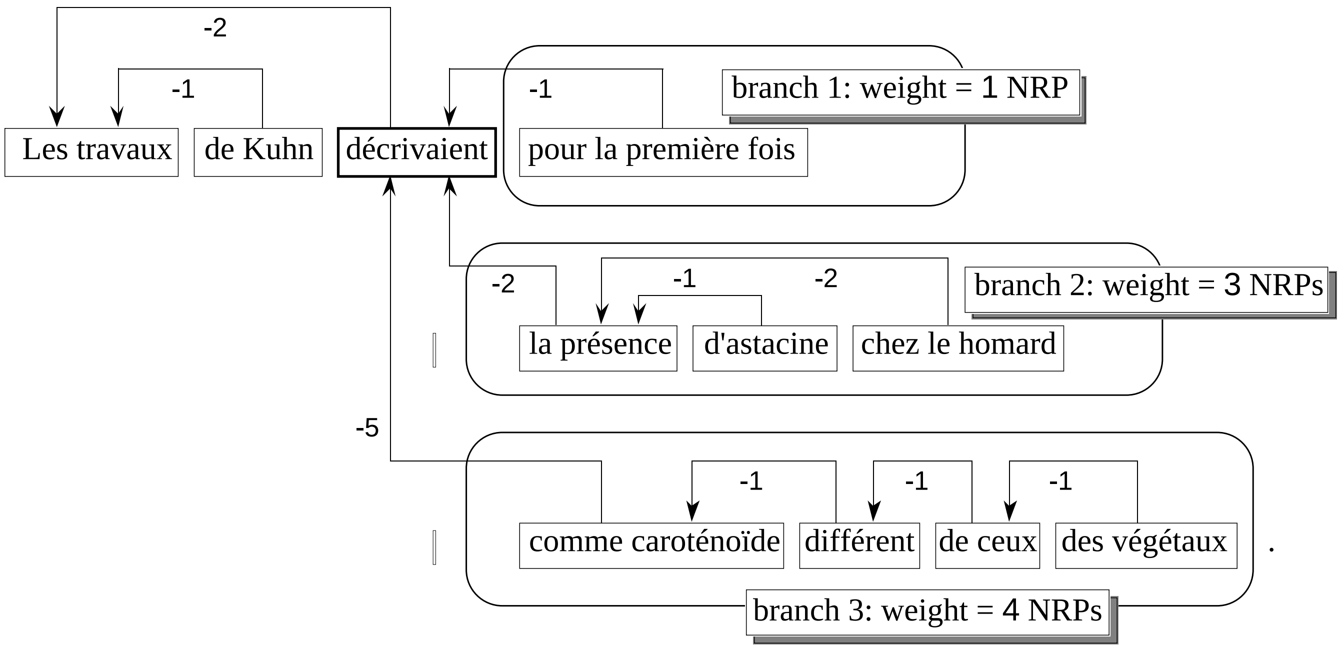

Figure 8: Linear order: verb - complement branch - object branch [11]

In both cases, the shortest branch is the first linearized branch: in the two possible linear orders between two branches, it is the one which minimizes the sum of distances (in absolute value) between linked nodes. This minimization of distances allows a minimization of the memory effort at emission (and reception) and will allow to compute links at reception.
We call this process the "optimized linearization of the dependency tree".

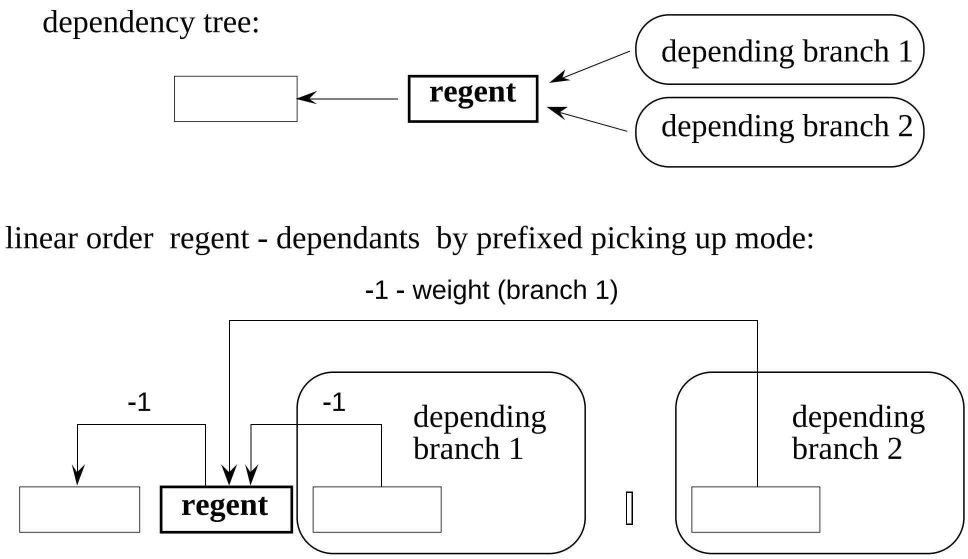

Figure 9: Linearization of a prefixed regent node with two depending branches

2.4 Computing the optimized linearization

We shall now demonstrate this property for French in the case of a prefixed regent node, while taking the abstract case of a regent node followed by its two depending nodes (a regent NRP prefixed before his two dependants). In figure 9 above, branch 1 is conventionally linearized first.
The dependency length of the first node of the depending branch 1 is -1 because this branch is produced the first one, contiguously to its regent. The dependency length of the first node of the depending branch 2 is -1 - weight (branch 1) because this branch is produced as soon as branch 1 is closed. Let us define the "weight" of a branch as its number of nodes, of NRPs, and it generalizes the length of a branch whatever its structure is.

Let us define the optimization criterion in the following way:

> The optimized linearization is the one which, in all possible linearizations, minimizes the sum of the absolute values of dependency lengths.

For the linear order branch 1 - branch 2 (as in figure 9), the sum of the absolute values of dependency lengths is: 3 + weight (branch 1).
For the linear order branch 2 - branch 1, the sum of the absolute values of dependency lengths is: 3 + weight (branch 2).
Therefore, the optimized linearization is the linear order branch 1 - branch 2 if :

$$3 + \text{weight (branch 1)} \leq 3 + \text{weight (branch 2)}$$

or $$\text{weight (branch 1)} \leq \text{weight (branch 2)}$$

> In the case of a regent node followed with its two depending nodes, the optimized linearization is the linear order which places the minimal weight branch as first branch.

While generalizing to any number of branches, the optimized linearization is the linear order which places branches by growing order of weights, as in figure 8, where we have the linear order of three branches of weight 1, 3 and 4.
If we compare the projection of a dependency tree to its linearization, projection implies a linear order which depends on the way the dependency tree is drawn; on the contrary, the optimized linearization by depth-first search does not depend on the dependency tree drawing, but only on the dependency tree structure and the picking up mode of every node.

3. The process: linear order → dependency tree

We shall now study how, during reception, the dependency tree is rebuilt from the linear order, by starting from a dynamic definition of dependency as a process founded on the use of memory.

3.1 Definition of the memorial dependency as a perception process

Let us remember the definition of the "*connexion*" by Lucien Tesnière in (Tesnière 59), page 11, § 3 (actually Tesnière's first page):
*Entre un mot and ses voisins, l'esprit aperçoit des **connexions**, dont l'ensemble forme la charpente de la phrase.* [12]
In this definition, Tesnière presents its "*connexion*" as:

(1) a **process** "*aperçoit*" (perceives)
(2) a mental **process** "*l'esprit*" (mind)
(3) a perception **process** "*aperçoit*"

In our definition of the memorial dependency, we keep these aspects of Tesnière's definition, but only these ones.

We propose the following definition:

> the **memorial dependency** is a relation between two NRPs (and not between two words), defined as a **linking process** during the reception by the hearer - reader, a **process** which is founded on his working memory.

All NRPs (nominal, verbal) are processed in the same way, and conjugated verbs are not always regents (as they are for Tesnière).

3.2 The linking process between two Non Recursive Phrases

We bring the hypothesis that the "memorial dependency" is computed in a two steps process, by the mean of memories which are specialized for a type of relations. We shall study the example of the relation between a nominal subject NRP and a verbal NRP. This relation, during reception, will be computed in two distinct and successive steps:
- **step 1**: a nominal NRP (heard or read) is memorized as an eventual subject, expecting an eventual verbal NRP (see figure 10),
- **step 2**: when a verbal NRP occurs (heard or read), it is linked to the expecting nominal NRP, which then does not expect a verbal NRP anymore, and therefore is "forgotten" from the memory of subjects which are expecting a verb (see figure 11 in next page).

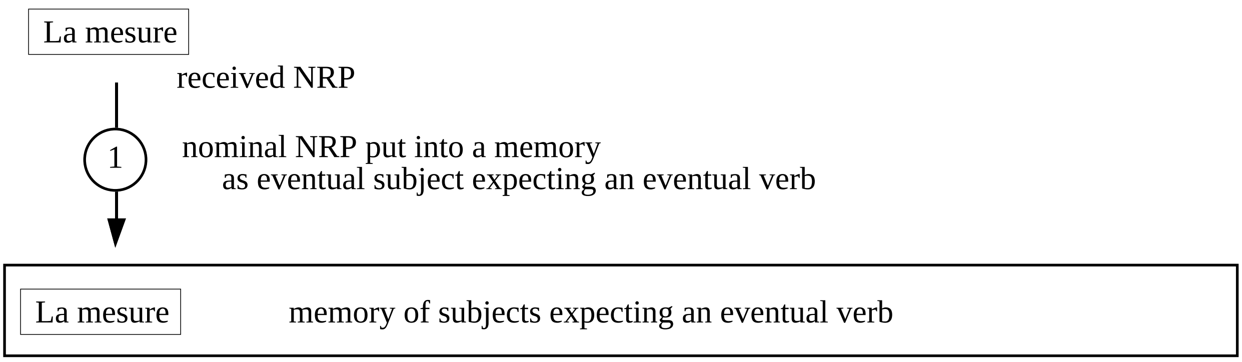

Figure 10: **Step 1** of the linking process between two NRPs

Between step 1 and step 2 of this subject - verb linking process, both nominal NRPs depending on the subject are linked to it by identical processes, but by the mean of the memory of nominal regents expecting nominal dependants.
Step 2 comprises four operations (see figure 11 in next page):
- 2a : the memory of subjects expecting a verb is consulted,
- 2b : a link is created between the verb and the expecting subject,
- 2c : the subject does not expect a verb anymore: it is forgotten from **this** memory,
- 2d : all the complements of the subject are forgotten from all the memories, because they do not expect a complement anymore. The subject has two depending branches: one for its complements, which is now closed, and one for its verb, which is just beginning. This is a consequence of the fact that the linear order is the result of a depth-first search in the dependency tree emitted by the speaker or the writer: in other words, a branch is linearized entirely before another one begins to be linearized.
This linking process in two distinct steps and in particular this operation 2d allow and make con-

crete the interdependency of different linking processes occurring simultaneously during reception.
All these interdependent linking processes give as a result the dependency tree of figure 12 below.

This process is now generalized with memories specialized for every type of link (dependency, co-ordination, antecedence, ...), and it is modelled in the frame of an automatic syntactic parser which outputs the dependency tree as shown in figure 12, with co-ordination and antecedence links.

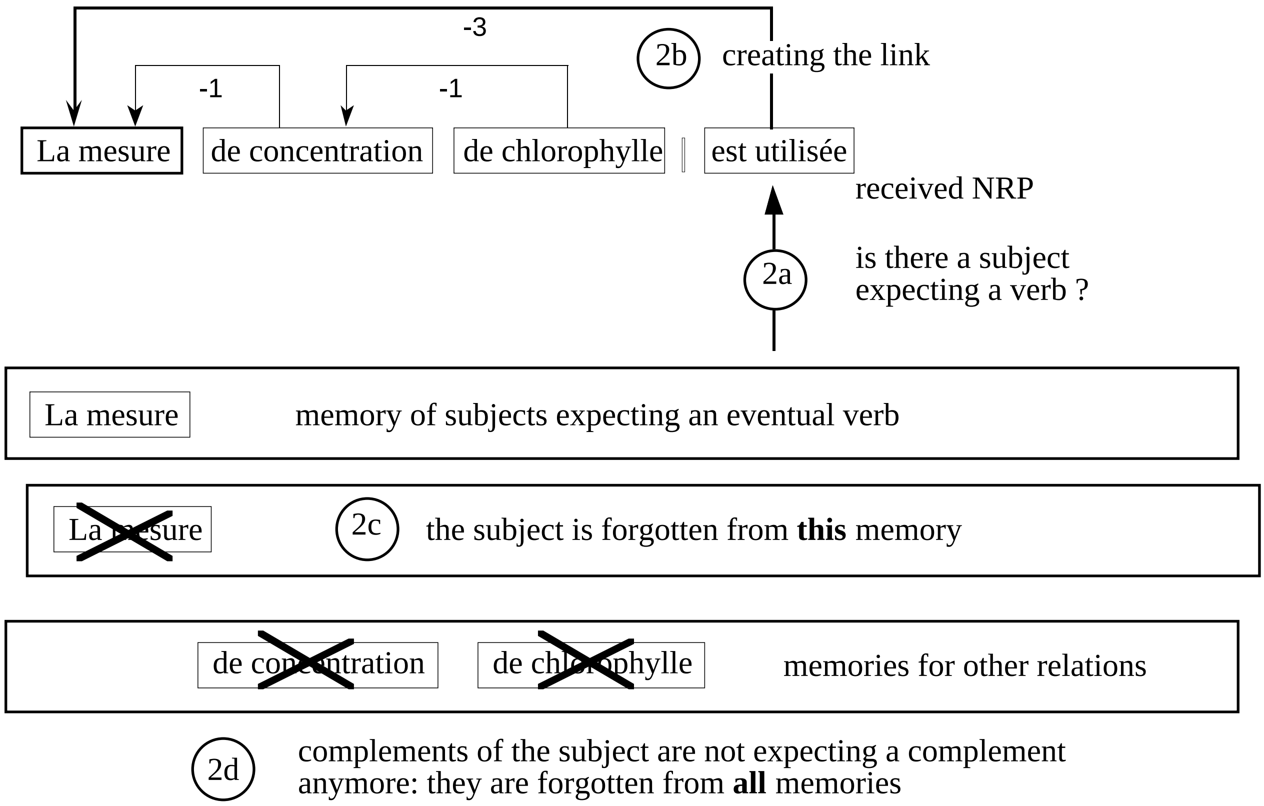

Figure 11: **Step 2** of the linking process between two NRPs

This parser is described in (Giguet-Vergne 97). Its efficiency is already important, and it is possible to see large parsed corpora in French: newspaper articles (Le Monde), literature, scientific texts, on internet at the address: http://www.info.unicaen.fr/~giguet. The parsing efficiency validates the principle of the linking process in two steps by the mean of memories specialized by type of relation. Let us notice that no hypothesis is explicitly done neither on syntactic structures situated between both linked segments, nor on distance between these two segments.
As the computer is a machine that chronologically executes actions, its use as a modelling tool focused us more on the explicitation of processes than on the explicitation of structures, and that drove us to design this linking process, and then enlarge it to the definition of the memorial dependency.

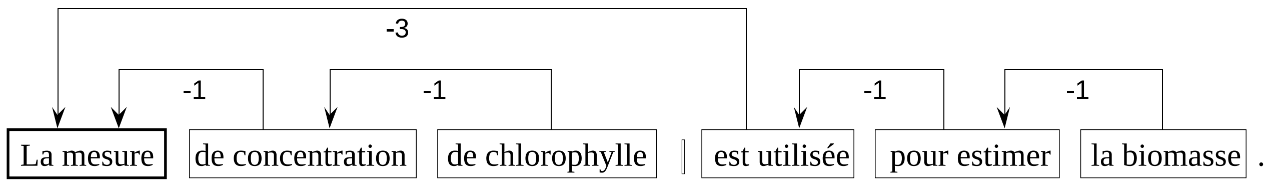

Figure 12: Dependency tree built by the linking processes [13]

4. Conclusion

In this article, we tried to present in a coherent whole both dependency tree and linear order, associated with both processes of production and reception which allow a human being to transmit a dependency tree to another human being, this tree being temporarily coded and compacted into the linear order. We have shown geometric, informational, chronological and memorial constraints which shape these two structures and these two processes.

References

Abney Steven (1996), "Part-Of-Speech Tagging and Partial Parsing", in Ken Church and Steve Young and Gerrit Bloothooft, editors, *An Elsnet Book*, Corpus-Based Methods in Language and Speech, Kluwer Academic, Dordrecht.

Collins Michael John (1996), "A New Statistical Parser Based on Bigram Lexical Dependencies", in *Proceedings of 34th Annual Meeting of the Association for Computational Linguistics (ACL'96)*, University of California, Santa Cruz.

Chomsky, Noam (1969), *Structures syntaxiques*, Point Seuil, Paris.

Chomsky, Noam (1971), *Aspects de la théorie syntaxique*, Seuil, Paris.

Chomsky, Noam (1980), *Essais sur la forme et le sens*, Seuil, Paris.

Giguet E. and Vergne J. (1997), "From part of speech tagging to memory-based deep syntactic analysis", in *International Workshop on Parsing Technologies 1997 Proceedings*, Boston.

Grunig, Blanche-Noëlle (1993), "Charges mémorielles et prédictions syntaxiques", in *Cahiers de Grammaire* n°18, ERSS, Toulouse, pp. 13-29.

Lucas, Nadine (1995), "Le style scientifique en Japonais et en Français", in *Japon Pluriel, actes du 1er colloque de la Société française des études japonaises*, éditions Picquier, Arles, pp. 393-402.

Mel'cuk, Igor (1988), *Dependency syntax : theory and practice*, State University Press NY, Albany, NY.

Stevens, Peter (1973), *Patterns in nature*, Little, Brown & Co. Inc.

Tesnière, Lucien (1953), *Esquisse d'une syntaxe structurale*, Klincksieck, Paris.

Tesnière, Lucien (1959). *Elements de syntaxe structurale*, Klincksieck, Paris.

Vergne, Jacques (1995a), "Les cadres théoriques des Traitements Automatiques des Langues syntaxiques: quelle adéquation linguistique et algorithmique? une étude et une alternative", in *Actes de "TALN'95" Conférence sur le Traitement Automatique du Langage Naturel*, Marseille, pp. 24-33.

Vergne, Jacques (1995b), "Esquisse d'une syntaxe des langues concrètes - Application à l'analyse syntaxique automatique", in *Les cahiers du GREYC*, numéro 11, université de Caen, 180 pages.

---

1 *Pretty marquise, your beautiful eyes make me pine for you.*

2 *8.- We call linear order the one with which words are arranged along the spoken chain. As the spoken chain, the linear order is one dimensioned.*

3 *1.- The structural order of words is the one according to which the connexions are set.*

4 *At the end of the meeting of his cabinet, the president declared that fights which began in December caused the flight of many refugees.*

5 *[...] We can say that [...]* ***speaking*** *a natural language is transforming its structural order into a linear order, and conversely that* ***understanding*** *a natural language is transforming its linear order into a structural order.*

6 *[...] on the syntactic point of view, the true sentence is the* ***structural*** *sentence, and the linear sentence is only its picture projected with difficulty, and with all the flattening disadvantages that this projection on the spoken chain comprises.*

7 *This fish can carry all these pigments to its tissues.*

8 *A distance will be, in such a space, computed by a formula.*

9 *Farmers have arranged the country with too restricting rules.*

10 *The author thanks Professor Hubert Ceccaldi for the interest he showed during this work.*

11 *Kuhn's work described for the first time the presence of astacin in the lobster as a carotenoid different from these of plants.*

12 *Between a word and its neighbours, the mind perceives* ***connexions****, and the whole of these connexions builds the framework of the sentence.*

13 *Measuring the concentration of chlorophyll is used to estimate the biomass.*