Cogn Neurodyn (2009) 3:331–346
DOI 10.1007/s11571-009-9094-0

RESEARCH ARTICLE

The Unification Space implemented as a localist neural net: predictions and error-tolerance in a constraint-based parser

Theo Vosse · Gerard Kempen

Received: 28 May 2009 / Revised: 27 August 2009 / Accepted: 31 August 2009 / Published online: 26 September 2009

Abstract We introduce a novel computer implementation of the Unification-Space parser (Vosse and Kempen in Cognition 75:105–143, 2000) in the form of a localist neural network whose dynamics is based on interactive activation and inhibition. The wiring of the network is determined by Performance Grammar (Kempen and Harbusch in Verb constructions in German and Dutch. Benjamins, Amsterdam, 2003), a lexicalist formalism with feature unification as binding operation. While the network is processing input word strings incrementally, the evolving shape of parse trees is represented in the form of changing patterns of activation in nodes that code for syntactic properties of words and phrases, and for the grammatical functions they fulfill. The system is capable, at least qualitatively and rudimentarily, of simulating several important dynamic aspects of human syntactic parsing, including garden-path phenomena and reanalysis, effects of complexity (various types of clause embeddings), fault-tolerance in case of unification failures and unknown words, and predictive parsing (expectation-based analysis, surprisal effects). English is the target language of the parser described.

Keywords Predictive parsing · Syntactic ambiguity resolution · Psycholinguistics · Unification Space · Localist neural network

T. Vosse
Donders Institute for Brain, Cognition and Behaviour, Nijmegen, The Netherlands
e-mail: TheoVosse@yahoo.com

T. Vosse · G. Kempen
Cognitive Psychology Unit, Leiden University, Leiden, The Netherlands

G. Kempen (✉)
Max Planck Institute for Psycholinguistics, Nijmegen, The Netherlands
e-mail: Gerard.Kempen@MPI.nl

Introduction

The Unification-Space model of parsing that we developed 10 years ago (henceforth U-Space2000; Vosse and Kempen 2000) is a dynamic model of syntactic parsing based on activation and inhibitory competition. Its dynamics enable it to simulate a considerable range of psycholinguistic phenomena related to the syntactic aspects of human sentence comprehension. In the decade that passed since its publication, two important developments took place that incited us to design an entirely new implementation. First, contrary to what we expected at the time of developing U-Space2000, *predictive* syntactic parsing has convincingly been shown to be an important component of human sentence comprehension (Kamide and Mitchell 1999; Konieczny 2000; Hale 2003; Van Berkum et al. 2005; Pickering and Garrod 2007; Levy 2008). Second, thanks to the ascent of novel neurophysiological research methods, there is a rapidly growing need for neural-network models of human cognition, including human syntactic parsing. Hence, we decided to embark on a project aiming at extending U-Space2000 with facilities for predictive parsing, and to re-implement it as a localist neural network. We expected, in addition, that this connectionist approach would bring nearer a parsing mechanism with the highly desirable but elusive property of "graceful degradation." In particular, we aimed at a parser that, when confronted with

an unknown word or an incorrectly inflected known word, does not halt immediately but attempts to guess the correct grammatical properties of that word. To our knowledge, no symbolic or connectionist model embodying these properties of human syntactic parsing has been implemented. (For pointers to recent literature on neurocomputational models of human sentence processing, we refer to van der Velde and de Kamps (2006), beim Graben et al. (2008), Mayberry et al. (2009), Huyck (2009); and several other articles in this special issue of Cognitive Neurodynamics.)

How to represent, in a neural net with a fixed pattern of connections, not only the declarative lexical and syntactic knowledge underlying linguistic competence but also the syntactic structures that are assembled online in the course of the parsing process? Since we wanted the network to compute (some equivalent of) real parse trees, we could not start out from familiar neural network architectures published in the literature. These typically aim at performing more limited linguistic tasks, e.g., predicting the next word of a sentence. Simple Recurrent Nets (SRNs), in particular, have often been shown capable of learning such tasks, even if the training set consists of sentences generated by a relatively complex grammar (Elman 1990, 1991). It is far from certain, though, that this type of performance requires an internal representation of sentence structure (van der Velde et al. 2004).

Therefore, we decided to start from scratch. Given that the U-Space2000 model centered around the notions of activation and competition, it should not come as a surprise that we ended up with a model that has some similarities with Interactive Activation and Competition (IAC) models which originated in the 1980s, for instance the TRACE model of auditory word recognition (McClelland and Elman 1986). The resulting system we dubbed SINUS, where US stands for Unification Space, N for Neural Network, I for IAC, and S for Simple or Semi (to bring out that it is anything but finished).

We start with an outline of the grammar formalism underlying the parsing system: Performance Grammar (PG; Harbusch and Kempen 2002; Kempen and Harbusch 2003). PG defines the structures to be formed during parsing. Then, we sketch the dynamic aspects of the parsing process, which we adopt from U-Space2000: Activation spreading and competitive inhibition (Vosse and Kempen 2000, 2009). The pièce de résistance of the paper is the description of the neural re-implementation: the wiring and the activation flow in the SINUS network, and the training/optimization procedure used to set the 30+ free parameters. Finally, we show that SINUS achieves the desired psycholinguistic effects, at least in a qualitative and rudimentary fashion, and conclude with some evaluative remarks and suggestions for improvements and extensions.

Structural assumptions: essentials of Performance Grammar

Performance Grammar is a "lexicalized" grammar: It assumes that the information needed to build grammatically correct sentences is associated with the individual lexical items. Syntactic trees result from the collaboration between two processing components—one dealing with the hierarchical structure of a tree, the other one with the linear order of the branches. The hierarchical component retrieves so-called *lexical frames* from the Mental Lexicon (see Fig. 1 for two examples) and links them together by "binding" the root node of one frame to a foot node of another frame. The two nodes being bound should carry the same phrase label. The binding operation underlying sentence (1) yields the tree in Fig. 2.

(1) *Money counts*

Associated with every root and foot node is a feature matrix, and every feature consists of an *attribute* (printed in capital letters) and a value. The value of an attribute is a non-empty disjunctive set of options. In Fig. 1, for example, the CASE feature of *money* has three possible options: nominative, dative and accusative. This means it can be used in syntactic environments that require either of these alternatives. In this paper, where the emphasis is on syntactic parsing, we distinguish *morphosyntactic* and *linear-order* features. A root node can bind to a foot node only if all of their features *unify*. We define unification here as a nonrecursive operation on features and feature matrices. Informally, a feature f_1 in feature matrix m_1 of some node unifies with feature f_2 in the feature matrix m_2 of another

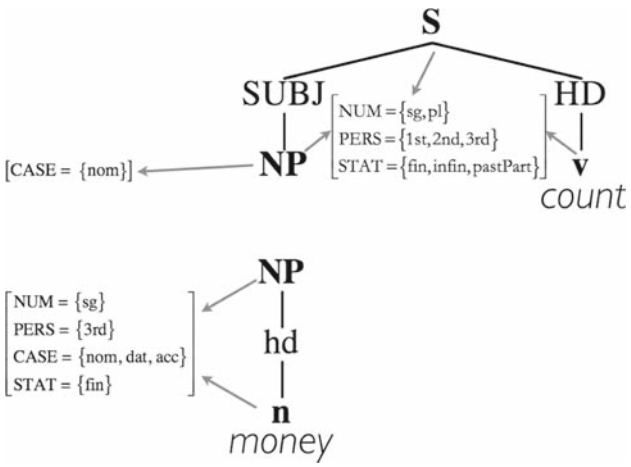

Fig. 1 Lexical frames for the words of sentence (1). Both frames are simplified here: Only branches whose terminal leaves are involved in a binding operation or carry a lexical label, are shown. Some less frequently used abbreviations: HD = Head; STAT = status of a clause/verb: finite, infinitival or participial (present or past). The TENSe feature of the verb is left out

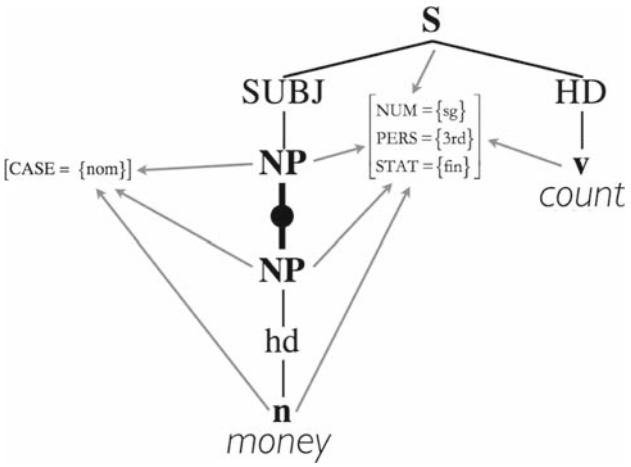

Fig. 2 Syntactic tree for example (1) after unification of the Subject footnote of *count* with the root NP node of *money*

node either if the attributes of f_1 and f_2 are different, or else (that is, they have the same attribute) if the intersection of the values of f_1 and f_2 is non-empty. If and only if all features of m_1 unify with all features of m_2, unification succeeds. Successful unification delivers a feature matrix that contains the union of the features in m_1 and m_2 (one exemplar of each feature); and the intersection computed as value of features with the same attribute counts as the value of that attribute in the resulting matrix. In Fig. 2, successful unification is indicated by the black dot in the thick line connecting the unified nodes. Note that the unified nodes do not merge/fuse; they only take the same feature matrix. If unification fails, the feature matrices remain unchanged. Comparison of Figs. 1 and 2 shows that unification of the two lexical frames yields a single value option for the CASE attribute of the root and foot nodes of *money*, and a single option for the NUMber, PERSon and STATus attributes of *counts*. The example also illustrates how unification determines Subject-Verb agreement.

PG's *linear* component works with so-called topologies (or topological fields). Paired with each lexical frame is a topology, that is, a one-dimensional array of slots that serve as placeholders for sentence constituents. In the current PG version for English, every lexical frame headed by a verb has a "clausal topology" with nine slots, as depicted in Fig. 3. Placement rules like those in Table 1 allocate a position in one of the slots to individual constituents; if a constituent has more than one placement option, the left-most one is preferred. Because every verb is treated as the

Table 1 Examples of topology slot fillers for English clauses

Slot	Filler
F1	*In declarative main clause*: Topic, Focus (one constituent only)
	In interrogative main clause: Wh-constituent
	In complement clause: Wh-constituent (including Complementizer *whether/if*)
F2	*In complement clause*: Complementizer *that*
F3	Subject (iff non-Wh)
M1	Pre-Infinitival *to* < Head verb (obligatory) < Verb particle
M2	*In interrogative main clause*: Subject (iff non-Wh)
	Direct Object (iff personal or reflexive pronoun); Subject < Direct Object
M3	Indirect Object < Direct Object (both non-Wh; and Direct Object not a personal/reflexive pronoun)
M4	Verb particle
E1	Non-finite Complement clause of Auxiliary verbs and other "Verb Raisers"
E2	Finite or non-finite Complement clause of 'VP Extraposition' verb
	Finite Complement Clause

Modifier constituents are not shown. Precedence between constituents landing in the same slot is marked by "<" (N. B. The simulations to be reported below use a somewhat simplified version of these placement rules. In particular, since the complementizer *that* is missing from the English grammar underlying the current version of SINUS, there is no F2 slot, and F3 is renamed F2.)

Head of a finite or nonfinite clause, and every clause has its own topology, it follows that example sentence (2) activates four clausal topologies, as depicted in Fig. 4. The resulting stack of topologies is "read out" from top to bottom and from left to right, in a depth-first manner.

(2) *Who would wish to deny that money counts?*

Importantly, linear order rules are applied at the same time as, and in conjunction with, binding (unification) decisions. Whenever a binding decision is not accompanied by a licit slot assignment, the derivation of the sentence fails. Furthermore, PG does not have movement rules: Slot positions are assigned "once and for all." For example, during the derivation of a sentence like *What did you say?*, there is not an intermediate stage where the Direct Object *what* occupies a position rightward of the Head verb *say*, followed during a later stage by a "fronting" operation which moves *what* to its definitive position at the beginning of the sentence. (For detailed movement-free PG

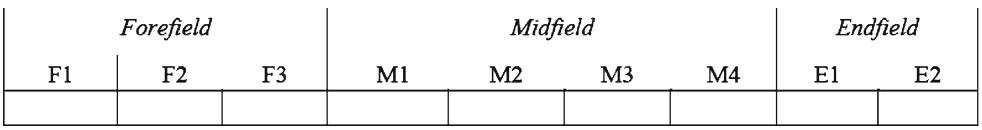

Fig. 3 Topology for English clauses. The names of the three fields are our translations of the original German names Vorfeld, Mittelfeld, and Nachfeld

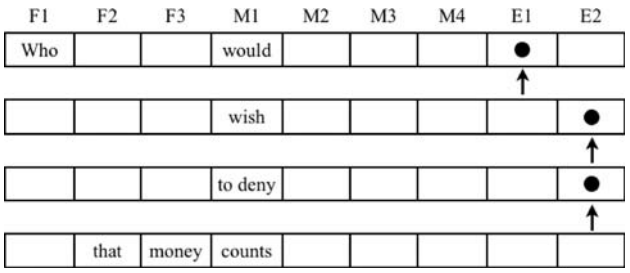

Fig. 4 Application of the placement rules in Table 1 to sentence (2). The *black dots* and the *arrows* pointing to them express the placement of embedded clauses: The complete content of the topology at the tail of an arrow is entered into the slot containing the dot. *Wish* and *deny* are treated as "VP Extraposition verbs" in the terminology of Generative Grammar

treatments of grammatical phenomena that often have been analyzed in terms of movement, see Kempen and Harbusch 2003 and Kempen 2009). We will not dwell on this topic here because only relatively simple cases have been implemented in the present SINUS version.

Dynamic assumptions: U-Space2000

A simple two-word sentence like (1) only allows one possible attachment between the lexical frames, provided the words are unambiguous. However, when the number of words increases, the number of syntactically allowed attachments (unifications) increases rapidly. Figure 5 provides an example with a relatively simple seven-word sentence. The first tree shows the lexical frames of the sentence and the unifications defining the correct parse tree. However, if we leave semantic and pragmatic constraints out of consideration, at least one additional attachment is allowed by the grammar: The PP *with a spot* might also be analyzed as a Modifier of the verb. Moreover, because we aim for a robust parser that degrades gracefully, it should always be on the alert for ill-formed input, in particular lexical, morphological, and word order errors. The middle tree shows additional attachment options when word order and morphology are left out of consideration. In order to find, within the total set of possible attachment patterns, a small set of syntactically plausible alternatives—preferably just one or two—, U-Space2000 stages competitions between mutually exclusive attachments (e.g., between the "low" and "high" attachment of PP *with the cat*). The outcome of these competitions is determined by an interplay of *dynamic* factors based on activation spreading and lateral inhibition—mechanisms that are frequently used in neurocognitive modeling. The third tree in Fig. 5 shows (most of) the inhibitory links that U-Space2000 creates internally in the course of parsing the sentence.

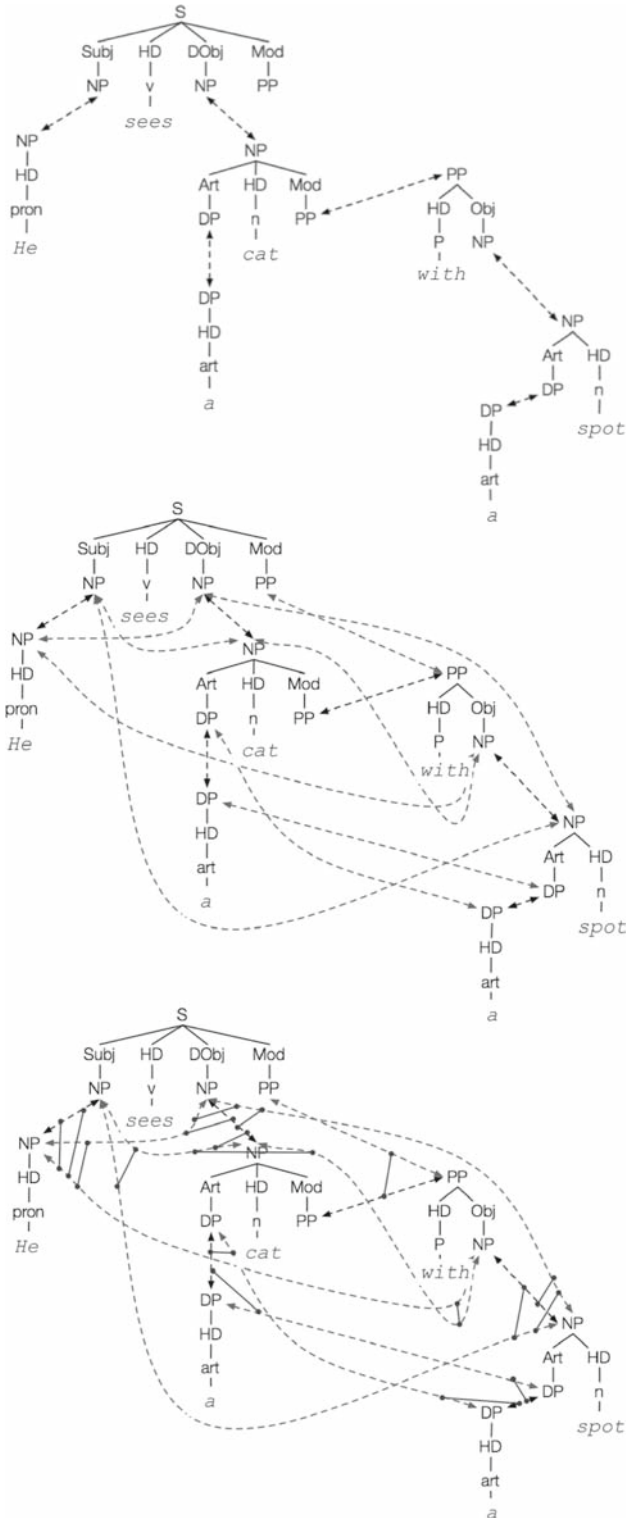

Fig. 5 *Above* PG lexical frames (*labeled nodes with continuous lines*) and ultimately correct unifications (*straight dashed arrows*) for a simple sentence. *Middle* alternative unification options. *Below* competitive inhibition between alternative unification options (*continuous lines* ending in *black dots*)

SINUS is built on similar dynamic principles which, as shown by U-Space2000, enable psycholinguistically plausible simulations of ambiguity resolution, garden-path phenomena, reanalysis, and effects of word (form) frequency and complexity. However, SINUS is not just a neural-net re-implementation of U-Space2000. It aims to achieve, at least in rudimentary form, two feats seldom seen in computational models of human sentence comprehension: predictive parsing, and graceful degradation (error-tolerance).

Connectivity

The SINUS network consists of a one-dimensional array of "columns." Every node in a column codes for a grammatical property of an incoming word (*localist representation*). These properties are either adduced from the lexicon or assigned during the parsing process. Among the former are morphosyntactic features (e.g., a node representing "Number = Singular," or a node coding for the fact that a personal pronoun projects—i.e., is the Head of—an NP); the latter include grammatical functions (e.g., the NP headed by an input noun receiving the role of Subject) and linear positions in a phrase or clause (e.g., a Subject NP selecting topology slot F3 as its destination). One column represents one word, and the columns are filled one by one, from left to right, according to word order in the input string. In its present version, SINUS has 12 columns. Because one column serves as receptacle for syntactic predictions concerning future input, SINUS can process sentences up to 11 words long. However, in the section "Removing the upper bound on sentence length," we describe an algorithm that removes this limitation. Intercolumnar connections enable the representation of grammatical relation between the words of a sentence.

With a few exceptions, all nodes in the columns function in the same manner. Every node has an *activation* level between zero and unity, set initially to zero. If the activation level of a node surpasses a difference threshold (i.e., exceeds the activation levels of competing nodes by more than a minimum value, the threshold), then the grammatical property represented by that node forms part of the current analysis of the input sentence. Activation spreads to other nodes via a pattern of connections that is derived from the PG grammar. This activation is added to the activation level of the node at the receiving end of an *excitatory* connection, but it is subtracted from that level if the connection is *inhibitory* (the equations describing these effects are introduced in the section "Computational aspects."). Whether a connection is excitatory or inhibitory depends on whether the nodes at either end belong to the same or to different *layers*. Every column is divided into six layers, each containing nodes that code for one type of grammatical information (word category, lexical frame, unification node, topology slot, etc.). Connections between nodes in the same layer are all inhibitory; this holds for intracolumnar as well as intercolumnar connections. Connections between nodes in different layers are all excitatory, whether upward or downward, and they only connect nodes in adjacent layers (but see footnote 2 for one exception to the latter).

The bank of interconnected word columns is the "Unification Space" where syntactic structures are built in the form of activation patterns. Upon initialization, the system is "empty" in the sense that all nodes have zero activation. The properties of input lexical items are adduced from a separate store, the Mental Lexicon. Via pathways from each lexical item to the bank of columns, every word recognized in the input can, in principle, reach every column: A special circuitry allocates every input words to the leftmost empty column in their order of arrival.[1] The lowest layer of a column is the first to receive activation from the word it represents, whereafter this activation spreads upward via "feedforward" links to the next higher layer, reverberating again via "feedback" links to the next lower layer.

We now describe the layers of a word column from bottom to top (see Fig. 6). Describing one word column suffices because all columns embody the same network.

The *Input* layer functions as the intermediary between Mental Lexicon and Unification Space. It represents the identity of "its" input word and passes the morphosyntactic properties of this word on to the next higher layer.

The *Word Category* layer represents the Head of the lexical frame associated with the input word (e.g., the noun of an NP lexical frame, the preposition of a PP lexical frame, the verb of a clausal lexical frame). It also contains nodes that code for the morphosyntactic properties of the input word (number, person, gender, case, etc.). In case of lexical ambiguity, several word category nodes and their associated features can be active simultaneously. (Due to competition via inhibitory connections, one of the categories will ultimately gain the upper hand.)

The word category and the morphosyntactic features are activated bottom-up by a short activation pulse. However, this pulse cannot keep the nodes in the Word Category layer alive for a long time. In fact, the network critically relies on excitatory feedback from the Lexical Frame layer: Category and feature nodes receive feedback from the phrasal nodes whose head they are (nouns and pronouns

[1] Recent neurophysiological evidence (e.g., Snijders et al. 2009) suggests that the Mental Lexicon is subserved by the Left Posterior Middle Temporal Gyrus, and the Unification Space by the Left Inferior Frontal Gyrus.

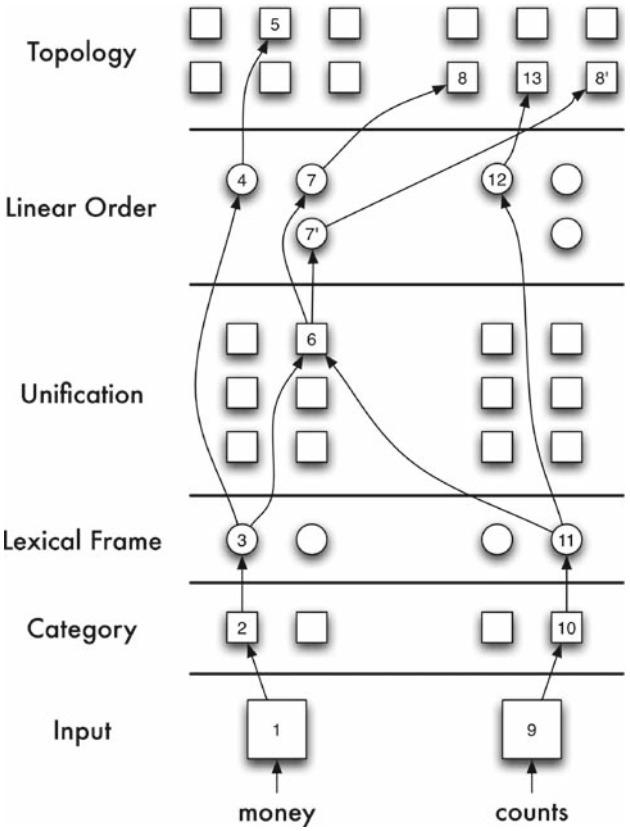

Fig. 6 Activation spreading through the SINUS network for the noun *money* and the verb *counts* in example (1)

from NP, verbs from S, adjectives and adverbs from AP, etc.; note that, in PG, every word, even the particle of a particle verb, is head of a phrase). Feedback may depend on combinations of activated features. For example, the word category Pronoun should not get top-down feedback from an NP phrasal node when the latter was activated bottom-up by an input Noun.

The *Lexical Frame* layer codes for the syntactic information in the lexical frame(s) associated with the input word, together with the feature matrices (cf. Fig. 1). Multiple frames may be active at the same time. Importantly, word columns do not systematically set apart syntactic elements belonging to different lexical frames (see the paragraph on lexical frame nodes in the next section for more details on the treatment of lexical ambiguity).

The *Unification* layer contains so-called Unification nodes (U-nodes). A U-node carries the name of a grammatical function (Subject, Direct Object, Modifier, etc.). It is connected to a root node in its "own" column and a foot node in another column, and its label specifies the grammatical function that the root node fulfills in the lexical frame to which the foot node belongs. For instance, node #6 in Fig. 6 codes for the Subject function of *money* in sentence (1)—more precisely, for the unification of the root node of NP *money* with the foot NP-node of the Subject of *counts* (cf. also the black circle in the thick line of Fig. 2). U-nodes receive bottom-up activation from the unification "partners" (i.e., the root and the foot node they bind) and from feature nodes. An example of the latter: The activation level of node #6 in Fig. 6 depends partly on the activation levels of the number and person features involved.

In example (1), the input word *money* leads to activation of several U-nodes—not only the U-node representing the option that the input word is going to play the role of Subject in the upcoming sentence, but also the U-node for Direct Object, Indirect Object, and Prepositional Object. Since these roles are incompatible, they compete by inhibiting each other. Differential feedback from the Linear Order layer (see below), may cause the activation levels of these U-nodes to diverge. For instance, in the current SINUS version, Subject U-nodes receive a higher amount of initial activation. Due to this head start, the first NP of a clause will often seize the Subject role, especially when helped by an ensuing finite verb.

Figure 7 shows that unification partners need not be situated in adjacent columns. However, a foot node belonging to a remote column exerts less influence on unification strength than a foot node in a nearby column. (This effect is achieved by weights on intercolumnar connections that decrease with increasing distance.) Importantly, a column includes several exemplars of every U-node: one exemplar for every other column. In the three-column system depicted in Fig. 7, every column contains two exemplars of the Subject U-node; the twelve-column system of SINUS includes eleven tokens of every U-node. In our descriptions of what happens in a word column, we will skip these details.

A U-node may link to several different types of root nodes. For example, Modifier nodes of S-frames (clauses) receive excitation from PP-nodes, AP-nodes, or other S-nodes (adverbial clauses).

The *Linear Order* layer contains nodes representing a licit "destination" of a U-node; more precisely, they try to assign one topology slot to one U-node. These nodes receive bottom-up activation from the U-nodes for which they seek a destination. The amount of bottom-up activation is tempered when not all placement conditions are met. For instance, as specified in Table 1, the Direct Object is allowed to land in slot M3 of a clausal topology only if it is neither a Wh-constituent nor a personal or reflexive pronoun.

The PG grammar (the rules in Table 1 in particular) may specify two or more placement options for a U-node. For instance, the Subject can go to slot F3 or to slot M2 of a clausal topology. In Fig. 6, these options are represented by nodes #7 and #7'. Linear Order nodes coding for competing options inhibit each other. Competition also arises

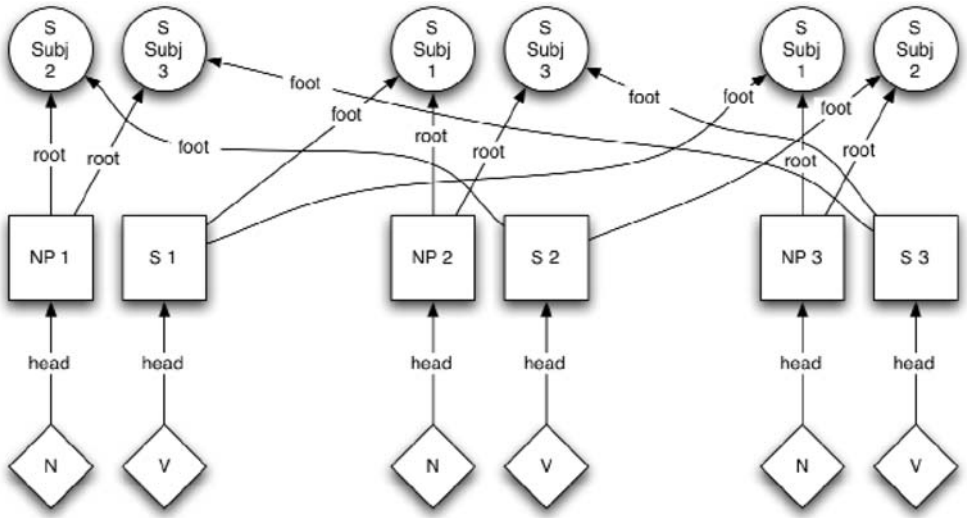

Fig. 7 Example of intercolumnar connectivity of Unification nodes. U-nodes (*circles*) in three word columns linked to, and receiving activation from, phrasal nodes (*squares*) in two other columns. The label "Subj" stands for "Subject Unification node." Note that each exemplar of a U-node connects to only one column, and that different columns connect to different U-node exemplars

between two or more Linear Order nodes that attempt to assign the same topology position to different U-nodes. The higher the current activation level of a Linear Order node, the more secure the placement option it tries to realize for its U-node.

The *Topology* layer consists of nodes that represent a position in a topology (i.e., a slot or a relative position within a slot). The activation level of a Topology node indicates the strength of an association between a U-node and a position in a topology. One of the factors influencing these levels are the competitions raging in the Linear Order layer. Nodes in the Topology layer do not compete with one another. Instead, nodes coding for early positions transmit activation to certain nodes further downstream in the same topology, in a manner that causes positions to be filled in a temporal order that roughly corresponds to their spatial (left-to-right) order in the topology. This transmission takes place indirectly, via nodes in the Linear Order layer, as will be explained in the next section.

The Topology layer specifies topologies for all types of lexical frames in the grammar, not only for the clausal one associated with S-nodes. For instance, it includes an NP topology consisting of four slots (for Determiner Phrase, prenominal Modifier, Head, and postnominal Modifier. The upmost row of nodes in the Topology layer of Fig. 6 codes for the three leftmost NP positions (hence, node #5 is the destination of the Head (pro)noun of an NP. The second row of nodes codes for three of the nine slots of clausal topologies. (The topologies for other phrase types, which are very small, are not shown in Fig. 6.)

Finally, outside of the word columns and their layers there is a "global" node, called the *Apex*, which serves as a "coat hanger" for the syntactic structure of the entire input sentence. It has a constant (nondecaying) activation and connects to the S-nodes in the various columns. These S-nodes enter into a competition to become the "unification partner" of the Apex, and the winner ultimately dominates the entire parse tree. The activation of the Apex node has the side-effect of "attracting" S-nodes: It slightly reduces the probability for the latter to attach at lower positions, e.g., as a relative or complement clause.

Flow of activation and inhibition

SINUS spends 20 processing cycles on every new input word. During a cycle, for every node in every column, the activation level of that node is updated by adding the activation transmitted to it from connected nodes in higher and lower adjacent layers, and by subtracting inhibition and decay (for details, see the section "Computational aspects"). The intercolumnar connections enable SINUS to pre-activate (or pre-inhibit) certain nodes in columns rightward of the last filled column. Then, when these columns are filled, the patterns of activation running there influence the way the input is processed. For instance, they can bias the analysis of a word-class ambiguous word towards the reading whose activation pattern conforms best to the pre-activated pattern. In the current SINUS version, pre-activation is restricted to a single column—the one immediately following the one that was filled last. Pre-activation endows SINUS with a rudimentary form of predictive parsing.

Figure 6 shows SINUS at work for example (1). The arrows show feedforward activation between nodes that turn out to win the competitions in the various layers (except for #7' and #8'; see below).

Node #1 in the Input layer is activated first. During subsequent cycles, activation spreads to node #2 (Noun) in the Word Category layer, which in turn activates node #3 (NP) in the Lexical Frame layer. From here, activations spreads in two directions, in parallel: to node #4 in the

Linear Order layer,[2] which selects a slot position for *money* in the NP-topology (node #5); and to node #6, which codes for the Subject role in a clause. The activation level of node #6 does not reach its maximum level yet since it only has support from a root node: No foot node that might serve as unification partner is on the horizon yet. Hence, up to this point there is partial support for the Subject role of *money*. Node #6 partially pre-activates nodes #7 and #7'in the Linear Order layer, which select topology slots F3 and M2, respectively. Nodes #7 and #7' compete with one another, and #7 emerges as the winner due to feedback from node #8 (this feedback activation route is not depicted in Fig. 6). Since topology node #8 codes for slot F3, the NP *money* may be said to land in Forefield slot F3.[3]

This raises the question why Linear Order slot #7' does not receive feedback from topology slot #8', thereby being doomed to lose the competition with node #7. The answer is that node #8' has to stay dormant until after topology node #13 has been woken up. Node #13 codes for slot M1, the landing site for verbs. So let us see what happens when the verb *counts* is input into column 2 (node #9). From here, activation propagates to node #10 (Verb), node #11 (S) and node #6, which was already partially active. Due to this added activation, node #6 now reaches its maximum activation level. This means there is sufficient support for the Subject role of NP *money*. The activation boost that the verb brings about to node #6, propagates to node #7 and indirectly to node #8 (slot F3).

In parallel, activation propagates from node #11 to Node #12, which consigns the verb to topology slot M1 (node #13). At that moment, SINUS may be said to have found a parse: a clause headed by *counts* in clausal topology slot M1, with *money* heading the Subject NP and placed in topology slot F3. SINUS can visualize the current parse tree by running a "tree extraction" procedure (which has no influence on the course of events in the columns; see next section for details).

Node #13 sends feedback to Linear Order nodes that code for downstream topology slot positions in the Midfield (M2 through M4; not shown in Fig. 6). One of them is node #7'. This activation could have worked as a trigger that raises the activation of node #7' to a level above the threshold. However, inhibition from competitor node #7 is too strong. As a consequence, NP *money* stays in Forefield slot F3 (node #8) where it was consigned by Linear Order node #7, and does not end up at M2 (represented by node #8').

Many lexical items have more than one set of syntactic properties. This raises the question how lexical ambiguity influences the course of events in a column. As already mentioned in the previous section, SINUS does not distinguish differing lexical frames associated with the current input word: Nodes in the Lexical Frame layer do not code for the lexical frame from which they received their activation. For instance, the verb *counts* can be intransitive—as in sentence (1)—or transitive (the lexical frame for the transitive *counts* includes a Direct Object branch that is missing from Fig. 1). In U-space columns, the distinction between the transitive and intransitive S-frames is lost: The Input, Word Category, and Lexical Frame layers of a column contain only one token of a node coding for a given type of (morpho-)syntactic element. Only after the parsing process has come to a halt, can one observe whether *counts* was used transitively or intransitively—by checking whether the resulting parse tree contains a Direct Object branch. An example of a word category ambiguity is the word *her*, which in SINUS is encoded both as a personal pronoun (heading an NP) and as a possessive pronoun (heading a Determinar Phrase). When seeing this word, SINUS honors this distinction by activating both an NP and a DP lexical frame (which henceforth inhibit one another).

Computational aspects

In this section, we present the essentials of the equations that specify the flow of activation through the SINUS network, and we sketch the procedure for extracting syntactic trees from the current activation pattern running in the bank of columns.

Activation spreading

The activation level of SINUS nodes can be modified in two different ways: *node updating* and *activation copying*. Nodes are updated quasi-simultaneously in discrete time slices. Basically the same update function is applied to all nodes (see also Fig. 8):

$$a_i(t+1) = H(\alpha_l m_l(f_{\text{forw}}(t)) + \beta_l a_i(t) - \gamma_l f_{\text{inh}}(t) + \delta_l f_{\text{back}}(t))$$

That is, the activation level of node i in layer l at time $t + 1$ is the sum of four terms at time t:

[2] This feedforward acivation from nodes that code for lexical frames, to Linear Order nodes coding for the position of the lexical Heads of these frames skips the Unification layer. This is the only exception to the rule that feedforward and feedback passes only through adjacent layers. See also the activation from node #11 to node #12 in the second column of Fig. 6.

[3] In the meantime, input node #1 has ceased to be active. However, this does not mean that the other nodes cannot stay alive: Since feedback activation flows down, followed by another wave of feedforward activation, the nodes sustain each other's activation.

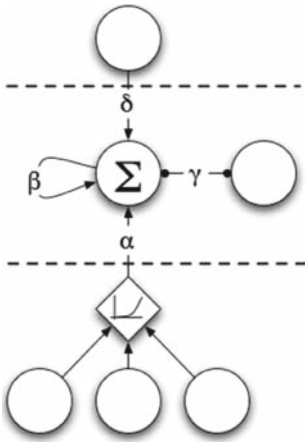

Fig. 8 General structure of a SINUS node

- the summed feedforward activation (f_{forw}) adduced from the adjacent lower layer multiplied by parameter α_l,
- plus the current excitation level (a_i) multiplied by decay parameter β_l,
- minus the summed inhibition (inhibitory activation, f_{inh}) adduced from within the current layer multiplied by parameter γ_l,
- plus the summed feedback (f_{back}) adduced from the adjacent higher layer, multiplied by parameter δ_l.

The feedforward activation is scaled by the following equation:

$$m_l(x) = (1 - e^{\lambda_l x}) \Big/ (1 - e^{\lambda_l \max(x)})$$

where $\max(x)$ is the maximum value x can reach, and λ_l a free parameter. This equation plays a role only in the three highest layers. $H(x)$ corrects outlying values of x to zero or unity: if $x < 0$ then $H(x) = 0$; if $x > 1$, then $H(x) = 1$. The value of parameters with subscript l are layer-dependent.

The second mode of spreading activation is simpler. It consists of copying the current level of a node to an identically labeled node in another column. Activation copying is part of the mechanism enabling SINUS to process sentences that consist of more words than the number of columns (explained in the next section).

Tree extraction

The total pattern of activation running in the bank of columns defines the grammatical relationships between the words of the sentence, that is, the parse tree. A parse tree can be extracted from the activation state of the network at the end of a processing cycle. The extraction procedure presupposes, as implied by the standard intercolumnar connections, that every root node in a lexical frame is connected, via a U-node, to all foot nodes in other columns that carry the same name (so that they could unify) and, that every foot node of a lexical frame is connected, via a U-node, to all its namesakes in other columns. Every U-node codes for the grammatical function that one lexical frame fulfills in the other frame. Given an input sentence, only a few U-nodes will reach an above-zero activation level. Tree extraction proceeds as follows:

1. Select the most active lexical frame in each filled column; more precisely, the most active root node (in example (1): an NP-node in column 1 and an S-node in column 2).
2. For every foot node in these lexical frames, select the most highly activated U-node that binds it to a root node in another column (in the example: the Subject U-node in column 2, which unifies the Subject NP of *counts* with the root NP of *money*).
3. If the activation level of this U-node surpasses the threshold value (which can be set by the user of the SINUS program), look up the topology slot where this U-node has landed.
4. For each thus selected U-node in each column, merge the two unification partners into a single tree node. Draw the resulting tree nodes and the names of their U-nodes from left to right in accordance with their positions in the topologies; the S-node that unifies with the Apex becomes the root node of the complete tree.

If a filled column has no U-node surpassing the activation threshold, it forms the root of a syntactic fragment covering only part of the input sentence. The analysis of a complete grammatical sentence should consist of precisely one fragment that should not include any cyclical attachments, and whose root is unified with the Apex. (An example of cyclical attachment: An NP is functioning as Subject of a clausal lexical frame—i.e., of a frame rooting in an S-node—and, at the same time, this S-node is attached as a relative clause within the NP.)

The procedure for extracting syntactic trees from an activation pattern running in SINUS is crude. It cannot fully represent every state of the network, which may embody a cyclical structure or a tree that does not belong to the set of trees generated by the PG grammar. An exotic example of the latter: a Determiner Phrase that (temporarily) seems to be the Subject of a Prepositional Phrase. Therefore, the quality of SINUS as a *syntactic* parser should be evaluated in the light of the final structures yielded at the end of processing complete or incomplete sentences.

Nonetheless, we stress that the syntactic structures assembled by SINUS in the course of parsing a sentence and delivered at end-of-sentence are merely auxiliary structures that contribute to reaching the ultimate goal of sentence comprehension: reconstructing the communicative intention

of the speaker/writer. We assume that this reconstruction process unfolds incrementally, running in parallel with syntactic parsing. Correct syntactic trees are not the final goal of the comprehension process. But, given that SINUS currently is not running in the context of a complete sentence comprehension system, we have no choice but to use (activation patterns underlying) parse trees as evaluation criterion.

Removing the upper bound on sentence length

The present version of SINUS contains $n = 12$ columns. In order to enable it to parse input strings containing more words than fit into the Unification Space, we implemented a mechanism that enables the activation patterns in a pair of columns to be "compressed" into one member of the pair, thereby creating space for an extra input word. We now describe this compression mechanism, using the five-word sentence in (3) as an illustration and presupposing (for explanatory purposes) that SINUS contains only $n = 4$ columns.

(3) *Little Johnny counts his money*

The mechanism needs a memory cache where part of the activation pattern running in the to-be-vacated column can be stored. To this purpose, SINUS utilizes a set of auxiliary columns, which we call *sidecolumns* in order to distinguish them from the *main columns* discussed so far. The bank of n main columns is paralleled by n sidecolumns, which are much simpler than main columns. More specifically, they consist of only two layers: a Unification and a Linear Order layer. The Unification layer contains one U-node for every grammatical function; this node is connected to one or more nodes in the Linear Order layer. Nodes in sidecolumns only inhibit incompatible U-nodes in main columns, and they are able to copy their activation level into their namesakes in the sidecolumn of the immediately preceding column. The activation levels of sidecolumn nodes are kept constant: They are not subject to decay and receive no feedforward or feedback activation, and no inhibition. As SINUS always keeps the column to the immediate right of the last filled column free for pre-activation (see the section "Predictive parsing"), the compressing mechanism is launched immediately after the processing cycles for the word in column n-1 have been completed (in the example of sentence (3): after *counts* has been processed). It basically comprises the following steps:

1. Select the leftmost main column that, in view of its current pattern of attachments to nodes in other columns, is a "suitable" candidate to be vacated. A column is suitable if it codes for an input word heading a lexical frame whose root node does not dominate any other word. In (3), the adjective *little* is a suitable candidate for vacation: It is the head of an AP whose other branches are currently empty (*little* being the only member of the Adjectival Phrase). As no other suitable candidates are available, the *little* column is chosen as the first one to be vacated.
2. The activation levels of the U-nodes and the Linear Order nodes in the to-be-vacated main column are *copied* into their namesakes in the sidecolumn of their unification partner (here, the sidecolumn of *Johnny*). (NB The to-be-vacated column need not be adjacent to the target column; and the latter need not be further downstream than the former.)
3. The activation patterns running in all main columns and sidecolumns to the right of the to-be-vacated column are copied into their namesakes within the left-hand neighbor.

The crucial effect achieved by the compression operation is to enable the "essential content" of an erased column to keep exerting its influence, in particular its inhibitory influence on competitors for U-nodes and topology slots. In (3), SINUS' first column now codes for the noun *Johnny*, and the Modifier U-node in its sidecolumn represents the fact that the NP headed by *Johnny* includes a Modifier. In order to enable SINUS to "remember" this information, it keeps activation levels in sidecolumns constant throughout the parsing process: no decay, no updating.

To continue example (3), after the first compression operation, *his* is entered into main column 3. The lexical frame associated with this word is a Determiner Phrase (DP), which cannot be strongly attached to the current tree but weakly activates the Determiner U-node in its column. Compression now looks for a suitable column to vacate and selects column 1: The Subject NP represented there does not dominate a subconstituent in any other column. Hence, *counts* is copied into main column 1, and its sidecolumn now codes for the fact that SINUS has already consumed a Subject. This information—or rather the inhibition emitted by the Subject U-node in the *counts* sidecolumn—serves to ward off other constituents that aspire to this grammatical role. After copying the activation patterns in the *counts* and *his* columns into their predecessors, the word *money* enters column 3 and adopts the function of Direct Object. If the sentence would not have finished here, the next candidate for compression would be *his*.[4]

[4] The compression procedure does not affect tree extraction: The user interface of the SINUS simulation program includes a procedure that stores the essential content of columns that fell victim to compression. This enables users to inspect trees dominating the complete input string.

Parameter estimation

The structure of the network of the current SINUS version was derived from a (simple) English PG-grammar that specifies the hierarchical structures and topologies for sentences consisting of a finite active or passive main clause, possibly including finite complement clauses. The NPs could include a postnominal finite relative clause or a reduced relative clause headed by a passive verb. (Table 1 gives an impression of the major constituents that could populate a clause, and of their linear order.)

The bank of 12 columns in the current version of SINUS consists of 4,632 nodes. The system includes 32 free parameters; all other parameters are either fixed in advance or their value is fully determined by other parameters. SINUS takes between 1 and 3 s to process a sentence once on a modern CPU (while parsing a 10-word sentence, SINUS needs to compute more than 430.000 activation levels). Parameter values were estimated through an optimization algorithm based on Simulated Annealing. We used the following set of 11 training sentences:

(T1) He knows her
(T2) He knows her boy
(T3) *He sleep
(T4) It is the elephant which hits the monkey
(T5) It is the elephant which the monkey hits
(T6) The elephant hits the monkey
(T7) The elephant hits the monkey which hugs the rabbit
(T8) The elephant is given to the monkey by the rabbit
(T9) The elephant is struck by the monkey
(T10) The elephant which hits the monkey hugs the rabbit
(T11) The elephant which the monkey hits hugs the rabbit

The set contains active sentences with one or two object NPs, passive sentences, subject and object relative clauses, right-branching and center-embedded relative clause, a lexical ambiguity (*her* as personal and possessive pronoun), and an ill-formed sentence (violation of Subject-verb agreement). Sentences T4 through T11 were taken from a study by Caplan et al. (1985) on sentence comprehension by aphasic patients. These sentences also played a role in our implementation of U-Space2000. A set of parameter values was judged to be suitable if SINUS yielded a correct parse tree for all training sentences (including a "corrected" parse tree for the ill-formed string).

A complete optimization run takes between 1.5 and 2 days of CPU time. This is the practical reason why we refrained from introducing noise into the system—in contrast to what we did when implementing U-Space2000. In order to obtain, for each member of the set of training sentences, a reliable estimate of the number of times the network settles down in a certain final state, the optimization procedure needs to run between 150 and 250 times. This would have taken between 8 and 18 months of CPU time. Hence, SINUS is fully deterministic: In case of ambiguity, it always yields the same solution; and, given an input sentence, it either always succeeds or always fails (no fine-grained measure of parsing difficulty).

The resulting system parses successfully not only the test sentences or sentences that embody the same syntactic structures couched in different words. For instance, the following sentences are parsed correctly as well:

S1 An elephant is an animal
S2 John gives a jukebox
S3 John is sweet
S4 He knows her small boy
S5 She knows the elephant hugs the sweet rabbit
S6 He sleeps
S7 Is John a man
S8 The elephant which sleeps hugs the rabbit
S9 The elephant hugs the rabbit which hits the monkey

In the following section, we focus on sentences that illustrate a number of important psycholinguistic phenomena.[5]

Parsing performance

SINUS can be characterized as an incremental, single-pass, bottom-up, constraint-based syntactic parser. In this section, we describe how, due to its internal dynamics, SINUS can simulate some important properties of human syntactic parsing. We start with three psycholinguistic phenomena that have been covered by other symbolic or neural models: effects of structural ambiguity, lexical ambiguity, and structural complexity. Then we explain how SINUS simulates three aspects of parsing behavior that, as far as we know, have not been addressed by other computational (psycho)linguistic models: predictive parsing, error tolerance after unification failure, and parsing sentences with unknown words.

Garden-path sentences and reanalysis

While processing garden-path sentences, SINUS spontaneously performs reanalysis, at least in case of mild garden-paths. Sentence (4), which includes a local syntactic ambiguity, is first analyzed as a simple main clause with *the monkey* as Direct Object of *sees*. However, when the

[5] A demonstration version of SINUS which runs on Apple Mac computers under OSX, is available from the corresponding author.

verb *hits* is consumed, this attachment is undone due to the inhibition emitted by the Subject U-node that is activated by the lexical frame of the latter verb. NP *the monkey* soon adopts the Subject role in the Complement clause headed by *hits*.

(4) *The elephant sees the monkey hits the rabbit*

Sentence (5a), which combines lexical and syntactic ambiguity, illustrates SINUS' sensitivity to the usage frequencies of the various readings of an ambiguous lexical item (cf. Ferreira and Clifton 1986). In English, the past-participle form of regular verbs is identical to the past-tense form. This fact occasions a well-know garden-path effect in Reduced Relative Clauses (RRC), provided that the past-participle form is infrequent or yields a locally implausible interpretation. In (5a), for example, the verb *examined* heads an RRC but, given the meaning of the preceding Subject NP and the lower frequency of the past-participle reading, there is a strong initial preference to interpret it as the Main Verb (MV) of a finite past-tense clause, as is correct in (5b).

(5a) *The lawyer examined by the court was found guilty*
(5b) *The lawyer examined the evidence*

The lexical frame associated with the finite forms of English verbs forms includes a Subject branch, which is absent from the lexical frame associated with infinitival and participial verb forms. The lexical frame of passive verbs includes a past participle as Head, an S-node as root, and a special grammatical function for the *by*-phrase, termed "Agentive Object" (AOBJ). The Agentive Object branch has a PP node as foot. When SINUS is processing a sentence with a word like *examined*, a PP headed by *by* in one of the columns boosts the activation level of an AOBJ Unification node, and indirectly the passive reading of *examined*.

In our simulation of the RRC/MV ambiguity, we looked at SINUS' behavior when the initial activation of the less frequent reading of the ambiguous verb form is varied while the activation of the frequent version is kept at a high value. When the ambiguous verb form enters a column, both the active and the passive lexical frames send their activation to the nodes that code for them, and incompatible nodes start inhibiting each other. In particular, the Subject and the Direct Object U-nodes both compete with the Agentive Object U-node. The outcome of this competition is partly determined by the activation levels that the two lexical frames and their components have when they enter a SINUS column. As is often assumed, these levels reflect—among other things—the usage frequencies of the two readings and are copied from the Mental Lexicon.

We fixated the initial activation level of the past-tense form at the maximum value of 1.0 and varied the corresponding level of the past-participle reading between .15 and .25. SINUS analyzes both sentences in (5) correctly for past-participle values in the range between .18 until .22. With lower initial past-participle activation, SINUS cannot parse the RRC version (5a) correctly, and with values above that range, it fails on the MV version (5b). For values within the range, SINUS has a short-lived preference for the RRC analysis (due to a relatively strong pre-activation of a postnominal modifier), but the MV analysis soon gains the upper hand—already before the next word (*the* or *by*) is seen. Figure 9 shows how long the Subject U-node stays dominant as a function of the activation level of the past-participle lexical frame. The more active the past-participle reading, the sooner the MV parse starts degrading.

Complexity effects and embedded clauses

The training set includes sentences with a single embedded relative clause. The embeddings are of two different types: center-embedding (T10, T11) and right-branching (T4, T5, T7). After training, SINUS parses the two types in roughly the same time. It is well-known that human language users experience less difficulty with doubly embedded right-branching clauses than with doubly embedded center-embedded ones. SINUS displays the same contrast by succeeding on sentence (6a) and (6b) but failing miserably on (6c).

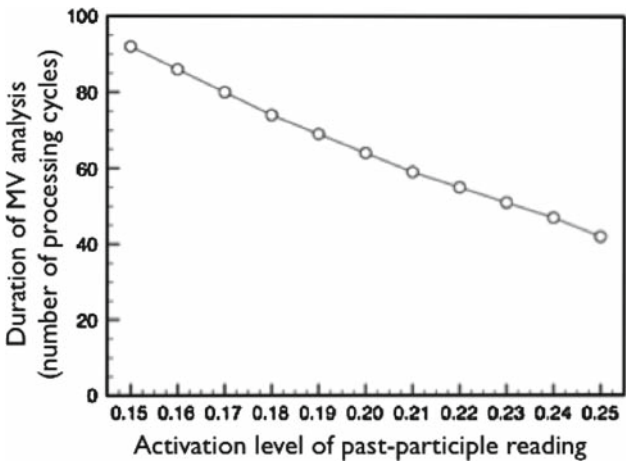

Fig. 9 Duration of the MV analysis of garden-path sentence (5a) as a function of the input activation level of *examined* as a past-participle. The vertical axis plots the time interval during which the (ultimately incorrect) MV analysis holds up, that is, the interval between the cycle in which the critical verb is entered, and the first cycle which delivers a parse tree that does not flawlessly represent an MV analysis of the input string consumed thus far. This interval ends after about 40 cycles, or 2 words, for a past-participle activation of .25, i.e., when the second article is being processed. When this activation is set to .15, it lasts considerably longer, until *was* is being processed

(6a) [S *The elephant hits the monkey* [S *which hugs the rabbit* [S *which shoots the dog*]]]
(6b) [S *The elephant* [S *which hugs the rabbit* [S *which shoots the dog*]] *hits the monkey*]
(6c) [S *The elephant* [S *which the monkey* [S *which hits the rabbit*] *hugs*] *shoots the dog*]

Error tolerance in case of feature mismatch

The activation of a Unification Node does not only depend on the activation of its root and foot nodes, but also on whether or not their feature matrices unify. In example (1), the person and number features of *money* and *counts* agree, so there is no penalty for the Subject U-node in the form of inhibition. However, had we changed the verb to the plural form *count*, then the outcome would have been different. Figure 10 shows the consequence. When the verb is introduced, the Subject U-node in the *money* column has low activation. Now, if the matrices of NP and S unify (the leftmost curve), the Subject U-node quickly reaches the maximum activation level, but if unification fails (the rightmost curve), it is suppressed by inhibition from other U-nodes. Nonetheless, it continues to receive activation from the verb, and after some time it manages to overcome the inhibition. The resulting analysis is identical to that of the correct sentence, but takes more time.

Interestingly, not only does SINUS reconstruct the intended parse tree when Subject-verb agreement is missing, it also settles down in a state that is similar to the final state reached after parsing the well-formed counterpart. At the end of the parsing process, the activation level of the feature node coding for third-person singular has reached a level that is comparable to that of the nodes coding for other number/person combinations. Hence, in a sense, the system not only recovers from an input error, it also comes close to correcting it. Apparently, in contrast to

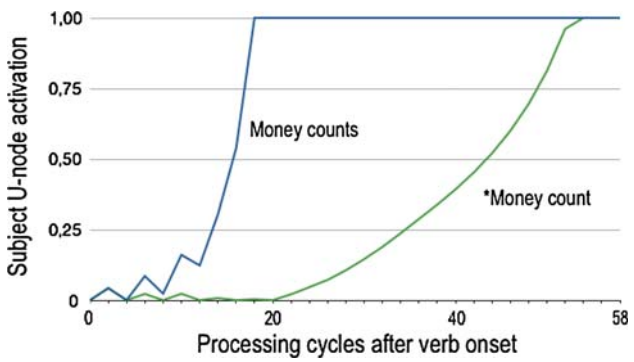

Fig. 10 Delayed unification due to missing Subject-verb agreement. Simulation with a lexicon where *count* and *counts* are represented as intransitive verbs

usual assumptions, the tasks of assembling syntactic structure and of monitoring the integrity of the resulting structures need not be consigned to distinct processing components.

Predictive parsing

SINUS offers the possibility of provisionally assigning a grammatical function to a root node before a suitable unification partner becomes available (cf. section "Flow of activation and inhibition"). Although NP *money*, taken in isolation, can function not only as Subject but also as Direct, Indirect, Prepositional or Agentive Object, in sentence (1) it is immediately analyzed as Subject, before the verb has entered. This is a consequence of feedback from the Linear Order layer and (indirectly) from the Topology layer, where the Subject NP is "expected" to be positioned in the Forefield. This feedback is added to the bottom-up activation the Subject U-node receives from NP *money*, and helps this U-node to win the competition with the other grammatical functions. However, a U-node that receives activation from only one unification partner reaches a much lower activation level than one receiving activation from two partners. This makes these early assignments relatively easy to overcome.

Once activated, a node in one column can spread activation to nodes in the next-higher or next-lower layer, not only of its own column but also of other columns. For instance, the Lexical Frame and Linear Order layers can feed into the Unification layers of subsequent columns and pre-activate one or more grammatical functions there. Simultaneously, inhibition from active U-nodes suppresses already active functions. This dynamic interplay between partly converging and partly opposing forces may be illustrated in terms of example (1). Suppose that, in the SINUS lexicon, *count* has been coded as an *in*transitive verb. After this verb has entered column 2 and its activation has reached the Topology node coding for slot M1, feedback from this node via the Linear Order Layer will reach U-nodes in subsequent columns—U-nodes coding for grammatical functions that may occupy the Midfield of a clause, in particular Direct and Indirect Object, and Modifier. However, none of these U-nodes receives bottom-up support from the intransitive verb *counts* in column 2: The lexical frame of this verb sends feedforward activation only to the Subject U-nodes in subsequent columns. These Subject U-nodes, however, are inhibited by the already active Subject-U-node in column 1. Consequently, the Modifier U-nodes will gain the highest level of activation (albeit a very modest level). Thus, SINUS may the be said to "expect" a Modifier after the intransitive *counts*, not a Direct or Indirect Object. If the second input word were the transitive verb *corrupts* instead, then the

activation pattern in the next column would be somewhat different: This verb also pre-activates Direct Object U-nodes (via feedforward connections), which in fact would become the highest activated U-nodes in the pre-activated column. This state represents the prediction of an upcoming Direct Object (in addition to a weaker expectation of a Modifier, which receives only top–down support).

The pre-activation mechanism enables SINUS to simulate, at least in a rudimentary fashion, a behavioral effect that recently has been accounted for in terms of "surprisal" (Hale 2003; see Levy 2008, for extensive discussion). The surprisal of a word is proportional to the negative log-probability of that word in its sentential context (Levy, o.c., p. 1130). Eye-movement studies have shown that a Head word is easier to process if it is preceded by more dependent constituents (Konieczny 2000; Konieczny and Döring 2003). When the number of dependent constituents increases, the Head is expected more and more strongly, that is, its surprisal values decreases. SINUS can be shown to simulate this effect when parsing a simple NP such as *A sweet rabbit*. The pre-activation in the Lexical Frame layer of an NP root node (which represents the expectation of an upcoming noun heading an NP lexical frame) increases substantially when going from one (only a Determiner) to two preceding dependents (a determiner and a prenominal adjectival Modifier), as depicted in Fig. 11. With more dependents preceding, the surprisal value gets lower, and due to the already high level of pre-activation, the noun can be processed more easily.

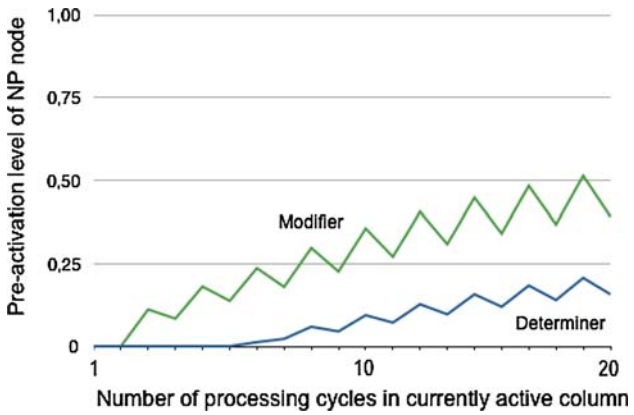

Fig. 11 Expectation-based parsing of NP *A sweet rabbit*. The curves show the development of the pre-activation of the NP node (in the pre-activated column) during processing the Determiner (*lower curve*) and the Modifier (*upper curve*). The higher the pre-activation level of the NP node, the stronger the expectation of an upcoming NP lexical frame. This expectation is relatively weak while the Determiner is being processed, and gets stronger while processing the Modifier. The jagged shape of the curves is a consequence of oscillatory feedback from mutually inhibiting U-nodes

Jabberwocky

Human comprehenders are able to reconstruct the syntactic structure of sentences that contain a high proportion of nonwords, especially if these nonwords have an internal structure reminiscent of the morphological structure of normal words, like in Lewis Carroll's Jabberwocky. Since SINUS has no morphological component, we cannot expect it to parse Jabberwocky. However, SINUS has a limited capability to parse sentences that include one or a few nonwords, purely on the basis of their position in an otherwise grammatically well-formed sentence. Sentence (6), with a single nonword, is one of the lines of Carroll's poem. Equipped with the syntactic properties of only the real English words, SINUS does not crash when parsing this sentence but delivers a virtually complete parse tree. The only elements missing are the word class and the phrasal node of *vorpal*. However, SINUS does attach the nonword as a prenominal Modifier of *sword*. Interestingly, if we replace *vorpal* by an adjective (e.g., *mighty*), the same parse tree results, but attaching *mighty* takes six processing cycles less than attaching *vorpal.*

(7) *He took his vorpal sword in hand*

In both the *vorpal* and the *mighty* version, the PP *in hand* is analyzed as a postnominal Modifier of *sword* rather than as a prepositional Modifier of *took*. This is a consequence of SINUS' preference for "recent attachment" and the absence of conceptual and idiomatic knowledge. Both *took* and *sword* compete for the PP but the latter wins because its activation level is higher than that of *took*, which in the meantime has decayed considerably.

Discussion

We have shown that SINUS fulfills quite a few important criteria that any neurocognitive model of the syntactic aspects of human sentence comprehension should meet. We hasten to add, though, that the coverage of psycholinguistic phenomena is smaller than that of U-Space2000, the predecessor. This shortcoming is compensated, we belief, by the fact that SINUS has a higher level of neurocognitive plausibility (being able to represent the online construction of syntactic trees in the form of patterns of activation in a localist neural net), and that it can simulate, at least qualitatively and rudimentarily, several psycholinguistic phenomena that are beyond reach of U-Space2000, viz. graceful degradation (error-tolerance) and predictive parsing.

Despite these achievements, much remains to be desired. Most urgent is the addition of a parallel *conceptual* processing component—maybe operating on similar principles—

that can evaluate syntactically proposed unifications against criteria of word meaning and world knowledge, and depending on the outcome, transmit excitatory or inhibitory signals to the U-nodes involved. Such a combination could help SINUS to succeed on longer sentences than the ones we have worked with thus far. Moreover, it would shift the criterion of success from delivery of a complete and grammatically correct parse tree for the input sentence to faithful reconstruction of the (or an) underlying communicative intention.

But staying within the syntactic domain, we would advocate a re-implementation where the relation between the topologies and the information represented there is less tightly linked to the information in the individual columns, so that linear-order constraints can have more impact. Presently, we have incipient ideas on how to accomplish this. Another fond wish is to extend the coverage of the parser with syntactic structures that—in terms of generative grammar—embody cross-clausal movement, as in *Which elephant did you say hugged the dog?* Performance Grammar analyzes these phenomena in terms of unification of left-peripheral slots in clausal topologies (Kempen and Harbusch 2003; Kempen 2009), but this part of PG has not yet been incorporated into SINUS. Finally, we would welcome attempts to implement SINUS-like parsers for languages other than English and German (as for the latter, see Vosse and Kempen 2008), especially for non-Germanic languages that have been targeted in empirical sentence comprehension research.

Acknowledgments The work reported here was supported by The Netherlands Organisation for Scientific Research (NWO), grant 051.04.031 (PLUS: Processing of Linguistic units in the Unification Space model). We are indebted to the members of the PLUS project for frequent discussions of the progress: Tineke Snijders, Jos van Berkum, and Peter Hagoort.

Appendix: Parameters

Each layer has its own set of parameters, and their values may differ from layer to layer.

General

Three general parameters are: number of cycles per word, number of wrap-up cycles (extra processing cycles after the final input word) and influence of feature mismatch on feedforward strength. The first two were fixed at 20 cycles. The third parameter, estimated by Simulated Annealing, reduces the amount of activation transmitted to a U-node from the unification partners (root and foot node) when unification fails. The reduced amount is a percentage of the amount that would have been transmitted in case of successful unification.

Input

Input nodes have two parameters: their default activation level at input, and the duration of their activation (fixated at three cycles). We assume that decay is absent during these cycles.

Word category

The nodes in the Word Category layer have three parameters: feedforward, feedback, and decay.

Lexical frame

This layer has five parameters: current activation, feedback from its involvement in unifications as root, feedback from its involvement in unifications as foot, inhibition, and decay.

Unification

The unification level has the highest number of parameters: 14 in total. They can be grouped under two headings: "backward looking" and "forward looking." The former concern unifications with a node in a word column to the left of the current column; the latter concern unifications with nodes in columns to the right. The rationale behind this split is that backward looking unifications yield an attachment to an already active node whereas the unification partner of a forward looking unification is dormant.

The seven parameters in each group are: feedforward, feedback, decay, inhibition, shape (λ) of the feedforward mapping function m, balance between root and foot activation levels, and distance between root and foot. Feedforward is the summed activation of the phrasal nodes that license the unification, mapped through the non-linear function m. The activation of the root and the foot node involved in a unification is weighted via the balance parameter: At balance $= 0$, the influence from the root is nullified; at balance $= 1$, only influence from the root counts; and at balance $= .5$, the influence from both root and foot count equally. The distance between root and foot influences the total amount of activation negatively, thus favoring short distance attachments.

Linear order

Nodes in this layer have eight parameters. Four of them determine the dynamics of the node: feedforward, the shape (λ) of the feedforward mapping m, decay, and inhibition. The fifth parameter determines the relative influence of to-be-placed U-nodes. The three remaining parameters ensure that the temporal order in which a topology slot is assigned to U-nodes, agrees with their left-to-right order in the topology (See also the paragraph on Linear Order nodes in section "Connectivity.").

Topology

Topology nodes are controlled by three parameters: feedforward, decay, and one that determines the shape (λ) of the mapping function m.

References

beim Graben P, Pinotsis D, Saddy D, Potthast R (2008) Language processing with dynamic fields. Cogn Neurodyn 2:79–88

Caplan D, Baker C, Dehaut F (1985) Syntactic determinants of sentence comprehension in aphasia. Cognition 21:117–175

Elman JL (1990) Finding structure in time. Cogn Sci 14:179–211

Elman JL (1991) Distributed representations, simple recurrent networks, and grammatical structure. Mach Learn 7:195–225

Ferreira F, Clifton C (1986) The independence of syntactic processing. J Mem Lang 25:348–368

Hale JT (2003) The information conveyed by words in sentences. J Psycholinguist Res 32:101–123

Harbusch K, Kempen G (2002) A quantitative model of word order and movement in English, Dutch and German complement constructions. In: Proceedings of the 19th international conference on computational linguistics (COLING-2002), Taipei. Morgan Kaufmann, San Francisco

Huyck CR (2009) A psycholinguistic model of natural language parsing implemented in simulated neurons. Cogn Neurodyn 3. doi:10.1007/s11571-009-9080-6

Kamide Y, Mitchell DC (1999) Incremental pre-head attachment in Japanese parsing. Lang Cogn Process 14:631–662

Kempen G (2009) Verplaatsingsvrije beregeling van werkwoordconstructies in het Nederlands: Een oefening in strikt monotone zinsbouw (submitted)

Kempen G, Harbusch K (2003) Dutch and German verb constructions in Performance Grammar. In: Seuren PAM, Kempen G (eds) Verb constructions in German and Dutch. Benjamins, Amsterdam

Konieczny L (2000) Locality and parsing complexity. J Psycholinguist Res 29:627–645

Konieczny L, Döring P (2003) Anticipation of clause-final heads: evidence from eye-tracking and SRNs. In: Proceedings of the 4th international joint conference on cognitive science (ICCS/ASCS-2003). University of New South Wales, Sydney

Levy R (2008) Expectation-based syntactic comprehension. Cognition 106:1126–1177

Mayberry MR, Crocker MW, Knoeferle P (2009) Learning to attend: a connectionist model of situated language comprehension. Cogn Sci 33:449–496

McClelland JL, Elman JL (1986) The TRACE model of speech perception. Cogn Psychol 18:1–86

Pickering MJ, Garrod S (2007) Do people use language production to make predictions during comprehension? Trends Cogn Sci 11:105–110

Snijders TM, Vosse T, Kempen G et al (2009) Retrieval and unification of syntactic structure in sentence comprehension: an fMRI study using word-category ambiguity. Cereb Cortex 19:1493–1503

van Berkum JJA, Brown CM et al (2005) Anticipating upcoming words in discourse: evidence from ERPs and reading times. J Exp Psychol Learn Mem Cogn 31:443–467

van der Velde F, de Kamps M (2006) Neural blackboard architectures of combinatorial structures in cognition. Behav Brain Sci 29:37–70

van der Velde F, van der Kleij G, de Kamps M (2004) Lack of combinatorial productivity in language processing with simple recurrent networks. Connect Sci 16:21–46

Vosse T, Kempen G (2000) Syntactic structure assembly in human parsing: a computational model based on competitive inhibition and a lexicalist grammar. Cognition 75:105–143

Vosse T, Kempen G (2008) Parsing verb-final clauses in German: garden-path and ERP effects modeled by a parallel dynamic parser. In: Love BC, McRae K, Sloutsky VM (eds), Proceedings of the 30th Annual Conference of the Cognitive Science Society (Washington DC, July 2008). Cognitive Science Society, Austin, TX

Vosse T, Kempen G (2009) In defense of competition during syntactic ambiguity resolution. J Psycholinguist Res 38:1–9