A HYBRID MODEL OF HUMAN SENTENCE PROCESSING: PARSING RIGHT-BRANCHING, CENTER-EMBEDDED AND CROSS-SERIAL DEPENDENCIES

THEO VOSSE
GERARD KEMPEN
N.I.C.I., University of Nijmegen
Montessorilaan 3
6525 HR Nijmegen
The Netherlands
Phone: +31 80 512621
Internet: vosse@psych.kun.nl
kempen@psych.kun.nl

ABSTRACT

A new cognitive architecture for the syntactic aspects of human sentence processing (called Unification Space) is tested against experimental data from human subjects. The data, originally collected by Bach, Brown and Marslen-Wilson (1986), concern the comprehensibility of verb dependency constructions in Dutch and German: right-branching, center-embedded, and cross-serial dependencies of one to four levels deep. A satisfactory fit is obtained between comprehensibility data and parsability scores in the model.

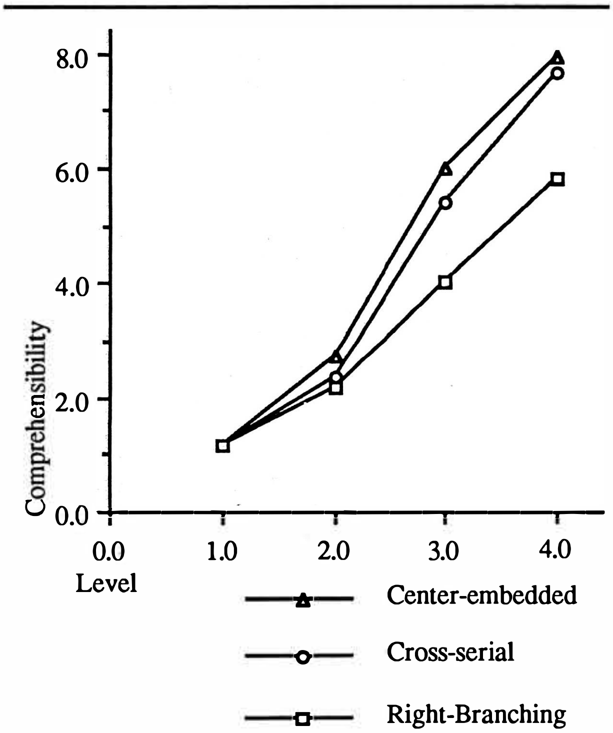

Figure 1. Comprehensibility ratings for various construction types and depths (1 = very easy, 9 = very hard).

INTRODUCTION

In a recent paper (Kempen & Vosse, 1990), we have proposed a new cognitive architecture for the syntactic aspects of human sentence processing. The model is 'hybrid' in the sense that it combines symbolic structures (parse trees) with non-symbolic processing (simulated annealing). The computer model of this architecture — called *Unification Space* — is capable of simulating well-known psycholinguistic sentence understanding phenomena such as the effects of Minimal Attachment, Right Association and Lexical Ambiguity (cf. Frazier, 1987).

In this paper we test the Unification Space architecture against a set of psycholinguistic data on the difficulty of understanding three types of verb dependency constructions of various levels of embedding[1].

[1] A recent paper by Joshi (1990) motivated us to do the present study. He succeeds in obtaining a good fit between Bach *et al.*'s data and a complexity measure deriving from his model, which is based on an Embedded Push-Down Automaton (EPDA) and Tree Adjoining Grammar (TAG).

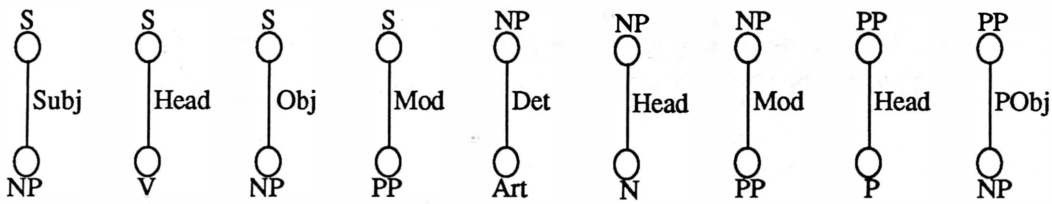

Figure 2. Various types of synactic segments.

The data were collected by Bach, Brown and Marslen-Wilson (1986) and concern comprehensibility ratings of cross-serial, center-embedded and right-branching constructions as illustrated by (1). Subjects rated two types of verb dependencies: right-branching and either center-embedded (German) or cross-serial (Dutch) dependencies.

Dependency type

(1a) ... when John saw Peter walk
Right-branching

(1b) ... als Johan Peter laufen sah
Center-embedded (nested)

(1c) ... toen Jan Peter zag lopen
Cross-serial (crossed)

The right-branching constructions are quite common in Dutch and German. German sentences were rated only by native speakers of German, Dutch sentences only by native speakers of Dutch. Figure 1 shows the obtained comprehensibility (or rather, *in*comprehensibility) ratings for four 'levels' (the term level refers to the depth of embedding; level 1: one clause, without embeddings; level 2: two clauses, one embedded in the other as in (1), etc.). Notice that the (Dutch) crossed dependencies were consistently rated easier to understand than the (German) nested dependencies. From level 3 onward, the right-branching structures were judged easier than their crossed or nested counterparts. Via a question-answering task Bach *et al.* verified that the comprehensibility ratings indeed reflect processing loads (real difficulties in comprehension).

In Section 2 we outline briefly the type of grammar we use to represent syntactic structures. The parsing mechanism capable of building such structures is described in Section 3. Section 4 is devoted to design and results of the computer simulation. In Section 5, finally, we evaluate our results and draw some comparisons with alternative computational models proposed in the psycholinguistic literature.

SEGMENT GRAMMAR

Kempen (1987) introduced Segment Grammar as a formalism for generating syntactic trees out of so-

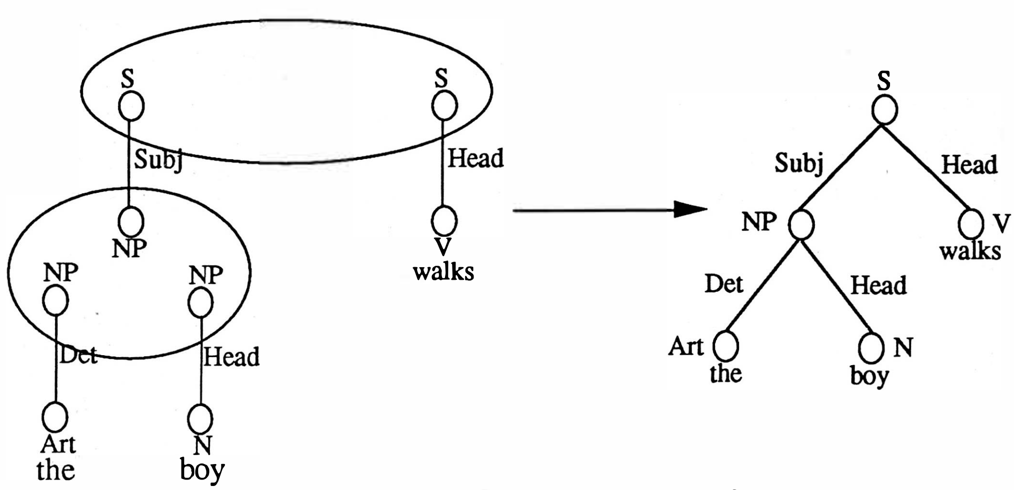

Figure 3. Building a tree through unification.

called segments. A segment is a node-arc-node triple, the top node being called 'root' and the bottom node 'foot'. Both root and foot nodes are labeled by a syntactic category (e.g. S, NP) and have associated with them a matrix of features (i.e., attribute-value pairs). Arc labels represent grammatical functions. See Figure 2 for some examples. All syntactic knowledge a segment needs (including ordering rules) is represented in features.

The basic tree formation operation is *unification* of the feature matrices of nodes which carry the same category label. In Figure 3 successful unification has been visualized as the merger of the corresponding nodes.

Segment Grammar is completely lexicalized. Every lexical entry specifies a single segment or a sub-tree consisting of several segments. For instance, one entry for the English verb *eat* looks like Figure 4. It specifies the subcategorization features for this verb, including the fact that it can take zero or more modifiers (Mod*) in the form of prepositional or adverbial phrases. For more details about Segment Grammar (including the Dutch sentence generator based on it) see De Smedt (1990).

THE UNIFICATION SPACE

The dynamics of the Unification Space model were inspired by the metaphor of bio-chemical synthesis. Think of the segments as molecules floating around in a test-tube and entering into chemical bonds with other molecules (unification of nodes). The resulting larger structure may be insufficiently stable and fall apart again. After a break-up, the segments continue their search for suitable unification partners until a stable 'conformation' — that is, the final parse tree — has been reached.

Henceforth, we denote the test-tube by the term Unification Space. Words recognized in the input string are immediately looked up in the mental lexicon and the lexical entry listed there is immediately entered into the Unification Space. In case of an ambiguous input word, all entries are fed into the system simultaneously.

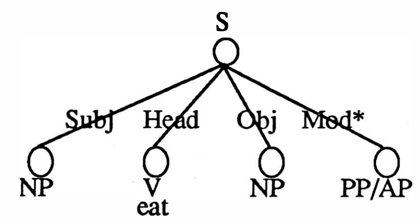

Figure 4. Lexical entry for the transitive verb 'eat'.

The following principles control the events in the Unification Space (see Kempen & Vosse, 1989, for details):

- *Activation decay.* When the nodes are entered into the Unification Space they are assigned an initial activation level by their lexicon entry. This activation level decays over time.

- *Stochastic parse tree optimization.* Generally, on the basis of its feature composition, a node could unify with several other nodes present in the Unification Space. In order to make the best possible choice, Simulated Annealing is used as a stochastic optimization technique (cf. Sampson, 1986). If two nodes *can* unify, they actually unify with probability p_U. This probability depends, among others, on the activation level of both nodes and on the grammatical 'goodness of fit'. Various syntactic and semantic factors are at stake here. Among the former are word order constraints. For instance, if during the analysis of *He gave that girl a dollar* the article *a* would attempt to unify with the noun *girl*, this would cause violation of a word order rule and drastically reduce the value of p_U. Assigning *a dollar* the role of indirect object would be evaluated as less good than as direct object, both for syntactic and semantic reasons.

  On the other hand, unified nodes may break up, with probability p_B. This probability increases accordingly as the activation of the nodes and/or their grammatical goodness of fit decrease. One consequence of this scheme is a bias in favor of semantically and syntactically well-formed syntactic trees encompassing recent nodes.

- *Global excitation.* Due to the spontaneous decay of node activation and the concomitant rising p_B, all unifications would ultimately be annulled in the absence of a mechanism for intercepting and 'freezing' high-quality parse trees. In standard versions of simulated annealing one obtains this effect by making both p_U and p_B dependent on a global 'temperature' variable T which decreases gradually according to the 'annealing schedule' which has been determined beforehand. We define a parameter E (for global Excitation) whose function is similar to that of temperature. However, E's value does not decrease monotonically as prescribed by some annealing schedule but is proportional to the *summed activations* of all nodes that currently populate the Unification Space.

The relation between E on one hand and p_U and p_B on the other is such that, after E has fallen below a threshold value ('freezing'), no unifications are attempted anymore nor can unified nodes become dissociated. If the resulting conformation consists of exactly one tree, the parsing process is said to have succeeded. If several disconnected, partial trees result, the parsing has failed.

It is important to note that the workings of the Unification Space prevent the parallel growth of multiple parse trees spanning the same input string. In other words, *structural* (syntactic) ambiguity is not reflected by multiple parse trees. Only in case of *lexical* ambiguity can there be parallel activation of several segments or subtrees. This agrees with the picture emerging from the psycholinguistic literature (cf. the survey by Rayner & Pollatsek, 1989).

We now describe the essence of the computer implementation of the Unification Space model. Mathematical details can be found in Kempen & Vosse (1989).

1. Time is sliced up into intervals of equal duration. During each cycle, one iteration of the basic algorithm is carried out. This process stops when E has fallen below the threshold value.

2. Words recognized in the input sentence are stored in an input buffer for a limited period of time, T_B. Individual words are read out from left to right at fixed intervals $T_w << T_B$. Their corresponding lexical entries are immediately entered into the Unification Space.

3. During each cycle, two nodes, n_1 and n_2, are picked at random. If their feature composition permits unification, they actually unify with a probability of p_U which covaries with n_1's and n_2's activation levels. The activation level of the resulting single node is higher than the activation level of either n_1 or n_2.

4. Then, for each segment in the Unification Space, it is determined whether or not it will dissociate from its unification partner (if any). This event takes place with probability p_B which correlates negatively with the activation level. Whenever *lexical* segments are are involved in a break-up (lexical segments have word classes rather than phrases as their foot labels), their lexical entries are reentered into the Unification Space without delay. Thus they are given a new chance to find a suitable unification partner. The activation levels of reentering nodes are reset to the initial value stored in the lexicon. However, if a word has already been dropped from the input buffer, its lexical entry is *not* reentered.

5. The activation levels of all nodes are adjusted on the basis of the decay parameter and the new value for E is computed.

THE SIMULATION STUDY

In our earlier study we obtained satisfactory simulation results for the sentences in (2).

(2a) The rat the cat chased escaped.
(2b) The cat chased the rat that escaped.
(2c) The rat the cat the dog bit chased escaped.
(2d) The dog bit the cat that chased the rat that escaped.

The Simulation Space had virtually no problems in parsing doubly embedded sentences (2a) and (2b): the number of correct solutions was close to 100 percent. However, this score dropped considerably for triply embedded clauses: to about 80 and 50 percent for righthand and center-embeddings respectively[2]. This pattern is in good agreement with psycholinguistic observations.

In order to avoid controversial assumptions about the syntactic structure underlying cross-serial dependencies, we have devised simple artificial grammars which generate right-branching, center-embedded and cross-serial dependencies among pairs of opening and closing brackets, e.g. '(){}', '({})' or '({)}'. The grammars contain two types of lexical segments (with arc labels Left and Right) and one optional type of non-lexical segments with arc label Mod. The number of Mod segments dominated by an S node is either zero or one The optional Mod segment is attached to the lexical entries of opening brackets as depicted in Figure 5. It is the Mod segments that give the grammar a recursive flavor.

The S nodes have associated with them a 'bracket type' feature whose value is 'round', 'curly', 'square', etc. This prevents unification of S nodes

[2] These numbers have been computed as described in footnote 3 below.

that dominate brackets of different types, e.g. S-Left-{ with S-Right-].

The sole difference between the three grammars rests in their word order constraints. Center-embeddings require the embedd*ed* subtree to be positioned inbetween the branches of the embedd*ing* S. (The constraints for both other grammars are easy to devise.) However, there was no need to have the Unification Space actually check word order constraints because we never used input strings which contained more than one pair of brackets of the same type (e.g. '{}{}') and/or more than one type of embedding (e.g. '[<>]{}'). Thus word order constraints are in effect encoded in the bracket type feature.

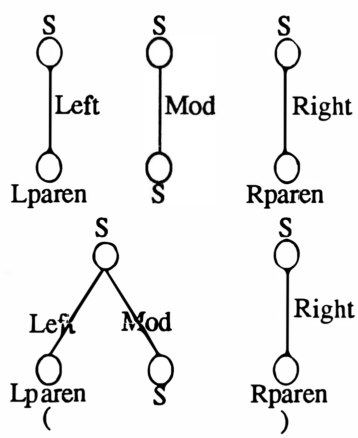

Figure 5. Segments of the grammar, and the lexical entries for '(' and ')'.

The actual simulations were run with 5 (levels) times 3 (dependency types) equals 15 different input strings. Each string was fed into the Unification Space 400 times. *The parameter settings were exactly equal to those used in the earlier Kempen & Vosse (1989) paper*[3]. No attempts have been made to find a set of parameter values yielding a better fit with Bach *et al.*'s empirical data.

The simulation results for the 15 sentences are displayed in Figure 7. They show the same general pattern as the comprehensibility ratings displayed in Figure 2 above. That is, (1) comprehensibility decreases with increasing depth of embedding, (2) center-embedded dependencies are harder than cross-serial dependencies, and (3) right-branching dependencies take a strong lead, being much easier to understand than both other constructions.

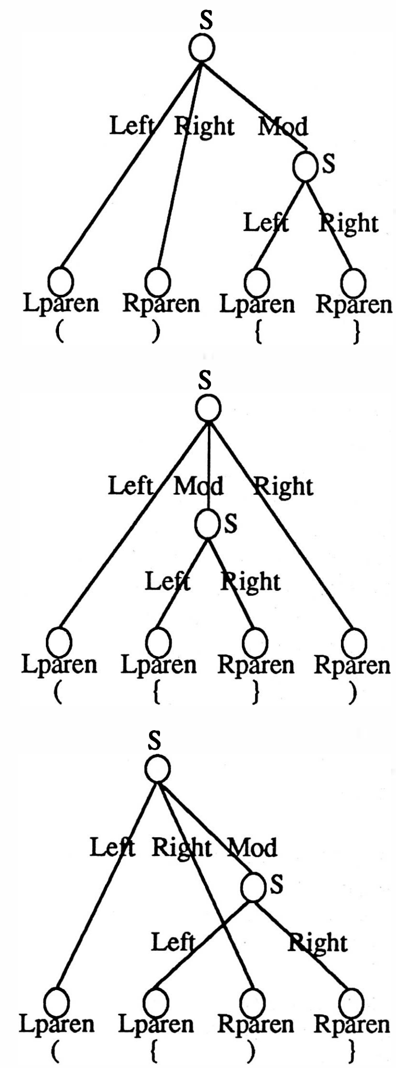

Figure 6. Example parse trees of level 2: respectively right-branching, center-embedded and cross-serial.

There are also differences between the human data and computer simulation, however. First of all, the comprehension scores for the three dependency types fan out more rapidly in our simulation than in the human subjects. Second, in the human data the first signs of a differentiation between sentence types manifest themselves already at level 2, whereas in our simulation the percentages start diverging at level 3 only. From our previous study we know that the Unification Space is rather sensitive to sentence length. If this applies to human readers as well, we could argue that our level 1 and level 2 scores are too

3 For Chaos parameter C (not discussed in the present paper) we had four different values: .1, .2, .3 and .4. There were 100 runs for each value of C. In Figure 7 we show percentages averaged over C values.

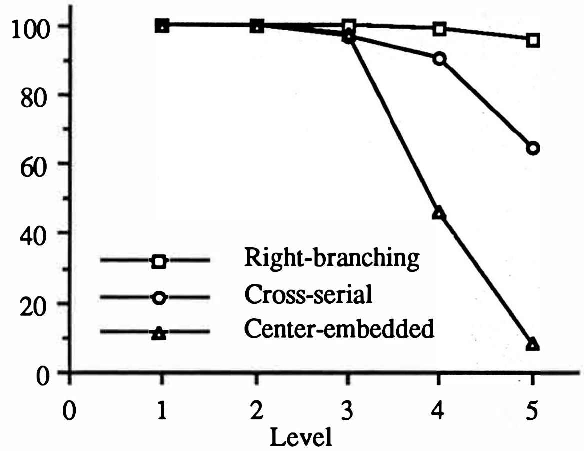

Figure 7. Percentages of correctly parsed strings for three types of dependency and five levels of depth.

good (in Bach *et al.*'s study, these levels were tested through sentence of 6 to 8 words long).

DISCUSSION

The simulation revealed a satisfactory fit between the empirical pattern of comprehensibility ratings observed by Bach *et al.* and parsability by the Unification Space. Since the model applied exactly the same grammar when processing the three types of dependencies, it follows that the empirical pattern can be explained in terms of the *different spatial-temporal arrangements* between the members of a dependency pair. No additional assumptions about differences between the syntactic structure underlying the three types of dependencies are needed.

To what extent are alternative computational models of human sentence processing capable of accounting for the empirical pattern? So far, Joshi's (1990) proposal is the only one reported in the literature. However, it is not clear how well this model behaves with respect to other psycholinguistic sentence processing phenomena such as Right Association, Minimal Attachment, Verb Frame Preferences and the like. Two other recent models (Gibson, 1990a,b,c; McRoy & Hirst, 1990) do address the latter phenomena but they pay no attention to cross-serial dependencies. So, as far as we know, there is no competing model of comparable wide coverage.

REFERENCES

Bach, Emmon, Colin Brown & William Marslen-Wilson (1986). Crossed and nested dependencies in German and Dutch: a psycholinguistic study. *Language and Cognitive Processes, 1*, 249-262.

De Smedt, Koen (1990). *Incremental sentence generation: Representational and computational aspects.* Ph.D. thesis, University of Nijmegen.

Frazier, Lyn (1987). Theories of sentence processing. In: Jay L. Garfield (Ed.), *Modularity in knowledge representation and natural language understanding.* Cambridge, MA: M.I.T. Press.

Gibson, Edward (1990a). Memory capacity and sentence processing. In: *Proceedings of the 28th Annual Meeting of the ACL*, Pittsburgh.

Gibson, Edward (1990b). Recency preferences and garden-path effects. In: *Proceedings of the 12th Annual Conference of the Cognitive Science Society*, Cambridge MA.

Gibson, Edward (1990c). A computational theory of processing overload and garden-path effects. *Proceedings of the 13th International Conference on Computational Linguistics* (COLING-90), Helsinki.

Joshi, Aravind K. (1990). Processing crossed and nested dependencies: an automaton perspective on the psycholinguistic results.. *Language and Cognitive Processes, 5(1)*, 249-262.

Kempen, Gerard (1987). A framework for incremental syntactic tree formation. In: *Proceedings of the 10th International Joint Conference on Artificial Intelligence* (IJCAI-87), Milan, 655-660.

Kempen, Gerard & Theo Vosse (1990). Incremental syntactic tree formation in human sentence processing: an interactive architecture based on activation decay and simulated annealing. *Connection Science, 1*, 273-290.

McRoy, Susan & Graeme Hirst. (1990). Race-based parsing and syntactic disambiguation. *Cognitive Science, 14*, 313-353.

Rayner, Keith & Alexander Pollatsek (1989). *The psychology of reading.* Englewood Cliffs, NJ: Prentice-Hall.

Sampson, Geoffrey (1986). A stochastic approach to parsing. In: *Proceedings of the 11th International Conference on Computational Linguistics* (COLING-86), Bonn.