A SHORT TERM MEMORY BASED ALGORITHM FOR LINKING SEGMENTS

Jacques Vergne
GREYC - CNRS URA 1526
Université de Caen
F-14032 Caen cedex
France
Jacques.Vergne@info.unicaen.fr

février 22, 2008

Summary

In this paper, we describe a short term memory based algorithm for linking segments. These segments are core phrases. At this step of the parsing, words are already tagged, and core phrases delimited and tagged; the sentence is represented as a chain of core phrases and connectors.

The purpose of the algorithm is to link core phrases, which are the nodes of a dependency tree.

The algorithm is conceived as the simulation of a reading process based on short term memory: the computer memorises a phrase A as waiting for another phrase B; while meeting a phrase B, it reminds A and links B with A. This process is generalised to all link types, while using a specialised memory for every link type.

We make the hypothesis that this algorithm is an organic simulation of a human reading process.

Subject Areas: syntax, cognitive linguistics, parsing

(a demonstration is proposed on French, English and Spanish)

Word Count: 3300

A SHORT TERM MEMORY BASED ALGORITHM FOR LINKING SEGMENTS

Summary

In this paper, we describe a short term memory based algorithm for linking segments. These segments are core phrases. At this step of the parsing, words are already tagged, and core phrases delimited and tagged; the sentence is represented as a chain of core phrases and connectors.

The purpose of the algorithm is to link core phrases, which are the nodes of a dependency tree.

The algorithm is conceived as the simulation of a reading process based on short term memory: the computer memorises a phrase A as waiting for another phrase B; while meeting a phrase B, it reminds A and links B with A. This process is generalised to all link types, while using a specialised memory for every link type.

We make the hypothesis that this algorithm is an organic simulation of a human reading process.

Subject Areas: syntax, cognitive linguistics, parsing

(a demonstration is proposed on French, English and Spanish)

Word Count: 3300

1. PURPOSE OF THE ALGORITHM

The purpose of the algorithm is to link core phrases (noun, verb, adjective or adverb core phrases), which are the nodes of the dependency tree to be built.

1.1. Input: sentence representation

At this step of the parsing, words are already tagged, and core phrases are delimited and tagged according to their type; the sentence is represented as a chain of core phrases and connectors (prepositions, conjunctions, punctuation).

Every core phrase is replaced by a one letter code according to its type: noun, verb, adjective or adverb core phrase, and when it is a verbal phrase, a subtype is coded: conjugated, infinitive, present or past participle, and whether it is transitive or not.

We obtain an abstracted representation, detached from word level, detached from language type; the linking process works on this representation.

1.2. Output of the linking process

The output of the linking process is the dependency tree: the nodes are not words but core phrases, and the link is the dependency between core phrases, in a new definition, given below in section #4.

Here is an example of the dependency tree:

sentence	representation in core phrases and connectors
la mesure	N
. de concentration	. pN
. . de chlorophylle	. . pN
. **est utilisée**	. V
. . pour **estimer**	. . pI
. . . la biomasse [1]	. . . N

(N: noun core phrase, V, I: conjugated, infinitive **verb core phrase**, p: preposition)

The following explicit dependencies between core phrase heads are output:

mesure de concentration	N pN
concentration de chlorophylle	N pN
mesure **est utilisée**	N V
utilisée pour **estimer**	V pI

[1] Translation: the chlorophyll concentration mesure is used to estimate the biomass.

estimer biomasse I N

2. PRINCIPLES OF THE ALGORITHM

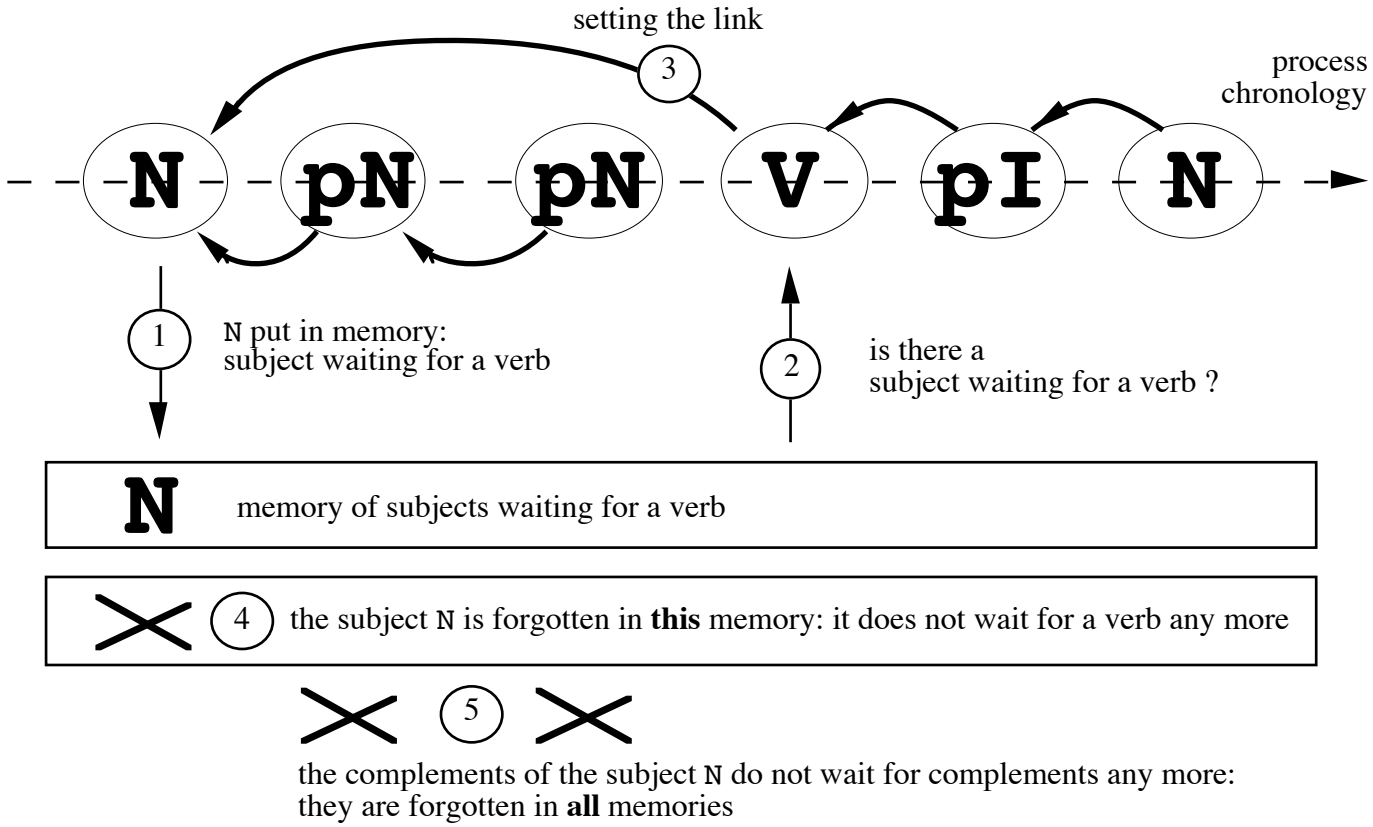
2.1. Process description

The process simulates a hypothetical human linking process (while listening or reading) based on short term memory:

process	an example
-1- the computer memorises a phrase A	(a subject N waiting for its verb V)
-2- it reminds it while meeting a phrase B	(a conjugated verb V)
-3- it links B to A (the most recent A)	(subject N <-- verb V)
-4- it forgets A	(this subject N does not wait for a verb any more)
-5- it forgets all phrases situated between A et B	(subject complements does not wait for complements any more)

Let us study how the link between subject and verb is computed: la mesure **est utilisée** .

At the beginning of the process, the sentence is represented of as a chain of core phrases and connectors. Here is a diagram which illustrates the process working on this representation:



the branch 1 of N subject (its complements) is closed, and V is the beginning of its branch 2

Phrases are processed from left to right, in the order in which a listener - reader receives the sentence.

The forgettings in operations 4 et 5:

- force, restrict every link which will be computed after,

- realise a property of the linearised dependency tree: a closed branch is definitely closed.

The basic idea of this linking process is that we do not look for a subject for a processed verb in the **past** (on the left in the structure), but we observe the **present** state of memories: phrases which wait for a link are put in a waiting memory at the moment of their processing. A link is not set in one operation while processing the second phrase and looking backwards, but in 3 operations (1, 2 et 3) by the mediation of a memory: putting in a memory, reading this memory, and linking.

2.2. Generalisation of the linking process

Every dependency and coordination link is computed in the same way; I intend to extend this process to link also anaphoric coreferents, and words as: "ne ... que", "plus ... que".

Specialised memories (at present 7) manage different link types:

• dependency links between subjects, verbs and objects:

- memory of subjects waiting for a verb,
  of verbs waiting for an object,
  of connectors waiting for a phrase,

• other dependency links:

- memory of regents waiting for a postposed dependant,
- memory of anteposed dependants waiting for a regent,

• coordination links:

- memory of connected (preposition, conjunction) phrases waiting for a coordinated phrase,
- memory of subject phrases waiting for a coordinated subject phrase,
- memory of non-connected (no preposition, nor conjunction) and non-subject phrases waiting for a coordinated phrase,
- memory of relative clauses waiting for a coordinated phrase (it may be a relative clause).

3. IMPLEMENTATION WITH DECLARATIVE RULES

3.1. Implementation of the linking process:

The linking process is implemented with interpreted declarative rules. These rules (about 80) are in 2 types, according to their function:

1) **memorising a left phrase** in a specialised memory, waiting for a right phrase: it is the operation 1 of

the process (about 30 rules),

2) **recognising a right phrase and linking it** to a left phrase found in a specialised memory: it is the operation 2 of the process (about 50 rules).

These rules have the following form:

filter of conditions on attributes of phrases and connectors (the elements of the representation)

=> actions on these elements

Here is the vocabulary used in these rules:

conditions: written form, lemma of the phrase head,
phrase type,
presence of a phrase in a specialised memory,
determined noun phrase, nominalised verb phrase,
phrase which agrees with, isomorphic (for coordination) phrase with,
phrase which depends on, phrase which is the regent of,
phrase which is contiguous to (before, after);

actions: putting a phrase into a specialised memory,
forgetting a phrase from a specialised memory,
pointing out the dependant,
pointing out the regent,
pointing out the coordinated.

Examples of some rule principles:

• type 1) **memorising a left phrase :**
- every non-connected phrase (no preposition, nor conjunction) is placed in the memory of subjects waiting for a conjugated verb;
- every phrase is placed in the memory of regents waiting for a postposed dependant;
- every subject noun phrase is placed in the memory of subjects waiting for a coordinated subject phrase.

• type 2) **recognising a right phrase and linking it** :
- if the current phrase is a conjugated verb phrase, and if there is a noun phrase in the memory of subjects waiting for a conjugated verb, then link the verb to the subject;
(if there are more than one waiting phrase in a memory, the link is set with the last one)
- if the current phrase is a conjugated verb phrase, and depends on "qui" ("who"), and if there is a noun phrase in the memory of regents waiting for a postposed dependant, and if this noun phrase is determined and agrees with the verb, then link "qui" to the noun phrase, its antecedent.

At every application of a rule of type 2 (link setting), some metarules manage the forgettings: a forgetting is an erasement in a memory (operations 4 and 5 of the linking process):

- forgetting the left phrase in a memory which has been used for linking (operation 4 of the process);

this forgetting implies the unicity of the link for the left phrase (a subject has an unique verb); this forgetting is done in every memory except the one of regents waiting for a postposed dependant: a regent may have more than one postposed dependant;

- forgetting a right object noun phrase in the memory of subjects waiting for a verb (an object cannot be a subject);
- forgetting every element situated between 2 linked phrases in every memory (operation 5).

The representation of the sentence is processed from left to right; let us call "current element" the processed element; the chronology of rule application made some problems, which have found a solution by applying some simple principles:

• The right phrase to be linked is not always the current phrase :

In the rule which links a verb to its subject, the current phrase is the right phrase to be linked; but in the rule which links a relative pronoun "qui" with its antecedent, the right phrase to be linked is not the current verb phrase: we must link "qui" to its antecedent, and not the current verb phrase to the antecedent. Conclusion: in the rules, we must dissociate the current phrase from the right phrase to be linked, they are 2 different elements of the rule.

• Moment of the decision: we must immediately decide with available data at the moment when the current phrase is processed, without waiting for a fact which can never occur. With this principle, it is sometimes necessary to go back over a decision, for example in these 2 cases where a human being goes back over his decision too:

- coordinated objects or coordinated clauses?

Le mot d'interprétation **signifie** une simple catégorisation de phénomène
, c'est-à-dire son affectation à une famille de phénomènes ressemblants ; [1]

N pN **V** N pN ,cN pN pN (c: coordination conjunction)
The coordination is immediately set between 2 objects (N cN)
without waiting for a verb which would have "affectation" for subject.

[1] Translation: The word "interpretation" means a simple phenomenon categorisation, that is its affectation to a resembling phenomena family.

La détermination de_ les équations [1] de ces hyperplans **a fait** l'objet de nombreux travaux , et les probabilités théoriques d'erreur **ont été largement étudiées** . [2]

N pN pN **V** N pN ,cN pN **V**

The coordination is immediately set between 2 objects (N cN)
without waiting for a verb which would have "probabilités" for subject.

This coordination is undone when the verb "**ont été largement étudiées**" arrives
and is replaced by the coordination of 2 clauses (NV cNV).

- case of the object relative pronoun: criterion = "que" saturates an object valency

On **verra** ce que leur utilisation **apporte** dans la résolution de certains problèmes . [3]

N V N P N V pN pN (P: subordination conjunction)

"que" is relative pronoun, object of "**apporte**" with "ce" as antecedent
without waiting for an object which would have "**apporte**" for verb.

Il **s'agit** de **programmer** l'ordinateur de telle sorte qu'il **soit capable** d'**interpréter** les sensations **reçues** à travers ces capteurs . [4]

NV pI N pN PN U pI N ù pN
(U: intransitive conjugated verb core phrase; ù: past participle verb core phrase)

"qu'" is relative pronoun, object of "**interpréter**" with "sorte" as antecedent
without waiting for an object which would have "**interpréter**" for verb.
"qu'" object relative pronoun is undone when the object "sensations" of "**interpréter**" arrives.

"sensations" = object of "**interpréter**" => "qu'" = subordination conjunction

Going back over a decision at the following element is also a decision: the linking process remains linear, and the rule propagation goes on; going back over a decision at the following element by another decision is not a backtracking:

going back over a decision:	*backtracking:*
- going forwards to the following element - data values are modified, and a datum is set to its value of the antepenultimate element	- going backwards to the preceding element - going backwards to the values of data of the preceding element

[1] The amalgam preposition-determiner "des" is cut into two parts: "de_ les", to be exactly processed as "de ces".

[2] Translation: The equation determination of these hyperplanes has been described in many works, and the theoretical error probabilities have been widely studied.

[3] Translation: We will see what their use brings in the resolution of some problems.

[4] Translation: We must program the computer in such a way it is able to interpret feelings received through these captors.

3.2. Evaluation of the quality of the linking process

There are 2 questions:

- has the parser made a decision?
  sometimes a regent or a coordinated phrase is not found
- has the parser made a right decision?

This evaluation is done by comparing links which are expected by a person who knows the language of the text, with links which are computed by the parser. The expected links are explicited by editing a file which had been output by the parser and contains the links; the edited file is then placed as input, to automatically compute the gap between expected and computed links. An error rate is a ratio equal to the number of errors divided by the total number of links.

On the 2 texts of the French corpus [1], there are error rates of respectively 9.7% and 12.9% for dependencies between core phrases (on 2300 and 2000 dependencies).

But these error rates are very different according to the link type:

- error rates of 2.1% and 1.3% for dependencies between subject, verb, object, and between connectors and phrases (on 1200 and 1000 dependencies)
- error rates of 17.2% and 20.7% for other dependencies between core phrases (on 1100 and 1000 dependencies).

Dependencies of prepositional phrases to their regent are about 40% of these other dependencies; their error rates are 12.2% and 13.9% (on 480 and 380 dependencies). The error rates on computing coordinations between core phrases are 39.4% and 32.6% (on 70 and 100 coordinations between core phrases); the coordination rate is 4.9% and 8.9% (coordinated phrases on total of phrases). It is probably possible to improve the quality of the linking process by finer linguistic studies on corpora, on every particular link type.

4. LINGUISTIC HYPOTHESIS: ANOTHER DEFINITION OF DEPENDENCY

The dependency used here is different from the classic definition of dependency:

- it is a link between core phrases and not between words
- it is defined by the linking process and not statically.

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

In a way, we come back to some aspects of the Tesnière's original definition:

"entre un mot et ses voisins, l'esprit aperçoit des connexions" (Tesnière, 1959, page 11) [1]

("connexion" is Tesnière's term for dependency)

He adds: "ces connexions ne sont indiquées par rien" [2].

But he wrote also (in the "esquisse", in Tesnière, 1953, page 4):

"deux mots en connexion sont généralement placés l'un à côté de l'autre sur la chaîne linéaire" [3]

Tesnière's definition has three common points with the one I propose:

(1) it is a process ("aperçoit"),

(2) a mental process ("l'esprit"),

(3) a process concerning perception ("aperçoit").

5. COGNITIVE HYPOTHESES

5.1. Limits of human short term memory while linking process

Memorisation limits of the speaker - writer and of the listener - reader are constraints which shape forms of natural languages. Mainly, these limits force the phrase linking process.

These memorisation limits are of 3 types:

Memorisation duration limit for a phrase waiting for a link:

After a certain time, probably very short, a waiting phrase is forgotten; this duration limit implies a distance limit in written form: linked segments have to be close, contiguous if possible, as words inside a core phrase, or linked core phrases inside a sentence.

Maximum number of core phrases waiting for a link of the same type:

There is generally zero or one subject waiting for a verb; very rarely, there are two stacked subjects waiting for two verbs, but I never observed three stacked subjects waiting for three verbs.

When we listen or read an inserted block (for instance, a prepositional phrase, or a relative clause after a subject), we must make a memory effort to remind the subject, while waiting for the verb, to link the verb to the subject:

[1] Translation: between a word and its neighbours, the mind perceives connections.

[2] Translation: these connections are given by nothing.

[3] Translation: two connected words are generally placed one beside the other on the linear chain.

structure: [N [pN] V]
insertion level: 0 1 0
link at level 0: <----------

If there is an inserted block at level 2 inside the inserted block at level 1, we must successively stack in memory the first subject at level 0, then the second subject at level 1, then wait for the verb at level 1, to link it to its subject (the second), and at last wait for the second verb, to link it to its subject (the first) at level 0:

structure: [N [O N [pN] V] V] (O: object relative pronoun)
insertion level: 0 1 1 2 1 0
link at level 1: <----------
link at level 0: <-------------------------

It seems that it is an important memory effort, and it could explain why the insertion at level 3 has never been observed, and why an inserted subordinated clause at level 2 is very rare.

Maximum number of phrases waiting for a link of any type:

If we cut the linearised three between two contiguous core phrases, we cut a certain number of links, equal to or greater than 1; this number is equal to or lower than the number of phrases waiting for a link of any type at a moment of the sentence; let us call it the number of simultaneous links.

In a very simple sentence, where every dependency between core phrases is marked by contiguity, this number is constant and equal to 1. As soon as a node (a core phrase) has more than one branch (more than one dependant), this number, just after this node, is equal to or greater than the number of branches (there may be another link).

Quantitative evaluation of the memory effort at a given moment in the sentence:

At a given moment in the sentence, between two contiguous core phrases, we can quantify the memory effort (for the speaker - writer and the listener - reader) with the following data:

- the number of simultaneous links (from 1 to about 4),
- for each of them, their length in number of core phrases (-1 or 1 in case of contiguity),
- the number of phrases waiting for a link of the same type, mainly subject - verb (from 0 to 2).

5.2. Programmed processes and human processes

A priori, the aim of programmed processes (automatic parsing) is either to be a research tool into natural language syntax, either to parse texts and to delimit, tag and link segments, but this aim is not to simulate human processes (listen - read), neither to study and understand these processes, neither to imitate functions and/or means of these human processes in an efficiency aim.

But programmed processes and human processes are not strangers ones to the others: programmed processes have functions which are comparable to human processes: the parsing situation with an input text, the cutting up into segments, the identification of segments, and their linkage; we can say that programmed processes make a **functional** simulation of these human processes. Let us call "**organic** simulation" the case where a programmed process imitates functions **and** means, algorithms of a human process.

Therefore, a priori, programmed processes of automatic parsing do not make an organic simulation of human processes of reading - listening.

But, a posteriori, we must wonder if programmed processes make, without wanting it, a certain **organic** simulation of human processes: as the 2 types of processes process the same object and have the same functions, then the same global constraints are exerted on them: the linear aspect of the objet, the properties of the linearisation of the dependency tree, the logic of the chronological feasibility of operations; at last, this same object is shaped by constraints of human origin: memory limits, the features of the vocal system, the features of the respiratory system, the features of the brain. This identity of the objet, of functions and constraints partly induces this organic simulation, even if we do not intend it.

Then, we can ask a question about connections between syntax and human processes of reading - listening: properties of the processed and observable object (the forms of a natural language) may be partly studied while considering the static object, but to go further into, it is necessary to view automatic parsing processes in their dynamic and constrained aspects because similar processes may be performed by a human being (with a certain organic simulation): forms are the result of a play of human and spatio-temporal constraints, and become more understandable if we study these forms not only in themselves but also as clues of this play of constraints.

The most flagrant instance is the dependency definition as a process, together linguistic definition, automatic parsing algorithm, and hypothesis on the human linking process.

REFERENCES

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

Lucien Tesnière. 1959. Éléments de syntaxe structurale. *Klincksieck*, Paris (last edition: 1982).

•