BEHAVIORAL AND BRAIN SCIENCES (2006) **29**, 37–108
Printed in the United States of America

Neural blackboard architectures of combinatorial structures in cognition

Frank van der Velde[a] and Marc de Kamps[b]
[a]Cognitive Psychology, Leiden University, Wassenaarseweg 52, 2333 AK Leiden, The Netherlands; [b]Robotics and Embedded Systems, Institut für Informatik, Technische Universität München, Boltzmannstrasse 3, D-85748 Garching bei München, Germany
vdvelde@fsw.leidenuniv.nl **kamps@in.tum.de**

Abstract: Human cognition is unique in the way in which it relies on combinatorial (or compositional) structures. Language provides ample evidence for the existence of combinatorial structures, but they can also be found in visual cognition. To understand the neural basis of human cognition, it is therefore essential to understand how combinatorial structures can be instantiated in neural terms. In his recent book on the foundations of language, Jackendoff described four fundamental problems for a neural instantiation of combinatorial structures: the massiveness of the binding problem, the problem of 2, the problem of variables, and the transformation of combinatorial structures from working memory to long-term memory. This paper aims to show that these problems can be solved by means of neural "blackboard" architectures. For this purpose, a neural blackboard architecture for sentence structure is presented. In this architecture, neural structures that encode for words are temporarily bound in a manner that preserves the structure of the sentence. It is shown that the architecture solves the four problems presented by Jackendoff. The ability of the architecture to instantiate sentence structures is illustrated with examples of sentence complexity observed in human language performance. Similarities exist between the architecture for sentence structure and blackboard architectures for combinatorial structures in visual cognition, derived from the structure of the visual cortex. These architectures are briefly discussed, together with an example of a combinatorial structure in which the blackboard architectures for language and vision are combined. In this way, the architecture for language is grounded in perception. Perspectives and potential developments of the architectures are discussed.

Keywords: binding; blackboard architectures; combinatorial structure; compositionality; language; dynamic system; neurocognition; sentence complexity; sentence structure; working memory; variables, vision

1. Introduction

Human cognition is unique in the manner in which it processes and produces complex combinatorial (or compositional) structures (e.g., Anderson 1983; Newell 1990/1994; Pinker 1998). Therefore, to understand the neural basis of human cognition, it is essential to understand how combinatorial structures can be instantiated in neural terms. However, combinatorial structures present particular challenges to theories of neurocognition (Marcus 2001) that are not always recognized in the cognitive neuroscience community (Jackendoff 2002).

A prominent example of these challenges is given by the neural instantiation (in theoretical terms) of linguistic structures. In his recent book on the foundations of language, Jackendoff (2002; see also Jackendoff, 2003) analyzed the most important theoretical problems that the combinatorial and rule-based nature of language presents to theories of neurocognition. He summarized the analysis of these problems under the heading of "four challenges for cognitive neuroscience" (pp. 58–67). As recognized by Jackendoff, these problems arise not only with linguistic structures but with combinatorial cognitive structures in general.

This paper aims to show that neural "blackboard" architectures can provide an adequate theoretical basis for a

FRANK VAN DER VELDE has a Ph.D. in Cognitive Psychology, an M.Sc. in Physics and an M.A. in Philosophy (epistomology, logic). He received a fellowship award from the Royal Dutch Academy of Science. van der Velde has published articles on topics ranging from experimental psychology to computation theory of cognition. He is the author of a forthcoming book on neural architectures of cognition. His main interest is the understanding of the neural mechanisms of cognition—in particular as related to the issue of productivity—as found in visual perception, working memory, reasoning, and language.

MARC DE KAMPS is a postdoc at the Chair for Robotics and Embedded Systems at the Department of Computer Science of the Technische Universität München. After completing an M.Sc. in theoretical physics and a Ph.D. in particle physics, he worked for the Psychology Department of Leiden University. Together with van der Velde, de Kamps constructed neural network models of visual attention, working memory, and language representation. His interests include the relationship between artificial and biological neural networks and the efficient simulation of large scale cortical networks. He is also deputy coordinator of the EU Network of Excellence nEUro-IT.net.

© 2006 Cambridge University Press 0140-525X/06 $12.50

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

neural instantiation of combinatorial cognitive structures. In particular, we discuss how the problems presented by Jackendoff (2002) can be solved in terms of a neural blackboard architecture of sentence structure. We also discuss the similarities between the neural blackboard architecture of sentence structure and neural blackboard architectures of combinatorial structures in visual cognition and visual working memory (van der Velde 1997; van der Velde & de Kamps 2001; 2003).

To begin with, we first outline the problems described by Jackendoff (2002) in more detail. This presentation is followed by a discussion of the most important solutions that have been offered so far to meet some of these challenges. These solutions are based on either synchrony of activation or recurrent neural networks.[1]

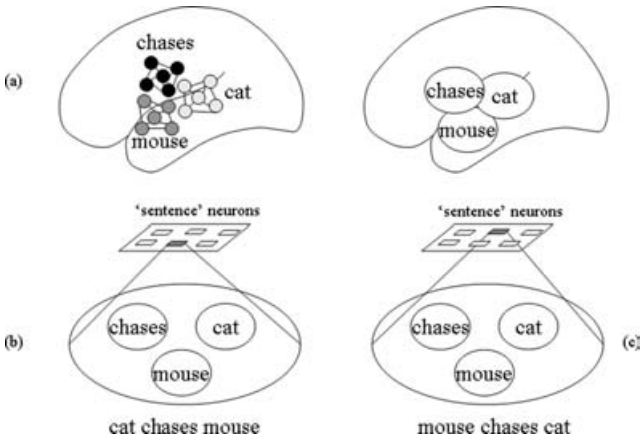


Figure 1. a. Two illustrations of neural structures ("neural word assemblies") activated by the words *cat*, *chases*, and *mouse*. Bottom: An attempt to encode sentence structures with specialized "sentence" neurons. b. A "sentence" neuron has the assemblies for the words *cat*, *chases*, and *mouse* in its "receptive field" (as indicated by the cone). The neuron is activated by a specialized neural circuit when the assemblies in its receptive field are active in the order *cat chases mouse*. c. A similar "sentence" neuron for the sentence *mouse chases cat*.

2. Four challenges for cognitive neuroscience

The four challenges for cognitive neuroscience presented by Jackendoff (2002; see also Marcus 2001) consist of the massiveness of the binding problem that occurs in language, the problem of multiple instances (or the "problem of 2"), the problem of variables, and the relation between binding in working memory and binding in long-term memory. We discuss the four problems in turn.

2.1. The massiveness of the binding problem

In neuroscience, the binding problem concerns the way in which neural instantiations of parts (constituents) can be related (bound) temporarily in a manner that preserves the structural relations between the constituents. Examples of this problem can be found in visual perception. Colors and shapes of objects are partly processed in different brain areas, but we perceive objects as a unity of color and shape. Hence, in a visual scene with a green apple and a red orange, the neurons that code for green have to be related (temporarily) to the neurons that code for apple, so that the confusion with a red apple (and a green orange) can be avoided.

In the case of language, the problem is illustrated in Figure 1. Assume that words like *cat*, *chases*, and *mouse* each activate specific neural structures, such as the "word assemblies" discussed by Pulvermüller (1999). The problem is how the neural structures or word assemblies for *cat* and *mouse* can be bound to the neural structure or word assembly of the verb *chases* in line with the thematic roles (or argument structure) of the verb. That is, how can *cat* and *mouse* be bound to the role of agent and theme of *chases* in the sentence *The cat chases the mouse* and to the role of theme and agent of *chases* in the sentence *The mouse chases the cat*?

A potential solution for this problem is illustrated in Figure 1. It consists of specialized neurons (or populations of neurons) that are activated when the strings *cat chases mouse* (Fig. 1b) or *mouse chases cat* (Fig. 1c) are heard or seen. Each neuron has the word assemblies for *cat*, *mouse*, and *chases* in its "receptive field" (illustrated with the cones in Figs. 1b and 1c). Specialized neural circuits could activate one neuron in the case of *cat chases mouse*, and another neuron in the case of *mouse chases cat*, by using the difference in temporal word order in both strings. Circuits of this kind can be found in the case of motion detection in visual perception (e.g., Hubel 1995). For example, the movement of a vertical bar that sweeps across the retina in the direction from A to B can be detected by using the difference in activation time (onset latency) between the ganglion cells in A and B.

A fundamental problem with this solution in the case of language is its lack of productivity. Only specific and familiar sentences can be detected in this way. But any novel sentence of the type *Noun chases Noun* or, more generally, *Noun Verb Noun* will not be detected because the specific circuit (and neuron) for that sentence will be missing. Yet when we learn that *Dumbledore is headmaster of Hogwarts*, we immediately understand the meaning of *Dumbledore chases the mouse*, even though we have never encountered that sentence before.

The difference between language and motion detection in this respect illustrates that the nature of these two cognitive processes is fundamentally different. In the case of motion detection, there is a limited set of possibilities, so that it is possible (and it pays off) to have specialized neurons and neural circuits for each of these possibilities. But this solution is not feasible in the case of language. Linguists typically describe language in terms of its unlimited combinatorial productivity. Words can be combined into phrases, which in turn can be combined into sentences, so that arbitrary sentence structures can be filled with arbitrary arguments (e.g., Chomsky 2000; Jackendoff 2002; Piattelli-Palmarini 2002; Pullum & Scholz 2001; Sag & Wasow 1999; Webelhuth 1995). In theory, an unlimited amount of sentences can be produced in this way, which excludes the possibility of having specialized neurons and circuits for each of these sentences.

However, unlimited (recursive) productivity is not necessary to make a case for the combinatorial nature of language, given the number of sentences that can be produced or understood. For example, the average English-speaking 17-year-old knows more than 60,000 words (Bloom 2000). With this lexicon and with a limited

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

sentence length of 20 words or less, one can produce a set of sentences in natural language in the order of 10^{20} or more (Miller 1967; Pinker 1998). A set of this kind can be characterized as a "performance set" of natural language in the sense that (barring a few selected examples) any sentence from this set can be produced or understood by a normal language user. Such a performance set is not unlimited but it is of "astronomical" magnitude (e.g., 10^{20} exceeds the estimated lifetime of the universe expressed in seconds). By consequence, most sentences in this set are sentences that we have never heard or seen before. Yet because of the combinatorial nature of language, we have the ability to produce and understand arbitrary sentences from a set of this kind.

Hence the set of possibilities that we can encounter in the case of language is unlimited in any practical sense. This precludes a solution of the binding problem in language in terms of specialized neurons and circuits. Instead a solution is needed that depends on the ability to bind arbitrary arguments to the thematic roles of arbitrary verbs in agreement with the structural relations expressed in the sentence. Moreover, the solution has to satisfy the massiveness of the binding problem as it occurs in language, which is due to the often complex and hierarchical nature of linguistic structures. For example, in the sentence *The cat that the dog bites chases the mouse*, the noun *cat* is bound to the role of theme of the verb *bites*, but it is bound to the role of agent of the verb *chases*. In fact, the whole phrase *The cat that the dog bites* is bound to the role of agent of the verb *chases* (with *cat* as the head of the phrase). Each of these specific bindings has to be satisfied in an encoding of this sentence.

2.2. The problem of 2

The second problem presented by Jackendoff (2002) is the problem of multiple instances, or the "problem of 2." Jackendoff illustrates this problem with the sentence *The little star is beside a big star.*[2] The word *star* occurs twice in this sentence, the first time related to the word *little* and the second time related to the word *big*. The problem is how in neural terms the two occurrences of the word *star* can be distinguished, so that *star* is first bound to *little* and then with *big*, without creating the erroneous binding of *little big star*. The problem of 2 results from the assumption that any occurrence of a given word will result in the activation of the same neural structure (e.g., its word assembly, as illustrated in Fig. 1). But if the second occurrence of a word results in only the reactivation of a neural structure that was already activated by the first occurrence of that word, the two occurrences of the same word are indistinguishable (van der Velde 1999).

Perhaps the problem could be solved by assuming that there are multiple neural structures that encode for a single word. The word *star* could then activate one neural structure in *little star* and a different one in *big star*, so that the bindings *little star* and *big star* can be encoded without creating *little big star*. However, this solution would entail that there are multiple neural structures for all words in the lexicon, perhaps even for all potential positions a word could have in a sentence (Jackendoff 2002).

More important, this solution disrupts the unity of word encoding as the basis for the meaning of a word. For example, the relation between the neural structures for *cat* and *mouse* in *cat chases mouse* could develop into the neural basis for the long-term knowledge ("fact") that cats chase mice. Similarly, the relation between the neural structures for *cat* and *dog* in *dog bites cat* could form the basis of the fact that dogs fight with cats. But if the neural structure for *cat* (say, cat_1) in cat_1 *chases mouse* is different from the neural structure for *cat* (say, cat_2) in *dog bites* cat_2, then these two facts are about different kinds of animals.

2.2.1. The problem of 2 and the symbol-grounding problem. It is interesting to look at the problem of 2 from the perspective of the symbol-grounding problem that occurs in cognitive symbol systems. Duplicating symbols is easy in a symbol system. However, in a symbol system, one is faced with the problem that symbols are arbitrary entities (e.g., strings of bits in a computer), which therefore have to be interpreted to provide meaning to the system. That is, symbols have to be "grounded" in perception and action if symbol systems are to be viable models of cognition (Barsalou 1999; Harnad 1991).

Grounding in perception and action can be achieved with neural structures such as the word assemblies illustrated in Figure 1. In line with the idea of neural assemblies proposed by Hebb (1949), Pulvermüller (1999) argued that words activate neural assemblies distributed over the brain (as illustrated with the assemblies for the words *cat*, *mouse*, and *chases* in Fig. 1). One could imagine that these word assemblies have developed over time by means of a process of association. Each time a word was heard or seen, certain neural circuits would have been activated in the cortex. Over time, these circuits will be associated, which results in an overall cell assembly that reflects the meaning of that word.

But, as argued above, word assemblies are faced with the problem of 2. Hence it seems that the problem of 2 and the symbol-grounding problem are complementary problems. To provide grounding, the neural structure that encodes for a word is embedded in the overall network structure of the brain. But this makes it difficult to instantiate a duplication of the word and hence to instantiate even relatively simple combinatorial structures such as *The little star is beside a big star*. Conversely, duplication is easy in symbol systems (e.g., if *1101* = *star*, then one would have *The little 1101 is beside a big 1101*, with *little* and *big* each related to an individual copy of *1101*). But symbols can be duplicated easily because they are not embedded in an overall structure that provides the grounding of the symbol.[3]

2.3. The problem of variables

The knowledge of specific facts can be instantiated on the basis of specialized neural circuits, in line with those illustrated in Figure 1. But knowledge of systematic facts, such as the fact that *own(y, z)* follows from *give(x, y, z)*, cannot be instantiated in this way, that is, in terms of a listing of all specific instances of the relation between the predicates *own* and *give* [e.g., from *give(John, Mary, book)* it follows that *own(Mary, book)*; from *give(Mary, John, pen)* it follows that *own(John, pen)*; etc.].

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

Instead, the derivation that *own(Mary, book)* follows from *give(John, Mary, book)* is based on the rule that *own(y, z)* follows from *give(x, y, z)*, combined with the binding of *Mary* to the variable *y* and *book* to the variable *z*. Marcus (2001) analyzed a wide range of relationships that can be described in this way. They are all characterized by the fact that an abstract rule-based relationship expressed in terms of variables is used to determine relations between specific entities (e.g., numbers, words, objects, individuals).

The use of rule-based relationships with variable binding provides the basis for the systematic nature of cognition (Fodor & Pylyshyn 1988). Cognition is systematic in the sense that one can learn from specific examples and apply that knowledge to all examples of the same kind. A child will indeed encounter only specific examples (e.g., that when John gives Mary a book, it follows that Mary owns the book) and yet it will learn that *own(y, z)* follows from all instances of the kind *give(x, y, z)*. In this way, the child is able to handle novel situations, such as the derivation that *own(Harry, broom)* follows from *give (Dumbledore, Harry, broom)*.

The importance of rule-based relationships for human cognition raises the question of how relationships with variable binding can be instantiated in the brain.

2.4. Binding in working memory versus long-term memory

Working memory in the brain is generally assumed to consist of a sustained form of activation (e.g., Amit 1989; Fuster 1995). That is, information is stored in working memory as long as the neurons that encode the information remain active. In contrast, long-term memory results from synaptic modification, such as long-term potentiation (LTP). In this way, the connections between neurons are modified (e.g., enhanced). When some of the neurons are then reactivated, they will reactivate the others neurons as well. The neural word assemblies illustrated in Figure 1 are formed by this process.

Both forms of memory are related in the sense that information in one form of memory can be transformed into information in the other form of memory. Information could be stored in a working memory (which could be specific for a given form of information, such as sentence structures) before it is stored in long-term memory. Conversely, information in long-term memory can be reactivated and stored in working memory. This raises the question of how the same combinatorial structure can be instantiated both in terms of neural activation (as found in working memory or in stimulus-dependent activation) and in terms of synaptic modification, and how these different forms of instantiation can be transformed into one another.

2.5. Overview

It is clear that the four problems presented by Jackendoff (2002) are interrelated. For example, the problem of 2 also occurs in rule-based derivation with variable binding; the massiveness of the binding problem is found in combinatorial structures stored in working memory and in combinatorial structures stored in long-term memory. Therefore a solution to these problems has to be an integrated one that solves all four problems simultaneously. In this paper, we discuss how all four problems can be solved in terms of neural blackboard architectures in which combinatorial structures can be instantiated.

First, however, we discuss two alternatives for a neural instantiation of combinatorial structures: the use of synchrony of activation (e.g., von der Malsburg 1987) as a mechanism for binding constituents in combinatorial structures, and the use of recurrent neural networks to process combinatorial structures, in particular sentence structures.

3. Combinatorial structures with synchrony of activation

An elaborate example of a neural instantiation of combinatorial structures in which synchrony of activation is used as a binding mechanism is found in the model of reflexive reasoning presented by Shastri and Ajjanagadde (1993b). In their model, synchrony of activation is used to show how a known fact such as *John gives Mary a book* can result in an inference such as *Mary owns a book*.

The proposition *John gives Mary a book* is encoded by a "fact node" that detects the respective synchrony of activation between the nodes for *John*, *Mary*, and *book* and the nodes for *giver*, *recipient*, and *give-object*, which encode for the thematic roles of the predicate *give (x, y, z)*. In a simplified manner, the reasoning process begins with the query *own(Mary, book)?* (i.e., does Mary own a book?). The query results in the respective synchronous activation of the nodes for *owner* and *own-object* of the predicate *own(y, z)* with the nodes for *Mary* and *book*. In turn, the nodes for *recipient* and *give-object* of the predicate *give(x, y, z)* are activated by the nodes for *owner* and *own-object*, such that *owner* is in synchrony with *recipient* and *own-object* is in synchrony with *give-object*. As a result, the node for *Mary* is in synchrony with the node for *recipient*, and the node for *book* is in synchrony with the node for *give-object*. This allows the fact node for *John gives Mary a book* to become active, which produces the affirmative answer to the query.

A first problem with a model of this kind is found in a proposition such as *John gives Mary a book and Mary gives John a pen*. With synchrony of activation as a binding mechanism, a confusion arises between *John* and *Mary* in their respective roles of *giver* and *recipient* in this proposition. In effect, the same pattern of activation will be found in the proposition *John gives Mary a pen and Mary gives John a book*. Hence, with synchrony of activation as a binding mechanism, both propositions are indistinguishable. It is not difficult to see the problem of 2 here. *John* and *Mary* occur twice in the proposition but in different thematic roles. The simultaneous but distinguishable binding of *John* and *Mary* with different thematic roles cannot be achieved with synchrony of activation.

To solve this problem, Shastri and Ajjanagadde allowed for a duplication (or multiplication) of the nodes for the predicates. In this way, the whole proposition *John gives Mary a book and Mary gives John a pen* is partitioned into the two elementary propositions *John gives Mary a book* and *Mary gives John a pen.* To distinguish between the propositions, the nodes for the predicate *give(x, y, z)*

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

are duplicated. Hence there are specific nodes for, say, *give*$_1$*(x, y, z)* and *give*$_2$*(x, y, z)*, with *give*$_1$*(x, y, z)* related to *John gives Mary a book* and *give*$_2$*(x, y, z)* related to *Mary gives John a pen*. Furthermore, for the reasoning process to work, the associations between predicates have to be duplicated as well. Hence the node for *give*$_1$*(x, y, z)* has to be associated with a node for, say, *own*$_1$*(y, z)*, and the node for *give*$_2$*(x, y, z)* has to be associated with a node for *own*$_2$*(y, z)*.

This raises the question of how these associations can be formed simultaneously during learning. During its development, a child will learn from specific examples; hence it will learn that when *John gives Mary a book*, it follows that *Mary owns the book*. In this way, the child will form an association between the nodes for *give*$_1$*(x, y, z)* and *own*$_1$*(y, z)*. But the association between the node for *give*$_2$*(x, y, z)* and *own*$_2$*(y, z)* would not be formed in this case, because these nodes are not activated with *John gives Mary a book* and *Mary owns the book*. Hence, when the predicate *give(x, y, z)* is duplicated into *give*$_1$*(x, y, z)* and *give*$_2$*(x, y, z)*, the systematicity between *John gives Mary a book* and *Mary gives John a pen* is lost.

3.1. Nested structures with synchrony of activation

The duplication solution discussed above fails with nested (or hierarchical) propositions. For example, the proposition *Mary knows that John knows Mary* cannot be partitioned into two propositions, *Mary knows* and *John knows Mary*, because the entire second proposition is the *y* argument of *knows(Mary, y)*. Hence the fact node for *John knows Mary* has to be in synchrony with the node for *know-object* of the predicate *know(x, y)*. The fact node for *John knows Mary* will be activated because *John* is in synchrony with the node for *knower* and *Mary* is in synchrony with the node for *know-object*. However, the fact node for *Mary knows Mary*, for example, will also be activated in this case, because *Mary* is in synchrony with both *knower* and *know-object* in the proposition *Mary knows that John knows Mary*. Hence the proposition *Mary knows that John knows Mary* cannot be distinguished from the proposition *Mary knows that Mary knows Mary*.

As this example shows, synchrony as a binding mechanism is faced with the "one-level" restriction (Hummel & Holyoak 1993), that is, synchrony can encode bindings at only one level of abstraction or hierarchy at a time.

3.2. Productivity with synchrony of activation

A fundamental problem with the use of synchrony of activation as a binding mechanism in combinatorial structures is its lack of productivity. Synchrony of activation has to be detected to process the information that it encodes (Dennett 1991). In Shastri and Ajjanagadde's model, fact nodes (e.g., the fact node for *John gives Mary a book*) detect the synchrony of activation between arguments and thematic roles. But fact nodes and the circuits that activate them are similar to the specialized neurons and circuits illustrated in Figure 1. Having such nodes and circuits for all possible verb-argument bindings that can occur in language, in particular for novel instances of verb-argument binding, is excluded. As a result, synchrony of activation as a binding mechanism fails to provide the productivity given by combinatorial structures.

The binding problems as analyzed here – the inability to solve the problem of 2, the inability to deal with nested structures (the "one-level restriction"), and the lack of systematicity and productivity – are typical for the use of synchrony of activation as a binding mechanism (van der Velde & de Kamps 2002b). The lack of productivity given by the need for "synchrony detectors" is perhaps the most fundamental problem for synchrony as a mechanism for binding constituents in combinatorial structures. True combinatorial structures provide the possibility of answering binding questions about novel combinations (e.g., novel sentences) never seen or heard before. Synchrony detectors (or conjunctive forms of encoding in general) will be missing for novel combinatorial structures, which precludes the use of synchrony as a binding mechanism for these structures. Synchrony as a binding mechanism would seem to be restricted to structures for which conjunctive forms of encoding exist and that satisfy the "one-level restriction" (van der Velde & de Kamps 2002b).

4. Processing linguistic structures with "simple" recurrent neural networks

The argument that combinatorial structures are needed to obtain productivity in cognition has been questioned (Churchland 1995; Elman 1991; Port & van Gelder 1995). In this view, productivity in cognition can be obtained in a "functional" manner ("functional compositionality"; van Gelder 1990) without relying on combinatorial structures. The most explicit models of this kind deal with sentence structures.

A first example is the neural model of thematic role assignment in sentence processing presented by McClelland and Kawamoto (1986). However, the model was restricted to one particular sentence structure and it could not represent different tokens of the same type, for example, dog_{agent} and dog_{theme} in *dog chases dog*. St. John and McClelland (1990) presented a more flexible model based on a recurrent network. The model learned presegmented single-clause sentences and assigned thematic roles to the words in the sentence, but it could not handle more complex sentences, such as sentences with embedded clauses.

A model that processed embedded clauses was presented by Miikkulainen (1996). It consisted of three parts: a parser, a segmenter, and a stack. The segmenter (a feed-forward network) divided the input sentence into clauses (by detecting clause boundaries). The stack memorized the beginning of a matrix clause, for example, *girl* in *The girl, who liked the boy, saw the boy*. The parser assigned thematic roles (agent, act, patient) to the words in a clause. All clauses, however, were two- or three-word clauses, because the output layer of the parser had three nodes.

The "simple" recurrent neural networks (RNNs) play an important role in the attempt to process sentence structures without relying on combinatorial structures (Elman 1991; Miikkulainen 1996; Palmer-Brown et al. 2002). They consist of a multilayer feed-forward network, in which the activation pattern in the hidden (middle) layer

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

is copied back to the input layer, as part of the input to the network in the next learning step. In this way, RNNs are capable of processing sequential structures. Elman (1991) used RNNs in a word-prediction task. For example, with *Boys who chase boy feed cats*, the network had to predict that after *Boys who chase*, a noun would follow, and that after *Boys who chase boy*, a plural verb would occur. The network was trained with sentences from a language generated with a small lexicon and a basic phrase grammar. The network succeeded in this task with both the sentences that were used in the training session and other sentences from the same language.

The RNNs used by Elman (1991) could not answer specific binding questions such as *Who feed cats?* Hence the network did not bind specific words to their specific roles in the sentence structure. Nevertheless, RNNs seem capable of processing aspects of sentence structures in a noncombinatorial manner. But RNNs model languages derived from small vocabularies (in the order of 10 to 100 words). In contrast, the vocabulary of natural language is huge, which results in an "astronomical" productivity, even with limited sentence structures (e.g., sentences with 20 words or less; see sect. 2.1). We discuss "combinatorial" productivity with RNNs in more detail below.

4.1. Combinatorial productivity with RNNs used in sentence processing

Elman (1991) used a language in the order of 10^5 sentences based on a lexicon of about 20 words. In contrast, the combinatorial productivity of natural language is in the order of 10^{20} sentences or more, based on a lexicon of 10^5 words. A basic aspect of such a combinatorial productivity is the ability to insert words from one familiar sentence context into another. For example, if one learns that *Dumbledore is headmaster of Hogwarts*, one can also understand *Dumbledore chases the mouse*, even though this specific sentence has not been encountered before. To approach the combinatorial productivity of natural language, RNNs should have this capability as well.

We investigated this question by testing the ability of RNNs to recognize a sentence consisting of a new combination of familiar words in familiar syntactic roles (van der Velde et al. 2004b). In one instance, we used sentences such as *dog hears cat, boy sees girl, dog loves girl*, and *boy follows cat* to train the network on the word-prediction task. The purpose of the training sentences was to familiarize the RNNs with *dog*, *cat*, *boy*, and *girl* as arguments of verbs. Then a verb such as *hears* from *dog hears cat* was inserted into another trained sentence, such as *boy sees girl*, to form the test sentence *boy hears girl*, and the networks were tested on the prediction task for this sentence.

To strengthen the relations between *boy*, *hears*, and *girl*, we also included training sentences such as *boy who cat hears obeys John* and *girl who dog hears likes Mary*. These sentences introduce *boy* and *hears* and *girl* and *hears* in the same sentence context (without using *boy hears* and *hears girl*).[4] In fact, *girl* is the object of *hears* in *girl who dog hears likes Mary*, as in the test sentence *boy hears girl*.

However, although the RNNs learned the training sentences to perfection, they failed with the test sentences. Despite the ability to process *boy sees girl* and *dog hears cat*, and even *girl who dog hears likes Mary*, they could not process *boy hears girl*. The behavior of the RNNs with the test sentence *boy hears girl* was similar to the behavior in a "word salad" condition consisting of random word strings based on the words used in the training session. In this "word salad" condition, the RNNs predicted the next word on the basis of direct word-to-word associations, determined by all two-word combinations found in the training sentences. The similarity between "word salads" and the test sentence *boy hears girl* suggests that RNNs resort to word-to-word associations when they have to process novel sentences composed of familiar words in familiar grammatical structures.

The results of these simulations indicate that RNNs of Elman (1991) do not possess a minimal form of the combinatorial productivity underlying human language processing. It is important to note that this lack of combinatorial productivity is not just a negative result of the learning algorithm used; the training sentences were learned to perfection. With another algorithm these sentences could, at best, be learned to the same level of perfection. Furthermore, the crucial issue here is not learning but the contrast in behavior exhibited by the RNNs in these simulations. The RNNs were able to process ("understand") *boy sees girl* and *dog hears cat*, and even *girl who dog hears likes Mary*, but not *boy hears girl*. This contrast in behavior is not found in humans, regardless of the learning procedure used. The reason is the systematicity of the human language system. If you understand *boy sees girl*, *dog hears cat*, and *girl who dog hears likes Mary*, you cannot but understand *boy hears girl*. Any failure to do so would be regarded as pathological.[5]

4.2. Combinatorial productivity versus recursive productivity

The issue of combinatorial productivity is a crucial aspect of natural-language processing, which is sometimes confused with the issue of recursive productivity. Combinatorial productivity concerns the productivity that results from combining a large lexicon with even limited syntactical structures. Recursive productivity deals with the issue of processing more complex syntactic structures, such as (deeper) center-embeddings.

The difference can be illustrated with the "long short-term memory recurrent networks" (LSTMs). LSTMs outperform standard RNNs on recursive productivity (Gers & Schmidhuber 2001). Like humans, RNNs have limited recursive productivity, but LSTMs do not. They can, for example, handle context-free languages such as $a^n b^m B^m A^n$ for arbitrary (n, m). However, the way in which they do this excludes their ability to handle combinatorial productivity.

An LSTM is an RNN in which hidden units are replaced with "memory blocks" of units, which develop into counters during learning. With $a^n b^m B^m A^n$, the network develops two counters, one for the ns and one for the ms. Hence the network counts whether a^n matches A^n and b^m matches B^m. This makes sense because all sentences have the same words, that is, they are all of the form $a^n b^m B^m A^n$. Sentences differ only in the value of n and/or m. So the network can learn that it has to count the ns and ms.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

But this procedure makes no sense in a natural language. The sentence *mouse chases cat* is fundamentally different from the sentence *cat chases mouse*, even though they are both *Noun-Verb-Noun* sentences. How could an LSTM capture this difference? Should the model, for example, count the number of times that *mouse* and *cat* appear in any given sentence? Consider the number of possibilities that would have to be dealt with, given a lexicon of 60,000 words, instead of four words as in $a^n b^m B^m A^n$. Furthermore, how would it deal with novel sentences, like *Dumbledore chases mouse*? Could it develop counters to match *Dumbledore* and *mouse* if it has never seen these words in one sentence before?

This example illustrates that combinatorial productivity is an essential feature of natural-language processing but is virtually nonexistent in artificial languages. The ability to process complex artificial languages does not guarantee the ability to process combinatorial productivity as found in natural language.

4.3. RNNs and the massiveness of the binding problem

Yet RNNs are capable of processing learned sentences like *girl who dog hears obeys Mary* and other complex sentence structures. Perhaps RNNs could be used to process sentence structures in abstract terms, that is, in terms of *Nouns (N)* and *Verbs (V)*. Hence *N-who-N-V-V-N* instead of *girl who dog hears obeys Mary*.

However, sentences such as *cat chases mouse* and *mouse chases cat* are *N-V-N* sentences and hence indistinguishable for these RNNs. But these sentences convey very different messages that humans can understand. In particular, humans can answer "who does what to whom" questions for these sentences, which cannot be answered using the *N-V-N* structure processed by these RNNs.

This raises two important questions for these RNNs. First, how is the difference between *cat chases mouse* and *mouse chases cat* instantiated in neural terms, given that this cannot be achieved with RNNs? Second, how can the structural *N-V* information processed by these RNNs be related to the specific content of each sentence? This is a "binding" problem, because it requires that, for example, the first *N* in *N-V-N* is bound to *cat* in the first sentence and to *mouse* in the second sentence.

However, even if these problems are solved, sentence processing in terms of *N-V* strings is still faced with serious difficulties, as illustrated by the following sentences:

The cat that the dog that the boy likes bites chases the mouse (1)

The fact that the mouse that the cat chases roars surprises the boy (2)

The abstract (*N-V*) structure of both sentences is the same: *N-that-N-that-N-V-V-V-N*. Yet there is a clear difference in complexity between these sentences (Gibson 1998). Sentences with complement clauses, such as (2), are much easier to process than sentences with center-embeddings, such as (1). This difference can be explained in terms of the dependencies within the sentence structures. In (1) the first noun is related to the second verb as its object (theme) and to the third verb as its subject (agent). In (2), the first noun is related to only the third verb (as its subject). This difference in structural dependency is not captured in the sequence *N-that-N-that-N-V-V-V-N*.

The difference between sentences (1) and (2) again illustrate the massiveness of the binding problem that occurs in linguistic structures. Words and clauses have to be bound correctly to other words and clauses in different parts of the sentence in line with the hierarchical structure of a sentence. These forms of binding are beyond the capacity of language processing with RNNs. Similar limitations of RNNs are found with the problem of variables (Marcus 2001).

5. Blackboard architectures of combinatorial structures

A combinatorial structure consists of parts (constituents) and their relations. The lack of combinatorial productivity with RNNs illustrates a failure to encode the individual parts (words) of a combinatorial structure (sentence) in a productive manner. In contrast, synchrony of activation fails to instantiate even moderately complex relations in the case of variable binding. These examples show that neural models of combinatorial structures can succeed only if they provide a neural instantiation of both the parts and the relations of combinatorial structures.

A blackboard architecture provides a way to instantiate the parts and the relations of combinatorial structures (e.g., Newman et al. 1997). A blackboard architecture consists of a set of specialized processors ("demons"; Selfridge 1959) that interact with each other using a blackboard ("workbench," "bulletin board"). Each processor can process and modify the information stored on the blackboard. In this way, the architecture exceeds the ability of each individual processor. For language, one could have processors for the recognition of words and for the recognition of specific grammatical relations. These processors could then interact by using a blackboard to process a sentence. With the sentence *The little star is beside a big star*, the word processors could store the symbol for *star* on the blackboard, first in combination with the symbol for *little* and then in combination with the symbol for *big*. Other processors could determine the relation (*beside*) between these two copies of the symbol for *star*. Jackendoff (2002) discusses blackboard architectures for phonological, syntactic, and semantic structures.

Here we propose and discuss a neural blackboard architecture for sentence structure based on neural assemblies. To address Jackendoff's (2002) problems, neural word assemblies are not copied in this architecture. Instead, they are temporarily bound to the neural blackboard in a manner that distinguishes between different occurrences of the same word and that preserves the relations between the words in the sentence. For example, with the sentence *The cat chases the mouse*, the neural assemblies for *cat* and *mouse* are bound to the blackboard as the subject (agent) and object (theme) of *chases*.

With the neural structure of *The cat chases the mouse*, the architecture can produce correct answers to questions like "Who chases the mouse?" or "Whom does the cat chase?" These questions can be referred to as "binding questions" because they test the ability of an architecture

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

to "bind" familiar parts in a (potentially novel) combinatorial structure. A neural instantiation of a combinatorial structure like *The cat chases the mouse* fails if it cannot produce the correct answers to such questions. In language, binding questions typically query "who does what to whom" information, which is characteristic of information provided by a sentence (Calvin & Bickerton 2000; Pinker 1994). Aphasic patients, for example, are tested on their language abilities using nonverbal "who does what to whom" questions (Caplan 1992). In general, the ability to answer binding questions is of fundamental importance for cognition because it is related to the ability to select information needed for purposive action (van der Heijden & van der Velde 1999).

6. A neural blackboard architecture of sentence structure

In the architecture, words are encoded in terms of neural "word" assemblies, in line with Pulvermüller (1999), as illustrated in Figure 1. It is clear that the relations between the words in a sentence cannot be encoded by direct associations between word assemblies. For example, the association *cat-chases-mouse* does not distinguish between *The cat chases the mouse* and *The mouse chases the cat*.

However, relations between words can be encoded – and Jackendoff's problems can be solved – if word assemblies are embedded in a neural architecture in which structural relations can be formed between these assemblies. Such an architecture can be formed by combining word assemblies with "structure" assemblies.

A word assembly is a neural structure that is potentially distributed over a large part of the brain, depending on the nature of the word (e.g., see Pulvermüller 1999). A part of that structure could be embedded in a "phonological" architecture that controls the auditory perception and speech production related to that word. Other parts could be embedded in other architectures that control other forms of neural processing related to other aspects of that word (e.g., visual perception, semantics).

Here we propose that a part of a word assembly is embedded in a neural architecture for sentence structure given by "structure" assemblies and their relations. A word assembly can be associated ("bound") temporarily to a given structure assembly so that it is (temporarily) "tagged" by the structure assembly to which it is bound. A word assembly can be bound simultaneously to two or more structure assemblies. The different structure assemblies provide different "tags" for the word assembly that distinguish between different "copies" of the word encoded with the word assembly. However, the word assembly itself is not "copied" or disrupted in this process, and its associations and relations remain intact when a word assembly is tagged by a given structure assembly. Hence any "copy" of a word is always "grounded" (as discussed in sect. 2.2.1).

Structure assemblies are selective. For example, nouns and verbs bind to different kinds of structure assemblies. Furthermore, the internal structure of structure assemblies allows selective activation of specific parts within each structure assembly. Structure assemblies of a given kind can selectively bind temporarily to specific other structure assemblies so that a (temporal) neural structure of a given sentence is created. Hence structure assemblies can encode different instantiations of the same word assembly (solving the "problem of 2") and they can bind word assemblies in line with the syntactic structure of the sentence.

Binding in the architecture occurs between word assemblies and structure assemblies and between structure assemblies. Binding between two assemblies derives from sustained ("delay") activity in a connection structure that links the two assemblies. This activity is initiated when the two assemblies are concurrently active. The delay activity is similar to the sustained activation found in the "delay period" in working-memory experiments (e.g., Durstewitz et al. 2000). Two assemblies are bound together as long as this delay activity continues.

Figure 2 illustrates the basic neural structure in the architecture of *cat chases mouse*. The structure consists of the word assemblies of *cat*, *mouse*, and *chases* and structure assemblies for noun phrases (NPs) and verb phrases (VPs), together with "gating circuits" and "memory circuits." Gating circuits are used to activate selectively specific parts within structure assemblies. Memory circuits are used to bind two assemblies together temporarily.

The assemblies for *cat* and *mouse* are bound to two different NP assemblies (N_1 and N_2), and the assembly for *chases* is bound to a VP assembly (V_1). The structure assemblies are bound to each other to encode the verb-argument structure of the sentence. For this purpose, each structure assembly is composed of a main assembly (N_i for NP assemblies and V_i for VP assemblies) and one

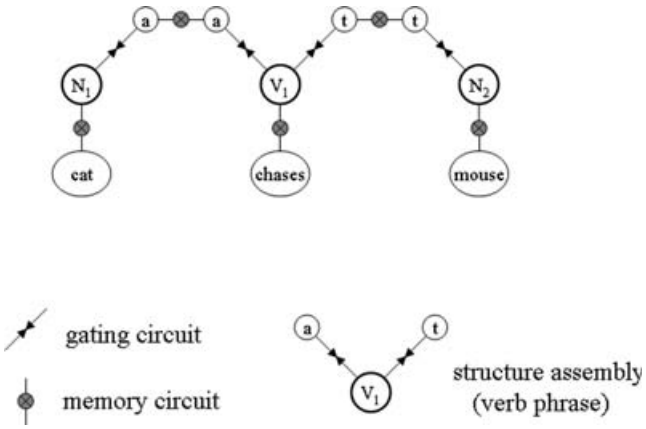


Figure 2. Illustration of the neural sentence structure of *cat chases mouse* in the neural blackboard architecture presented here. The words are encoded with the word assemblies illustrated in Figure 1 (sect. 2.1). Sentence structure is encoded with "structure assemblies" for noun-phrases (NP assemblies) and verb-phrases (VP assemblies). A structure assembly consists of a main assembly and a number of subassemblies connected to the main assembly by means of gating circuits. The labeled subassemblies represent the thematic roles of agent (a) and theme (t). Binding between assemblies is achieved with active memory circuits. Here the assembly for *cat* is bound to the NP assembly N_1, the assembly for *chases* is bound to the VP assembly V_1, and the assembly for *mouse* is bound to the NP assembly N_2. N_1 and V_1 are bound by means of their agent subassemblies, and V_1 and N_2 are bound by means of their theme subassemblies.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

or more subassemblies. In Figure 2, the NP and VP assemblies have subassemblies for agent (a) and theme (t).[6] To encode *cat* as the agent of *chases*, N_1 is bound to V_1 with their agent subassemblies. In turn, N_2 is bound to V_1 with their theme subassemblies, encoding *mouse* as the theme of *chases*.

Main assemblies and subassemblies also have the ability of reverberating ("delay") activity so that they remain active for a while after they have been activated. Subassemblies are connected to main assemblies with gating circuits, which control the flow of activation within structure assemblies. For example, a main assembly can be active but its subassemblies not. Control of activation in structure assemblies is of crucial importance in the architecture. Before illustrating this in more detail, we discuss the gating and memory circuits in the architecture.

6.1. Gating and memory circuits

A gating circuit consists of a disinhibition circuit (e.g., Gonchar and Burkhalter 1999). Figure 3 (left) illustrates a gating circuit in the direction from assembly *X* to assembly *Y*. The circuit controls the flow of activation by means of an external control signal. If *X* is active, it activates an inhibition neuron i_x, which inhibits the flow of activation from *X* to X_{out}. When i_x is inhibited by another inhibition neuron (I_x) activated by an external control signal, *X* activates X_{out}, and X_{out} activates *Y*. A gating circuit from *Y* to *X* operates in the same way. Control of activation can be direction-specific. With a control signal in the direction from *X* to *Y*, activation will flow in this direction (if *X* is active) but not in the direction from *Y* to *X*. The symbol in Figure 3 (left) will be used to represent the combination of gating circuits in both directions (as in Figure 2).

A memory circuit consists of a gating circuit in which the control signal results from a "delay" assembly. Figure 3 (right) illustrates a memory circuit in the direction of *X* to *Y*. However, each memory circuit in the architecture consists of two such circuits in both directions (*X* to *Y* and *Y* to *X*). The delay assembly (which controls the flow of activation in both directions) is activated when *X* and *Y* are active simultaneously (see below) and it remains active for a while (even when *X* and *Y* are no longer active) because of the reverberating nature of the activation in this assembly.

A memory circuit has two possible states: active and inactive. Each state will be represented with the symbol in Figure 3 (right). If the memory circuit is inactive, activation cannot flow between the assemblies the circuit connects. If the memory circuit is active, activation will flow between the assemblies the circuit connects if one of these assemblies is active. In this way, an active memory circuit binds the two assemblies it connects. This binding lasts as long as the activation of the delay assembly in the memory circuit. The memory circuits in Figure 2 are active, binding word assemblies and structure assemblies (temporarily) in line with the sentence structure.

6.2. Overview of the blackboard architecture

Figure 4 illustrates the part of the architecture in which nouns can bind as arguments to verbs (Fig. 2). This part is illustrative of the overall architecture.

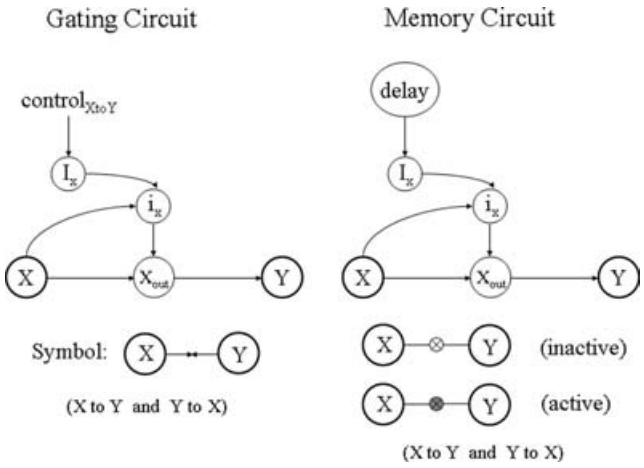


Figure 3. Left: A gating circuit in the direction from assembly *X* to assembly *Y*, based on a disinhibition circuit. The circles depict neural assemblies, except for the circles labeled with I_x or i_x. These circles depict (groups of) inhibitory neurons. A combination of two gating circuits in the directions *X* to *Y* and *Y* to *X* is depicted in other figures with the symbol illustrated at the bottom. Right: A memory (gating) circuit in the direction from assembly *X* to assembly *Y*, based on a gating circuit with a delay assembly for control. A combination of two memory circuits in the directions *X* to *Y* and *Y* to *X* is depicted in other figures with the symbols illustrated at the bottom, one for the inactive state and one for the active state of this combined memory circuit.

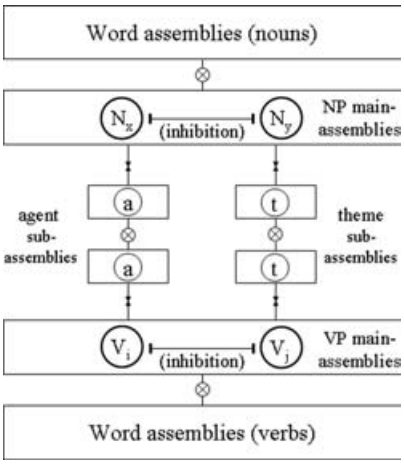


Figure 4. A neural blackboard architecture for verb-argument binding. Word assemblies for verbs are connected to the main assemblies of VP structure assemblies by means of (initially) inactive memory circuits. Word assemblies for nouns are connected to the main assemblies of NP structure assemblies by means of (initially) inactive memory circuits. The agent (a) and theme (t) subassemblies of the VP and NP structure assemblies are connected by means of (initially) inactive memory circuits. Only subassemblies of the same kind are connected to each other. VP main assemblies are connected to a population of inhibitory neurons that can initiate competition between the VP main assemblies; the same applies to NP main assemblies.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

Each noun (word) assembly is connected to the main assembly of each NP assembly with a memory circuit (initially inactive). Likewise, each verb (word) assembly is connected to the main assembly of each VP assembly with a memory circuit (initially inactive). Main assemblies of the same kind are mutually inhibitory. Each NP and VP main assembly is connected to a number of subassemblies with gating circuits. The gating circuits can be selectively activated by neural control circuits (not shown). For example, the gating circuits between the main assemblies and the agent subassemblies can be activated without activating the gating circuits for the theme subassemblies. Finally, all subassemblies of the same kind are connected through memory circuits. For example, each agent subassembly of the NP assemblies is connected to each agent subassembly of the VP assemblies with a memory circuit (initially inactive).

A new NP assembly will be activated when a new noun in a sentence is processed. The NP assembly is arbitrary but "free," that is, not already bound to a sentence structure (i.e., all its memory circuits are inactive).[7] The active NP assembly will remain active until a new NP assembly is activated by the occurrence of a new noun in the sentence.[8] The selection of a VP assembly is similar.

When several structure assemblies have been activated, the ones activated first will return to the inactive state because of the decay of delay activity in their memory circuits. In this way, only a subset of the structure assemblies will be concurrently active and "free" structure assemblies will always be available. As a result, a limited set of VP assemblies and NP assemblies is needed in this architecture.

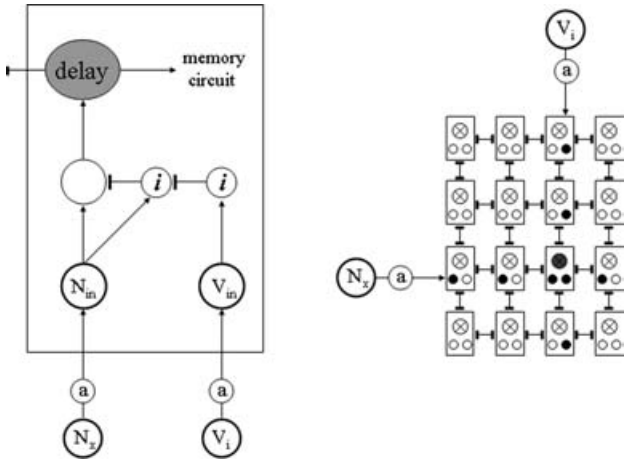


Figure 5. Connection structure for the agent subassemblies in Figure 4. Left: a delay assembly in a memory circuit (Fig. 3, right) is activated when the subassemblies connected by the memory circuit are concurrently active (using a disinhibition circuit). Right: Each agent subassembly of all NP assemblies is connected to each agent subassembly of all VP assemblies with a specific "column" in an array of columns. Each column consists of the memory circuits that connect both subassemblies, together with the circuit in on the left. The active subassembly of N_x will activate all N_{in} neurons in its horizontal row of columns. In the same way, the active subassembly of V_i will activate all V_{in} neurons in its vertical row of columns. This results in the activation of the delay assembly in the memory circuit in the corresponding column of N_x and V_i. In this way, the connection structure instantiates a logical AND operation. Columns in horizontal and vertical rows are mutually inhibitory. Inhibition is initiated by active delay assemblies in the memory circuits.

6.2.1. Connection structure for binding in the architecture. Figure 5 (right) illustrates that the connection structure between the agent subassemblies in Figure 4 consists of a matrix-like array of "columns." Each column contains a memory circuit (in both directions) and the delay assembly that can activate the memory circuit. Each column also contains a circuit to activate the delay assembly (Fig. 5, left). This circuit is a disinhibition circuit that activates the delay assembly if the neurons N_{in} and V_{in} are active at the same time. These neurons are activated by the respective agent subassemblies of an NP assembly and a VP assembly.

In the structure of *cat chases mouse* (Fig. 2), the NP assembly for *cat* (N_1) is bound to the VP assembly for *chases* (V_1) with their agent subassemblies. This binding process is illustrated in Figure 5. The activated agent subassembly of the (arbitrary) NP assembly N_x activates the N_{in} neurons in a horizontal row of columns. Likewise, the activated agent subassembly of the (arbitrary) VP assembly V_i activates the V_{in} neurons in a vertical row of columns. The delay assembly in the column on the intersection of both rows will be activated if the agent subassemblies of N_x and V_i are active simultaneously, which results in the binding of these agent subassemblies.

The columns within each horizontal and vertical row (Fig. 5, right) are mutually inhibitory. Inhibition is initiated by the active delay assemblies (Fig. 5, left).[9] Hence, when the agent subassemblies of N_x and V_i are bound by an active memory circuit, the active delay assembly in their mutual column inhibits all columns in the same horizontal and vertical row. This prevents a second binding of N_x with another VP assembly or of V_i with another NP assembly, with agent subassemblies.

The connection structure in Figure 5 is illustrative of every connection structure in the architecture in which assemblies are (temporarily) bound, including the binding of V_1 and N_2 (Fig. 2) with their theme subassemblies.

In the binding process of the sentence in Figure 2, the assembly for *cat* is bound to an arbitrary ("free") NP assembly by the activated memory circuit that connects the two assemblies. Likewise, the assembly for *chases* is bound to a VP assembly. The binding of *cat* as the agent of *chases* results from activating the gating circuits between the NP and VP main assemblies and their agent subassemblies. The active NP and VP main assemblies (N_1 for *cat* and V_1 for *chases*) will then activate their agent subassemblies, which results in the binding together of these two agent subassemblies (as illustrated in Fig. 5).

Gating circuits will be activated by neural control circuits. These circuits instantiate syntactic (parsing) operations based on the active word assemblies and the activation state of the blackboard. In the case of *cat chases mouse*, these circuits will detect that in *cat chases* (or *N-V*), *cat* is the agent of the verb *chases*. In response, they will activate the gating circuits for the agent subassemblies of all NPs and VPs. The binding of *mouse* as the theme of *chases* proceeds in a similar manner. We present an example of a control circuit in section 6.8.4 below.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

6.2.2. The effect of gating and memory circuits in the architecture. When a memory circuit is active, activation can flow between the two assemblies it connects (Fig. 3, right). The two connected assemblies are then temporarily associated, or "merged," into a single assembly. Figure 6a illustrates the merging of assemblies for the structure of *The cat chases the mouse* (Fig. 2). In Figure 6a, the word assemblies are directly connected (merged) with the main assemblies of their structure assemblies. Likewise, the agent subassemblies and theme subassemblies are merged into single assemblies (one for agent and one for theme). The resulting structure shows that the backbone of a neural sentence structure in this architecture is given by the gating circuits.

Figure 6b illustrates what happens if the gating circuits are removed. Subassemblies (agent, theme) are now also directly merged with their main assemblies, and structure assemblies of a different kind are directly connected through associative links. As illustrated in Figure 6c, the neural sentence structure in Figures 2 and 6a is hence reduced to a single merged assembly that results from directly associating each of the assemblies involved in the original sentence structure of Figures 2 and 6a. In particular, Figure 6c shows that the word assemblies for *cat*, *chases*, and *mouse* are now directly associated, so that the distinction between the sentences *The cat chases the mouse* and *The mouse chases the cat* is lost.

With the use of gating circuits, the neural blackboard architecture for sentence structure can solve the "four challenges for cognitive neuroscience" presented by Jackendoff (2002) (see sect. 2), as discussed below.

6.3. Multiple instantiation and binding in the architecture

Figure 7a (left, right) illustrates the neural structures of the sentences *The cat chases the mouse*, *The mouse chases the cat*, and *The cat bites the dog* in the neural blackboard architecture (in the manner of Fig. 6a). The words *cat*, *mouse*, and *chases* occur in more than one

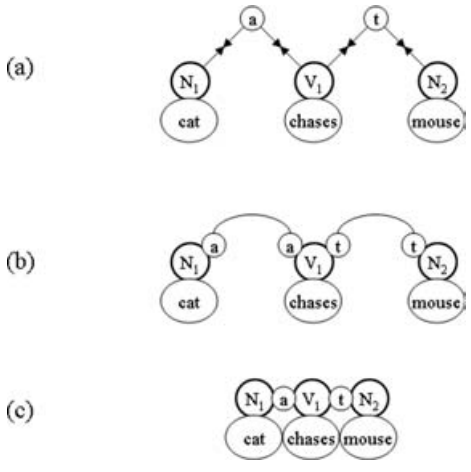


Figure 6. a. The structure of *The cat chases the mouse* in Figure 2, represented by merging the assemblies that are connected through active memory circuits. b. The structure of *The cat chases the mouse* in Figure 2 that results when the gating circuits are removed. c. The structure in Figure 6b with the merging of the assemblies involved.

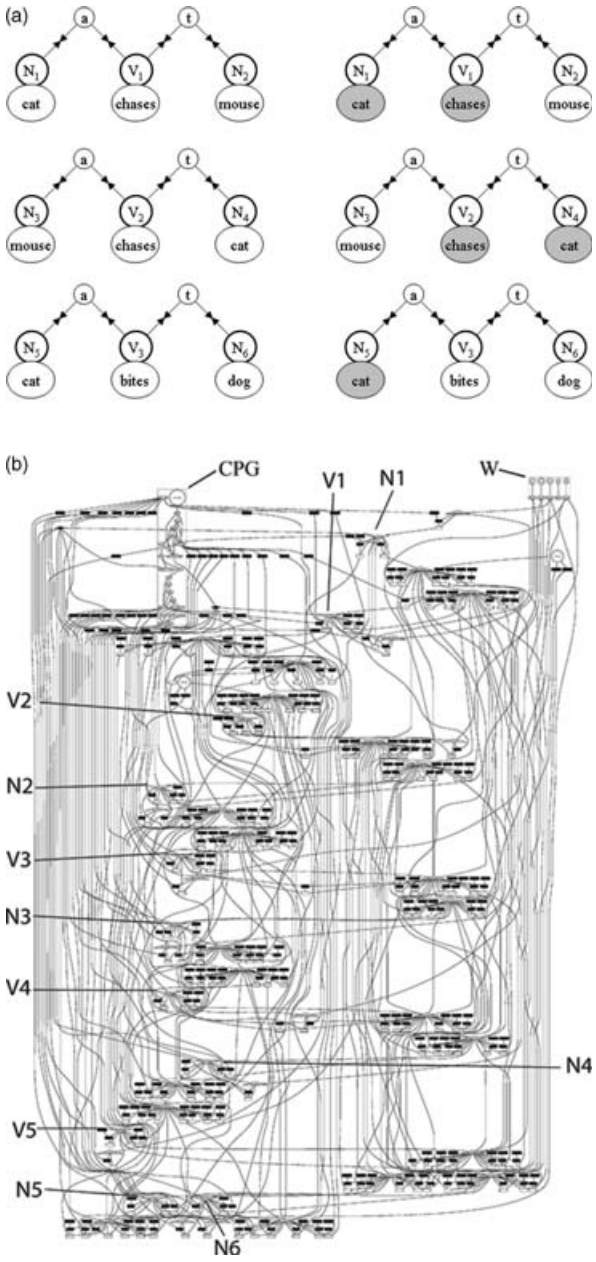


Figure 7. a. Left: Combined instantiation of the sentences *cat chases mouse*, *mouse chases cat*, and *cat bites dog* in the architecture illustrated in Figure 4. The multiple instantiations of *cat*, *chases*, and *mouse* in different sentences (and in different thematic roles) are distinguished by the different NP or VP structure assemblies to which they are bound. Right: The activation of the word assembly for *cat* and the word assembly for *chases* due to the question "Whom does the cat chase?" b. Network of the sentence structures in Figure 7a. Neurons in gating and memory circuits are represented by black boxes (inhibitory neurons) and white boxes (excitatory neurons). Main assemblies (noun or verb) are represented as inverted triangles. Subassemblies (agent, theme) are represented as upright triangles. The circles represent the input (control) for the gating circuits, which activate a gating circuit. These include the delay assemblies in the memory circuits. Their activity is constant during simulation. The network also contains inhibitory populations that initiate winner-takes-all competition between verb (noun) assemblies. The labels V_1 to V_3 and N_1 to N_6 refer to the VP and NP assemblies in Figure 3. V_4 and V_5 are two "free" VP assemblies. W refers to word assemblies. CPG refers to the central pattern generator.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

sentence, which creates the problem of multiple instantiation (the problem of 2) for their word assemblies.

Figure 7a shows that this problem is solved by the use of structure assemblies. For example, the word assembly for *cat* is bound to the NP assemblies N_1, N_4, and N_5. Similarly, different VP assemblies (V_1 and V_2) encode the verb *chases* in different sentences. In this way, *cat* can be the agent of *chases* in one sentence (binding N_1 and V_1 with their agent subassemblies) and the theme of *chases* in another sentence (binding N_4 and V_2 with their theme subassemblies). Furthermore, *cat* can also be the agent of another verb (*bites*) in a third sentence, using N_5.

The internal structure of the NP and VP assemblies given by the gating circuits is of crucial importance. Without this internal structure, the neural structures in Figure 7a would collapse into direct associations between neural assemblies, which would result in a failure to distinguish between, for example, *The cat chases the mouse* and *The mouse chases the cat* (as illustrated in Fig. 6b and c). Using the gating circuits, the neural structures of these two sentences can be selectively (re)activated.

6.3.1. Answering binding questions. Selective reactivation of a sentence structure in Figure 7a is necessary to retrieve information from the blackboard architecture, that is, to answer specific binding questions such as "Whom does the cat chase?" This question provides the information that *cat* is the agent of *chases*, which activates the assemblies for *cat* and *chases* (Fig. 7a, right) and the gating circuit for agent. Furthermore, the question asks for the theme of *chases* (i.e., *x* in *cat chases x*).

The answer is produced by a competition process between the VP assemblies in which V_1 emerges as the winner. Figure 7a (right) shows that V_1 is activated by *chases* and N_1 (through *cat*), whereas V_2 is activated only by *chases* and V_3 is activated only by N_5 (through *cat*). This results in V_1 as the winner of the VP competition. The activation of the gating circuits for theme will then result in the activation of N_2 by V_1, and hence in the activation of *mouse* as the answer to the question.

In contrast, the question "Who chases the cat?" will result in a VP competition in which V_2 is the winner. The difference results from the selective activation of the gating circuits. Both questions activate the assemblies for *cat* and *chases*, but they activate different gating circuits. The first question defines *cat* as the agent of *chases*, which produces the activation of the gating circuits for agent. The second question defines *cat* as the theme of *chases*, which activates the theme-gating circuits so that N_4 (activated by *cat*) can activate V_2. This route of activation was blocked in case of the first question. With the second question, V_2 emerges as the winner because it receives the most activation. Then *mouse* can be produced as the answer, because the question asks for the agent of *chases* (i.e., *x* in *x chases cat*).

6.3.2. Simulation of the blackboard architecture. We have simulated the answer of "Whom does the cat chase?" with the sentences in Figure 7a stored simultaneously in the architecture. The simulation was based on the dynamics of spiking neuron populations (i.e., average neuron activity). In all, 641 interconnected populations were simulated, representing the word assemblies, main assemblies, subassemblies, gating circuits, and memory circuits used to encode the sentences in Figure 7a. The 641 populations evolved simultaneously during the simulation. Appendix A1 provides further details of the simulation.

An overview of the network as simulated is presented in Figure 7b. We used a visual tool (*dot* by Gansner et al. 1993) to represent the network. The program *dot* aims to place nodes (neurons) at a reasonable distance from each other and to minimize the number of edge (connection) crossings. The network presented in Figure 7b is the same as the network presented in Figure 7a (i.e., the three sentence structures). Both networks can be converted into each other by successively inserting the structures presented in Figures 3, 4, and 5 into the structures presented in Figure 7a.

Figure 8 (left, middle) illustrates the simulation of the network with activation of the VP and NP assemblies labeled in Figure 7b. The Figure also shows (middle) two "free" VP main assemblies (V_4, V_5), not used in the sentence encoding of Figure 7a, to illustrate the activation of free assemblies in this process. The simulation starts at $t = 0$ ms. Before that time, the only active assemblies are the delay assemblies in the memory circuits (as in Fig. 2).

The question "Whom does the cat chase?" provides information that *cat* is the agent of *chases* and it asks for the theme of the sentence *cat chases x*. The production of the answer consists of the selective activation of the word assembly for *mouse*. Backtracking (see Fig. 7a), this requires the selective activation of the NP main assembly N_2, the theme subassemblies for N_2 and V_1, and the VP main assembly V_1 (in reversed order).

This process proceeds in a sequence of steps. We used a central pattern generator (CPG) to control the sequence. Basically, a CPG consists of a sequence of neurons in which a neuron is active for a while before activation is passed on to the next neuron in the sequence. CPGs can be used in motion control (e.g., Kling and Szekély 1968). Forms of sequential motion control could provide a

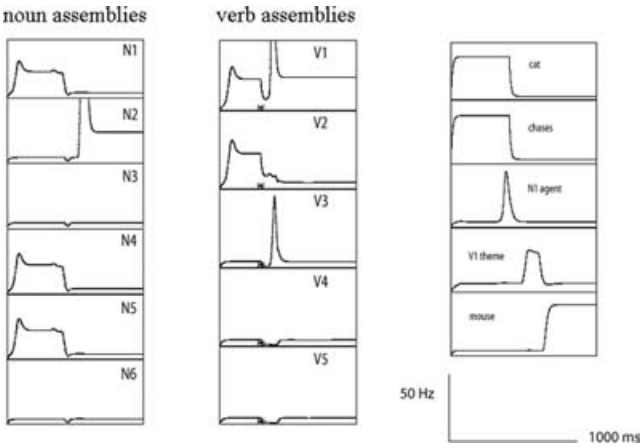


Figure 8. Activation of the neural assemblies in Figure 7a and 7b (in Hz/ms). Left panel: The NP main assemblies N_1 to N_6. Middle panel: The VP main assemblies V_1 to V_5. Right panel: The word assemblies for *cat*, *chases*, and *mouse* and the subassemblies for N_1 agent and V_1 theme. The asterisks indicate the onset of the control sequence that initiates the competition process.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

basis for analogous functions in language as well (e.g., Dominey 1997). We assume that a question of the type *cat chases x?* activates a specific (type-related) CPG. That is, the same CPG will be initiated with any question of the form *noun verb x?*. The CPG used consisted of 15 neurons, each one active for 25 ms. This provides 15 time steps of 25 ms. The CPG was activated at t = 300 ms (indicated with the asterisks in Fig. 8). It initiated the following sequence: activate VP competition (step 1–5), activate agent-gating circuits (step 1–4), inhibit word assemblies (step 5), activate NP competition (step 6), activate theme-gating circuits for VP assemblies (step 8–11), and activate theme-gating circuits for NP assemblies (step 13–15).

Figure 8 shows the activation of the assemblies for *cat* and *chases* (beginning at t = 0 ms). To produce the selective activation of the word assembly for *mouse*, other word assemblies cannot be active at that moment. Therefore word assemblies are inhibited after a certain time (step 5). The activation of *cat* results in the activation of the NP main assemblies N_1, N_4, and N_5 (Fig. 8, left), and the activation of *chases* results in the activation of the VP main assemblies V_1 and V_2 (Fig. 8, middle). As long as V_1 and V_2 are both active, the question "Whom does the cat chase?" cannot be answered. To produce the answer, the gating circuits for the theme subassemblies of the VP assemblies have to be activated, because the question asks for the theme of *cat chases x*. However, when both V_1 and V_2 are active, this will result in the activation of the theme subassemblies for V_1 and V_2 and, in turn, of *mouse* and *cat* (via N_2 and N_4) as the answer to the question. Therefore, to produce *mouse* as the only answer to the question, a winner-takes-all (WTA) competition between V_1 and V_2 has to occur, with V_1 as the winner.

The competition process between the VP assemblies proceeds as follows. VP main assemblies are connected to a population of inhibitory neurons. The competition between the VP assemblies is initiated by activating this population. The competition between the VP assemblies is decided by activating the gating circuits for the agent subassemblies. This results in the activation of the agent subassemblies of N_1, N_4, and N_5 because they are the active NP assemblies (Fig. 8, left). The activation of the N_1-agent subassembly is illustrated in Figure 8 (right).

The active agent subassemblies of N_1 and N_5 are bound to the VP assemblies V_1 and V_3 respectively (see Fig. 7a). Hence the VP assemblies V_1 and V_3 receive activation from these NP assemblies when the "agent"-gating circuits are activated. (The agent subassembly of N_4 is not bound to a VP assembly, because N_4 is bound to a VP assembly with its theme subassembly; see Fig. 7a). As a result, V_1 wins the competition between the VP assemblies, because V_1 receives activation from *chases* and N_1, whereas V_2 receives activation only from *chases*, and V_3 receives activation only from N_5. (The initial drop in activation of V_1 results from the fact that verb competition and activation of the agent-gating circuits start at the same moment, but the activity from N_1 has to pass through the gating circuit to reach V_1). Figure 8 (middle) shows that V_1 is the only active VP assembly after this competition process. After a transient period, the activation of V_1 is given by its delay (reverberating) activation. The activation of V_2 and V_3 is reduced to the level of the "free" assemblies V_4 and V_5.

When V_1 remains as the only active VP assembly, the answer *mouse* can be produced by activating the theme-gating circuits. This will produce the selective activation of N_2, which is the NP assembly bound to *mouse* in Figure 7a, provided that the active NP main assemblies (N_1, N_4, and N_5 in Fig. 8) are inhibited first (step 6 of the CPG). After the inhibition of the active NP assemblies, the theme-gating circuits can be activated. As a result, the theme subassembly of V_1 and the main assembly N_2 are now selectively activated as well. Finally, the word assembly for *mouse* will be activated (Fig. 8, right).

The dynamics of the network is straightforward, despite its apparent complexity. The simulation of the network in Figure 7b covers all aspects of the dynamic interaction in the blackboard that is needed to answer binding questions. This results from the modular nature of the architecture and the chainlike nature of sentences structures. Only assemblies of the same kind interact in the architecture, for example, VP assemblies with VP assemblies and NP assemblies with NP assemblies. The same is true for other types of structure assemblies that will be introduced. Furthermore, different kinds of assemblies interact only through the specific subassemblies with which they are connected (i.e., temporarily bound) and which are selectively activated in the interaction process. Each of these bindings in the architecture is of the same kind. That is, with a specific subassembly (e.g., theme), a structure assembly of a given kind (e.g., VP) can bind (and hence interact) with only one other structure assembly, which has to be of a different kind (e.g., NP). These interactions have also been simulated with the network in Figure 7b.

Figures 7 and 8 illustrate that the neural blackboard architecture can solve the binding problem in language on the level of verb-argument binding. However, extensions of the neural blackboard architecture presented so far are needed to handle the more massive form of binding found in linguistic structures (Jackendoff 2002). For the remainder of the paper we use structures like those in Figure 7a to discuss the encoding of sentence structures in the architecture. However, each of the structures we present can be transformed into a network as illustrated in Figure 7b.[10]

6.4. Extending the blackboard architecture

A first extension of the architecture is introduced in Figure 9. As Bickerton argued (Calvin & Bickerton 2000), an important step in the evolution of language consisted of the transformation illustrated in the top half of Figure 9. The left diagram represents a sentence structure in protolanguage. The diagram on the right represents a basic sentence structure in modern language. One argument of the verb is placed outside the verb's direct influence (i.e., the verb phrase), in a controlling position of its own (as the subject).

The bottom half of Figure 9 shows a similar transition in terms of our neural architecture. The left structure is the structure of *cat chases mouse* of Figure 2. For convenience, we have introduced a (further) shorthand presentation of this structure in Figure 9. As in Figures 6a and 7, memory circuits are not shown, and bounded subassemblies are presented as one. Here the gating circuits are not shown as well, and words are simply written close to their

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

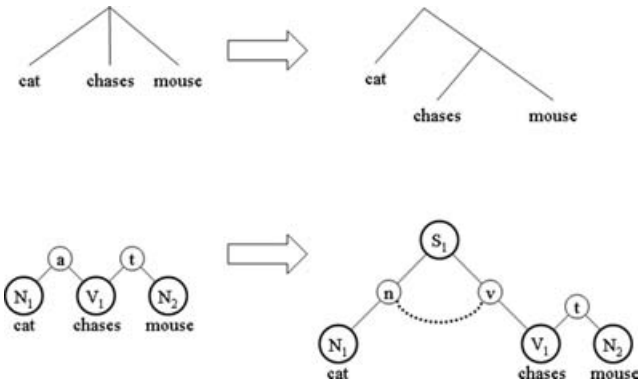


Figure 9. Top: Transformation of sentence structure in protolanguage (left) to sentence structure in modern language (right), after Calvin & Bickerton (2000). Bottom: similar transformation in terms of neural sentence structures. The neural sentence structure of *cat chases mouse* on the left is the same as in Figure 2, but in a shorthand presentation. The neural sentence structure of *cat chases mouse* on the right (also in shorthand presentation) consists of a new structure assembly for sentence (S), with subassemblies for noun (n) and verb (v). The dotted line between the noun and verb subassemblies represents the possibility of encoding agreement between subject and verb by means of these subassemblies.

structure assemblies.[11] However, the full structure of Figure 2 is still implied. The shorthand version does not result in ambiguities: subassemblies are always connected to their main assemblies with gating circuits, subassemblies are always bound to other subassemblies with active memory circuits, and word assemblies are always bound to structure assemblies of the corresponding type (e.g., nouns to NP assemblies).

The left sentence structure in the bottom half of Figure 9 resembles the left diagram in the top half of the Figure. In turn, the sentence structure on the right in the bottom half of the Figure (also in shorthand presentation) resembles the diagram on the right in the top half of the Figure. In this sentence structure, the NP of *cat* is not directly bound to the VP of *chases*. Instead, it is bound to a new "sentence" structure assembly (S). Binding is achieved through the noun subassembly (n) of the NP assembly (not shown in Fig. 4) and the corresponding noun subassembly of the S assembly. Likewise, the VP assembly is bound to S with verb subassemblies (v).

The connection structures of the noun subassemblies and the verb subassemblies are similar to the connection structure illustrated in Figure 5. Furthermore, the S main assemblies can inhibit each other as the NP and VP main assemblies can (Fig. 4). This does not mean that only one sentence at a time could be stored in the blackboard. As illustrated in Figure 2, information is stored in the blackboard by means of active delay assemblies, which are not mutually inhibitory (except in the manner illustrated in Fig. 5). The inhibitory interaction between main assemblies of the same kind is needed when information is stored or retrieved, as in answering binding questions (Fig. 8).

The dotted line between the noun and verb subassemblies in Figure 9 indicates that these subassemblies can be used to encode agreement between the subject *cat* and the verb *chases* (as in *cat chases* versus *cats chase*). For example, S assemblies could have different noun and verb subassemblies for single and plural, which can be activated selectively. Once a noun is bound to the noun subassembly for single, this subassembly will enforce a binding of the verb to a verb subassembly for single as well.

Further extensions of the architecture proceed along similar lines. They consist of the introduction of new structure assemblies and new subassemblies needed for appropriate binding. New assemblies can be added because of the modular structure of the architecture.

6.4.1. The modular nature of the blackboard architecture. The modular nature of the blackboard architecture is illustrated in Figure 10 with the structure for Jackendoff's (2002) sentence *The little star is beside a big star*. The new structure assemblies here are determiner assemblies (D_1 and D_2), adjective phrase assemblies (Adj_1 and Adj_2), and prepositional phrase assemblies (P_1). The D_i assemblies are bound to NP assemblies with determiner subassemblies (d), the Adj_i assemblies are bound to NP assemblies with adjective subassemblies (adj), and the P_i assemblies are bound to VP assemblies with preposition-verb subassemblies (pv) and to NP assemblies with preposition-noun subassemblies (pn). The connection structure of each of these new kinds of subassemblies is again similar to the connection structure in Figure 5. Main assemblies of the same kind are again mutually inhibitory.

This sentence structure again illustrates the solution of the problem of 2 provided by the architecture, and it illustrates the solution of the massiveness of the binding problem in linguistic structures. The word assembly for *star* can participate in two different constituents of the sentence because it is bound to two different NP assemblies. Bound to N_1, *star* has the determiner *the* and the adjective *little*, and it constitutes the subject of the sentence, bound directly to S. Bound to N_2, *star* has the determiner *a* and the adjective *big*, and it is bound to the preposition *beside*, which is bound to the verb of the sentence.

Questions can be again be answered by selectively activating structure assemblies and gating circuits. For example, the question "Which star is beside a big star?" can be answered if S_1 is activated together with the gating circuits for the noun subassemblies (the question asks for the adjective of the subject). When N_1 is activated, D_1 and Adj_1 can also be activated, which produces the answer *the little star*. S_1 will be activated because of the information *is beside a big star* provided by the question. The phrase *a big star* activates N_2, which hence initially wins the competition over N_1. However, after the selection of S_1, N_1 will be activated because of the activation of the "subject"-gating circuits. Conversely, the question "Where is the little star?" produces the activation of S_1 and V_1, and it asks for the prepositional phrase of the sentence. The answer will result from activating the gating circuits for the preposition-verb subassemblies.

The sentence structure in Figure 10 raises the question of how many different kinds of structure assemblies would be needed in the neural blackboard architecture. A preliminary answer is that the architecture would have a particular kind of structure assembly for each kind of constituent that can occur in a linguistic structure. In section

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

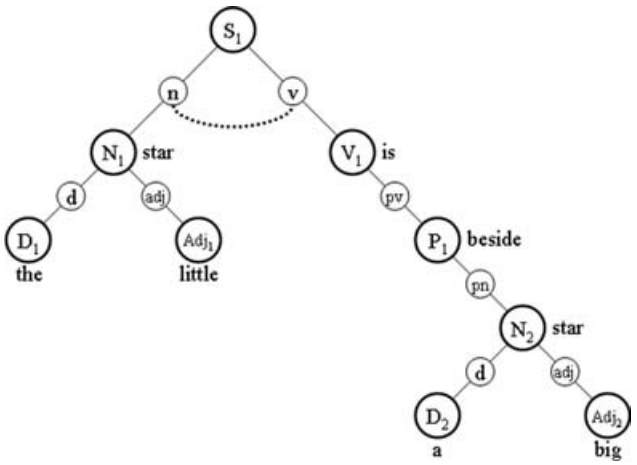


Figure 10. Neural sentence structure for *The little star is beside a big star*. The structure assemblies are similar to those in Figure 9 (bottom right), with new structure assemblies for determiner (D_i), adjective phrase (Adj_i) and prepositional phrase (P_i), and new subassemblies for determiner (d), adjective (adj), preposition-verb (pv) and preposition-noun (pn).

6.8.1 below, we illustrate this point with the encoding of embedded clauses. First, however, the two remaining problems presented by Jackendoff (2002) have to be solved: the problem of variables and the problem of how a combinatorial structure encoded in neural activity can be stored in long-term memory (i.e., constituent binding with activation versus constituent binding with synaptic modification). We begin with the latter problem.

6.5. Constituent binding in long-term memory

An important role in the process of storing information in long-term memory is played by the hippocampus and surrounding areas (hippocampal complex; Nadel & Moscovitch 2001). The hippocampal complex (HC) has the ability for rapid storage of information by means of synaptic modifications (Rolls & Treves 1998), depending on, for example, long-term potentiation (LTP).

In the view of the "Hebb-Marr" model (McNaughton & Nadel 1990), HC neurons form a conjunctive encoding of neurons that are concurrently active in the cortex (e.g., O'Reilly & Rudy 2001; Rolls & Treves 1998). The encoding results from the modification of the synapses between the active neurons in the cortex and the active neurons in the HC. Combined, the neurons form an autoassociator (Marr) or a cell assembly (Hebb) that can be reactivated as a whole after activating a part of it. In this way, the HC forms a "snapshot-like" memory of an event with a duration of about a second (Rolls & Treves 1998). Given the "sparse connectivity" structure of the HC, different events or episodes can be separated in memory because they can be encoded with different and nonoverlapping groups of neurons in the HC (O'Reilly & Rudy 2001).

A crucial aspect of encoding with HC neurons is the unstructured nature of the information stored (Roll & Treves 1998); that is, the HC acts as a simple binding device forming a conjunctive encoding of the input that is concurrently available. The HC does not, by itself, encode systematic relations within the input (O'Reilly & Rudy 2001). Therefore, as described by O'Reilly & Rudy (2001, p. 320): "all relationship information must be present in the inputs to the hippocampus, which can then bind together the relational information with other information about the related items in a conjunction."

Figure 11 (left) illustrates what this means in terms of the word assemblies activated (within a second or so) by the sentence *The cat chases the mouse*. The HC will form a conjunctive encoding of the word assemblies but not of their relations.[12] The same conjunctive encoding of the word assemblies will be formed with the sentence *The mouse chases the cat*. Hence HC conjunctive encoding of word assemblies creates the familiar binding problem. Reactivation by the HC will reactivate the word assemblies for *cat*, *mouse*, and *chases* but not the structure of either sentence.

The problem can be solved by including relationship information in the input to the HC, as described by O'Reilly & Rudy (2001). This will occur if the activity in the neural blackboard architecture is included in the input to the HC (Fig. 11, right). In this way, the HC can reactivate a neural sentence structure by reactivating the neural blackboard. Figure 11 (right) illustrates that a neural blackboard architecture plays a crucial role in the process of storing combinatorial structures in long-term memory (i.e., in terms of synaptic modification). Even a conjunctive encoding as provided by the HC is sufficient if the activity in the blackboard is included in the encoding.

With longer sentences, the HC will encode the sentence structure in terms of a sequence of events, each consisting of a conjunctive encoding of a part of the sentence structure. Figure 12 illustrates this process for the structure of *The little star is beside a big star* presented in Figure 10. Figure 12 also illustrates that encoding in the HC will be a form of distributed encoding. Here two partly overlapping sets of HC neurons encode two different parts of the sentence, which could also be partly overlapping. The whole sentence structure can be reactivated if some of the HC neurons reactivate the part of the sentence structure they encode. The overlap between the HC encoding and the two sentence structures can then result in the activation of the remaining part of the sentence structure.

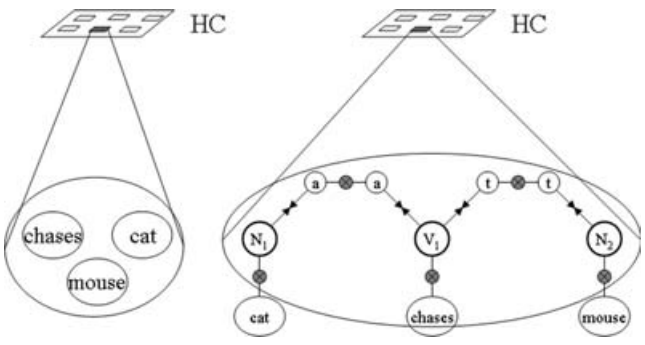


Figure 11. Left: Conjunctive encoding of the assemblies for *cat*, *chases*, and *mouse* with a neuron (or group of neurons) in the hippocampal complex (HC). Right: Conjunctive encoding of the neural sentence structure of *cat chases mouse* with a neuron (or group of neurons) in the HC.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

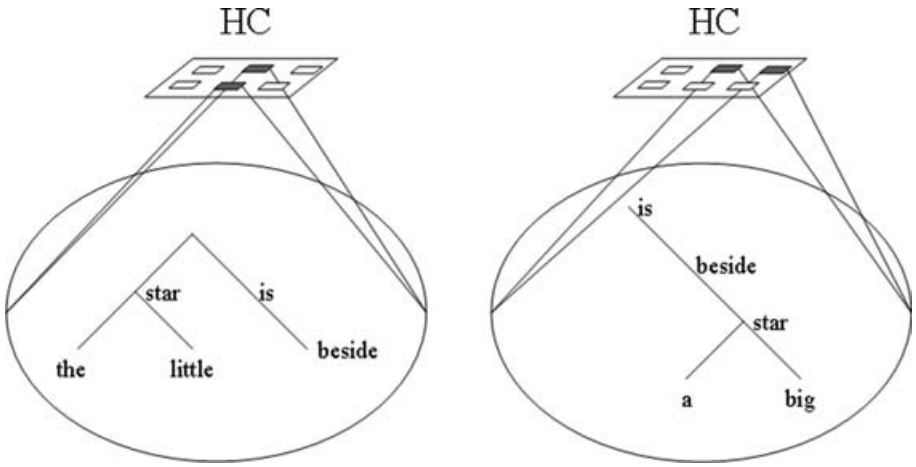


Figure 12. Encoding of the neural sentence structure of *The little star is beside a big star* (Fig. 10) with partly overlapping sets of neurons in the hippocampal complex (HC). Each set of neuron encodes a part ("episode") of the sentence structure. Both parts can be overlapping.

6.5.1. One-trial learning. In particular, the activity of the delay assemblies in the memory circuits has to be included in the input to the HC, because the structure of a sentence is completely determined by the set of active delay assemblies. In fact, as hinted at in Figure 11 (right), the HC encoding would not have to include all (or even any) of the word assemblies of the sentence. The overall structure can be retrieved (i.e., binding questions can be answered) as long as the delay assemblies can be reactivated by the HC.

The fact that HC encoding of the active delay assemblies is sufficient to store the sentence structure in memory constitutes an important aspect of the use of delay activity as a binding mechanism. The delay assemblies in the blackboard can remain active concurrently without causing interference, unlike the word and structure assemblies. The reverberating activity of delay assemblies will then provide sufficient time for the process of synaptic modification to proceed (e.g., long-term potentiation takes in the order of 1–4 seconds, and spoken sentences are processed in the order of 3–4 words per second).

In particular, this solves the problem of one-trial learning, as described by Jackendoff (2002, p. 66):

> It is usually argued that transient connections have the effect of gradually adjusting synaptic weights (so-called Hebbian learning). But what about cases in which one trial is sufficient for learning? For example, you say to me, *I'll meet you for lunch at noon*. I reply, *OK*, and indeed we do show up as agreed. My long-term memory has been laid in on the basis of one trial; there hasn't been any opportunity to adjust synaptic weights gradually.

Figure 11 (right) illustrates how one-trial learning can proceed by means of the blackboard architecture. The word assemblies in *The cat chases the mouse* are indeed activated briefly, to prevent the interference effects that would otherwise occur. But the delay assemblies can remain active for a longer period because they do not interfere with each other. This provides the opportunity to adjust the synaptic weights between the HC and the delay assemblies gradually, in line with Hebbian learning. In this way, a long-term memory of a sentence structure can be formed on the basis of one trial.

6.5.2. Explicit encoding of sentence structure with synaptic modification. Although the conjunctive encoding of the blackboard by the HC provides an encoding of sentence structure in terms of synaptic weights, retrieval of information from long-term memory would require that the blackboard activation of the sentence structure is reactivated by the neurons in HC, probably in a sequence as illustrated in Figure 12. One could imagine that a more explicit encoding of a sentence structure in terms of synaptic weights would be possible, which on its own could be used to retrieve information. An important function of the HC is indeed to provide a quick but temporal storage of information, so that the interaction between the HC and the cortex can result in a (slower) transference of that information to the cortex, where it can be incorporated in the existing knowledge base (O'Reilly & Rudy 2001). After such a process, a sentence structure could be encoded explicitly in the cortex in terms of synaptic modification.

Figure 13 presents a neural structure of *The cat chases the mouse* in terms of synaptic modification (the structure in the brackets represents the shorthand version). As in Figure 2, the structure consists of word assemblies, structure assemblies, and the appropriate bindings between the assemblies. The word assemblies in Figure 13 are the same as those in Figure 2. The structure assemblies in Figure 13 are of the same kind as those in Figure 2 (NP and VP). Structure assemblies in Figure 13 also consist of main assemblies and subassemblies connected to gating circuits. However, binding in Figure 13 is not achieved by memory circuits (as in Fig. 2) but instead consists of synaptic modification. In this way, the word assemblies are directly connected to the main assemblies. Subassemblies of the same kind are also directly connected to each other, effectively forming a single assembly.

The structure assemblies in Figure 13 do not belong to the blackboard architecture illustrated in Figure 4. Binding in the architecture of Figure 4 is always temporary, lasting only as long as the activity of the delay assemblies in the memory circuits. When the delay assemblies in the memory circuits connected to a structure assembly are no longer active, the structure assembly can be

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

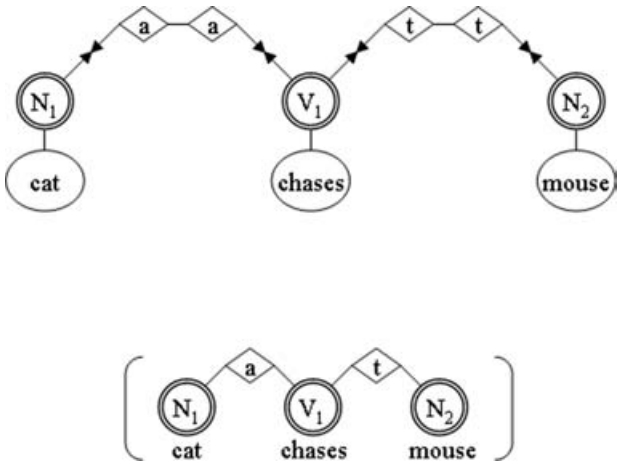


Figure 13. Explicit encoding of neural sentence structure in long-term memory, illustrated with the sentence *cat chases mouse*. Word assemblies are bound to main assemblies of structure assemblies with synaptic modification, with nouns to noun-phrase (NP) assemblies and verbs to verb-phrase (VP) assemblies. Subassemblies of the same kind are bound with synaptic modification. This effectively results in a single subassembly, as illustrated with the agent (a) and theme (t) subassemblies of NP and VP assemblies. A shorthand presentation of the sentence structure is given in brackets.

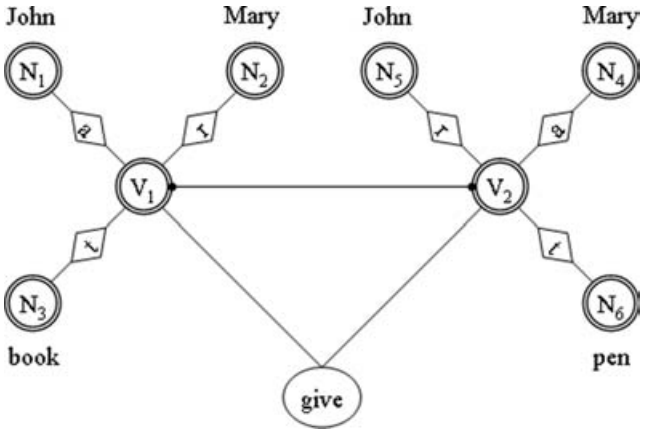


Figure 14. The explicit encoding of the (combined) neural structures of *John gives Mary a book* and *Mary gives John a pen* in long-term memory, in the manner of the structure presented in Figure 13 (with shorthand presentation). The subassemblies include a new subassembly for recipient (r). VP main assemblies are mutually inhibitory.

reused in the encoding of a different sentence structure (again temporarily). This characteristic is the basis of the productivity of the architecture in Figure 4. With the ability to reuse the structure assemblies again and again, the architecture can encode arbitrary and novel sentence structures on the fly.

In contrast, the structure assemblies in Figure 13 cannot be reused in this way. Because of binding with synaptic modification, the structure in Figure 13 is of a more permanent nature created specifically for this particular sentence structure. A knowledge base of this kind can only consist of a collection of sentence structures ("facts") that have actually been encountered. Furthermore, each structure will be created after a (prolonged) learning process in line with the transference of information between the HC and the cortex discussed above. Hence it is possible that the sentence *The cat chases the mouse* belongs to this knowledge base but the sentence *The mouse chases the cat* does not.

6.6. Variable binding

The knowledge base illustrated with the sentence structure in Figure 13 can be used in a rule-based derivation with variable binding, such as the derivation that *own(Mary, book)* follows from *give(John, Mary, book)*. Here we discuss how the binding question "What does Mary own?" can be answered on the basis of the fact (proposition) *John gives Mary a book and Mary gives John a pen*. In section 3, we argued that the model of Shastri and Ajjanagadde (1993b), based on synchrony of activation, is faced with serious difficulties in the case of such a proposition because of the multiplication of the arguments *John* and *Mary* in different roles in the proposition (i.e., the problem of 2).

Figure 14 shows how the combination of the facts *John gives Mary a book* and *Mary gives John a pen* will be encoded in terms of the neural structure introduced in Figure 13 (using the shorthand presentation). The verb *give(x, y, z)* has three arguments (agent, recipient, and theme); hence the VP and NP assemblies have an additional subassembly for recipient (r). The word assembly for *give* is connected to two VP main assemblies (V_1 and V_2), which are mutually inhibitory. V_1 is bound to the NP assemblies for *John* (N_1), *Mary* (N_2), and *book* (N_3) in such a manner that it encodes the fact *give(John, Mary, book)*. Similarly, V_2 is bound to the NP assemblies for *Mary* (N_4), *John* (N_5), and *pen* (N_6) in such a manner that it encodes the fact *give(Mary, John, pen)*.

Even though the fact *Mary owns a book* does not belong to the knowledge base, the question "What does Mary own?" can be answered on the basis of the fact *John gives Mary a book* by transforming the information provided by the question into information related to *give(x, y, z)*. The question "What does Mary own?" provides the information that *Mary* is the agent of *own*, and it asks for the theme in the proposition. In short, the question provides information of the form *own(Mary,?)*. In terms of *give(x, y, z)*, the question provides the information that *Mary* is the recipient of *give*, and it asks for the theme in the proposition. In short, the question provides information of the form *give(-, Mary, ?)*. In general, information of the form *own(X, ?)* can be transformed into information of the form *give(-, X, ?)* on the basis of a long-term association between *own-agent* and *give-recipient* (as in the model of Shastri and Ajjanagadde 1993b).

In line with the process of answering binding questions (sects. 6.3.1. and 6.3.2.), the information of the form *own(X, ?)* will produce the activation of the assembly for *own* and the gating circuits for agent. In contrast, the information of the form *give(−, X, ?)* will produce the activation of the assembly for *give* and the gating circuits for recipient. Therefore the activation produced by *own(X, ?)* cannot be concurrently active with the activation produced by *give(−, X, ?)*. In Figure 14, this would result in the activation of *give* and the combined

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

activation of the gating circuits for agent and recipient. The VP assemblies V_1 and V_2 would then receive an equal amount of activation when the assembly for *X* (*Mary*) is active, so that *book* and *pen* would have an equal probability of being produced as the answer to the question.

Concurrent activation produced by *own(X, ?)* and *give(−, X, ?)* would be prevented if the activation produced by *own(X, ?)* consists of an "attractor" state (Amit 1989) of a control network, which is associated with the attractor state in the control network produced by *give(−, X, ?)*. First, the control network will be in the attractor state related to *own(X, ?)*. But when an answer is not produced in this way (because *own(Mary, book)* does not belong to the knowledge base), the attractor state in the control network would change into the associated attractor state that corresponds with *give(−, X, ?)*.[13]

When the information related to *give(−,Mary,?)* is singled out, the answer can be produced by activating *Mary* and *give* and the gating circuit for recipient. As illustrated in Figure 14, this will result in V_1 as the winner of the competition between the VP assemblies (as in Fig. 8). After that, the answer can be produced by activating the gating circuits for theme.

The transformation of the information related to *own(X, ?)* into information related to *give(−, X, ?)* does not depend on *Mary* or on any of the other word assemblies in Figure 14 (i.e., *book*, *pen*, or *John*). It depends only on the association between *own-agent* and *give-recipient*. Hence the derivation of *own(Mary, book)* from *give(John, Mary, book)* is a rule-based derivation with variable binding. The same process can operate on the blackboard architecture in Figure 4, so that a novel structure like *give(Dumbledore, Harry, broom)* can result in the answer to the question "What does Harry own?"

6.6.1. Neural structure versus spreading of activation. In the neural structure illustrated in Figure 14, the fact *give(John, Mary, book)* can be used to answer the question "What does Mary own?" even though the fact *give(Mary, John, pen)* is also instantiated in the architecture. The two facts do not interfere with each other because the gating circuits control the flow of activation in the structure assemblies.

Figure 15 shows the collapse of the structure presented in Figure 14 when the gating circuits are removed, to illustrate again the importance of activation control provided by the gating circuits in the neural structures presented here (see also Fig. 6b and c). Without the gating circuits, a main assembly and its subassemblies merge into a single assembly. In fact, the NP assemblies can be omitted altogether, because the word assemblies for the nouns are now directly connected to the VP main assemblies V_1 and V_2. Because all assemblies are now directly connected through excitatory or inhibitory connections, processing depends only on spreading of activation. The information related to *give(−,Mary,?)* results in the activation of the assemblies for *give* and *Mary*. Because of the uncontrolled spreading of activation, the activation of *give* and *Mary* results in an equal activation of V_1 and V_2, so that a correct answer to the question cannot be given without ambiguity or error.

In fact, any question will result in ambiguities or error in this uncontrolled spreading of activation network. For example, a question like "Who gives a book?" will result in the activation of both *John* and *Mary* as potential answers, even though V_1 will win the competition over V_2. In contrast, in the structure in Figure 14, the question "Who gives a book?" will result in *John* as the answer, because the question will result in the activation of the gating circuits for agent after V_1 has won the VP competition.

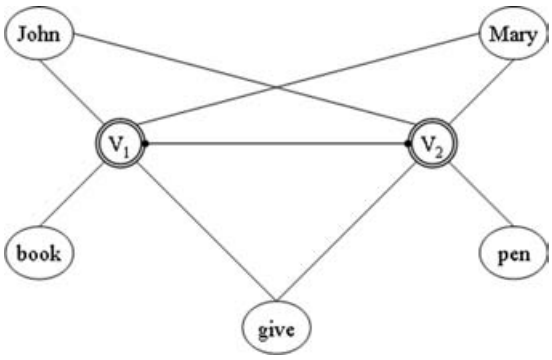


Figure 15. Illustration of the collapse of the neural structures presented in Figure 14 when the gating circuits are removed. The result is a network of assemblies based on spreading of activation.

6.7. Summary of the basic architecture

The ability of the architecture to encode arbitrary sentence structures is based on the fact that binding between a word assembly and a structure assembly and between two structure assemblies is only temporal. The duration of the binding between two assemblies is given by the duration of the reverberating activity of the delay assembly in the memory circuit that connects the two assemblies. When the reverberating activity in the delay assemblies disappears, the structure assemblies are "free" again, which means that they can be used to encode another sentence structure. Furthermore, only a small set of structure assemblies is needed in the architecture (enough to account for the memory span of language users and the duration of reverberating activity). As a result, an arbitrary number of sentence structures can be encoded without an explosion of structure assemblies.

Binding in this architecture is not a state of the system that needs to be observed for read-out purposes (as in the case of binding with synchrony of activation). Instead, it is a process in which binding relations can be retrieved by posing "binding questions." A system can only answer a binding question on the basis of information that is available to the system itself. Hence the answer to a binding question shows that the system has solved the binding problem implicated in the question. For example, *mouse* is bound to *chases* as its theme when it emerges as the answer to the question "Whom does the cat chase?" as illustrated in Figures 7 and 8. The process of answering binding questions affects (manipulates) the activity of the structure and word assemblies, as illustrated in Figure 8. But the activity of the delay assemblies is not disrupted by the process of answering binding questions. This is crucial for retrieving the binding information in the architecture, because that information is based on

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

the activity of the delay assemblies (and not on the activity of the word and structure assemblies). Hence read-out is always possible as long as the delay assemblies remain active. Subsequent processing stages can retrieve information in this way from the architecture and use that, for example, to produce a long-term memory structure of the binding relations in the sentence, as illustrated in Figures 13 and 14.

The process of encoding a sentence structure is controlled by neural circuits that can selectively activate a specific type of gating circuits, such as the gating circuits for the VP agent subassemblies. An example of a control circuit will be given below. The type of gating circuit that is activated at a given moment is determined by syntactic structure of the sentence. Hence the neural circuits that control the encoding of a sentence structure instantiate basic parsing operations. Activated gating circuits will activate subassemblies, which can then bind in the manner illustrated in Figure 5. Binding between subassemblies depends on the concurrent activation of these assemblies at a given moment. In that case, they activate a specific delay assembly by means of a circuit that instantiates an AND operation, as illustrated in Figure 5. Other circuits that instantiate AND operations (e.g., see Koch 1999) could also be used for this purpose.

The process of retrieving information (answering binding questions) in the architecture requires a form of dynamic control. As noted, this form of control does not depend on the information stored in the blackboard. Instead, the sequence depends on the type of the question asked. This form of control is not unlike that found in motor behavior, which also depends on a precise sequence of excitation and inhibition of muscle innervations.

The activation in the architecture is a form of working memory that is specific for encoding language structure. To show how information encoded in this way can be transferred to long-term memory, we introduced the hippocampus (and surrounding areas) as a "simple" binding device, in line with the literature on that subject. This does not mean that the hippocampus would be necessary for encoding sentence structure in the architecture itself. Sentence encoding in the blackboard architecture can occur independently of the hippocampus. In fact, the activity of the delay assemblies that encodes a sentence structure in the blackboard is needed for the use of the hippocampus as a simple binding device, as illustrated in Figures 11 and 12.

6.8. Structural dependencies in the blackboard architecture

As Jackendoff (2002) noted, a solution of the "four challenges for cognitive neuroscience," as presented above, would allow a more productive interaction between neural network modeling and linguistic theory to begin. To illustrate the possibility of such an interaction, we will discuss the neural blackboard structures of the sentences (1) and (2), discussed in section 4.2. They are repeated here for convenience:

The cat that the dog that the boy likes bites chases the mouse (1)

The fact that the mouse that the cat chases roars surprises the boy (2)

In section 4.3, we argued that these two sentences pose a problem for models that process sentences in terms of strings of word category labels (*N-V* strings). Both sentences have the same word category structure (*N-that-N-that-N-V-V-V-N*) but they are different in terms of complexity (Gibson 1998), with (1) rated as far more complex than (2). The difference in complexity between the sentences is related to the different bindings between the constituents in both sentences. In (1), the subject of the main clause (*cat*) is also an argument (theme) of a verb in an embedded clause (*bites*), whereas in (2), the subject of the main clause (*fact*) is not an argument of any of the verbs in the embedded clauses (*chases* and *roars*). The contrast between (1) and (2) forms an interesting example of the massiveness of the binding problem that occurs in language.

A neural instantiation of sentence structure has to account for the differences in constituent binding illustrated with sentences (1) and (2), as any linguistic theory of sentence structure would have to do. But a neural instantiation of sentence structure should also provide an explanation of the observed differences in complexity between these sentences (and other performance effects; van der Velde 1995).

As noted, the structural difference between sentences (1) and (2) is related to the nature of the embedded clauses they contain. Therefore we first have to discuss how embedded clauses can be instantiated in the neural architecture presented here.

6.8.1. Embedded clauses in the blackboard architecture.

Figure 16a presents the structure of the sentence *The cat that bites the dog chases the mouse* (without the determiners *the*). This sentence contains the subject-relative clause *that bites the dog*. To encode and bind this clause, a new clause structure assembly (C) is introduced, with a new clause subassembly (c). C assemblies play a role in the encoding of a clause that is similar to the role played by S assemblies in the encoding of the main sentence (*cat chases mouse* in Fig. 16a). However, there are a few differences between the roles played by S and C assemblies, which motivate their distinction.[14] C assemblies have to be bound to one of the structure assemblies in the sentence, as illustrated with the binding between C_1 and N_1 in Figure 16a, which requires a new kind of subassembly (c). Furthermore, the word assemblies of complementizers can bind to C assemblies, as illustrated by *that* in Figure 16a.

A verb (verb-phrase) can only have a single argument for each of its thematic roles, but a noun (noun-phrase) can be the argument of two verbs. That is, a noun can bind to the verb on the same level as the noun in the sentence structure (the "sister" of the noun) and it can bind to a verb in a subordinate clause. Because binding is achieved with subassemblies in this architecture, different subassemblies will be needed for binding a noun to its "sister" verb and to a subordinate verb. In Figure 16a, to encode that *cat* is the subject of the verb in its subordinate clause (*bites the dog*), N_1 binds through a n^c (noun-clause) subassembly to the n subassembly of C_1. The n^c subassembly of N_1 is similar to its n subassembly, except that it is activated under the influence of the (subordinate) clause C_1, introduced with *that*.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

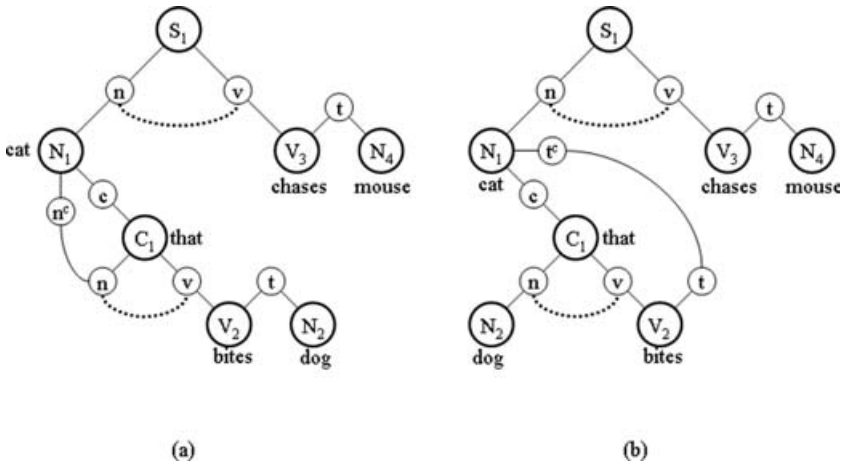


Figure 16. a. Illustration of the neural sentence structure of *The cat that bites the dog chases the mouse* (without the determiners). The structure is based on the sentence structure presented in Figure 9 (bottom right) with the addition of a clause structure assembly (C) and a clause subassembly (c). The dotted lines represent agreement between subject and verb. b. Illustration of the sentence structure of *The cat that the dog bites chases the mouse*, using the same kind of structure assemblies as in a.

Like S assemblies, C assemblies can be used to encode agreement between subject and verb. In the case of a subject-relative clause, agreement exists between subject of the main sentence (*cat*) and the verb of the relative clause (*bites*), as indicated with the dotted line between the noun and verb subassemblies of the C assembly in Figure 16a.

Figure 16b presents the structure of the sentence *The cat that the dog bites chases the mouse*. This sentence contains the object-relative clause *that the dog bites*. In this case, *dog* is the subject of *bites*, so it is bound to the noun subassembly of C_1 by its n subassembly (because *bites* is a sister of *dog* in the clause). As before, agreement between *dog* and *bites* is encoded by agreement between the noun and verb subassemblies of C_1, as indicated with the dotted line. In an object-relative sentence such as *The cat that the dog bites chases the mouse*, the subject of the main sentence is the theme of the verb in the relative (subordinate) clause. To this end, the t^c (theme-clause) subassembly of N_1 (*cat*) is used to bind to the t subassembly of V_2 (*bites*). The t^c subassembly of N_1 is similar to its t subassembly, except that it is activated under the influence of the subordinate clause, as in the case of the n^c subassembly of N_1.

The activation of the t^c subassembly of N_1 poses a problem for the control of binding in this sentence. When V_2 is active, N_2 is the active NP assembly, not N_1. Therefore the t^c subassembly of N_1 has to be activated before the activation of N_2 (a subassembly can remain active even if its main assembly is deactivated). Hence the gating circuits for theme clause have to be activated before the activation of the main assembly of N_2. With the object-relative sentence in Figure 16b, the control circuits could conclude from the sequence *cat that dog* (or *N that N*) that *cat* is the theme of a verb in the clause, so the gating circuits for t^c have to be activated before the activation of N_2. This control of activation is not needed for the subject-relative sentence in Figure 16a. Furthermore, the theme of *bites* (V_2) in Figure 16a presents itself directly with the occurrence of *dog* (N_2), resulting in a direct binding between V_2 and N_2. In Figure 16b, the theme of *bites* (*cat*-N_1) can bind only to V_2 because of the prolonged activation of t^c. These activation differences between the structures in Figure 16b and Figure 16a could be the basis for the fact that object-relative sentences are more difficult to process than subject-relative sentences (Gibson 1998).

Because a verb can have only one argument, the distinction between t and t^c subassemblies needed for NP assemblies does not occur with VP assemblies. For the same reason, the distinction between n and n^c subassemblies needed for NP assemblies does not occur with C assemblies. In this way, the verb *bites* can have only one subject (either *cat*-N_1 in Fig. 16a or *dog*-N_2 in Fig. 16b) and only one theme (either *dog*-N_2 in Fig. 16a or *cat*-N_1 in Fig. 16b).

In the case of a sentence *The cat that the dog bites chases the mouse* (Fig. 16b), the question "Who bites the cat that chases the mouse?" can be answered by activating the word assemblies and the gating circuits in the direction from *mouse* (theme) to *chases* (subject) to *cat* (theme) to *bites*. This will result in *bites* (V_2) winning the competition (in particular, because it receives more activation from *cat* (N_1) than *chases* (V_3) receives from *mouse* (N_4).

An interesting comparison can be made with answering the question "Whom does the cat that chases the mouse bite?" with the sentence *The cat that bites the dog chases the mouse* (Fig. 16a). The difficulty here is that *cat* is the subject (agent) of both *chases* and *bites*. So it has to be figured out that *cat bites dog* is an embedded clause, that is, that activating the gating circuits in the direction from *mouse* to *chases* to *cat* results in the activation of the S assembly instead of the C assembly. In other words, the architecture predicts that answering this question will be more complex than answering "Who bites the cat that chases the mouse?" with the sentence *The cat that the dog bites chases the mouse*. Notice that this difference in complexity is the reverse of that of the sentences involved. That is, the object-relative sentence *The cat that the dog bites chases the mouse* is more complex than the subject-relative sentence *The cat that bites the dog chases the mouse* (Gibson 1998).

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

6.8.2. Multiple embedded clauses. Figure 17 illustrates the structure of *The boy sees the mouse that likes the dog that chases the cat*.

The right-branching nature of this sentence structure is a straightforward extension of the structure in Figure 16a. In this case, each embedded clause is attached to the theme of its superordinate clause. The structure can easily be constructed in an incremental manner by binding each new C assembly to the last active NP assembly. This is in agreement with the fact that strictly right-branching sentences are easy to process in English (Gibson 1998).

Figure 17 illustrates (again) how the constituent structure of a sentence can be instantiated in the neural architecture presented here. The phrase (*mouse*) *that likes the dog that chases the cat* is a constituent because it is "dominated" by N_2. This results from the fact that, for example, N_2 is bound to C_1 with c subassemblies, whereas N_3 is (indirectly) bound to C_1 with v subassemblies. The nature of binding, that is, the subassemblies used, determines the dominance relations in the structure: N_2 dominates C_1, whereas C_1 dominates N_3 (which, in turn, dominates C_2).

Figure 18a presents the structure of sentence (1), *The cat that the dog that the boy likes bites chases the mouse*. Sentence (1) contains the double center-embedded object-relative clause *that the dog that the boy likes bites*. Sentences of this type are notoriously hard to process, to the point that they can be classified as unprocessable (Gibson 1998). The encoding of the phrase *The cat that the dog* proceeds in the same way as in Figure 16b, so the t^c subassembly of N_1 (*cat*) will be activated to bind to the theme subassembly of the next verb. However, another embedded clause is introduced instead of a verb. The phrase *the dog that the boy* is structurally similar to the phrase *the cat that the dog*, so the t^c subassembly of N_2 (*dog*) will be activated to bind to the theme subassembly of the next verb. Hence, when the first verb (*likes*) appears, there are two subassemblies that can bind to the theme subassembly of this verb, whereas the verb should bind to *dog* (N_2) as its theme argument. The situation is similar with the second verb (*bites*), which should bind to *cat* (N_1) as its theme argument. The two problematic bindings are indicated with the dashed lines in Figure 18a.

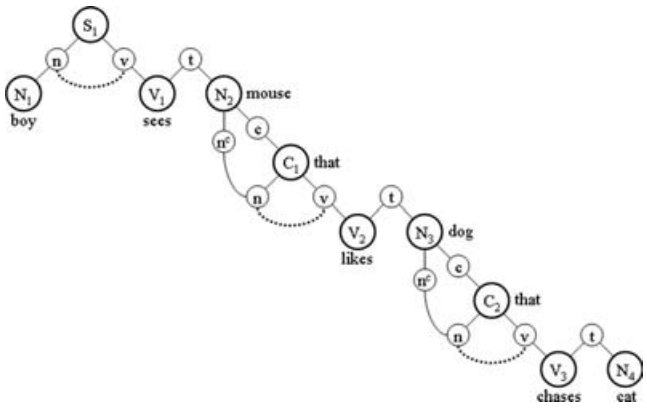


Figure 17. Illustration of the neural sentence structure of *The boy sees the mouse that likes the dog that chases the cat* (ignoring *the*), with the same kind of structure assemblies as in Figure 16a.

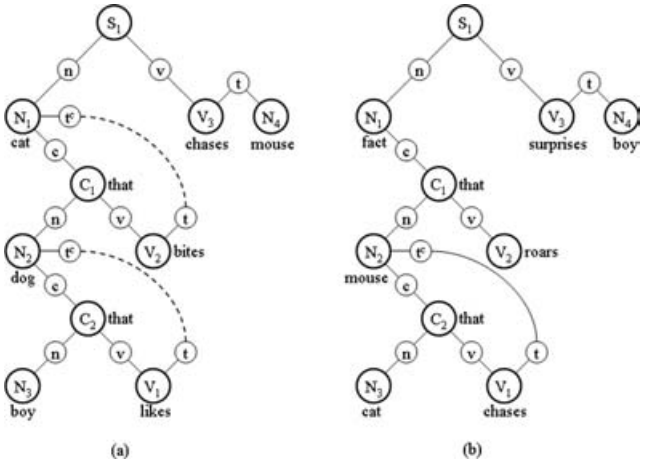


Figure 18. a. Illustration of the neural sentence structure of *The cat that the dog that the boy likes bites chases the mouse*, with the same kind of structure assemblies as used in Figure 16b. b. In the same way, the neural structure of the sentence *The fact that the mouse that the cat chases roars surprises the boy*.

Figure 18b shows the structure of sentence (2), *The fact that the mouse that the cat chases roars surprises the boy*. The structure of (2) is very similar to the structure of (1), except for the fact that *roars* (V_2) does not have a theme argument. A phrase beginning with *The fact that* will be interpreted as a complementary clause, so the t^c subassembly of N_1 (*fact*) will not be activated. When the object-relative clause in *the mouse that the cat chases* appears, the t^c subassembly of N_2 (*mouse*) will be activated to bind *mouse* as the theme of the verb in its subordinate clause (*chases*), as in Figure 16b. However, in contrast to the structure of (1) in Figure 18a, the binding of *mouse* (N_2) to the first verb (*chases*) as its theme can succeed because the theme subassembly of N_2 is the only active theme subassembly at that moment.

Hence the difference in complexity between (1) and (2) results from a difference in structural dependency between both sentences. In (1), the subject of the main sentence (*cat*) is also the theme of a verb in an object-relative clause. In combination with the second object-relative clause, this results in an ambiguity of the binding of *cat* (N_1) or *dog* (N_2) as the theme of *likes* (V_1) or *bites* (V_2). In contrast, in (2) the subject of the main clause (*fact*) is not bound to any of the verbs in the embedded clauses, so the ambiguities in (1) do not arise in (2). Hence sentence complexity in (1) results from binding problems that arise when a number of structure assemblies of the same kind have to bind in sequence with the overall sentence structure (in line with the notion of similarity-based interference as the basis of sentence complexity; Lewis 1999).

At face value, the binding problem that arises with the theme subassemblies of the sentence structure in Figure 18a would also have to arise with the verb subassemblies in both sentence structures in Figure 18, in particular for the verb subassemblies connected to the C assemblies (the verb subassembly of the S assembly could be activated after the binding of C assemblies has been completed). The activation of C_2 will inhibit the

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

activation of C_1 in both sentence structures; hence the verb subassembly of C_1 has to be activated before C_2 is activated. But the first verb in the sentence (*likes* or *chases*) has to be bound to C_2, which requires the activation of the verb subassembly of C_2 as well. However, the binding problem with the verb subassemblies can be solved in terms of the dynamics of the binding process, as discussed below.

6.8.3. Dynamics of binding in the blackboard architecture. The binding of subassemblies occurs in a connection structure as illustrated in Figure 5. Figure 19 illustrates the process of subassembly binding between two arbitrary structure assemblies A and B.

In Figure 19a, the subassembly of A_{i-1} has activated its horizontal row of columns in the connection structure. If the B_j subassembly would activate its vertical row of columns in the connection structure, a binding would result between A_{i-1} and B_j, in the manner as discussed in section 6.2.1. However, the subassembly of A_i is activated first, which results in the activation of a second horizontal row of columns.

In Figure 19b, the subassembly of B_j activates its vertical row of columns in the connection structure. At this moment, a conflict arises between the binding of A_{i-1} to B_j and the binding of A_i to B_j. Because of the inhibitory interaction between the columns in the vertical row of B_j (initiated by activated delay assemblies), only the stronger of these two bindings will survive (as in the VP competition illustrated in Fig. 8). Figure 19c illustrates that A_i will bind to B_j if the activation in the horizontal row of A_i is stronger than the activation in the horizontal row of A_{i-1}.

When the binding process of A_i and B_j has been completed, the columns in the horizontal row of A_i (and the vertical row of B_j) will be inhibited because of the active delay assembly in the column that binds A_i to B_j. However, as illustrated in Figure 19d, the columns in the horizontal row of A_{i-1} are still active (with the exception of the column in the vertical row of B_j). Hence the subassembly of A_{i-1} can bind to another B subassembly if that is activated.

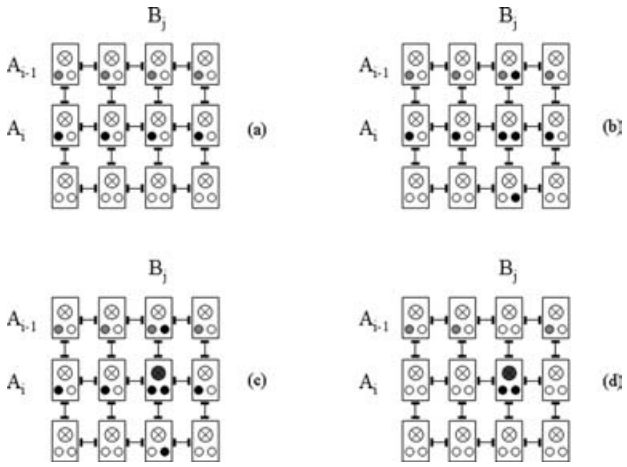


Figure 19. Four stages in the process of subassembly binding between arbitrary structure assemblies A and B, with the connection structure as illustrated in Figure 5. a. The subassemblies of A_{i-1} (first) and A_i (second) have activated their horizontal row of columns. b. The subassembly of B_j has activated its vertical row of columns. c. Binding occurs between A_i and B_j, because the activation in the row of A_i is stronger than the activation in the row of A_{i-1}. d. After completion of the binding process in c, A_{i-1} can bind to another B assembly. In this way, the connection structure can operate as a pushdown stack.

The process illustrated in Figure 19 shows that two A subassemblies can bind in sequence to B subassemblies if there is a clear difference in activation strength between the two A subassemblies. In that case, the stronger activated A subassembly will bind to the first activated B subassembly and the other A subassembly will bind to the second activated B subassembly. In theory, one could have a whole series of A subassemblies that can bind in sequence with B subassemblies if the A subassemblies have distinguishable differences in their activation strengths.

Pulvermüller (1999) suggested that a gradual decay of activation in reverberating assemblies (such as the delay assemblies in the memory circuits) could form the basis of a neural pushdown stack. Figure 19 illustrates this possibility. If the subassemblies of A_{i-n} to A_i have been activated that order, and if the activation strength of the subassemblies decays over time, then the subassembly of A_i would have the strongest activation and it would bind to the first B subassembly, as illustrated in Figure 19. Then the subassembly of A_{i-1} would bind to the next B subassembly, as illustrated in Figure 19d. In the same manner, all the subassemblies of A_{i-n} to A_i would bind to B subassemblies in the reverse order of their activation, in line with the notion of a pushdown stack.[15]

It is not clear whether such a distinctive and reliable decay of reverberating activity will be found in the brain because of the fluctuations that can occur in this kind of activity (Amit 1989). However, in one circumstance one can find a clear difference in activation strength between reverberating assemblies. Fuster and colleagues (1985) investigated the relation between reverberating activity in the prefrontal cortex and in the visual cortex. First they identified neurons in both areas of the cortex that responded to the same objects and that maintained their activation in a delay period. Then they applied a technique of reversible cooling of one of the areas involved. In this way, the activity of the neurons in that area can be blocked temporarily, but the activity will reappear when the temperature is increased to a normal level. Fuster and colleagues observed that blocking the activity of neurons in one area also reduced the activity of the neurons in the other area. The activity in the second area increased again when the activity in the first area reappeared (by terminating the cooling in that area).

The results of the study by Fuster and colleagues (1985) indicate that reverberating activity in a neural assembly is stronger when the assembly also receives activation from outside. In this way, the binding of the verb subassemblies in the sentence structures in Figure 18 can be explained. The main assembly of C_2 is active when the first verb (*likes* or *chases*) appears. Therefore the verb subassembly of C_2 is also activated by the main assembly, unlike the verb subassembly of C_1. As a result, the activity of the verb subassembly of C_2 is stronger than the activity of the verb subassembly of C_1. In line with the binding process illustrated in Figure 19, the verb subassembly of C_2 will bind to the verb subassembly of V_1 (*likes* or

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

chases), and the verb subassembly of C_1 will bind to the VP assembly (V_2) of the next verb (*bites* or *roars*).

In contrast, the main assembly of N_2 in Figure 18a is not active because of the activation of N_3 (*boy*), which is needed to bind *boy* with *likes*. Without a clear distinction in activation strength between the theme subassemblies of N_1 and N_2, the binding process illustrated in Figure 19 will produce a conflict, which results in the complexity associated with sentence (1).

6.8.4. Control of binding and sentence structure. Figure 20 illustrates the structure of the sentence *The boy says that the dog knows that the cat chases the mouse*, combined with a circuit that can be used to control the binding of the constituents in the sentence structure.

The control circuit is given in the form of a connectionist network consisting of I-nodes (input nodes), E-nodes ("expectation" nodes), and C-nodes ("conjunction" nodes). The I-nodes are activated by the words in the sentence, based on their lexical type. It is assumed that one I-node is active at a time (with the exception of the I-node *S*). The I-node *S* is inhibited by the E-nodes (not shown in Fig. 20). The E-nodes are activated by specific I-nodes or C-nodes. They remain active until inhibited. The C-nodes are activated by a specific conjunction of an I-node and an E-node. They activate the gating circuits that result in a specific binding of constituents in the sentence structure, and they inhibit the E-node by which they are activated.

When *boy* is presented, it binds to N_1 and it activates the I-node *N*. Furthermore, *boy* will be seen as the beginning of a sentence, because there are no active E-nodes that would force a binding of *boy* in an existing sentence structure (see below). As a result, *boy* also activates S_1 and the I-node *S*, which in turn activates the E-nodes $\mathbf{S_n}$ and $\mathbf{S_v}$. The conjunction node $N\mathbf{S_n}$ is then activated, which results in the binding of S_1 and N_1 to their noun (n) subassemblies. The activation of the E-node $\mathbf{S_v}$ reflects the expectation of a verb for the main (matrix) sentence.

When the verb *says* is presented, it activates V_1 and the I-node V_c, which entails that *says* is interpreted as a verb that requires a complement clause (given by the lexical information related to *says*). In turn, V_c activates the E-node $\mathbf{V_c}$. The combined activation of the I-node V_c and the E-node $\mathbf{S_v}$ results in the activation of the conjunction node $V\mathbf{S_v}$. This node activates the gating circuits for the binding of V assemblies and S assemblies with their verb (v) subassembly, which results in the binding of V_1 with S_1. The word *that* activates C_1 and the I-node *C*, which in combination with the E-node $\mathbf{V_c}$ activates the C-node $C\mathbf{V_c}$. In turn, $C\mathbf{V_c}$ produces the binding of C_1 with V_1, and activates the E-nodes $\mathbf{C_n}$ and $\mathbf{C_v}$ (i.e., the expectation of a clause subject and verb).

Continuing in this manner, *dog* (N_2) and *knows* (V_2) will bind to C_1. The verb *knows* again activates the I-node V_c, which results in the binding of *that* (C_2) to *knows* (V_2). Then *cat* (N_3) and *chases* (V_3) will bind to C_2. The verb *chases* is a verb that requires a theme. This lexical information related to *chases* will activate the I-node V_t, which in turn activates the E-node $\mathbf{V_t}$. The word *mouse* will bind to N_4 and will activate the I-node *N*. In combination with the active E-node $\mathbf{V_t}$, this results in the binding of *mouse* (N_4) as the theme of *chases* (V_3).

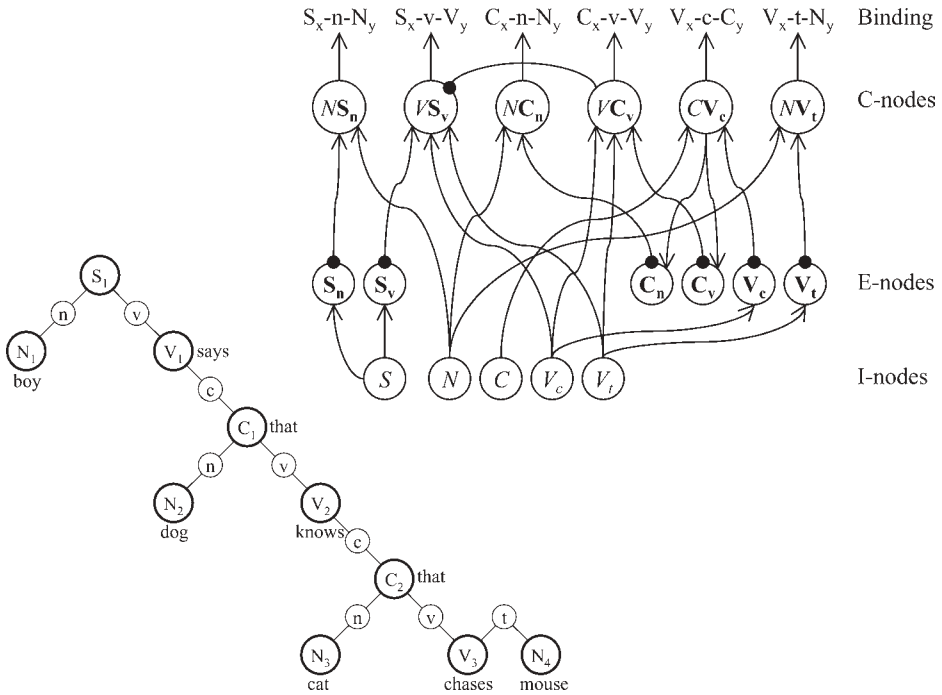


Figure 20. Illustration of the neural sentence structure of *The boy says that the dog knows that the cat chases the mouse*, combined with a control circuit for this sentence. I-nodes are the input nodes for the circuit, activated by the words in a sentence. E-nodes are expectation nodes with sustained (delay) activation, and C-nodes are conjunction nodes that activate the gating circuits for a specific binding. Connections with an arrowhead are excitatory. Connections with a dot are inhibitory. Connections with an arrowhead and a dot represent two (bidirectional) connections. The binding symbol S_x-n-N_y represents the binding of an arbitrary active S assembly with an arbitrary active N assembly by means of their noun (n) subassemblies. The other binding symbols represent similar forms of binding.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

Notice that the nouns *dog*, *cat*, and *mouse* are bound in an existing sentence structure because of the active E-nodes in the circuit.

The circuit in Figure 20 illustrates how the process of binding could proceed in the architecture for sentence structure presented here. However, it is clear that a more detailed account of this process is a topic for further research, as described below.

6.9. Further development of the architecture

The neural blackboard architecture for sentence structure outlined here provides a solution to the "four challenges for cognitive neuroscience" presented by Jackendoff (2002). The discussion in section 6.8 also illustrates that the architecture can potentially account for structural and performance aspects of language processing. However, further research is clearly needed to fulfill this potential. A few directions of further research can be indicated with the architecture presented so far.

One line of research would concern the development of the architecture, both in terms of evolution and in terms of growth and learning. As Figure 7b illustrates, the architecture consists for the most part of gating circuits, which can be seen as the first level of organization in the architecture. The second level consists of gating circuits organized in structure assemblies. The third level of organization consists of the distinction between different kinds of structure assemblies and the way they interact. Gating mechanisms are found in the brain (e.g., Newman et al. 1997). So the study of the development of the architecture would be focused on the way in which the second and third level of organization arise.

In terms of evolution, an important issue is the development of the connection structure presented in Figures 5 and 19. A benefit of an explicit model such as the one in Figure 5 is that the model can be used as a target in computer simulations. Hence, starting with more elementary structures, one could investigate whether such a connection structure could develop in an evolution-like process. In terms of growth and learning, an important issue is the question of how specific bindings with connection structures like the one in Figure 5 could develop. That is, assuming that an undifferentiated connection structure exists for undifferentiated assemblies, one can investigate whether a learning process could reorganize the undifferentiated connection structure into a connection structure in which distinctions are found between different kinds of structure assemblies and subassemblies, as illustrated above. Furthermore, one could investigate whether different languages used in the learning process would result in a different reorganization of the initial connection structure.

Another line of research concerns the issue of parsing in this architecture. Parsing will result from the neural circuits that control the binding process in the architecture. An example is presented in Figure 20. As noted earlier, the control circuits instantiate basic parsing operations. Hence they will be sensitive to the coding principles used in languages to express structural information, such as word order or case marking in languages with free word order (van Valin 2001). However, the neural control circuits will also be sensitive to the pattern of activation that arises in the blackboard during sentence processing. Figure 19 provides an illustration. An active subassembly produces a significant amount of activation in its connection structure (i.e., its row of columns), which provides the information that a specific binding is required. This information can be used by the control circuits to initiate the activation of a subassembly of the same kind (e.g., a VP theme subassembly when an NP theme subassembly is active).

One topic in the study of parsing will be the question of how constraints are implemented in the control circuits and how they relate to aspects of sentence ambiguity. An example of a constraint is presented in the circuit in Figure 20. It consists of the inhibitory connection from the C-node $V\mathbf{C_v}$ to the C-node $V\mathbf{S_v}$. This connection implements the hierarchical constraint that verb-binding in a clause precedes verb-binding in a main (matrix) sentence (as in Fig. 18). This form of constraint is a "hard" constraint in which one demand (verb-binding in a clause) overrides another (verb-binding in a matrix sentence). An important issue will be how "soft" constraints can be implemented, including those given by statistical regularities and semantic information.

It is clear that the study of parsing in the architecture presented here is just beginning. However, the examples illustrated so far suggest that the neural control circuits that control the binding process in the architecture are engaged in a form of pattern recognition and pattern completion in which the current state of activation in the blackboard, together with the active word assemblies, constitutes the input pattern, and the new state of activation in the blackboard constitutes the output pattern. Pattern recognition is a core capability of networks (Bechtel & Abrahamsen 2002). The fact that a neural blackboard architecture of sentence structure could transform parsing operations into forms of pattern recognition is an attractive prospect for further research.

In combination with parsing, the study of sentence complexity and other psycholinguistic effects is also an important line of further research. Figures 16 and 18 illustrate the potential of the architecture to account for complexity effects, but there are number of other complexity issues that should be accounted for as well (see, e.g., Gibson 1998). An interesting topic here will be the complexity related to answering binding questions, as discussed in section 6.8.1. The architecture we present suggests that this can be a source of complexity of its own that needs to be investigated further.

A fourth line of investigation consists of relating the structure and dynamics of the architecture to observable brain structure and activation. The connection structure in Figure 5 is a prediction of how the brain could realize combinatorial productivity. The activation in the architecture during sentence retrieval, as illustrated in Figure 8, is also a prediction of activation that would occur in the brain. Dynamic causal modeling (Friston et al. 2003) can be used to compare activation simulated in a model with activation observed with neuroimaging. As an example, this approach could be used to investigate the sustained activation needed to handle long-distance dependencies. Figure 16b illustrates a long-distance dependency with object-relative clauses in the architecture, which can be simulated in terms of the neural dynamics illustrated in Figure 8. Ben-Sachar and colleagues (2003) observed activation in specific brain regions produced by sentences of this type (as compared to sentences with complement

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

clauses). With the approach of dynamic causal modeling, we can begin to compare the activation in the architecture produced by these sentence types to the activation observed in the brain.

Likewise, the neural activity that occurs in a simulation, as illustrated in Figure 8, can be transformed mathematically into an electroencephalogram (EEG) signal and compared with observations. This is technically demanding, but in conceptual terms it can be done. An important issue here is the topology of the network, which will affect the EEG signal. The model could be used to test and develop specific topologies by deriving the imaging signal using a given topology and comparing it with empirical results.

A fifth line of research is the relation between the architecture for sentence structure and other architectures for combinatorial structures in language (Jackendoff 2002) and cognition in general. For example, words can have an internal structure of their own that does not seem to agree with word encoding by means of (unstructured) word assemblies (Bierwisch 1999). However, the word assemblies used here can be seen as the interface between word structure (e.g., phonological structure) and sentence structure. That is, a word assembly is the part of a neural word structure that connects (or "anchors") that structure within the sentence structure. An example is given in Figures 2 and 13, in which the assemblies for *cat*, *chases*, and *mouse* form the interface between sentence structures in working memory (Figure 2) and sentence structures in long-term memory (Figure 13).

The interaction between a neural architecture for combinatorial (e.g., phonological) word structure and a neural architecture for sentence structure could explain how new words can be embedded easily in a sentence structure. A new word is itself a novel combination of familiar constituents (phonemes) instantiated in its own (assumed) phonological neural blackboard. Hence a new word would create a word structure in this word architecture similar to the way in which a new sentence creates a sentence structure in the neural sentence architecture. In this way, a new word would be temporarily encoded with, say, a W_x assembly, just as a sentence is temporarily encoded with an S_x assembly. The W_x assemblies could bind to the structure assemblies in the sentence architecture, which would result in the embedding of the new word in the sentence structure. Over time, the new word would form a direct link with the kind of structure assemblies in the sentence architecture to which it belongs.

Word assemblies could also form the interface between sentence structures and cognitive structures outside language, such as structures in visual cognition. This issue is addressed in more detail below.

7. Neural blackboard architectures of combinatorial structures in vision

The aim of this paper is to show that combinatorial structures can be encoded in neural terms by means of neural "blackboard" architectures. Although combinatorial structures are the "quintessential property" of language (Pinker 1998), they can also be found in visual cognition. Therefore we briefly discuss neural blackboard architectures of combinatorial structures in visual cognition, in particular for binding visual object features such as shape, color, and (relative) location. In this way we can investigate the differences and similarities that exist between neural architectures of combinatorial structures in two different domains such as language and vision. Furthermore, we discuss how the architectures for visual cognition and for language can be combined in a combinatorial structure like *The little star is beside a big star*.

As in the case of the architecture for sentence structure, we discuss the issue of binding in the "vision" architecture, such as the binding of color and shape, in terms of the process that answers binding questions, such as "What is the color of this shape?" The reason we discuss the binding problem in this way is related to the coordinate system or frame of reference in which the binding problem should be solved. As outside observers, we could see some form of related (e.g., concurrent) activity in brain areas that are involved in processing information in a given task, such as binding the color and shape of visual objects. But it is not clear that the observed relation in activity is used by these brain areas to solve the binding problem at hand; that is, it is not clear that these brain areas "know" that they are, say, concurrently active with each other, so that they can use that information effectively. What is needed is information that is available within the system itself (instead of only from an outside perspective). A binding question like "What is the color of this shape?" probes for information that is available within the system itself, because the system generates behavior when it answers such a question, which it can only do by using information that is available within the system itself. Investigating the process that results in answering binding questions is, in our view, the best way to study (and solve) the issue of binding in combinatorial structures, including the binding of color and shape (and the binding of words in a sentence structure).

In a blackboard architecture for visual cognition, one would have processors for the recognition of shape, color, location, and other visual-object features. Combined, these processors would correctly process a visual display of objects, such as a blue cross on the left and a yellow diamond on the right, if they could communicate with each other through a blackboard. In this way, the architecture could answer binding questions like "What is the color of the cross?" or "What is the shape of the yellow object?"

A neural blackboard architecture for combining visual object features in this manner is illustrated in Figure 21. The architecture is based on the pathways that determine the structure of the visual cortex (e.g.,Farah et al. 1999; Felleman & van Essen 1991; Livingstone & Hubel 1988; Oram & Perrett 1994).

The ventral pathway in the visual cortex includes the areas V2, V4, the posterior inferotemporal cortex (PIT) and the anterior inferotemporal cortex (AIT). This pathway is involved in the processing and selection of "object features" (e.g., shape and color). Objects are identified through a feed-forward network of areas, going from the primary visual cortex (V1) to the higher areas in the temporal cortex (e.g., AIT). The network gradually transforms retinotopic encoding in the lower areas (e.g., V2 to PIT) into a location-invariant identity (e.g., shape, color) encoding in the higher areas (e.g., AIT). The dorsal pathway in the visual cortex leads to the posterior parietal cortex (PP). This pathway is involved in the processing and

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

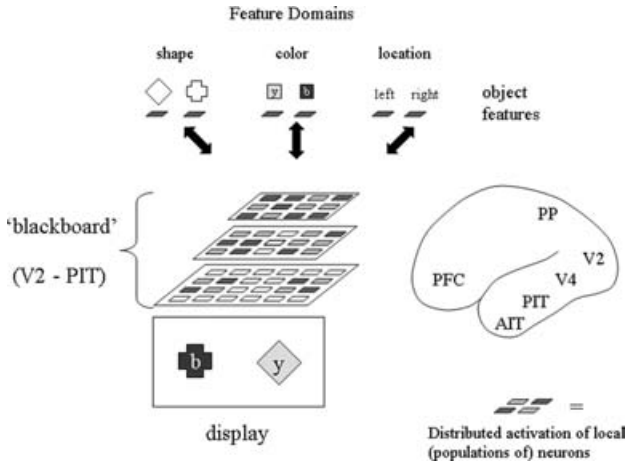


Figure 21. A neural blackboard architecture of combinatorial structure in visual cognition. The "blackboard" consists of the retinotopic areas in the visual cortex (e.g., V2 to PIT). Information about visual features (color, form, motion, location) is processed in feed-forward pathways leading to "feature domains" in specialized areas in the visual cortex (e.g., AIT for shape information, PP for location information). In turn, the feature domains send information to the retinotopic areas by means of feedback connections. (AIT, anterior inferotemporal cortex; PIT, posterior inferotemporal cortex; PFC, prefrontal cortex; PP, posterior parietal cortex; b, blue; y, yellow).

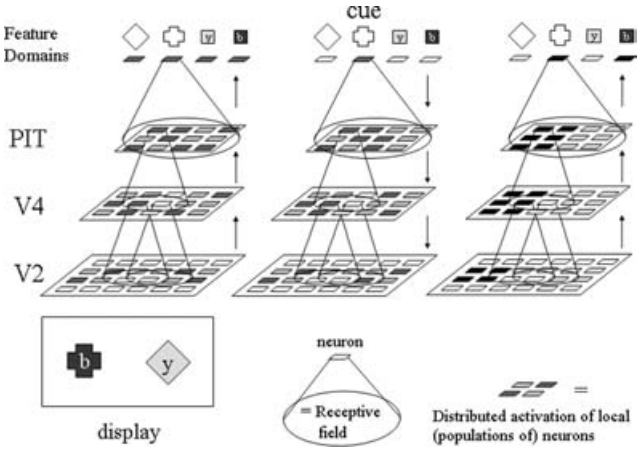


Figure 22. The process of answering the binding question "What is the color of the cross?" in the neural blackboard architecture of Figure 21. Left: The shapes, cross and diamond, and the colors, blue (b) and yellow (y), of two objects are processed in feed-forward pathways in the retinotopic areas. The receptive field size of neurons increases in higher areas (as indicated with the cones) until encoding is location-invariant in the feature domains. Middle: The shape of the target object (the cross) is selected as a cue in the shape feature domain. The selected cue initiates feedback activation in the retinotopic areas. Right: Interaction between feed-forward and feedback activation in the retinotopic areas results in the selection (enhancement) of the activation related to the target object in these areas. In turn, this results in the selection of the other features of the target object (its color, in this example) in the feature domains. In this way, the features of the target object ("cross" and "blue") are bound by the interaction in the neural blackboard architecture.

selection of spatial information (e.g., location of objects) and spatial transformations (e.g., for making eye movements). Both pathways start from the primary visual cortex (V1), but they are also interconnected on the levels of V2, V4, and PIT. Both pathways project to the prefrontal cortex.

Figure 22 (left) illustrates how the shape and the color of two objects, a blue cross and a yellow diamond, would be processed in this architecture. After the primary visual cortex V1 (not shown), the features are processed initially in a feed-forward manner (Oram & Perrett 1994). Each object produces a pattern of distributed activation in the areas V2 to PIT that corresponds to the retinotopic location of the object. The activated neurons could respond to one feature (e.g., shape) or to conjunctions of features, such as conjunctions of elementary shapes and color (Motter 1994).

The retinotopic object information in the lower layers is gradually transformed into location invariant information because of the increase in the receptive field size from layer to layer (illustrated by cones in Fig. 22). Furthermore, feature encoding is separated in the higher levels of the architecture, where distinctions are made between, for example, color encoding (e.g., blue versus yellow) and shape encoding (e.g., cross versus diamond). The distinctions between object features at this level form the basis for the constituents (parts) that are used to identify combinatorial visual structures.

In human cognition, object features as illustrated in Figures 21 and 22 form the basis for conceptual knowledge (e.g., Barsalou 1999; Barsalou et al. 2003). Human language provides ample evidence for the ability to encode object features like shape and color separately, that is, independent of any conjunction of these features. For example, we can use a word (e.g., red) to instruct a viewer to select an object in a visual display based on its color irrespective of its shape or location (see, e.g., van der Heijden et al. 1996).

7.1. Feature binding

Figure 22 illustrates the binding of shape and color in the blackboard architecture (binding of other features proceeds in a similar manner). The shape of the cross is given as a cue, for example, by the binding question "What is the color of the cross?" The binding process in the architecture consists of an interaction between a feed-forward network and a feedback network.

The feed-forward network (Fig. 22, left) processes the visual display, which results in the identification of the features of the cross and the diamond in the feature domains. The activation pattern in the feed-forward network that produces object identification is object-selective; that is, when an object is presented on a particular location in the display, it produces a pattern of (distributed) activation in the retinotopic areas in the feed-forward network. This pattern of activation is sufficiently different from the pattern of activation produced by another object presented on the same location in the display (otherwise a selective identification of the object could not succeed).

The feedback network in Figure 22 (middle) carries information about the selected feature (cue) from the feature domains back to the lower retinotopic areas in the architecture. The feedback network should be seen as lying "on top of" the feed-forward network; that is,

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

neurons in the retinotopic areas of the feed-forward network have corresponding neurons in the retinotopic areas of the feedback network. The corresponding neurons in both networks could belong to different layers of the same cortical column. Feedback connections are found between almost all areas in the visual cortex (e.g., Felleman & van Essen 1991).

Through the feedback connections, information processed at the level of object features (Fig. 21) can interact with information processed in the lower retinotopic areas. Hence the blackboard nature of the visual cortex, as discussed here, results basically from the feedback connections in the visual cortex (Bulier 2001; van der Velde 1997). The activation patterns in the feedback network are also object-selective. This can be achieved by adapting the connections in the feedback network with Hebbian learning (see below) using the selective activation patterns in the feed-forward network that occur in the process of object identification (van der Velde & de Kamps 2001).

The cue-related information in the feedback network (Fig. 22, middle) interacts with the processing of the display in the feed-forward network (Fig. 22, left). The interaction enhances ("selects") the neural activation related to the cue (cross) in the retinotopic areas. The selection of cue-related activation results from the match between the object-selective activation in the feed-forward network and the object-selective activation in the feedback network, as produced by the Hebbian learning procedure in the feedback network described above. The enhanced (selected) cue-related activation in the retinotopic areas can be used to select the color (and the other object features) of the cued object (cross) in the feature domains, as illustrated in Figure 22 (right).

The process presented in Figure 22 is an illustration of the fact that the basis for the blackboard architecture in Figure 21 is given by the interaction between the retinotopic areas, in which elementary information about the features of an object is combined, and the feature domains, in which identity information of object features is separated. In general terms, the visual features of an object can be bound in a combinatorial manner by selecting a feature (e.g., its shape or color) in one of the feature domains. Using an interaction process as described above, the activation related to a selected object feature will be enhanced in the retinotopic areas. In turn, this enhanced activation can be used to produce the selection of the other features of the object in the feature domains. In particular, a novel combination of familiar visual features (e.g., a purple cow)[16] can be identified in this way.

7.1.1. A simulation of feature binding. A simulation of the process of feature binding illustrated in Figure 22 is presented in Figures 23 and 24. (The simulation is based on van der Velde & de Kamps 2001 and de Kamps & van der Velde 2001b.) The display in this case consists of a red cross (at the location top left) and a green triangle (at the location bottom right). The figure illustrates the binding of the color (red) and the shape (cross) of an object in the display when the shape is given as a cue (i.e., the answer to the binding question "What is the color of the cross?").

The left panel in Figure 23 shows two layers in a feed-forward network that can identify shapes and colors of the objects in a display. The network also contains an

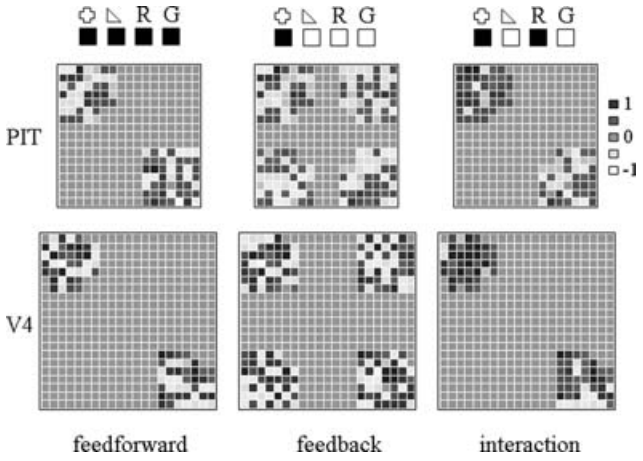


Figure 23. Left panel: Distributed retinotopic activation produced in two layers (V4 and PIT) of a feed-forward network that can identify shapes and colors (here, cross, triangle, R, red; G, green). Middle panel: Distributed retinotopic activation in the two layers of the corresponding feedback network when the shape of one object (cross) is selected as a cue. Right panel: Match between the retinotopic activation in the feed-forward network and the retinotopic activation in the feedback network. The match can be used to select (bind) the shape (cross) and the color (red) of the same object (see Fig. 24). Each small square within a layer represents the activation of one neuron. The object display consists of a red cross at the location top left and a green triangle at the location bottom right.

input layer (V1) and the layer V2 between V1 and V4 (not shown here; see de Kamps & van der Velde 2001b). Each small square within a layer represents the activation of one neuron. The input layer (V1) consists of a 24-by-24 matrix in which each element represents a V1 receptive field (RF). For each RF in V1 there are four input neurons that each encode one of four line orientations (vertical, horizontal, left diagonal, or right diagonal) and three input neurons that each encode one of three colors (red, green, or blue). The area V2 (not shown) consists of 529 neurons, arranged in a 23-by-23 matrix. Each of these neurons has an RF that covers a (unique) 2-by-2 submatrix of V1 RFs (529 in all). Hence each neuron in V2 is connected to the 4-by-7 V1 neurons in its RF. In turn, V4 consists of a 21-by-21 matrix of neurons (441 in all). Each V4 neuron has an RF that covers a (unique) 4-by-4 submatrix of RFs in V1 (441 in all). Hence a V4 neuron is connected to all (9) V2 neurons that have RFs that are fully covered by the RF of the V4 neuron. PIT consists of a 17-by-17 matrix of neurons (289 in all). Each neuron in PIT has an RF that covers a (unique) 8-by-8 submatrix of RFs in V1 (289 in all). Hence a PIT neuron is connected to all (25) V4 neurons that have RFs that are fully covered by the RF of that PIT neuron. The RFs of the identity neurons in the network (cross, triangle, red, and green in Fig. 23) fully cover all RFs in V1. Hence each neuron in the top layer is connected to all neurons in PIT.

The feed-forward network is trained (with back-propagation; Rumelhart et al. 1986) to identify the shapes and colors separately on all of the four potential locations in the display (top left, top right, bottom left, bottom right). The network used is an artificial network in which activation values of neurons range from −1 to 1. Networks of this kind can be transformed into networks of

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

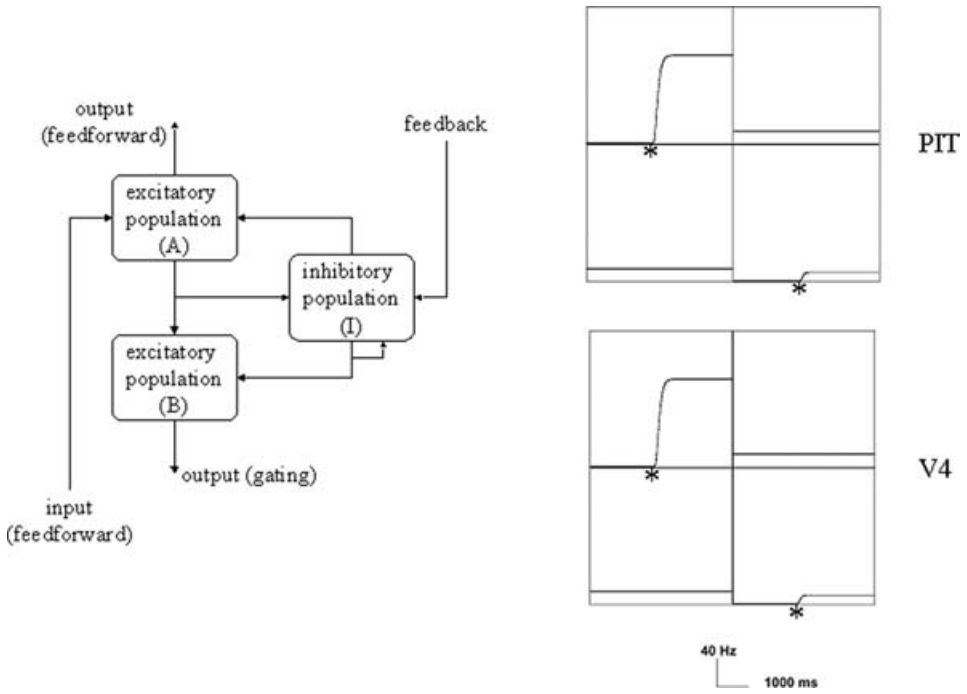


Figure 24. Left: Local circuit for the interaction between the feed-forward and feedback networks in Figure 23. The circuit consists of populations of excitatory (A, B) and inhibitory (I) neurons. Each population receives a small background activation from outside the circuit. Right: Retinotopic activation of population B produced by the circuit in the areas V4 and PIT. The activation is related to the four potential object locations in the display. Feed-forward activation is present from the beginning (and causes inhibition of the B population due to the increased activation of the I population in the circuit). The onset of feedback activation is indicated with an asterisk.

populations of spiking neurons (see de Kamps & van der Velde 2001a). The left panel in Figure 23 illustrates the pattern of activation in the feed-forward network that results when a display of a red cross (at the location top left) and a green triangle (at the location bottom right) is presented to the network. Each object produces distributed retinotopic activation in V4 and PIT (and V2, not shown here) and it activates the identity neurons for cross, triangle, red, and green.

The middle panel in Figure 23 shows the corresponding layers in a feedback network that propagates (top down) cue-related activation to the lower areas. The feedback network has the same connection structure as the feed-forward network, but with reciprocal connections. The feedback network is trained with the activation in the feed-forward network as input. In this procedure, the feed-forward network identifies a shape or a color presented on a given location. The resulting distributed activation pattern is then used to modify the connection weights in the feedback network with Hebbian learning (e.g., Brunel 1996). The procedure is repeated for various shapes and colors on all potential locations (for details, see de Kamps & van der Velde 2001b).

The rationale behind this procedure is that a feed-forward network can identify a shape or a color only if that shape or color produces a selective activation pattern in that network (selective enough for correct identification to occur). The selective activation pattern produced in the feed-forward network is the result of learning in that network. The actual learning procedure used is in fact irrelevant, because any learning procedure will have to produce a selective activation pattern in the feed-forward network for identification to occur. By using the Hebbian learning procedure, as described above, the shape or color selectivity in a feed-forward network can be transferred to a corresponding feedback network.

The middle panel in Figure 23 illustrates the distributed retinotopic activation in the feedback network that is produced when the cross is selected as a cue. The retinotopic activation in the feedback network is related to all four potential object locations in the display. This "fan-out" of activation results from the "fan-out" connection structure of the feedback network. It also reflects the fact that top-down selection of a cue is location-invariant.

The right panel in Figure 23 illustrates the match between the retinotopic activation in the feed-forward network and the retinotopic activation in the feedback network. Local match in activity is given by the product of the activation in both networks, and it varies from positive match (+1) to negative match (−1). Inspection of the figure suggests that there is a higher match of retinotopic activity between the feed-forward network and the feedback network related to the cued object (red cross) in the display, compared to the match of retinotopic activity between the feed-forward network and the feedback network related to the distractor object (green triangle) in the display.

Figure 24 illustrates that there is indeed a higher match of retinotopic activity between the feed-forward network and the feedback network related to the cued object in the display (the red cross at the location top left). Figure 24 (left) illustrates the circuit that the feed-forward network and the feedback network interact with locally (i.e., for each retinotopic activation related to one

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

of the potential object locations in the display). The circuit consists of interacting populations of excitatory neurons (A, B) and inhibitory neurons (I). The (inhibitory) connection from I to I represents the existence of disinhibition circuits in population I. The stimulus produces feed-forward activation of stimulus- (shape, color) selective neurons in population A. The cue produces feedback activation of cue-selective neurons in population I in a manner that results in disinhibition of stimulus-selective neurons in the circuit when the cue matches the stimulus, and inhibition of stimulus-selective neurons in the circuit when the cue does not match the stimulus. Further details of the circuit are presented in Appendix A2.

Figure 24 (right) illustrates the retinotopic activation of the B population produced by the circuit in the areas V4 and PIT. The retinotopic activation is related to the four potential object locations in the display. The activation related to the cued object (the red cross at the location top left in Fig. 23) clearly stands out, which demonstrates the selection of cue-related activation by means of the interaction between the feed-forward network and the feedback network. The role of the B population in the circuit is comparable to the role of the neuron X_{out} in the gating circuit presented in Figure 3. In this case, the cue-related activation of the B population can be use to produce (additional) activation within each (feed-forward) network that processes cue-related information. The (additional) activation will result in the selection (binding) of the features of the cued object, such as the color (and shape) of the cued object illustrated in Figure 23.

As in the case of the architecture for sentence structure, binding in the visual architecture illustrated in Figure 23 is not a state of the system but the result of a process initiated with a "binding question" (i.e., the selection of one of the features of the object as a cue).

A similar interaction process as illustrated in Figures 23 and 24 was used by van der Velde and de Kamps (2001) in a model of object-based location selection (e.g., for making eye movements). The model integrated the results of a number of experiments that investigated aspects of object-based selection in the monkey visual cortex. In particular, the selection of a cue in the higher levels (AIT) of the ventral pathway (e.g., Chelazzi et al. 1993), the interaction between cue and stimulus information in the retinotopic areas of the ventral pathway (e.g., Motter 1994), and the selection of the location of the cue-related object in the dorsal pathway (e.g., Gottlieb et al. 1998).

7.2. A neural blackboard architecture of visual working memory

Feature binding, as discussed above, could also occur in visual working memory. Neuroimaging studies in humans have shown overlapping areas of activation in the prefrontal cortex (PFC) with spatial and object memory tasks (e.g., D'Esposito 2001; Prabhakaran et al. 2000). Neurons that selectively respond to both identity and location information have been found in monkey PFC as well (Rainer et al. 1998; Rao et al. 1997). These results indicate an integrative role for (lateral) PFC in memory tasks (Fuster 2001; Duncan 2001).

A combined selectivity of spatial and object information in PFC is in line with the notion of a blackboard architecture for visual working memory. The neurons in a blackboard of visual working memory will respond selectively to combined (elementary) object and location information in a similar way to the neurons in the retinotopic areas of the visual cortex. Figure 25 (left) illustrates a putative connection between both blackboard architectures. One or more areas in the blackboard of the visual cortex (e.g., PIT) could be connected to a working memory (WM) blackboard in lateral PFC. A display of objects could then be encoded in both areas in a similar manner. The difference between the two areas will be found in the nature of the activation. Whereas the activation in the blackboard of the visual cortex results from the processing of the visual display, the activation in the WM blackboard is a form of self-sustained or reverberating activity in line with WM activity found in PFC (Durstewitz et al. 2000; Fuster 1995).

7.2.1. Feature binding in visual working memory. The reverberating activity in the WM blackboard can be used to retrieve (select) and bind the features of the objects in a visual working memory task. Figure 25 (right) illustrates that selection and binding of features (again) results from interactions between a blackboard and neurons that encode object features. These neurons could be located in PFC as well (e.g., Wilson et al. 1993), but they could also consist of the neurons that encode object features in the visual cortex. In the latter case, visual working memory will consist of interactions between neurons in PFC and neurons in posterior visual areas (Ruchkin et al., 2003).

The nature of the WM blackboard produces the behavioral effects reported by Luck and Vogel (1997), who observed that the number of objects that can be maintained in working memory is limited, but the number of their features is not. In terms of the WM blackboard, too many objects in a display will cause an interference between their distributed activations in the WM blackboard. This interference results in a limit to the number of objects that can be maintained in working memory, as

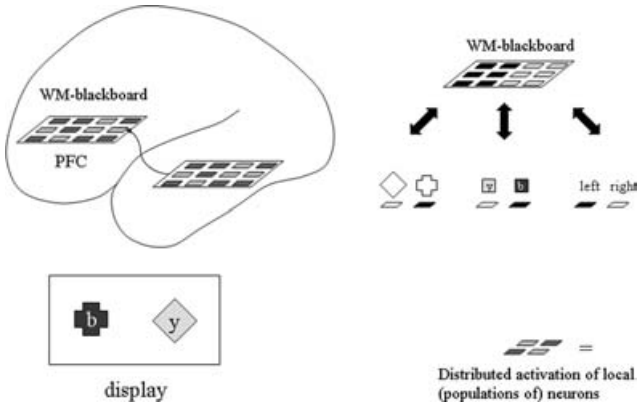


Figure 25. Left: A putative relation between the neural blackboard architecture in the visual cortex and a neural blackboard architecture in visual working memory (WM) in the prefrontal cortex (PFC). Right: An interaction between feature domains and the WM blackboard in PFC can be used to bind the features of an object (e.g., "cross," "blue," and "left") in working memory, similar to the binding process illustrated in Figure 21. (b, blue; y, yellow).

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

simulated by van der Voort van der Kleij et al. (2003). However, the number of features for each object is not limited. That is, all features of an object can be selected by means of the interaction with the blackboard (Fig. 25, right) as long as the object activations in the WM blackboard do not interfere.

7.3. Feature binding in long-term memory

Feature binding in visual working memory, as described above, is instantiated in terms of the sustained activation in the WM blackboard. As discussed in section 2.4, this raises the question of how feature binding can be achieved in terms of synaptic modification, which forms the basis of long-term memory. In the case of linguistic structures, this question was answered in terms of the process illustrated in Figure 11. The answer proceeds along similar lines for visual feature binding.

Figure 26 (left) illustrates the role of the HC in the case of visual features. A neuron in the HC forms a conjunctive encoding of the object features that are activated by a display of two objects (a blue cross on the left and a yellow diamond on the right). In this way, the neurons that encode the object features can be reactivated when the neuron in the HC is reactivated. However, it is clear that the conjunctive encoding by the HC neuron results in the familiar binding problem (von der Malsburg 1987), because the relations between the object features are lost in this form of encoding. A display of, say, a yellow cross and a blue diamond (at any of the two locations) would activate the same object features and would hence be encoded in the same way by the HC neuron in Figure 26 (left).

However, as in Figure 11, the relations between the object features can be encoded by the HC neurons if a neural blackboard (e.g., the WM blackboard) is included in the conjunctive encoding, as illustrated in Figure 26 (right). In this case, the relationship information is part of the input to the HC (as described by O'Reilly & Rudy 2001), so the HC can encode the relationship information (the blackboard) together with the object features. When the HC neurons reactivate the blackboard and the object features, the relations between the features of the objects in the display can be retrieved in the manner illustrated in Figure 25. The encoding of different events (episodes) in this architecture can proceed in a manner similar to the process illustrated in Figure 12.

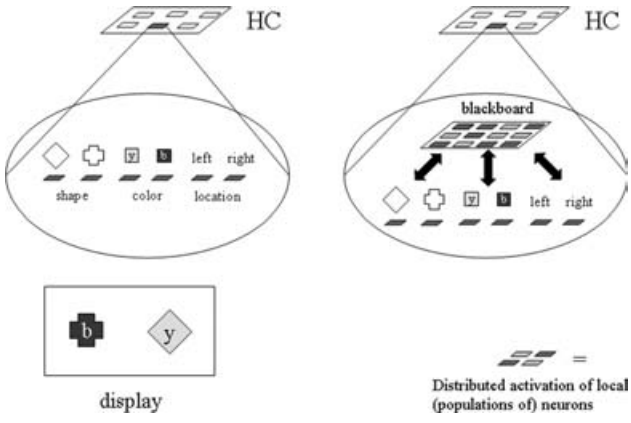


Figure 26. Left: Conjunctive encoding of the object features (e.g., shape, color, location) of two objects with a neuron (or group of neurons) in the hippocampal complex (HC). Right: Conjunctive encoding of the object features and a neural blackboard with a neuron (or group of neurons) in the HC. (b, blue; y, yellow).

Figure 26 illustrates again how a blackboard architecture can play an important role in the storage of combinatorial structures in long-term memory (i.e., in terms of synaptic modification). Even a conjunctive encoding as provided by the HC is sufficient if the blackboard activations are included in the encoding. In fact, as in Figure 11, the encoding of the blackboard alone would suffice.

Furthermore, Figure 26 again illustrates the importance of using delay activity as a binding mechanism. The sustained activity in the WM blackboard provides the time for the synaptic modifications (e.g., LTP) to occur (e.g., in the order of a second; Rolls & Treves 1998). In contrast, if synchrony of activation is used to bind features in visual working memory (e.g., Luck & Vogel 1997; Raffone & Wolters 2001), it is not clear how the relations between the features can be preserved in the transition from working memory to long-term memory, that is, how information encoded with synchrony of activation can be stored in terms of synaptic modifications. If the HC forms a conjunctive encoding of the neurons that are active in a time window (event) of about 1 second (Rolls & Treves 1998), it will form a conjunctive encoding of the features of all objects in a display, in the manner illustrated in Figure 26 (left). In that case, the relations between the features expressed with synchrony of activation are lost in the transition from working memory to long-term memory.

7.4. Integrating combinatorial structures in language and vision

A long-standing issue in cognition is the relation between visual processing and language processing (e.g., Bloom et al. 1996). We briefly touch upon that issue by discussing the combinatorial structure of *The little star is beside a big star* in terms of the architectures in Figures 21 and 25, as illustrated in Figure 27. The visual structures in Figure 27 should be combined with the sentence structure in Figure 10. In particular, the neural assemblies for words will be connected to neurons that encode visual features or visual operations (e.g., translations, visual selections). Figure 27 illustrates in a schematic fashion how the question "Where is the little star?" can be answered in this way.

In Figure 27a, the word *star* has selected the shape of the star as a cue in the shape feature domain. As a result, the cue-related activation in the feature domain is enhanced (in line with Chelazzi et al. 1993). In terms of the process illustrated in Figures 22, 23, and 24, the selection of the cue initiates an interaction in the visual blackboard (B), where the information of the visual display is processed (or maintained, as in Fig. 25). The interaction produces an enhancement of cue-related activation in the blackboard, which results in the selection (enhancement) of the neurons in the location domain (L) that encode the locations of the stars (van der Velde & de Kamps 2001). The neurons in this domain also encode the spatial magnitude of the objects in a display. The word *little* selects the location of the little star in L (Fig. 27b).

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

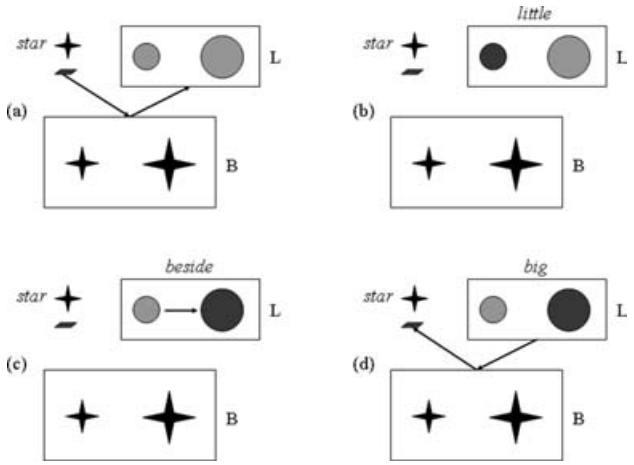


Figure 27. The structure of *The little star is beside a big star* in the neural blackboard architectures of Figures 21 and 25. a. Selection of the shape of the star (related to the word *star*) in the shape feature domain, which results in an interaction in the blackboard (B) and a selection of location and size information in the location (L) domain. b. Selection within the location domain of the smaller size (related to the word *little*), which is now the focus of attention. c. Shift of attention (related to the word *beside*) in the location domain to the location beside the attended location in b. The newly selected location is now the focus of attention. d. Feedback activation from the location domain interacts with the activation in the blackboard. This results in the selection of the feature in the shape domain (star, related to the word *star*) that corresponds with the location (and the size, related to the word *big*) of the newly attended object in c.

In Figure 27c, a shift of spatial attention is produced in the location domain. As a result, the neural activity that encodes the location of another object in the vicinity of the attended location in Figure 27b is enhanced. If spatial information is (initially) encoded in eye-centered coordinates (Batista et al. 1999), a shift of spatial attention will produce a spatial transformation in terms of eye-centered coordinates. The spatial transformation involved can be used to activate the associated word assembly (*beside*).

In Figure 27d, the newly selected neural activity in the location domain can be used to obtain a measure of the spatial magnitude of the newly attended object (big, in comparison with the previous object). It can also be used to influence processing in the blackboard so that the shape of the newly attended object can be selected in the shape feature domain. Both selections can produce the activation of their associated word assemblies (*big* and *star*).

The process illustrated in Figure 27 would operate in a similar manner with a structure like *The little triangle is beside a big triangle*. The only difference would be the selection of the shape of the triangle in the shape feature domain. The processes in the other domains are not affected by the change from star to triangle. Hence, for example, the attention shift in Figure 27c operates in the same manner for the shift from a little star to a big star as for the shift from a little triangle to a big triangle, because it is based on only the information in the location domain. Likewise, in a structure like *The little diamond is above a big square*, the process in Figure 27a is affected by only the (initial) selection of the shape of the diamond (instead of the star or the triangle), and the process in Figure 27c is affected by only the nature of the spatial transformation (above, instead of beside).

The similarity between these examples emphasizes the combinatorial nature of the process illustrated in Figure 27. Each of the individual processes operates only on information that is available in its own domain. However, by using the blackboard, a process in one domain can influence the processes in the other domains. In this way, a combinatorial structure can be produced by the architecture as a whole. For example, with *The little diamond is above a big square*, the attention shift in Figure 27c will produce the square as the second object selected in the shape feature domain (instead of the star or the triangle in the other examples) by the interaction process in the blackboard illustrated in Figure 27d.

7.5. Related issues

The architecture for combinatorial structures in vision discussed here is related to a number of issues that we can only briefly mention. The interactions between object (feature) information and spatial information, illustrated in Figure 27, have a clear relation to attentional processes in the cortex (e.g., as in the "bias competition" model of attention; Desimone & Duncan 1995). The blackboard architecture in the visual cortex (Fig. 21) and the blackboard architecture of visual working memory (Fig. 25) can be combined in a "closed-loop attention model" (van der Velde et al. 2004a). Another issue is the question of how and to what extent the architectures for combinatorial structures discussed here are related to processing of object identification (e.g., Biederman 1987; Edelman & Intrator 2003). Finally, the interaction between neural sentence structures (Fig. 10) and visual blackboard structures, as illustrated in Figure 27, could also form the basis of a model that combines pictorial and propositional aspects of mental imagery (for a discussion on that topic, see Pylyshyn 2002; van der Velde and de Kamps 2002a).

8. Conclusion and perspective

The aim of our paper is to show that the problems described by Jackendoff (2002) can be solved by means of neural blackboard architectures. To this end, we have discussed and illustrated how two kinds of combinatorial structures (linguistic and visual) can be instantiated in terms of neural blackboard architectures. As can be expected, there are clear structural differences between these architectures, which derive from the nature of the information processing in which they are involved (in this case, the spatial arrangement of visual features in a visual display versus the sequential arrangement of words in a sentence). However, there are also important similarities between the different blackboard architectures.

One similarity concerns the solution to the binding problem in each architecture. In both architectures, the

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

binding problem is solved in terms of a process that answers specific "binding questions" related to the binding at hand. This process consists of a selective flow of activation. Hence the binding of features in the visual blackboard architectures consists of a selective flow of activation from one feature domain to another, determined by the interaction process in the blackboard. Likewise, the blackboard architecture for sentence structure produces a selective flow of activation in the process of answering a "binding" question.

Another similarity between the architectures concerns the transition from working memory to long-term memory and the role of delay activity as a binding mechanism. Combinatorial structures can be stored in long-term memory (using synaptic modification) when the blackboard activity is included in a conjunctive form of encoding as provided by the HC. The delay activity in the blackboard provides the time for the synaptic modifications (Hebbian learning) to occur, even in the case of one-trial learning.

We also discussed and illustrated the potential for further development of the architectures we presented. It is clear that a substantial amount of work is needed to fulfill this potential. However, concerning the problems he discussed, Jackendoff (2002, p. 64) noted that "some further technical innovation is called for in neural network models ... upon the development of such an innovation, the dialogue between linguistic theory and neural network modelling will begin to be more productive." The examples of the neural sentence structures we discussed illustrate how such a dialogue could proceed.

We would argue that this technical innovation is also needed for the development of cognitive neuroscience models. In the preface to *The Cognitive Neurosciences*, Gazzaniga (1995, p. xiii) described the aim of cognitive neuroscience as follows:

> At some point in the future, cognitive neuroscience will be able to describe the algorithms that drive structural neural elements into the physiological activity that results in perception, cognition, and perhaps even consciousness. To reach this goal, the field has departed from the more limited aims of neuropsychology and basic neuroscience. Simple descriptions of clinical disorders are a beginning, as is understanding basic mechanisms of neural action. The future of the field, however, is in working toward a science that truly relates brain and cognition in a mechanistic way.

If the ultimate aim of cognitive neuroscience is to deliver detailed neural models of cognitive processes, the question arises as to how such models can be developed and tested.[17] In some cases, such as visual processing, an animal model exists that can be studied with the kind of experiments described in section 7.1.1. But for language and other aspects of high-level human cognition, an animal model is missing, which excludes the kind of rigorous investigation of neural activity on a cell level that is possible with visual processing. In Figure 8, we presented the neural activity of some of the assemblies presented in Figure 7b. As the lines and labels in Figure 7b suggest, one can see the activity presented in Figure 8 as the result of virtual electrodes inserted in the model. In a similar way, one could compare the model with observations made with real electrodes, except for the considerations that prevent such a testing. On the other hand, neuroimaging methods such as EEG and functional magnetic resonance imaging (fMRI) provide rather coarse observations of neural activity, not (yet) on the level of the detailed neural mechanisms needed to relate brain and cognition in a mechanistic way.

If the aim of cognitive neuroscience is to relate brain and cognition in a mechanistic way, then the gap that seems to exist between detailed neural mechanisms and the current methods of observation is a problem for both modeling and neuroimaging.[18] In our view, the best way to close this gap is the integration of modeling and neuroimaging that we described in section 6.9. In short, on the basis of detailed models, activation processes can be derived that can be compared with neuroimaging observations. In turn, these observations can be used to modify or change the existing models. This process is technically demanding and can proceed only in a step-by-step manner, but it may be the only way to fulfill the aim of cognitive neuroscience as described by Gazzaniga.

However, the success of this process depends on further advances in both neuroimaging and modeling. In neuroimaging, further development of techniques such as dynamic causal modeling (e.g., Friston et al. 2003) and further integration of EEG and fMRI are needed. In modeling, neural models are needed that can capture important aspects of human cognition, such as the productivity and systematicity that derives from the ability to process combinatorial structures.

Hence it is clear that a substantial amount of work, both theoretically and empirically, is required to develop an understanding of the neural basis of combinatorial structures in human cognition. However, the similarities between the potential neural instantiation of combinatorial sentence structures and the (more familiar) neural instantiation of combinatorial structures in visual cognition, as described here, provide the hope that such a development can be successful in the near future.

APPENDIX

A1. The architecture for sentence structure

The simulations presented in Figure 8 are based on excitatory (E) and inhibitory (I) neuron populations (e.g., Gerstner 2000; Wilson & Cowan 1972). The activation of population i is modeled with the population rate A_i, defined as the fraction of neurons that fire in the time interval [t, t + dt] divided by dt. A_i is given by:

$$\tau_\alpha \frac{dA_i}{dt} = -A_i + F(\Sigma_j w_{ij} A_j) \quad (1)$$

τ_E ($\alpha = E$) = 10 ms and τ_I ($\alpha = I$) = 5 ms are the time constants for the excitatory and inhibitory populations. The w_{ij} (or $w_{j\to i}$) is the efficacy from population j onto population i (w_{ij} is negative if j is an inhibitory population). $F(x)$ is given by:

$$F(x) = \frac{f_{max}}{(1 + e^{-\beta(x-\theta)})} \quad (2)$$

with $f_{max} = 500$ Hz, $\beta = 1$ and $\theta = 5$.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

The gating circuit in Figure 3 (in the direction *X* to *Y*) is given by the following equations, derived from (1) and (2):

$$\tau_E \frac{dX}{dt} = -X + F(input_X + w_{Y_{out}\to X} Y_{out}) \quad (3)$$

$$\tau_E \frac{dX_{out}}{dt} = -X_{out} + F(w_{X\to X_{out}} X - w_{i_x \to X_{out}} i_x)$$

$$\tau_I \frac{di_x}{dt} = -i_x + F(w_{X\to i_x} X - w_{I_x \to i_x} I_x)$$

$$\tau_I \frac{dI_x}{dt} = -I_x + F(control_{XtoY})$$

The efficacies used were $w_{X\to X_{out}} = w_{X\to i_x} = 0.20$, $w_{i_x\to X_{out}} = w_{I_x\to i_x} = 1$. $w_{Y_{out}\to X}$ is variable; its value depends on where the gating circuit is employed in the network. It will be referred to as the efficacy of the gating circuit. The gating circuit can be activated by the input signal $control_{XtoY}$ from an outside population, with activation $f = 30$ Hz and $w_{control} = 1$. The memory circuit in Figure 3 was simulated as a gating circuit, with control signal 0 ("off") or f ("on"), and $w_{X\to X_{out}} = 0.2$. The gating and memory circuit in the direction *Y* to *X* follow from the above equations by changing the labels *X* to *Y*, *Y* to *X* and *x* to *y*.

We did not explicitly model delay activity. Instead, we assumed that a delay population will be activated if its input is above threshold θ_{delay}. Time constant $\tau_{delay} = \tau_E$ in this case. With high input, activity will decay until it reaches a default $\theta_{default}$. It will be deactivated if the total input is below a negative threshold θ_{deact} (i.e., net inhibition). Memory activity will decay within approximately τ_E ms. We took $\theta_{deact} = -0.2$, $\theta_{delay} = 4$, $\theta_{default} = 2.0$ and $\tau_{delay} = 10000$ ms. We also used a parameterized version of the CPG, which reproduces observed behavior of CPGs but which allows for a simple treatment of them within our simulation.

A structure assembly consists of a central delay population (main assembly) and three external populations: an input population (word assembly), an agent population (subassembly), and a theme population (subassembly), as in Figure 2. The external populations are connected to the main assembly by gating circuits, with efficacies $\varepsilon_{input} = 0.05$, and $\varepsilon_{agent} = \varepsilon_{theme} = 0.01$. Each assembly consists of eleven populations, including eight constituents of the gating circuits.

An assembly group consists of a number of assemblies that belong together and that share a central inhibitory population (Fig. 4). There are two assembly groups, one for the nouns (six assemblies) and one for the verbs (five assemblies). They are indicated by the N_x and V_i symbols in Figure 8. The agent, theme, and input population of each assembly feed the inhibitory population with $efficacy_{competition} = 0.005$. The inhibitory population acts on all main assemblies with a "competition parameter," controlled by the CPG ("on" = 0.2; "off" = 0).

The six agent subassemblies of the noun assembly group are each connected to (memory) gating circuits with efficacy $\varepsilon_{agent} = 0.008$ to each of the five agent subassemblies of the verb assembly group as shown in Figures 4 and 5. Similarly for the theme subassemblies, with $\varepsilon_{theme} = 0.015$.

In all, the simulated model consisted of 641 populations. Integration of the system of equations (3) evolved simultaneously for the entire model, using fourth-order Runge-Kutta integration with an integration time step h = 0.01 ms.

A2. The local interaction circuit

The equations for the populations A, B, and I in the local interaction circuit in Figure 24 are:

$$\tau_s \frac{dI_A}{dt} = -I_A + J_{A\to A} F(I_A) - J_{I\to A} F(I_I) + I_{ff} + I_{bg}$$

$$\tau_s \frac{dI_B}{dt} = -I_B + J_{B\to B} F(I_B) - J_{I\to B} F(I_I) + J_{A\to B} F(I_A) + I_{bg}$$

$$\tau_s \frac{dI_I}{dt} = -I_I - J_{I\to I} F(I_I) + J_{A\to I} F(I_A) + (-m + nm) I_{fb} + I_{bg}$$

Here I_x is input current in population x, and τ_s is the synaptic time constant (5 ms). The efficacies J_x are: $J_{A\to A} = 0.01$, $J_{A\to B} = 0.07$, $J_{A\to I} = 0.3$, $J_{B\to B} = 0.01$, $J_{I\to I} = 0.01$ and $J_{I\to A} = 0.01$, $J_{I\to B} = 0.1$. The background activity to all populations is 3.4 Hz ($I_{bg} = 0.01$). The stimulus is given by a feed-forward input activity $I_{ff} = 25$ Hz to population A. Feedback is given by the cue-related activity $I_{fb} = 25$ Hz to population I, divided in the fractions match n and nonmatch nm (with n + nm = 1) as determined on the basis of Figure 23 (interaction).

NOTES

1. Tensor networks could perhaps be included here. However, tensor networks fail to instantiate combinatorial structures (Fodor & McLaughlin 1990). Basically, this results from the fact that a tensor is just a list of constituents organized in a particular fashion (i.e., as an n-dimensional list for a rank-n tensor). Any operation on a tensor consists of selecting a k-dimensional subset of the constituents in the tensor (with k £ n). But all selected subsets have to be listed in the tensor beforehand, which limits the instantiation of novel structures. Furthermore, adding constituents to the tensor increases the dimensions of the tensor, which requires adjustments to all components in the cognitive system that can interact with the tensor.

2. The sentence presented by Jackendoff (2002) is *The little star's beside a big star*, with the clitic z ('s) to emphasize the phonological structure of the sentence. Phonological structure is not discussed here; therefore the clitic z is omitted.

3. When a symbol is copied and moved elsewhere, it is detached from its network of relations and associations. One could try to reestablish these relations and associations from time to time, but this requires an active process executed by a control structure. Active control would be needed constantly to decide how many of these relations and associations have to be reestablished (and how often).

4. Pilot simulations showed that RNNs are very good at reproducing learned word-to-word associations. Hence, with the test sentence *boy hears girl*, we wanted to avoid combinations like *boy hears* and *hears girl* in the training sentences. Other than that, we wanted to train as many relations between these words as possible. In the case of this test sentence, the RNNs learned the relation *boy Verb girl*. Furthermore, they learned the relation *dog Verb girl* with *dog hears Noun*, and the relation *boy Verb cat* with *Noun hears cat*.

5. We are not aware of pathological behavior of this kind. Broca's aphasics, for example, often fail on sentences like *girl who dog hears obeys Mary*, but they can still understand sentences like *boy hears girl* (Grodzinsky 2000).

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

6. Verbs can have one, two, or three arguments, or thematic roles. Although in semantic terms many different kinds of arguments can be distinguished, they can be grouped into "semantic macroroles" (van Valin 2001). We will refer to these as "agent," "theme," and (later on) "recipient."

7. When an NP assembly is bound to a sentence structure, at least one of its memory circuits is active. This activation can be used as a signal that the NP assembly is not free, or one could have an "inhibition of return" that prevents the reactivation of a structure assembly activated recently.

8. The inhibition of the active NP assembly could result from initiating a competition between the NP assemblies. Because of its high transient activity, frequently found with a new stimulus, the new NP assembly could win the competition, or the occurrence of a new noun could result in the inhibition of the active NP assembly before a new NP assembly is generated.

9. The delay assemblies produce inhibition by activating inhibitory interneurons.

10. If treelike structures capture important aspects of sentence structures, neural instantiations of sentence structures can always be transformed into treelike structures. This is a direct consequence of having a neural instantiation of a sentence structure. It can be compared with the gradual transformation of one computer language into another. It is a matter of choice which of the intermediary structures are used to illustrate the transformation. A comparison of Figures 7a and 7b shows that it is more useful to discuss the architecture in terms of the structures illustrated in Figure 7a than those in Figure 7b. Yet both figures represent the same architecture and can be transformed into one another.

11. In linguistics, words are the terminal notes placed at the bottom of a sentence structure. Here they are placed close to their structure assemblies.

12. The structures in Figures 11 (left) and 1 (b and c) are not the same. Figure 11 (left) represents a conjunctive encoding that results from direct associations between each of the word assemblies and the HC neuron. In Figure 1 (b and c), a specific neural circuit will activate a "sentence" neuron when the word assemblies have been activated in the correct order. Circuits of this kind are much harder to develop than the conjunctive encoding in Figure 11.

13. Transitions from an attractor state into an associated attractor state have been observed in the cortex (Yakovlev et al. 1998).

14. A difference could be found in clause-based closure principles (see, e.g., Gibson 1998). For example, a relative clause can be closed without closing a matrix clause, but a matrix clause cannot be closed without closing a relative clause.

15. The connection structure illustrated in Figure 19 was not specifically designed to operate as a pushdown stack. Instead, it was designed to satisfy two constraints. First, the constraint of combinatorial productivity, which entails that every A subassembly should be able to bind with every B subassembly (and vice versa). This constraint is satisfied with the matrix-like array of columns in the connection structure. Second, the uniqueness constraint, which entails that a given A subassembly can bind to only one B subassembly (and vice versa). This constraint is satisfied with the inhibition within the horizontal and vertical rows of columns. The resulting connection structure operates as a pushdown stack if the reverberating activity in the structure decays over time.

16. The image of a purple cow is used in an advertisement campaign of a brand of milk chocolate bars sold in a purple wrap.

17. Perhaps we do not need to study the neural instantiation of cognition. In the view of "classical" cognitive psychology (e.g., Fodor & Pylyshyn 1988), implementation is irrelevant for the nature of a computational model of a cognitive process. A mathematical critique of this position can be found in van der Velde (2001). Here we point out that Fodor (2000) has argued that the computational theory of mind is incomplete – in fact, only a fraction of the truth. If so, it makes even more sense to study how computational processes can be implemented in the brain. We know that the brain produces cognition (it is the only example we are certain of). So if we had an idea of how computational processes could be matched onto brain processes, we could also get a clearer view of what could be missing in the computational account.

18. An alternative would be to reformulate the aim of cognitive neuroscience, or, which amounts to the same thing, to abandon cognitive neuroscience as described by Gazzaniga. We do not advocate such a position.

Open Peer Commentary

Conscious cognition and blackboard architectures

Bernard J. Baars

The Neurosciences Institute, San Diego, CA. 92121 ***baars@nsi.edu www.nsi.edu***

Abstract: van der Velde &de Kamps make a case for neural blackboard architectures to address four questions raised by human language. Unfortunately, they neglect a sizable literature relating blackboard architectures to other fundamental cognitive questions, specifically consciousness and voluntary control. Called "global workspace theory," this literature integrates a large body of brain and behavioral evidence to come to converging conclusions.

When Alan Newell and coworkers developed the first blackboard architectures in the 1970s, they also called them "global workspace architectures" (Newell 1990/1994). Global workspace theory (GWT) applied Newell's blackboard architecture to the empirical evidence bearing on consciousness and related issues in the 1980s and '90s and has since gained dramatic support from a wide range of brain imaging studies (Baars 1988; 1997; 2002; Baars et al. 2004). Recently, neural net models have expanded the theory and made it more testable in cognitive neuroscience experiments (Dehaene et al. 2003). Franklin and colleagues have built the first large-scale GWT-based model that is able to perform quite complex human tasks (Franklin 2003). Shanahan and Baars (in press) have applied GWT to the frame and modularity problems, widely discussed in cognitive science, and Baars and Franklin (2003) have shown how GWT can account for cognitive working memory. Finally, GWT converges well with neuroscience approaches to the problem of conscious cognition (e.g., Edelman & Tononi 2000).

The fact that two different terms are used for the same cognitive architecture may cause confusion. Because the target article is mainly concerned with language, this commentary will focus on ways in which GWT has made testable claims about other aspects of language as well. It should be noted in passing that the first large-scale application of a blackboard architecture in cognitive science, the Hearsay System by Newell and colleagues, was specifically designed to deal with speech recognition in noisy natural environments (Newell 1990/1994). Hence language has been a concern of blackboard theorists for thirty years.

The Hearsay blackboard architecture was chosen because in the 1970s the speech signal was highly ambiguous. The solution was to allow multiple autonomous knowledge sources – involving acoustics, vocabulary, phonotactics, syllable structure syntax, and so on – to cooperate and compete with each other in proposing "hypotheses" about the acoustical signal on a shared memory resource, the blackboard or global workspace. This approach was

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

surprisingly successful. Even though today's speech recognition systems are more predictive of acoustical formant patterns, language has a principled problem of ambiguity that would not be solved even by perfect acoustical processing. The reason, of course, is the massive ambiguity of the natural lexicon (as well as many other aspects of language). Neither speech recognition nor production is therefore fully determinate. "Expectation-driven processing" involving a mixture of semantic, pragmatic, discourse, lexical, world-knowledge, rhetorical, intonational, interpersonal, syntactic and phonetic-acoustical knowledge sources is required to solve this problem (Baars 1988).

Global workspace or blackboard architectures are especially useful for resolving ambiguities at *any* level of language, which may be addressed by knowledge at *any other* level, or what I have called the "any-any" problem (Baars 1988). Furthermore, it is easy to show that the problem of multiple ambiguities interacts with conscious cognition. Essentially all the common words of English are ambiguous, as one can see simply by consulting a dictionary. The *Oxford English Dictionary* devotes some 75,000 words to the many meanings of the word "set," for just one example. Yet it is striking that skilled speakers of English are rarely conscious of the sea of ambiguity in which they swim. Most input ambiguities are resolved unconsciously, though one can show both behavioral and brain correlates of the degree of ambiguity that needs to be resolved. Ambiguities can sometimes become conscious, as in the case of jokes, puns, and language games as well as genuine difficulties in understanding speech. But these are relatively rare events. More commonly, unconscious expectation-driven processing selects the most plausible interpretation, which is the one that becomes conscious.

These basic facts can be handled naturally by GWT; because competition between incompatible knowledge sources makes it unlikely for ambiguous "hypotheses" to survive on the common blackboard. The hypotheses that survive long enough to be widely distributed ("broadcast") are the ones that command at least tacit acceptance form multiple knowledge sources. This is precisely what Newell's group found with the Hearsay system, and it has recently been found to work as well with other problems of choice and ambiguity in Franklin's large-scale Intelligent Distribution Agent (IDA) architecture. Hence there is a plausible functional fit between GWT and the question of conscious and unconscious language processing.

On the language output side there is a precisely analogous problem, but here it is the problem of choosing between near-synonyms in expressing a semantic proposition. Writers encounter this problem consciously, thinking about different ways of expressing a thought or using thesauri to find just the right word to fit the thought. But natural language universally allows for expressions that differ in significant but relatively subtle ways, including such considerations as politeness, active-passive constructions, suitability for different listeners, rhetorical conventions, and so on. The point is that *there is principled indeterminacy on the output side of language, just as there is on the input side*. As Franklin and colleagues have shown, the problem of action selection in autonomous agents, which is formally identical to the speaker's problem of choosing among near-synonyms, can also be resolved by a blackboard/GW architecture. It is again striking that humans cannot report on rapid expectation-driven speech selection processes but can become quite conscious of the final coherent speech plan, as shown by standard measures such as reportability.

In sum, the authors' arguments language could be strengthened by taking into account global workspace theory. Questions such as conscious cognition and voluntary control are testable and subject to theoretical analysis. They are central to language as well as other brain functions. It is most encouraging that the same fundamental architecture can account for aspects of language as well as other fundamental human capacities, including conscious cognition.

On the structural ambiguity in natural language that the neural architecture cannot deal with

Rens Bod,[a,b] Hartmut Fitz,[c] and Willem Zuidema[a]

[a]*Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam, 1018TV, The Netherlands;* [b]*School of Computing, University of Leeds, Leeds, LS2 9JT, UK;* [c]*Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam, 1012CP, The Netherlands.*

rens@science.uva.nl http://www.science.uva.nl/~rens
fitz@science.uva.nl http://amor.cms.hu-berlin.de/~h2784i25
jzuidema@science.uva.nl http://www.science.uva.nl/~jzuidema

Abstract: We argue that van der Velde's & de Kamps's model does not solve the binding problem but merely shifts the burden of constructing appropriate neural representations of sentence structure to unexplained preprocessing of the linguistic input. As a consequence, their model is not able to explain how various neural representations can be assigned to sentences that are structurally ambiguous.

Any native speaker's knowledge of language is productively combinatorial, in that from a finite set of building blocks (depending on one's linguistic theory, words, morphemes, rules, or constructions), an unbounded set of possible sentences can be generated. A major challenge for cognitive neuroscience is to explain how this *capacity* can be implemented and acquired in neural tissue. van der Velde's and de Kamps's (vdV&dK's) target article reinterprets this challenge as a question about *representations*: how single, unified neural representations of the building blocks can function as symbols, such that they can be used in many different structures, in multiple structures at the same time, and in multiple instantiations within the same structure.

It is not easy to evaluate whether they have succeeded in answering that question, not only because of a regrettable lack of clarity in the article about technical details but also because of an equally regrettable lack of clarity in the field about the evaluation criteria. For instance, in connectionism, showing how representations are *learned* is usually seen as an integral part of cognitive modeling. In theoretical linguistics, formalisms are judged on their ability to model syntactic phenomena such as agreement, long-distance dependencies, and cross-serial dependencies. The target article addresses neither of these problems thoroughly, but we believe it should not be dismissed for those reasons. Explanatory models do need to simplify – but without relegating the core issues to unexplained processes.

Unfortunately this is exactly the situation with "binding" in the vdV&dK model. The authors repeatedly point out that the binding mechanisms in their model instantiate primitive parsing operations and that their model is intended to be a process model. We find, however, that the crucial decisions to arrive at the appropriate binding are relegated to an unspecified preprocessor. This is particularly clear when we consider the widespread structural ambiguity of natural-language sentences – a property that we see as a major challenge for any neural architecture of natural-language processing. How do we construct the appropriate neural representations of the multiple syntactic structures that can be assigned to a single natural-language sentence?

vdV&dK demonstrate how sentences with the same sequence of word categories can be distinguished on a representational level. The authors fail to show, however, how such syntactic parses are constructed procedurally. In their model, the representations of all possible sentence structures are prewired in a gridlike fashion. Connections between structure assemblies (syntactic variables) are physically present but they are initially inactive. The process of binding thus boils down to activating the right gating and memory circuits.

Section 6.8.4 attempts to clarify this process, but how exactly the activation of these circuits is controlled remains unclear. Obviously, the juicy parts of this mechanism are the detection of lexical categories and the assignment of thematic roles to sentence constituents. This information is required to activate the

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

right kind of input nodes of the neural control circuits in order to bind word assemblies to structure assemblies and to bind different structure assemblies via their subassemblies. For instance "cat chases mouse" and "mouse chases cat" both have the same surface N-V-N structure and are distinguishable by the blackboard model only if agent and theme are assigned inversely. The underlying event structure, however, cannot simply be read off the word order as the authors seem to suggest. This is clear from notorious examples like "The man frightens the child" versus "The child fears the man." Both sentences describe the same event, but neither the position in the sentence nor the syntactic category signals the thematic role of "man" and "child."

Moreover, concurrent activation of assemblies for "mouse" and "chases" does not resolve whether "mouse" is the actor or the theme of "chases." Hence simultaneous activity as a binding mechanism, that is, "and-binding," works only if the assignment of thematic roles is already accomplished by a preprocessor. vdV&dK do not sufficiently comment on how this task is achieved. The proposed model therefore does not solve the binding problem but merely shifts the burden of constructing appropriate neural representations of sentence structure to unexplained preprocessing of the linguistic input. Consider, for instance, the sentence "The fact that the boy comprehends surprises the teacher." Unlike the very similar examples in the target article, this sentence does have two readings for the same N-that-N-V-V-N sequence. How the model "parses" this sentence depends entirely on the preprocessor's choices on whether "the fact" should be assigned the theme role of "comprehends" or not.

This shortcoming is particularly problematic for real-world ambiguous sentences. If vdV&dK's model already runs into trouble for simple three-word sentences, what would it make of real-world sentences for which there can be an interaction of several types of ambiguities, such as attachment ambiguities (like PP attachments and reduced clause attachments), functional ambiguities (like subject, object and indirect object assignments), and scope ambiguities (like different scope of adjectives). Any processing or parsing model for natural language should be able to tell us how various (neural) representations can be assigned to ambiguous sentences, and many parsing models even go further and also tell us how to select the preferred representation (e.g., Bod 1998; Charniak 1997);. vdV&dK's model does neither; as a parsing model for real-world sentences the approach is thus inadequate.

Should we, then, interpret the model as just a neurally inspired model of representation? The problem with such an interpretation is that without specifying the processes that do something with the representations – learning, interpretation, production, inference – it is unclear what the requirements on representations are. At the very least, representations should distinguish between all sentences that have different phrase structure. Unfortunately, the authors do not show that their formalism can make these distinctions for more than a handful of simple examples, nor do they analyze the expressive power of the formalism proposed and compare it with established results in theoretical linguistics (e.g., Chomsky 1957; Joshi et al. 1991). The real question that we believe remains unanswered is how networks of neurons, of the type that are found in the human brain, can construct appropriate representations and put them to use in actual language processing.

ACKNOWLEDGEMENT

Willem Zuidema is funded by the Netherlands Organisation for Scientific Research (Exacte Wetenschappen), project number 612.066.405.

How neural is the neural blackboard architecture?

Yoonsuck Choe

Department of Computer Science, Texas A&M University, College Station, TX 77843-3112. ***choe@tamu.edu*** ***http://faculty.cs.tamu.edu/choe***

Abstract: The target article does not provide insight into how the proposed neural blackboard architecture can be mapped to known neural structures in the brain. There are theories suggesting that the thalamus may be a good candidate. However, the experimental evidence suggests that the cortex may be involved (if in fact the blackboard is implemented in the brain). Issues arising from such a mapping will be discussed.

In the target article, van der Velde &de Kamps (vdV&dK) present a neural blackboard architecture for the representation of combinatorial structures in cognition. However, they do not provide much insight into how their proposed architecture (Fig. 3 in target article) can be mapped onto known neuroanatomy and neurophysiology.

In this commentary, I will assess the idea that the thalamus is a key structure to which the neural blackboard architecture can be mapped. In fact, there are existing theories in this line: In addition to the work of Newman and colleagues (1997), which is cited in the target article, Harth and colleagues (1987) and Mumford (1991; 1995) also propose that the thalamus may be a good candidate for such an internal blackboard or a sketchpad. The rational is that the thalamus receives dense feedback from all parts of the cortex and hence is an ideal place for the results of cortical processing to be written and combined. Hence the thalamus on its own may take on an integrative role under these theories. (Also see Choe 2004 for a possible role of the thalamocortical circuit in processing analogy.)

However, neuroscientists have argued that the thalamus does not play such an integrative role in which signals from multiple sources in the cortex converge to a small, local region of the thalamus (Sherman & Guillery 2001, p. 102). On the other hand, there is also evidence that there may be intrathalamic interactions that can allow some form of integration through the thalamic reticular nucleus containing inhibitory neurons (Crabtree et al. 1998). Hence it is unclear whether a blackboard, if it is in fact implemented in the brain, can be found solely within the thalamus or if other parts of the brain such as the cortex can also serve a role.

Here I will assess the compatibility of the gating circuit in the target article's Figure 3 with the thalamocortical circuit. The basic finding is that the model of vdV&dK needs both the cortex and the thalamus, as well as the thalamic reticular nucleus (TRN) as a neural substrate. Figure 1(a) shows a schematic diagram of the thalamocortical circuit based on known experimental findings (see Sherman & Guillery 2001 for a review). Figure 1(b) shows how the gating circuit in the target article's Figure 3 can be nicely mapped onto the thalamocortical circuit, where the major representations, such as part-of-speech and thematic role, and the control all come from the cortex, and the disinhibitory gating happens at the thalamic reticular nucleus (TRN) and the thalamic relays. A connection new in (b) compared to (a) is the connection from X_{out} to Y, which can be seen as an ascending branch of the axon from T_2 to C_2 (see, e.g., Macchi et al. 1984). The mapping in Figure 1(b) suggests that the blackboard architecture may not be confined to the thalamus as Newman and colleagues, Harth and colleagues, and Mumford and colleagues suggest. Rather, the cortex would serve as the major blackboard, and the thalamus would only play the role of gating.

Several issues become apparent when the architecture is put into a concrete context as shown in Figure 1(b). One key assumption in the neural blackboard is that duplication of structure assemblies is easy compared to that of word assemblies. But what does it mean for an assembly to be duplicated? There are two possibilities: (1) the anatomical structure is duplicated on the fly, which seems highly unlikely (but Figure 7b in the target article indicates this may be the case); or (2) the pattern of activation that represents a structure assembly is duplicated, which seems too biologically underconstrained and abstract (i.e., how can various binding and control reach the dynamically changing patterns?).

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

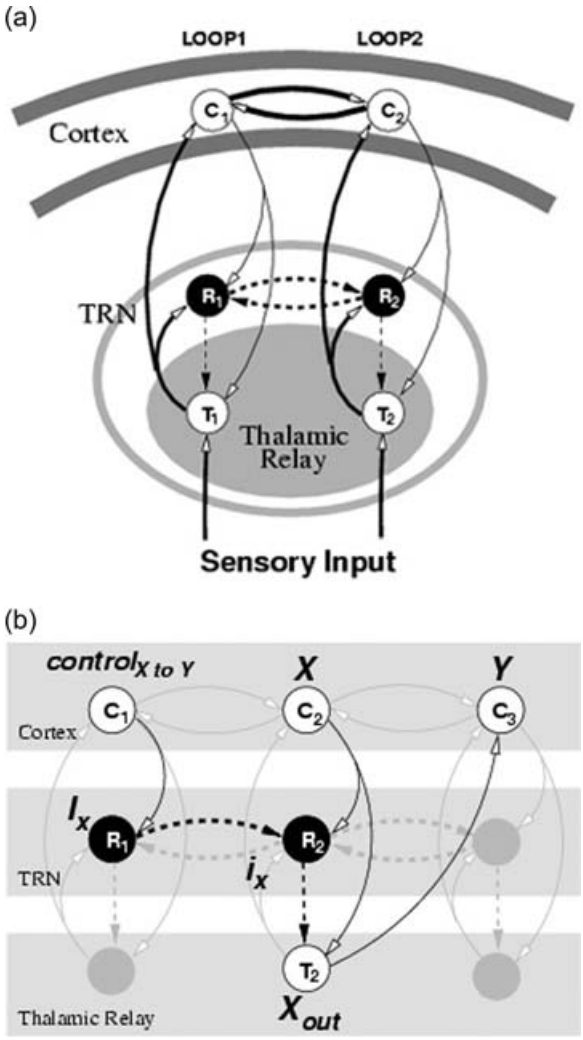


Figure 1 (Choe). Thalamocortical circuit and the gating circuit mapped onto it. (a) A schematic diagram of the thalamocortical circuit is shown (adapted from Choe 2004). White discs are excitatory neurons and black ones are inhibitory neurons, whereas solid edges are excitatory connections and dashed ones inhibitory. (b) The gating circuit in the neural blackboard architecture is mapped onto the thalamocortical circuit. The neurons and connections that may not participate in the gating circuit are shaded out in dark gray. Besides that, the only difference between the circuit here and that in (a) is the ascending thalamocortical projection from T_2 to C_3, i.e., X_{out} to Y (see text for a discussion).

Another issue is about how different word assemblies can be bound to a particular structure assembly construct such as N_1 in the target article's Figure 2 (e.g., cat -x- N_1). For this to work, potentially all cortical areas representing a word assembly (taking the place of X in Fig. 1[b]) need to have a physical connection to the relevant gating circuit in the thalamus (i_x and X_{out}) that project to N_1 (Y): a combinatoric explosion of connections. A related problem is about how control from the central pattern generator ($control_{X\ to\ Y}$) gets attached to the gating circuit (I_X). For the central pattern generator precisely to control the flow of activation in all the gating circuits, it needs to project to a large number of participating TRN neurons, which may, again, become too resource-intensive.

In sum, (1) it is possible that the gating circuit can be mapped to a concrete neural circuit in the thalamocortical network; but (2) with this concrete view of the gating circuit, the flexibility or dynamic nature of the proposed architecture comes under question, thus undermining the claimed productivity in the blackboard architecture. The blackboard architecture may be sufficient when it remains abstract, but once it becomes grounded in neural terms, its combinatoric power may suffer. Hence the question: How neural is the neural blackboard architecture?

How anchors allow reusing categories in neural composition of sentences

William J. Clancey

NASA Ames Research Center, Intelligent Systems Division, Moffett Field, CA 94035. ***William.J.Clancey@NASA.gov*** ***http://bill.clancey.name***

Abstract: van der Velde's &de Kamps's neural blackboard architecture is similar to "activation trace diagrams" (Clancey 1999), which represent how categories are temporally related as neural activations in parallel-hierarchical compositions. Examination of other comprehension examples suggests that a given syntactic categorization (structure assembly) can be incorporated in different ways within an open composition by *different kinds of anchoring relations* (delay assemblies). Anchors are categorizations, too, so they cannot be reused until their containing construction is completed (bindings are resolved).

Conceptual Coordination (Clancey 1999, hereafter CC) attempted to bridge connectionism and symbolic theories of cognition by analyzing slips, autism (Clancey 2003), learning in well-known architectures (e.g., MOPS, EPAM, SOAR), dream phenomenology (Clancey 2000), and natural-language comprehension difficulties. The neural blackboard architecture of van der Velde & de Kamps (vdV&dK) (pp. 15ff) has properties that directly map to "activation trace diagrams" (CC, p. 6), including: (1) representing a conceptualization of a sentence on a blackboard; (2) not copying structures (i.e., one physical instantiation in a given sentence diagram); (3) preserving temporal relations of neural activation; (4) composing word assemblies by structural relations (e.g., a noun phrase, NP); (5) creating word structures (e.g., "the cat - NP") by temporary binding/tagging (CC, p. 273); (6) binding words to multiple structures in different roles (e.g., both agent and theme); and (7) holding a partially completed structure active (a "delay" activity in vdV&dK; "anchor" in CC, p. 246).

Figure 1 represents conceptualization of a double-centered embedded sentence from vdV&dK using the notation from CC. During the composition process, the first noun phrase (NP1) is held active by an anchor, contained within subsequent phrases, and then resolved in an agent role by a right branch. A verb phrase is represented (in vdV&dK's terms) as an agent and verb on a line segment (e.g., "boy-likes"), with a theme (e.g., object, "dog") indicated by an arrow below. This sentence is analogous to "The cat that the bird that the mouse chased scared ran away" (CC, fig. 10.14, p. 256). vdV&dK (pp. 47ff) indicate that these sentences are difficult to comprehend because there are two NPs (NP2 and NP3) that could bind to two different theme assemblies (e.g., "likes" and "bites").

Based on analysis of other convoluted examples (provided by Lewis 1996), the binding problems involved in sentence comprehension can be restated and generalized. The following principles postulate restrictions on how word (semantic) and structural (syntactic) categorizations are employed in a given construction (from CC, pp. 261–62):

(P1) A semantic categorization can play multiple syntactic roles but not the same role twice in an active construction (e.g., "cat" cannot be an agent in two verb phrases).

(P2) A syntactic categorization may "index" multiple semantic categorizations but may be active only once at a given time (e.g., only one active NP; the others are in "completed" verb phrases).

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

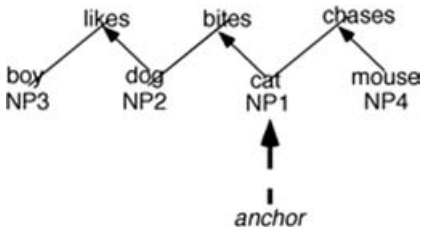


Figure 1 (Clancey). Activation trace diagram of "The cat that the dog that the boy likes bites chases the mouse" (vdV&dK, sect. 2.2, para. 7; cf. CC, fig. 17, p. 46).

(P3) The use of an anchor allows a syntactic categorization to be "open" as *the head of a sequence* twice at a given time (e.g., in Fig. 1 the anchor holds the NP at the head, allowing subsequent NP categorization activations). The anchor is itself a categorization relating syntactic categories (i.e., this anchor represents the relation "agent-NP"); the neural processes representing the syntactic categorization NP are actually released.

(P4) Meaning construction (comprehension) is a sequential behavior, not holding a parse tree or "moving" items in a data structure.

(P5) Two general kinds of anchors are possible, one holding an embedded sentence and another holding a relative pronoun or NP within a relative phrase (see example below). Anchors serve as "stacks" that allow one level of recursion, in the computational sense that the stack may be "popped" and compositional processing returns to the previously anchored element. A higher-order construction must be closed off before returning to an anchor categorization.

(P6) Multiple anchors may appear in one utterance, but each must categorize a different syntactic relation (e.g., embedded proposition, relative pronoun object, noun object, noun subject).

(P7) Multiple conceptual dimensions are operating simultaneously, including different kinds of NP categorizations, and conceptual organizers also include rhythm, stress, and visualization. These dimensions allow and facilitate reactivation and composition of multiple object and event references during the comprehension construction process (consistent with vdV&dK's relation of visual and verbal processing, sec. 8, para 1).

These principles imply a refinement of vdV&dK's neural architecture. In particular, vdV&dK say that a set of structure assemblies, whose number relates to memory span (sec. 6.7, para 1), can be simultaneously active. My analysis suggests that these are not multiple activations of a given structure assembly but categorizations of different relations constituting *different kinds of anchors* (delay assemblies connecting structure assemblies). For example, illustrating P5, the sentence in Figure 2 is impossible to comprehend because the opening ambiguous role of Fred (subject or indirect object?) requires two agent-NP anchors (for Fred and "the woman") to be active. Other examples suggest that the anchors shown for NP2 and NP3 are different syntactic relations so they are allowed (Principle 6) – anchor1 is "relative pronoun-NP," and anchor2 is agent-NP. Accordingly, we can comprehend "The woman whom the man who hired me married is tall." But the sentence in Figure 1 is confusing because it requires the agent-NP anchor to be active in different ways at the same time (for "the boy" and "the dog").

I believe the restriction on anchors follows naturally from vdV&dK's architecture (sec. 6.7, para 1). The generalization I propose is that any kind of "assembly" (categorization) can be active only once until, in the temporal process of comprehension, the binding relations are resolved in its contained categorization (e.g., a VP), allowing the constituent categorizations to be freed for reuse. Hence, just as a word can play a role (e.g., agent) in only one way in a sentence, an anchor (a delay connecting a structure assembly) is a categorization, too, and it can be active

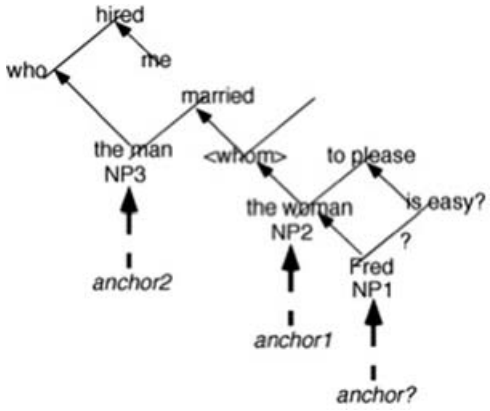


Figure 2 (Clancey). Activation trace diagram of "Fred is easy for the woman whom the man who hired me married to please" (example from Lewis 1996; see CC, fig. 10.18, p. 264).

(incorporated in an open construction) in only one way at a time. All this amounts to the rule: "no copying."

vdV&dK provide a neural model that supports and refines the postulates of CC. But the CC analysis provides many more examples that can be used to refine a theory of neural mechanisms. The process of combining structures on a "neural blackboard" can be mapped to problems of autistics in conceiving agent relations (Clancey 2003) as well as categorizing and sequencing discoordinations in verbal slips (CC, fig. 6.7, p. 161). Indeed, our dream of relating neural and "symbolic" analyses of cognition is progressing nicely.

NOTE

The author of this commentary is employed by a government agency and as such this commentary is considered a work of the U.S. government and not subject to copyright within the United States.

The problem with using associations to carry binding information

Leonidas A. A. Doumas, Keith J. Holyoak, and John E. Hummel

Department of Psychology, University of California, Los Angeles, Los Angeles, CA 90095-1563. ***adoumas@psych.ucla.edu holyoak@lifesci.ucla.edu. jhummel@psych.ucla.edu***

Abstract: van der Velde & de Kamps argue for the importance of considering the binding problem in accounts of human mental representation. However, their proposed solution fails as a complete account because it represents the bindings between roles and their fillers through associations (or connections). In addition, many criticisms leveled by the authors towards synchrony-based bindings models do not hold.

van der Velde & de Kamps (vdV&dK) argue forcefully for the importance of addressing the binding problem in accounts of human mental representation. One of the strengths of structured representations is that they can support reasoning based on relational (rather than simply featural) similarity. Although the representations in the neural blackboard architecture proposed by vdV&dK do allow specialized associations to be formed between bound elements (via structure assemblies), the resulting representations do not explicitly represent the bindings between relational elements and their arguments. This is because binding information is carried in connections between assemblies of neurons, which, as noted by von der Malsburg (1981/1994; 1986), are not sufficient to make bindings explicit

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

because information about the associations between neurons upstream are not available to neurons downstream.

For example, when the neural blackboard architecture represents the proposition *chase* (cat, mouse), it (in essence) connects the representation of cat to the *chase* relation via an agent structure assembly and the representation of mouse to the *chase* relation via a theme structure assembly. This is, as the authors demonstrate, sufficient to answer questions of the form, *Who is the cat chasing?* However, to neurons downstream, all that is available is a pattern of active neurons (i.e., active representations of cat, mouse, *chase*, and both agent and theme structure assemblies), not the associations between these upstream neurons and neuron assemblies. As a result, to these downstream neurons, the proposition *chase* (cat, mouse) is indistinguishable from the proposition *chase* (mouse, cat).

This limitation of binding by associative connections has led Hummel and Holyoak and their colleagues (e.g., Doumas & Hummel 2005; Holyoak & Hummel 2000; 2001; Hummel 2000; Hummel & Biederman 1992; Hummel & Holyoak 1992; 1996; 1997; 2003; Hummel et al. 2004) to argue that neural systems require a degree of freedom above and beyond activation and association weight in order to model human reasoning based on structured representations. vdV&dK argue that synchrony-based systems are not sufficient to model binding because they cannot solve the "problem of 2" or represent hierarchical relational propositions. However, the Learning and Inference with Schemas and Analogies (LISA) model of Hummel and Holyoak (1997; 2003) is a synchrony-based system that has no difficulty representing multiple instantiations of the same object (or relation) in a proposition (i.e., solving the problem of 2) or representing hierarchical propositions (see Hummel & Holyoak 1997; 2003). In addition, LISA makes bindings explicit (by means of the synchrony relations between bound elements) and thus can account for numerous phenomena from the literature on relational reasoning, relational generalization, schema induction, and relational retrieval.

One way to modify the vdV&dK model to allow it to capture structure might be to have the agent and theme nouns time-share. For example, let the agent noun fire first, followed by the theme for each represented relational proposition. However, this modification would simply be a way to use time to carry binding information (i.e., a form of binding by synchrony), which the authors explicitly eschew. LISA provides a more straightforward and detailed account of how time can be used to carry binding information in an architecture that can also account for numerous phenomena related to relational processing in human cognition.

Has the brain evolved to answer "binding questions" or to generate likely hypotheses about complex and continuously changing environments?

Birgitta Dresp[a] and Jean Charles Barthaud[b]

[a]*CNRS, 34293 Montpellier Cedex 5, France;* [b]*Laboratoire "Etudes des Mécanismes Cognitifs (EMC)," Université de Lyon 2, Campus Porte des Alpes, 69676 Bron Cedex, France.* ***Birgitta.Dresp@univ-lyon2.fr jean-charles.barthaud@univ-lyon2.fr***

Abstract: We question the ecological plausibility as a general model of cognition of van der Velde's & de Kamps's combinatorial blackboard architecture, where knowledge-binding in space and time relies on the structural rules of language. Evidence against their view of the brain and an ecologically plausible, alternative model of cognition are brought forward.

The authors present a model that addresses the question of how a brain could learn to process meaning conveyed by sentence structures of a modern European language. Claiming that language symbols are "grounded in perception," they argue further that sentence structures and visual object properties are controlled by the same top-down process. The idea that consistent "binding" of knowledge in space and time would, at some stage of cognitive development, generate specific brain states is readily dismissed. Binding, instead, is conceived as a computational operation designed to mimic a brain process that selectively links words or object properties via combinatorial rules imposed by the structure of the English language. Whether this approach provides a plausible general model for cognition, as claimed here, is questionable. It remains to be justified whether the "binding problem" is solved on the implicit assumption that the human brain has evolved to process information according to rules imposed by language structures.

The authors claim that binding results from a process which is initiated with a "binding question," such as "Who does what to whom?" or "Where is what?," leading to temporary selective activation/inhibition of verb-noun or object-feature connections. What is not explained is that "binding questions" in human cognition cannot be reduced to simplistic "queries" like the examples given. During cognitive development, the brain deals with far more complex "interrogations," such as "Who does what to whom, where, when, why and with what consequences?" for example. The authors touch on this by mentioning that with increasingly longer sentences, their model would have to encode entire sentence structures in terms of event sequences. This inevitably implies that their system would ultimately bind increasingly complex "event assemblies" rather than word-and-noun or object-and-feature assemblies. How the relatively simplistic combinatorial rules of language discussed in the framework of their article handles such increasing complexity remains an open question.

How plausible is it to assume that brains would use a process of the kind suggested here to bind knowledge in space and time? Brain learning is certainly aimed at generating behavioural success. Intelligent systems like the human brain have evolved to be able to generate the most likely hypotheses about complex and continuously changing environments. To be successful, that is, to enable fast behavioural adaptation to changes occurring in the physical world, such hypotheses have to rely on simple, fast, and foolproof mechanisms (i.e., "dumb mechanisms"; Anstis 2005). Visual recognition of complex cognitive events such as natural scenes is essentially holistic and incredibly fast in human and nonhuman primates (Delorme et al. 2004; Mace et al. 2005). This strongly suggests that the binding of familiar visual input reflects a brain state where expectation signals preset lower levels of visual pathways to the learned events, objects, or features.

How such a brain state could be computed is suggested in adaptive resonance theory (ART), an alternative theory of cognition not mentioned in the target article. According to ART (e.g., Grossberg 1999), binding would correspond to a resonant state resulting from positive versus negative matches between incoming external information (bottom-up signals) and the most likely hypothesis of what that information could be at a given time (top-down expectation signals). Resonant states would correspond to states of activation in the "what" and "where" processing streams (Ungerleider & Mishkin 1982) of the primate brain. Resonant states, as shown in ART, are generated by signal exchanges at the perceptual level and do not require a language-based, combinatorial architecture.

van der Velde & de Kamps conceive the brain as a general-purpose computer that can be programmed, unprogrammed, and reprogrammed to do all sorts of tasks according to a given number of limited rules. They state that language symbols are "grounded in perception," but what does that mean in terms of brain mechanisms? Empirical findings point toward functionally specialized, separate pathways for the processing of language and visual information such as faces,

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

shapes, or colours (Anstis 2005; Damasio 1983; Moscovitch et al. 1997; Stroop 1935). Moreover, the fact that human adults are capable of visually representing event sequences described by sentences and, conversely, are able to use sentences to describe events represented in visual scenes does not imply that language and vision produce unified knowledge controlled by a single top-down process generating "binding questions." The authors, however, defend such a conception of the human brain by arguing that observers can be cued verbally to select visual objects on the basis of colour irrespective of shape or location. This implies that language determines which features of an object will be encoded by the brain.

Evidence from object-recognition experiments using visual repetition priming (Barthaud et al. 2003) shows that such a conclusion cannot be sustained. We presented visual targets with varying shapes and colours, preceded by visual primes with the same colour as the target but different shape, the same shape but different colour, the same shape and colour, and different shape and colour. In the control condition, no prime preceded the target. When observers are verbally cued to decide on the shape of the target irrespective of colour, results unambiguously show that they recognize the shape of the target significantly faster when it is preceded by a prime with the same shape *and* the same colour. This clearly demonstrates that although only shape was cued for verbally, observers still used and therefore processed colour to recognize the targets as quickly as possible. Such findings add empirical support to models of cognition such as ART (e.g., Grossberg 1999), where external contextual events, not internalized cues or "binding questions," determine what is encoded by the brain. In conclusion, there is doubt about the ecological plausibility of the approach presented here as a general model of cognition.

Engineering the brain

Daniel Durstewitz

Centre for Theoretical and Computational Neuroscience, University of Plymouth, Plymouth, PL4 8AA, United Kingdom.

daniel.durstewitz@plymouth.ac.uk

http://www.bio.psy.ruhr-uni-bochum.de/cnl/cnl.htm

Abstract: The target article presents a stimulating account for some of the most challenging issues in cognitive neuroscience. The theory solves in neural terms cognitive problems beyond the scope of previous models. But in many aspects the neural implementation is a quite literal translation of symbolic descriptions and therefore still lacks some of the truly self-organizing properties characteristic of biological networks.

In their innovative framework, van der Velde and de Kamps (vdV&dK) exploit sustained activity for expressing temporary bindings and gating circuits for controlling the flow of activation in accordance with syntactical structures. Thereby they manage to account for some of the most challenging problems of human cognition using neurobiologically plausible components. Yet in many aspects, their approach follows an engineering, top-down-driven view on neural computation, where abstract linguistic components are, more or less, directly mapped onto neural components like labeled nodes (cell assemblies) and "labeled" links. This is not a drawback per se, as many valuable insights have been gained that way in the past. But although it may suit the cognitive side, some questions arise regarding biological validity.

Certainly, as said above, the neural ingredients used by dvV&dK, like gating circuits, are somewhere present in the brain. But *any* cognitive machine could be designed by freely combining elements from the rich repertoire of neurophysiologically established mechanisms (AND- and OR-like gates, for example, come in many different variants, ranging from molecular to network levels). For that same reason, the criticisms by the authors of previous linguistic neural models such as the ones based on synchronization or artificial recurrent networks (RNN) are at best criticisms of these specific models rather than of these neurocomputational approaches in general. For example, using synchronization (in the sense of fixed phase relations between spike times) as a mechanism, different bindings may be established within different oscillatory subcycles, and hierarchical structures might somehow be embedded in nested oscillations (like theta- and gamma-oscillations in the hippocampus). For read-out, as the authors point out for their own mechanisms, all that is really required is a network dynamics associated with a computational process (like answering a binding question) that is sensitive to different subpopulations of neurons being synchronously active. Devoting explicitly semantically predefined coincidence detectors to specific constellations of (phase-) synchronous neurons may not be necessary.

Finally, more intricate relational codes than phase-lag zero synchronization may be employed by the brain to encode hierarchical structures (Ikegaya et al. 2004). Or, taking RNNs, flexible encoding of relational structures could benefit from allowing multiplicative inputs (e.g., Hanson & Negishi 2002). Moreover, the fact that RNNs cannot properly *learn* the required generative properties, given their specific learning rules, does not imply that they cannot *represent* such knowledge. And self-organizing rules that build up the structures proposed by vdV&dK have yet to be suggested.

Given that neural biophysics offers such a rich repertoire of elementary mechanisms, and assuming that we are interested in the "true" neural implementation of cognitive processes, additional criteria may be used to compare vdV&dK's theory to other approaches:

1) *Simplicity and compatibility with physiological, anatomical, and behavioral data*. To which degree is the model supported yet by empirical data? The more that explicitly specialized mechanisms and elaborated anatomical arrangements are proposed, the more the current lack of neurobiological data for higher-level processes may become a problem. RNNs, for example, require fewer assumptions here. Another potential problem could be precise temporal coding. There is accumulating evidence that the brain produces precise temporal codes (Fellous et al. 2004; Ikegaya et al. 2004; Riehle et al. 1997; Wehr & Laurent 1996), although their role is still debated. If the brain does so, it would be surprising if they were not employed in the "highest" of cognitive functions, like language. In this case, some assumptions about neural computation might unfortunately also not be testable at the level of EEG and fMRI.

2) *Self-organization*. Could the proposed architecture emerge naturally from physiological learning rules or anatomical givens? How could it self-organize during online processing, and how are new between-assembly connections and gating circuits set up in the first place when novel words or syntactical components are encountered? Although vdV&dK are aware of this problem (sect. 6.9), solving it in the case of their model seems more challenging than, for example, with RNNs. For these, local learning rules at least can be defined. Likewise, phase-locking states (synchronization) often naturally appear as attractor states of the network dynamics. In contrast, I have the feeling that there are still many homunculi at work in the vdV&dK model. One such homunculus, for example, is the "central pattern generator" (CPG): Who "programs" the CPG for a given task?

3) *Evolutionary emergence*. How could the proposed carefully designed structure, with many specialized syntactic components wired through specific gating circuits, evolve? Evolution often proceeds by reusing components or slightly modifying already present machinery. The comparatively homogeneous structure of the neocortex suggests that language capabilities are also based on some more general principles of neural information processing. For example, processing of motor sequences may

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

rest on similar neurocomputational mechanisms. If so, animal models and physiological properties in motor cortex may be used as hints to the neural mechanisms of language processing. Such potential continuity is not that obvious in the vdV&dK theory – for example, although the proposed blackboard mechanisms for visual and sentence processing are conceptually similar, their suggested implementations seem quite different.

4) *Response speed and algorithmic efficiency*. Time is a serious constraint in biology, and evolutionary pressure should strongly favor the faster one of two potential neural mechanisms, all other things being equal. Phase coding may have an advantage here over the firing rate–based mechanisms suggested. For example, phase coding allows for fast, robust, and genuinely scale-invariant pattern recognition (Brody & Hopfield 2003; Hopfield 1995).

5) *Noise- and fault-tolerance*. Just for completeness, it should be mentioned that any neuronally plausible mechanism should tolerate considerable levels of noise in the input and possibly also intrinsically generated, and should tolerate destruction of its elements to some degree. These constraints are probably fulfilled by the vdV&dK theory.

vdV&dK address a hard neuro-cognitive problem for which neurobiological data are still scarce. They come up with a potential solution which appeals by the way it achieves the high flexibility required for language through combining a few neurobiological building blocks. Yet biological plausibility and self-organization remain an issue, and vdV&dK's achievement might have come at the price that addressing the above-mentioned challenges in their current framework seems more difficult than in other, more bottom-up-driven approaches.

ACKNOWLEDGEMENTS

Many thanks to Mark van Rossum, Thomas Wennekers, and Sabine Windmann for reading this commentary and discussing related ideas with me.

Will the neural blackboard architecture scale up to semantics?

Michael G. Dyer

University of California at Los Angeles, Computer Science Department, UCLA, Los Angeles, CA 90095. ***dyer@cs.ucla.edu***
http://www.cs.ucla.edu/~dyer/

Abstract: The neural blackboard architecture is a localist structured connectionist model that employs a novel connection matrix to implement dynamic bindings without requiring propagation of temporal synchrony. Here I note the apparent need for many distinct matrices and the effect this might have for scale-up to semantic processing. I also comment on the authors' initial foray into the symbol grounding problem.

van der Velde's and de Kamps's (vdV&dK's) neural blackboard architecture (NBA) is a highly structured localist model. The semantic processing portion of NBA is not well developed, but the authors imply that semantic knowledge would be organized in a manner similar to the SHRUTI model (Shastri 1999) except that propagation of dynamic bindings via synchronous firings (as done in SHRUTI) would not be needed; instead, NBA would employ its "connection structure matrix" approach (CSM). Unless I have misunderstood the CSM model, the following appears to be the case:

(1) Binding nodes from two categories, such as V-to-N, require a *different* CSM from the CSM needed for binding N-to-V.

(2) Hence for each different syntactic category binding pair, such as D-to-N (determiner-to-noun), S-to-N (sentence to noun), S-to-V, Det-to-N, N-to-Adj, etc., a *different* CSM is also required.

(3) Dynamic bindings are actually formed between role-nodes associated with a node of given syntactic category. (I will refer to role-nodes in **bold**.) For example, if category V has two roles (let $\mathbf{aV}$ = agent-role of V and let $\mathbf{tV}$ = theme-role of V) then one CSM is required to bind two agent-roles (across two categories) and a *different* CSM is needed to bind two theme-roles.

Consider how a CSM is implemented. To bind, say, the $\mathbf{aN_2}$ node to the $\mathbf{aV_3}$ node in the N-V CSM, a specific memory-delay binding element, $E_{2,3}$, *must already exist* with connections to it from both $\mathbf{aN_2}$ and $\mathbf{aV_3}$. Hence the temporary binding problem is "solved" by supplying a unique element E (itself a structured assembly of connectionist units) *for every possible node in a role-category that might be bound to every other possible node in another category with the same role*.

When a sentence, such as "The dog licks the cat near the dog," is input, some process must assign "dog" to the N-to-V CMS and also to the P-to-N CMS. It is unclear how this assignment is made. If it is done by activating all N-agent nodes in all CMSs that take N-agent nodes as input, then it seems that massive cross-talk would occur. If "dog" were assigned to the N-to-V CMS by activating all $\mathbf{aN_i}$ in that CMS, then it appears that massive cross-talk again would arise. If these problems are handled by the winner-take-all inhibitory connections in each CSM, then two consequences would seem to be: (a) only one binding pair could be active at a time; and (b) propagation of bindings would be slow (because of the need for settling into a single winner).

Will the CMS approach scale up for semantics? Unlike syntax (which has a relatively small number of categories), semantic systems tend to have a great many predicates. In addition to the authors' example of the roles of GIVE passing dynamic bindings to the roles of OWN, there would be HUNGRY to EAT; TELL to KNOW; HIT to HURT, and so on. For every pair of related *instances* of GIVE and OWN (HIT and HURT, etc.) there would have to exist already a binding element (in a distinct CMS) connected to just those two instances.

The authors have arrived at their highly structured solution partly as a result of rejecting an alternative approach, namely, distributed representations manipulated by recurrent neural networks (RNNs). One of their arguments against RNNs is that new tokens (such as "Dumbledore") cannot be handled in such systems. In response, consider the DISCERN model (Miikkulainen 1993). Through supervised training, DISCERN *learns* to do the following: read script-based stories, generate paraphrases of stories, and answer questions about story events. The distributed patterns, which represent word-meanings, self-organize by a process of extending back-propagation into the lexicon (Miikkulainen & Dyer 1991). DISCERN handles novel words by use of an *ID-content* technique. Each word-meaning consists of an ID portion and a content portion. The content portion self-organizes during training (so that words with similar meanings in the lexicon end up with similar distributed representations) whereas the RNN modules that manipulate these representations learn to pass on the ID portions without alteration. A new name, such as "Dumbledore," would be represented as an ID-content activity pattern, where the ID is a new pattern, whereas the content portion remains as a distributed representation that was formed for all humans (because of reading stories involving human characters during training). The content portion enables statistically based generalizations to occur in the various RNN modules, whereas the ID portion in passed, unaltered, from RNN module to RNN module.

I am not arguing here that DISCERN handles all aspects of dealing with novelty in multimodule RNN systems, but that vdV&dKs rejection of RNN approaches on the grounds of novelty-processing is premature (especially because multimodule RNN systems can *learn* all of their tasks whereas NBA is mostly hand-coded for its task).

With respect to the grounding problem, vdV&dK's comments are mostly speculative. In response I refer them to the DETE

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

spiking neuron model (Nenov & Dyer 1993; 1994). DETE learned to associate word sequences with simple image/motor sequences. Given an image sequence of a red triangle moving upward, hitting a blue circle, and bouncing downward, DETE would generate the word sequence "red triangle moves up hits blue circle and bounces." DETE could learn words such as "accelerate" by watching objects speed up in its visual field and it could learn to push a virtual finger at a virtual object by associating a motor sequence with a word sequence like "push red ball." Once DETE learned individual color and object words (by association with images of colored objects), DETE could immediately handle novel color-object combinations. Word-image-motor sequence associations were learned after just a few training repetitions, and multiple objects with different colors in the same visual field were handled by an attentional subsystem using synchronous firing.

The authors' argument against synchrony assumes simple forms of synchrony. If synchrony can exhibit period doubling (Sougne 2001) or be chaotic (Raffone & van Leeuwen 2001), then several instances of the same predicate may be handled. Synchrony can also be augmented by propagating activity patterns that are distributed both temporally *and spatially*. Such patterns have been referred to as "signatures" (Lange & Dyer 1989; Lange & Wharton 1995) and, when encoded as distributed representations, they can be learned and can also control their own propagation (Sumida & Dyer 1992).

Vector symbolic architectures are a viable alternative for Jackendoff's challenges

Ross W. Gayler

Camberwell, Victoria 3124, Australia. ***r.gayler@mbox.com.au***

Abstract: The authors, on the basis of brief arguments, have dismissed tensor networks as a viable response to Jackendoff's challenges. However, there are reasons to believe that connectionist approaches descended from tensor networks are actually very well suited to answering Jackendoff's challenges. I rebut their arguments for dismissing tensor networks and briefly compare the approaches.

van der Velde &de Kamps (vdV&dK) have proposed neural blackboard architectures (NBAs) in response to Jackendoff's (2002) challenges. Their note 1 dismisses tensor networks (Smolensky 1990) as a viable alternative. However, Gayler (2003) argues that vector symbolic architectures (VSAs) – connectionist approaches descended from tensor networks – are very well suited to answering Jackendoff's challenges. There is not space here to repeat those arguments. Rather, I will rebut vdV&dK's arguments for dismissing tensor networks and briefly compare the approaches. Regardless of the ultimate relative success of NBAs and VSAs, the field of cognitive neuroscience will benefit from having plausible alternatives that can be compared and contrasted.

vdV&dK's note 1 claims that "tensor networks fail to instantiate combinatorial structures." Tensor product binding was developed specifically to address the issue of combinatorial representation. Although Smolensky's (1990) presentation was somewhat abstract and mathematical, he chose his primitive representations and operations to be easily implemented in connectionist systems. Items are represented by numerical vectors (distributed representation) and the operators are simple vector sums and products. Smolensky demonstrated tensor network implementation of variables and binding and also, as an example of the computational power of tensor networks, demonstrated a connectionist implementation of the CONS, CAR, and CDR operators of the LISP programming language. Halford and colleagues (1998) discuss cognitive symbolic operations that can be implemented in tensor networks.

vdV&dK's note 1 cites Fodor and McLaughlin (1990) in support of the claim that "tensor networks fail to instantiate combinatorial structures," but this is not the focus of their paper. Their focus is Fodor's and Pylyshyn's (1988) claim that connectionism either cannot account for cognition or, if it can, is a mere implementation of a "classical" theory. Fodor and McLaughlin (1990) is a response to attempts by Smolensky to rebut Fodor and Pylyshyn (1988). In that context, Fodor and McLaughlin concede that "It's not . . . in doubt that tensor products *can represent* constituent structure" (p. 200).

Fodor and McLaughlin argue that although tensor networks represent constituent structure, the constituents are not causally effective in processing of the composite. They claim (correctly) that "When a tensor product vector . . . is tokened, its components are not" (1990, p. 198) and (incorrectly) that "Merely counterfactual representations have no causal consequences; only actually tokened representations do" (p. 199). Binding and unbinding by a tensor network (Smolensky 1990) suffices as a counterexample to demonstrate the falsity of the latter claim.

vdV&dK claim that "tensor networks fail to instantiate combinatorial structures. . . [because] a tensor is just a list of constituents organized in a particular fashion (i.e., as an n-dimensional list for a rank-n tensor)" (note 1). However, exactly the same claim could be made of the storage tape of a Turing machine or the CONS cells of a LISP program, yet no one would dispute the ability of a Turing machine to represent combinatorial structures.

The final argument in note 1 is the most significant; vdV&dK point out that "adding constituents to the tensor increases the dimensions of the tensor, which requires adjustments to all components in the cognitive system that can interact with the tensor." This is the major problem with tensor networks as an implementation rather than an abstract formalism. However, there have been 15 years of further development since Smolensky (1990), and this problem was soon solved (Plate 1991).

In a tensor network, two items are bound by forming the outer product of the vectors representing the items. That is, if each primitive item is represented by a vector of n elements, their combination contains n^2 elements. If this composite item were bound with another primitive item, the result would contain n^3 elements, and so on. This has major practical consequences for resource requirements and the connections between processing components of a tensor network. The number of elements increases dramatically with the binding order, which means that the resource requirements may be excessive and that bindings of arbitrarily high order cannot be represented on a fixed set of resources. Also, the connections between processing components must be dimensioned to accommodate the highest order representation chosen by the designer.

An abstract solution to this problem was proposed by Hinton (1988; 1990) and a specific, practical implementation demonstrated by Plate (1991; 1994; 2003). Hinton introduces the idea of "reduced descriptions" as a means to represent compositional structures in fixed size connectionist architectures. A reduced description is a representation of a composite item that is the same size as any of the component items and from which the component items can be generated. Plate demonstrates holographic reduced representations (HRRs) as a specific implementation of reduced descriptions. HRRs can be conceptualised as a compression of the tensor product to a vector the same size as each of the components being bound.

HRRs use a specific, highly ordered compression of the tensor product. However, Wilson and Halford (1994) show that the majority of tensor product elements can be destroyed without compromising performance, and Plate (2000; 2003) shows that many alternative compressions of the tensor product, including randomly disordered compressions, would suffice. Other compressions were developed independently (Gayler 1998; Gayler & Wales 1998; Kanerva 1994; 1997; Rachkovskij & Kussul 2001). Collectively, these are the vector symbolic architectures,

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

and various members of this family are compared in Gayler (1998) and Plate (1997).

Having responded to the arguments of note 1, I feel that a brief comparison of NBAs and VSAs is worthwhile. NBAs use localist representation, so effort is required to connect the representations that need to interact. VSAs use distributed representations, so many representations are simultaneously present over the same connectionist units. This replaces the problem of connecting the interacting representations with the problem of keeping most superposed representations logically separated to avoid unwanted interactions. Another difference is the treatment of novel items. Localist representation requires the recruitment of neural units to new assemblies, whereas distributed representation requires only a novel pattern which is implemented over the same units as familiar patterns. VSAs do not require a pool of unallocated units or a recruitment process. These two comparisons suggest the breadth of issues that can be explored in the range of possible connectionist cognitive architectures spanned by NBAs and VSAs.

ACKNOWLEDGEMENTS

My thanks go to Graeme Halford, Pentti Kanerva, Steven Phillips, Tony Plate, and Roger Wales for their helpful comments.

Distributed neural blackboards could be more attractive

André Grüning and Alessandro Treves

Cognitive Neuroscience Sector, Scuola Internazionale Superiore di Studi Avanzati (SISSA), 34104 Trieste, Italy. ***gruening@sissa.it*** ***http://www.sissa.it/~gruening*** ***ale@sissa.it*** ***http://www.sissa.it/~ale/limbo.htm***

Abstract: The target article demonstrates how neurocognitive modellers should not be intimidated by challenges such as Jackendoff's and should explore neurally plausible implementations of linguistic constructs. The next step is to take seriously insights offered by neuroscience, including the robustness allowed by analogue computation with distributed representations and the power of attractor dynamics in turning analogue into nearly discrete operations.

van der Velde and de Kamps (vdV&dK) add a new perspective to the neural mechanisms underlying syntactic binding in human sentence processing (and perhaps in visual processing). The neural blackboard model implements binding as an active and dynamic process and hence it dispenses in a satisfying and elegant way with a homunculus observing synchronous network activity, as posited by some previous approaches. We regard this as an important first step that can lead to important breakthroughs if combined with an understanding and effective use of cortical attractor dynamics.

vdV&dK commendably aim toward a nonsymbolic representation of words by distributed neural assemblies (Pulvermüller 1999). In their implementation of the blackboard, however, they stop short of describing words or, in general, items in semantic memory in terms of local network attractors (O'Kane & Treves 1992), and therefore they cannot utilise either the partially *analogue* nature of attractor states or the correlational structure of composite attractors, such as varying overlap with competing assemblies, context-dependence of assembly activity, or robustness to noise and partial disruption. Hence they use word assemblies in a monolithic, *symbolic* way, much like lexical word entries in a pure symbolic approach. The activation of their "subassemblies" is all-or-nothing (gradual activation is, however, used to control binding in recursive structures), and it is hard to see how it could be modulated, for example by contextual information, even though the activation of different subassemblies could be interpreted, perhaps beyond the authors' wish, as a coarse model of composite attractors. The symbolic nature of the proposed computations emerges also in the dichotomy between the blackboard, which temporarily stores a "parsing table," and the central pattern generator (CPG), which serves as a finite control. The CPG hides much of the complexity required to operate the blackboard. What would be a neurally plausible implementation of the CPG?

vdV&dK helpfully contrast *combinatorial* with *recursive* productivity, an important distinction that is not so transparent in the later Chomskyan view (Hauser et al. 2002). Their combinatorics, however, disappointingly appears to rely on an implausibly vast number of specialised processors, although disguised behind the thin veneer of "word assemblies." Assuming, for example, that a third of all the words the cited English 17-year-old knows are nouns, and that there are 10 main noun assemblies, we are left with still 200,000 specialised gating assemblies (just for nouns) that all do more or less the same job. Although this number is certainly much smaller than the total number of neurons or synapses in cortex, it appears to reflect an unimaginative view of assemblies as equivalent to local units – perhaps driven by an anxiety to respond to Jackendoff's challenges – that does not do justice to the idea of *distributed* processing in the brain. Composite cortical attractors, which allow for *quantitative* analyses of nondynamic, long-term memory storage (Fulvi Mari & Treves 1998) also lead, with a simple adaptive dynamics, to combinatorial productivity, without any recourse to specialised units, as in the *latching* model (Treves 2005).

It is highly praiseworthy that vdV&dK ground the blackboard operations in the concrete domain of neural network operations. A genuine challenge posed by neural mechanisms, however, is to describe how they could possibly be generated without recourse to innate knowledge or to explicit programming. Associative synaptic plasticity is the dominant – although not the only – concept emerging from neuroscience to account for learning processes, and it is, for example, sufficient to generate combinatorial productivity in the frontal latching networks model (Treves 2005). It would be exciting, and perhaps not far-fetched, to see vdV&dK's blackboards self-organise through an *associative learning* paradigm.

vdV&dK also briefly address *recursive* productivity, making use of decaying activity in reverberating assemblies (Pulvermüller 1999). This fits naturally with the blackboard architecture once there is a discrete set of structure assemblies. However a more distributed mechanism of recursion would be less ad hoc, for example the one discussed by Grüning (in press) for trainable networks. Perhaps one can integrate aspects of distributed network dynamics with composite attractor models in order to allow for distributed and compositional properties at the same time. Such a model combining a distributed blackboard for combinatorial structure and a distributed push-down (or more powerful) storage for recursion would be most interesting.

Last, we would invite a dynamic interaction in a wider sense. Jackendoff (2002) invokes a conceptual innovation in cognitive neuroscience that would allow for a more productive dialogue between neural network models and linguistic theory, and vdV&dK appear to respond to the call. While meeting the challenge of the linguists, however, cognitive modellers should not neglect to appreciate the conceptual advances in systems and theoretical neuroscience, especially relating to assembly and attractor dynamics. Without these advances, it seems difficult to understand how linguistic and, in general, cognitive representations can emerge from the underlying neural dynamics, and what qualitative or quantitative changes in the properties of the cortical ware, which had already been shaped by 200 million years of mammalian evolution, led to the distinctiveness of human cognition.

In sum, we regard vdV&dK's article as an important progress in the direction of a fruitful interdisciplinary exchange; we

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

suggest following this program through with the inclusion of distributed and dynamical attractor representations, thus avoiding some of the shortcomings and the neural implausibility of the current model.

ACKNOWLEDGEMENTS

André Grüning was supported by the Human Frontier Science Program RGP0047-2004/C during the writing of this commentary.

Neural circuits, matrices, and conjunctive binding

Robert F. Hadley

School of Computing Science and Cognitive Science Program, Simon Fraser University, Burnaby, British Columbia, V5A 1S6, Canada. ***hadley@cs.sfu.ca http://www.cs.sfu.ca/~hadley/***

Abstract: It is argued that van der Velde and de Kamps employ binding circuitry that effectively constitutes a form of conjunctive binding. Analogies with prior systems are discussed and hypothetical origins of binding circuitry are examined for credibility.

van der Velde and de Kamps (vdV&dD) offer a daring solution to Jackendoff's fourfold challenge. Viewed in abstraction from the neurological details of binding circuitry, their proposal may well provide a promising direction for a convincing solution. However, once the ramifications of their detailed architecture are set forth, substantial doubts emerge regarding possible origins of the numerous matrices of memory circuits required (as I explain below). Arguably, such doubts would be reduced were we to abstract away the neurological details of the binding circuitry and regard their binding (memory) devices as a variety of conjunctive coding. This suggestion may seem puzzling in light of vdV&dK's casual dismissal of conjunctive binding, for they remark, merely in passing, that conjunctive methods share fatal difficulties with synchronous binding methods. Nevertheless, there are striking analogies between the abstract functionality of vdV&dK's binding circuits and the functional roles of conjunctive nodes employed in prior connectionist, language models. Moreover, consider vdV&dK's own description of the functionality of binding circuits. They remark: "In this way, the connection structure instantiates a logical AND operation." (See their Fig. 5.)

vdV&dK employ competitive inhibition to ensure that within a matrix, the most active assembly along the vertical dimension will bind with the most active assembly along the horizontal dimension. This strategy is precisely analogous to the manner in which conjunctive nodes create bindings in the connectionist parsing model of Stevenson (1994) and the language acquisition models of Hadley and Hayward (1997) and Hadley and Cardei (1999). All three of these systems are capable of processing deeply embedded sentence structures. The latter two also exhibit strong forms of systematicity and, even in their present form, solve two of Jackendoff's binding problems.

One significance of these points of analogy is that if, as I anticipate, vdV&dK's elaborate circuitry is seen to be dubious from an evolutionary and/or learnability standpoint, one may yet find a largely analogous solution to Jackendoff's challenge by exploring alternative implementations of conjunctive binding (e.g., by using conjunctive neurons with looping self-feedback, subject to suitable decay). Of course, a commendable aspect of vdV&dK's entire proposal is that their binding devices are fleshed out in neurological detail, whereas the systems of Stevenson and of Hadley and colleagues employ binding nodes whose neurological implementations are left unspecified. Arguably, though, the neurological detail of vdV&dK's proposal is also its Achilles' heel. To appreciate this fully, we must consider not only the complexity of each of vdV&dK's binding circuits but the number and arrayment of these devices.

To begin with, consider just the matrix of binding circuits required to enable the binding of NP subassemblies to the *agent* subassemblies of verbs, as displayed in vdV&dK's Figure 5. Between each NP subassembly and verb subassembly, there must physically reside a "hardwired" memory circuit of the kind displayed in Figure 3. This alone entails that the matrix of Figure 5 must be vast, for within a language such as English, there are thousands of verbs and thousands of NPs that could fill the agent role of each of these verbs. Assuming a minimum of just 5,000 nouns and 3,000 verbs, at least 15 million memory circuits will be required for this "agent matrix" alone.

A separate huge matrix will be required to accommodate the theme role of all verbs that accept NPs as themes. Similar remarks apply to the indirect-object, instrumental, and location thematic roles of verbs whose semantics permit this. Furthermore, as vdV&dK later explain, NP subassemblies must be able to bind not only with the thematic assemblies of top-level verbs but also with clause assemblies, prep-nouns, and noun-clause, and theme-clause subassemblies, among others. In addition, where verbs are concerned, there must be separate matrices to accommodate both singular and plural nouns that relate to singular and plural verbs, respectively. Thus, despite vdV&dK's remark that "an arbitrary number of sentence structures can be encoded without an explosion of structure assemblies", the numbers of distinct binding circuits, whose positions are fixed in space, appear to be well over 60 million. Although this number may not be problematic in itself, the fact that each circuit must be intricate, functionally correct, and situated between appropriate subassemblies does raise serious worries about their genesis.

Concerning this genesis, three possibilities require comment (others exist, but space wanes). The first is that each of the complete matrices is innately endowed within the human brain. Yet, where language is concerned, this possibility is excluded by the fact that several of vdV&dK's subassembly types are language-specific (e.g., prep-noun, noun-clause, and theme-clause subassemblies are language-dependent, and consequently, so must the placement be of binding circuits that attach to those subassemblies).

A second possibility is that the combinatorial matrices are created by learning, in which the connectivity of each binding circuit is sculpted from "an undifferentiated connection structure" (to use vdV&dK's words). Such "sculpting" of dendritic and axonic connectivity would arise, presumably, from competition and Hebbian synaptic tuning, including the atrophy of connections from inactivity. Yet, given an "undifferentiated connection structure," the prospects of creating, via Hebbian tuning (together with competitive inhibition), millions of functionally identical structures as complex as the binding circuits at issue will seem remote to those who have lengthy experience with these methods. Moreover, without considerable *initial, coherent structure to guide the flow* of spreading activation, Hebbian/competitive learning is extremely unlikely to yield *distinct*, regular matrices containing *only* identically functioning circuits linked together in just the prescribed fashion. Of course, one cannot exclude the possibility that non-Hebbian (perhaps chemical) "sculpting methods" will eventually be discovered that will satisfy vdV&dK's needs. At present, however, this seems a slender reed.

The third possibility would be to assume that there exist, innately, a plethora of matrices, each containing appropriate binding circuits, but that experience determines the roles played by subassemblies within each matrix. Experience would, moreover, need to determine the timing of activation of gating circuits and other dynamics of parsing. Although it would be surprising to discover that our cells are genetically programmed to create the kinds of matrices vdV&dK have posited, this supposition is not wildly implausible. I submit that credibility would be increased, however, if vdV&dK were to replace their intricate memory circuits with a simpler form of conjunctive binding based upon conjunctive neurons having a suitable activation decay.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

Blackboards in the brain

Ralph-Axel Müller

Department of Psychology, San Diego State University, MC 1863, San Diego, CA 92120. ***amueller@sciences.sdsu.edu***
http://www.sci.sdsu.edu/~amueller/index.html

Abstract: Although van der Velde's de Kamps's (vdV&dK) attempt to put syntactic processing into a broader context of combinatorial cognition is promising, their coverage of neuroscientific evidence is disappointing. Neither their case against binding by temporal coherence nor their arguments against recurrent neural networks are compelling. As an alternative, vdV&dK propose a blackboard model that is based on the assumption of special processors (e.g., lexical versus grammatical), but evidence from the cognitive neuroscience of language, which is, overall, less than supportive of such special processors, is not considered. As a consequence, vdV&dK's may be a clever model of syntactic processing, but it remains unclear how much we can learn from it with regard to biologically based human language.

Although most linguists consider morphosyntax a core component of language, progress on elucidating its brain bases has been moderate. van der Velde's & de Kamps's (vdV&dK's) attempt to model combinatorial cognition as emerging from more elementary perceptual principles is commendable in this context. In principle, it has two major strengths. First, it approaches morphosyntax from a broad angle of combinatorial cognition, which may be found in domains outside language (e.g., visual processing). Second, it is implemented on a "neuronal" level, with the implied ambition of mimicking some known features of real neuronal circuits.

Construing the issue as a special case of the binding problem opens up treasures of empirical evidence, because binding has been more than just thought about—it has been studied in many well-controlled neuroscientific experiments, with some surprising success. For example, Singer and his colleagues (Nase et al. 2003; Singer 2001) have made a very good case for perceptual binding in the visual domain (e.g., perception of a gestalt) to be associated with phase-locked oscillatory firing patterns of neurons, even across distal brain regions. It is, therefore, disappointing that vdV&dK leave this extensive work unmentioned and discard computational models inspired by it (Shastri & Ajjanagadde 1993b) because they cannot account for productivity and hierarchical complexity of syntactic structures. vdV&dK's criticism appears to be related to their incomplete concept of "synchrony of activation." The potential representational complexity through synchronous multineuronal spiking is much greater than assumed by vdV&dK when parameters such as latency shifts (Fries et al. 2001) or coherence in different frequency bands (e.g., Bastiaansen et al. 2002) are considered. Such additional parameters may allow a neuron to participate in multiple assemblies of coherent firing at the same time. vdV&dK seem all too willing to discard a model that is at least partially consistent with established functional principles of neuronal assemblies, rather than to adapt it by enhancing its neural plausibility.

I am similarly unimpressed by vdV&dK's dismissal of recurrent neural networks (RNNs). Again, the argument that models proposed by Elman and others that incorporate a single hidden layer cannot account for the combinatorial productivity of natural languages completely misses the point. No one denies that the RNNs under scrutiny are simplistic caricatures compared to the actual complexity of brains. How many "hidden layers" do real neural networks have? The number is clearly greater than one and, with regard to language processing, probably more on the order of hundreds. Do vdV&dK truly claim that RNNs with hundreds of hidden layers cannot model combinatorial productivity? If not, as I suspect, their whole argument is unconvincing. The conclusion should be to push the envelope of more realistic RNNs rather than to discard the approach.

But following for now vdV&dK's argument that temporal coherence and RNNs are out and blackboards are in, the question arises: What can we tell about the blackboards in the brain? Blackboards are assumed to consist of specialized processors. Specifically for language, vdV&dK propose that "one could have" processors for words and for specific grammatical relations. This quotation from the article reflects the surprising nonchalance with which crucial assumptions are made. Cognitive neuroscientists have for many years painstakingly attempted to segregate lexical and grammatical processing, using mostly ERP and hemodynamic imaging techniques. Although a few authors have considered parts of the left inferior frontal gyrus (LIFG) to be a special processor for morphosyntax (e.g., Caplan et al. 1998)—sometimes supported by unusual statistical decisions (see Müller 2000)—the literature overwhelmingly shows that LIFG (and other perisylvian "language areas") show differential *modulation* of activity (Just et al. 1996). For example, there is enhanced activity in LIFG associated with syntactic violations, but activity in the identical region is also present for semantic violations (e.g., Embick et al. 2000). Such modulation would probably suggest shared rather than special processors, although it remains possible that special processors are too small to be resolved in neuroimaging.

One way to go beyond these limits of spatial resolution is to examine the cellular architecture in human LIFG. A few cytoarchitectonic studies of LIFG have yielded intriguing developmental patterns in dendritic complexity (Simonds & Scheibel 1989) as well as neuronal density and its asymmetries (Amunts et al. 2003). These studies speak against the idea of a unique type of cortex in LIFG that might itself explain unique functionality—morphosyntactic, lexicosemantic, or other. Instead, the area's cytoarchitecture appears to be best characterized as sharing some features with premotor cortex and some with polymodal association cortex (Amunts et al. 1999; Jacobs et al. 2001).

It remains possible that cellular or architectonic parameters yet to be investigated will in the future yield evidence for special processors. However, based on the current evidence, it appears much more likely that a "language area" such as LIFG can assume a set of specific functions, not based on specific architecture of modular processors but because it has a unique pattern of connectivity with other brain regions that permit the convergence of information critical to linguistic processing (Müller & Basho 2004). It is hence a unique profile of distributed neural connectivity rather than the local architecture of special processors that makes language happen in LIFG and other regions of left perisylvian cortex.

In summary, although vdV&dK's blackboard model draws a little bit from neuroscience regarding gating circuits, most of their arguments make insufficient use of the relevant evidence. This criticism does not imply that neural network modelers are asked to derive their models directly from neuroscientific evidence. However, those who believe that an ultimate understanding of cognition lies in the biology of the brain can expect that neuroscientific evidence be accepted as a set of *constraints* in neural network models. Otherwise, what are we to make of a model that optimally describes a functional domain, such as combinatorial processing, making assumptions inconsistent with the workings of the brain? Useful it may be, but could it be explanatory?

Constituent structure and the binding problem

Colin Phillips and Matthew Wagers

Department of Linguistics, University of Maryland, College Park, MD 20782.
colin@umd.edu **mwagers@umd.edu**
http://www.ling.umd.edu/colin **http://www.ling.umd.edu/matt**

Abstract: van der Velde's & de Kamps's model encodes complex word-to-word relations in sentences but does not encode the hierarchical constituent structure of sentences, a fundamental property of most accounts of sentence structure. We summarize what is at stake and suggest two ways of incorporating constituency into the model.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

We are impressed by the attempt by van der Velde & de Kamps (vdV&dK) attempt to take seriously the challenge of capturing the complexity of human language in a neurally plausible model. Their model makes it possible to ask questions about the encoding of the details of sentence structure that it was difficult even to ask previously. This is no mean achievement. Nevertheless, we are concerned that the authors' model avoids one of the most fundamental properties of sentence structure and that this could seriously restrict the scope of the model.

Although many of the figures in the target article bear a superficial resemblance to the phrase structure trees of linguistics, the sentence structure representations in the neural model lack the hierarchical constituent structure encoded in phrase structure trees. Phrase structure trees encode bindings between primitive elements (words) that create constituents and also bindings between constituents that form larger constituents. In vdV&dK's model, in contrast, only bindings between the basic word-level structural assemblies are encoded. A verb's theme subassembly may be temporarily bound to a noun's theme subassembly to form the equivalent of a simple verb phrase, but the verb phrase does not itself combine with other subassemblies to form larger constituents. The S and C structural assemblies that are employed in the encoding of main clauses and embedded clauses, respectively, do not delimit clause-sized constituents. Rather, they are word-level structural assemblies whose subassemblies bind with the subassemblies of other word-level units.

The binding of words and phrases to form hierarchically organized constituent structures is a property shared by a wide variety of linguistic models that differ in many other respects (e.g., Bresnan 2001; Chomsky 1965; Frank 2002; Goldberg 1995; Pollard & Sag 1994; Steedman 2000), and it plays a crucial role in explanations of many linguistic phenomena. These include the following:

i. Coordination rules. In most cases, like categories can be combined with the conjunction *and* to form a larger instance of the same category: nouns coordinate with nouns, verbs with verbs, verb phrases with verb phrases, and so on. In the absence of a mechanism for encoding recursive constituent structures in vdV&dK's model, it is difficult to capture the fact that *John and Mary* is a noun phrase that governs plural verb agreement, or the fact that *The managers and the pilots who supported the strike* is a noun phrase in which the relative clause may modify only *pilots* or both *managers* and *pilots*.

ii. Anaphoric relations. Languages make extensive use of anaphoric expressions that are interpreted as taking the meaning of another constituent in the sentence or discourse. Pronouns such as *he* or *them* may corefer with another noun phrase constituent, and forms like *it* or *so* may be anaphoric to a clause-sized constituent, as in *The sun was shining, but Sue couldn't believe it*. The expression *do so* is anaphoric to a verb phrase, which may be a larger constituent, as in *Bill finished his homework on Tuesday and Sally did so too*, or a smaller constituent, as in *Bill finished his homework on Tuesday and Sally did so on Thursday*. It is difficult to capture such dependencies in a model that lacks hierarchically organized constituents.

iii. Long-distance dependencies. The long-distance discrepancies found in wh-questions, topicalization, relativization, passivization, raising, and scrambling structures consistently involve the appearance of a constituent in a noncanonical position. It is difficult to capture such rules without constituents.

iv. Scope relations. Recursive formation of constituents makes it straightforward to capture the fact that the expression *second-longest American river* refers not to the Amazon – the second longest river and also an American river – but rather to the Mississippi-Missouri, which is the second longest among American rivers.

v. Command relations. Many syntactic relations are restricted to constituents that stand in a c-command relation, the relation that holds between a constituent and its sister and all subparts of its sister. For example, negative polarity items such as *ever* and *any* must be c-commanded by a negative expression. This constraint captures the acceptability of *Nobody thinks that Bill ever sleeps* and the unacceptability of **A man that nobody likes thinks that Bill ever sleeps*. The absence of constituents in vdV&dK's model makes it more difficult to capture structural generalizations of this kind.

vi. Recursive modification. Modifier expressions such as adjectives and relative clauses may be freely combined with the categories that they modify, in any quantity, as in *six big red India rubber balls*. In grammars with hierarchically organized constituents, this can easily be captured using a recursive rule such as $N' \rightarrow Adj\ N'$. In vdV&dK's model, however, modifier expressions are bound to the categories that they modify by means of dedicated subassemblies, and multiple modifiers require multiple dedicated subassemblies. It strikes us as inefficient to require all noun structural assemblies to include a special adjective subassembly that is exploited only in noun phrases with six or more adjectives.

vdV&dK correctly note that combinatorial productivity and recursive productivity are separable issues. Combinatorial productivity can obtain in the absence of recursive productivity so long as there is arbitrary binding between fillers and roles. Recursive productivity, they note, "deals with the issue of processing more complex syntactic structures, such as (deeper) center-embeddings" (sect. 4.2). The above discussion illustrates, we hope, that, at least for natural language, recursive productivity – that is, constituent depth – is at issue even for simple syntactic structures.

We can imagine at least two ways in which the neural blackboard architecture could be extended to encode hierarchical constituent structure without sacrificing the main insights of the model. One possibility would be to add new structural assemblies that correspond to nonterminal nodes in a phrase structure tree. For example, assemblies for the categories NP and VP would bind with other categories and not with individual words. All NP assemblies would then need to have a number of subassemblies that would allow them to bind with any potential mother or daughter node of NP. An alternative possibility would be to directly exploit the delay assemblies that are activated in a memory circuit when a pair of subassemblies is bound together. If the delay assembly could double as a structural assembly for a constituent node that could bind with other constituent nodes, then this might allow encoding of hierarchical constituent structure. Indeed, vdV&dK hint that a pair of bound structural subassemblies can themselves be bound to another pair of bound subassemblies, when they draw a dotted line between the n and v subassemblies in Figure 10 as a means of capturing subject-verb agreement. Crucially, the model must be able to encode not only the first-order relationships between word-level primitives but the second-order relationships between relationships that characterize constituency in natural language.

Whether these or any other solutions turn out to be most feasible, we suggest that the neural blackboard architecture cannot properly address the challenge of the "massiveness of the binding problem" (Jackendoff 2002) unless it is able to recursively encode constituents and bindings among constituents.

On the unproductiveness of language and linguistics

David M. W. Powers

School of Informatics and Engineering, Flinders University of South Australia, Adelaide SA 5001, Australia. ***David.Powers@flinders.edu.au***
http://www.infoeng.flinders.edu.au/people/pages/powers_david

Abstract: van der Velde & de Kamps (dvV&dK) present a response to Jackendoff's four challenges in terms of a computational model. This commentary supports the position that neural assemblies mediated by recurrence and delay indeed have sufficient theoretical power to deal

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

with all four challenges. However, we question the specifics of the model proposed, in terms of both neurophysiological plausibility and computational complexity.

It is often assumed that language is vastly different from and largely independent of other cognitive processing, and this modularity fallacy is apparent even in this target article, notwithstanding its attempt to explain aspects of language in more general cognitive and perceptual mechanisms exemplified in the examples given. In particular, in introducing the massiveness of the binding problem (sect. 2.1), it is suggested that *cat, mouse*, and *chases* activate specialized populations of neurons and distinguish word order in a manner similar to the operation of motion detectors for a vertical bar. But then it is argued that language is fundamentally different because motion detection is able to be specialized for motion of a limited set of possibilities, but language has unlimited combinatorial productivity. In fact, the productivity has nothing to with language or linguistic theory. Rather the complexity derives from our environment, and even without language, an organism is continually faced with new situations, new entities, and new interactions and interrelationships.

The fact that many things can chase a mouse has nothing to do with language. In a book, Dumbledore would be introduced with a sentence, whereas in real life he would be introduced verbally and mutimodally by a person in a direct personal encounter or might just be seen chasing the mouse without his name or profession being known.

The visual system detects the motion of Dumbledore, the cat, or anything else irrespective of any linguistic bindings. The fact of motion is detected at a very primitive level, and work such as Hubel's (1995) indicates that there are neural detectors for salient features such as edges in motion. The mechanisms by which an object is recognized, its motion is recognized, and its name is recognized all seem to be similar and are often theorized to be associated by neural synchrony on the basis of empirical evidence of synchronous neural activity (Roelfsema et al. 1997; Sarnthein et al. 1999; Shastri & Ajjanagadde 1993b; von Stein et al. 1999).

The problem of variables (sect. 2.3) is in essence an artefact of the assumption of rule-based structures, and both are linguistic constructs that probably have no concrete correlate in brain function. Rules and variables, moreover, do not necessarily occur in modern statistical language learning approaches: rules are implicit in supervised approaches involving tree-banks (Marcus 1991), probabilistic grammars (Charniak 1993), and/or data-oriented parsing (Bod 1995) but are supplanted by a more general concept of prosodic, phonological, morphological, syntactic, and semantic patterns in unsupervised approaches (Clark 2001; Hutchens 1995; Powers 1983). The underlying phenomenon whereby variables get attached to values (in a rule-based approach) or abstracted patterns get matched with current sensory-motor or linguistic content is again a matter of binding or association, which is commonly dealt with by theories of synchrony (Weiss et al. 2001).

However, van der Velde & de Kamps (vdV&dK) do not see synchrony or recurrence as a panacea for Jackendoff's challenges but rather show how various early models exhibit exactly these problems. They point out that that the Shastri and Ajjanagadde solution to the multiple binding problem is duplication and that this then faces problems with nested structures and implies a "one-level restriction." This is technically incorrect, but the argument does indeed imply a "finite levels restriction" which is consistent with Miller's (1956) Magical Number Seven constraints, with the inability of humans to cope with arbitrarily complex embedding, with phenomena such as subjacency, and with the observation that there is insufficient space in our heads for the infinite stack implied by linguistic theories that see language as strictly more complex than context-free.

Synchrony involves a specific pattern that is present in each neuron triggered as a result of a specific object or event, and this pattern represents a temporal encoding that would seem to encode low (<20 Hz) frequencies as well as information correlated with spatiotemporal location that results in higher-frequency components and evidently has a reverberatory or recurrent origin. A Hebbian neural assembly is intrinsically a spatiotemporal association, and the concept of synchrony adds the potential for overloading in the sense that the same neurons can synapse into multiple assemblies with different characteristic signatures. The circles or triangles that represent terminal or nonterminal symbols linguistically in vdV&dK are in fact intended to represent assemblies neurologically, and these are intrinsically dynamical structures that exhibit synchrony and provide evidence of recurrent processes (Hebb 1949; Pulvermüller 2002), although this is not made explicit in the target article.

There are, moreover, alternatives to the duplication approach as well as refinements such as a homunculus model built upon evidence of a propensity for spatiotemporal reception fields and projections that reflect information-distorted sensory-motor representations of the opposite half of the body (Powers & Turk 1989). Plausible models should also take account of attention and saccade and the evidence that we maintain a relatively high-level spatiotemporal model of our environment that is informed by attended events and peripheral change (e.g., in the visual domain, motion or lightening; in the auditory domain, modulation or softening). The spatiotemporal representations have very much the function of the more abstract pretheoretic blackboard metaphor. Powers (1997) envisions the spatiotemporal model as being like the columns of blackboards in a lecture theatre – different columns for different sensory-motor or cognitive modalities, different levels for different times and levels of complexity. In the lecture theatre, a board is pushed up after it has been written on, and a clean board emerges from below. While working on the current layer of boards, one can refer to any board on any higher layer that is still visible.

vdV&dK's model is largely consistent with this kind of model but is more reminiscent of sequential digital logic circuits and in fact makes the analogy explicit through the use of gate terminology. Synchrony, reverberation, and recurrence would seem to be important mechanisms in realizing their model, although there is an interacting pair of neural attributes that they neglect: delay and decay. Delay is clearly implicit in reverberatory and recurrent models, but delayed copies of a signal can exist at different parts of the network even without recurrence. These delayed copies create the temporal dimension of a blackboard-like spatiotemporal representation. Hence a neuron is as much a short-term memory element as a processing element and functions something like a dynamic memory cell that maintains its memory while it is being refreshed as relevant. A complementary effect is represented by the refractory period and its role in habituation. Powers (1983) used this delay-decay model (as well as a more abstract statistical model) for induction of both grammatical and ontological structure.

Although presented in an unhelpful notation that is nonstandard for both neurons and gates, the vdV&dK gating model is similar to sequential digital logic circuits, and the resulting model of memory acts like a static memory cell. Whilst the model is sufficient to provide solutions for Jackendoff's problems, it is considerably more complex than the simple delay-decay model, and there is no direct neurological support for this kind of model except in terms of the ongoing synchronic recurrence between features triggered for the same event that forms the Hebbian assembly.

The elaboration of the vdV&dK model is intrinsically syntactic in nature and this extends to their models of short-term (blackboard) and long-term (knowledge base) memory. There is no ontology, no semantics, no meaning captured in distinguishing *The cat chases the mouse* from *The mouse chases the cat.* There is no discussion of how *cat* is recognized or the verb *chases* is understood, and the representation of words with circles, supposedly representing neural assemblies, fails to capture the

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

inherent overlap of the distributed representation of a Hebbian assembly (Pulvermüller 2002). A more realistic model involves abstraction of entire scenes and association of words with those scenes in the same way as any other attributes of the objects or events involved, these structures reflecting (Piaget 1954) the multimodal multilevel spatiotemporal short-term delay-decay representation of the current ontological context in the blackboard network. The hearing or reading of a sentence generates a sequence of neural patterns that starts off being perceptual in nature and becomes increasingly abstract and conceptual as well-attested neural processes extract features and relationships. The sentence and its context, both linguistic and ontological, are thus represented in different layers of the network at different complexity levels and time points – the spatiotemporal neural blackboard. The function of recurrence and inhibition is not to implement short-term memory but to allow the representation of complex spatiotemporal relationships.

Linguists tend to get this backwards. It is not that language determines who is the subject or the object in a sentence, but rather that in the environment there is an actor and an undergoer (Pike & Pike1977). The reality has primacy, and the underlying representation is arguably there not to represent language but to represent past, present, and potential experience. The issues Jackendoff raises are not primarily problems of linguistics but matters of ontogenesis. In dealing with problems that Jackendoff poses from a linguistic perspective, vdV&dK tend to force their cognitive model into a linguistic mould rather than producing a proper ontological model and showing how linguistic relationships, semantic, phonetic, morphological, and syntactic, can be represented – or, indeed, can be emergent.

It is therefore highly appropriate that vdV&dK conclude by looking at how their blackboard architecture maps onto vision, which is arguably representative of the entire array of sensory-motor and cognitive modalities. This would have been a better starting point, as understanding this kind of feature-binding model can potentially lead to a better understanding of syntactic and semantic binding.

Comparing the neural blackboard and the temporal synchrony-based SHRUTI architectures

Lokendra Shastri

International Computer Science Institute, Berkeley, CA 94707.

shastri@icsi.berkeley.edu **http://www.icsi.berkeley.edu/~shastri**

Abstract: Contrary to the assertions made in the target article, temporal synchrony, coupled with an appropriate choice of representational primitives, leads to a functionally adequate and neurally plausible architecture that addresses the massiveness of the binding problem, the problem of 2, the problem of variables, and the transformation of activity-based transient representations of events and situations into structure-based persistent encodings of the same.

Table 1 compares two sets of solutions to the challenges posed by Jackendoff (2002). One set of solutions is provided by the SHRUTI architecture, which uses temporal synchrony for encoding dynamic bindings (Mani & Shastri 1993; Shastri 1999; Shastri & Ajjanagadde 1993b; Shastri & Wendelken 2000), and the other by the target article. This comparison is clearly at odds with what is stated in the target article. The following discusses the bases of the comparison and point out some of the factual errors and faulty analyses underlying the flawed evaluation of the temporal synchrony approach proposed in the target article.

The SHRUTI architecture represents relations (or predicates), types, entities, and causal rules using *focal-clusters*. Figure 1 depicts focal-clusters for relations (e.g., give), types (e.g., Person), entities (e.g., John), and the rule *give(x, y, z)* ⟨*own(y, z)*. Within a focal-cluster, the activity of the + node represents a degree of belief, the activity of the ? node represents querying of information, and the synchronous firing of a role node (e.g., *giver*) and an entity's (or type's) + node represents the dynamic binding of the role and the entity (or type). Type + nodes are further differentiated to encode quantification (*e* for existential and *v* for universal). Thus the sustained activity of +: *give* together with the firing of +: *John*, +: *Mary*, and +*e: Book* in synchrony with *giver*, *recipient*, and *give-object*, respectively, encodes the *active belief*: "John gave Mary a book." This activity immediately leads to the inference "Mary owns a book" because of synchronous activity propagating along connected nodes (e.g., *owner* synchronizes with *recipient* and hence with +:*Mary*).

Table 1 (Shastri). *An evaluation of two architectures that provide solutions to Jackendoff's four challenges to cognitive neuroscience*

Jackendoff's Challenges to Cognitive Neuroscience	Temporal Synchrony-based SHRUTI Architecture	Neural Blackboard (NBB) Architecture
Massiveness of the binding problem	A	A
Problem of 2 (multiple instantiation)	B	B
Problem of variables (reasoning with abstract rules)	A	D
Activity-based (dynamic) bindings and long-term bindings	A	I (incomplete)

Contrary to what is claimed in the target article (sect. 3, para 2; sect. 3.2, para 1), *no preexisting fact nodes or synchrony detectors* are required for the activity-based encoding of "John gave Mary a book," and no such nodes/detectors are required for drawing inferences based on this fact. Furthermore, as long as SHRUTI is told that Dumbledore is an entity or an instance of an existing type, it will have no problem encoding the novel event "John gave Dumbledore a book" and productively inferring that Dumbledore owns the book.

This brings out the first major error in the authors' understanding of the temporal synchrony-based SHRUTI architecture. Contrary to their claim, SHRUTI does not require prewired fact nodes for *all possible* facts. SHRUTI requires fact nodes (actually, fact circuits) only for encoding *memorable facts in its long-term memory*.

Problem of 2. Although the simple network shown in Figure 1 permits an entity simultaneously to fill multiple roles in different relations, it cannot simultaneously encode multiple instances of the same *event-type* (e.g., "John gave Mary a book" and "Mary gave John a pen") without binding confusions. Shastri and Ajjanagadde (1993b) present a solution to this problem within the temporal synchrony framework. The solution requires having a small number of copies of each relational focal-cluster and encoding rules by interconnecting antecedent and consequent focal-clusters via a switching circuit (Mani & Shastri 1993; see Wendelken & Shastri 2004 for an alternate solution).

The authors correctly point out that the use of multiple copies of focal-clusters makes it difficult to learn regularities between relations. Because an occurrence of a situation wherein a *give* event leads to an *own* event would engage only one focal-cluster each of *give* and *own*, respectively, only this pair of focal-clusters will learn the causal link between *give* and *own*.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

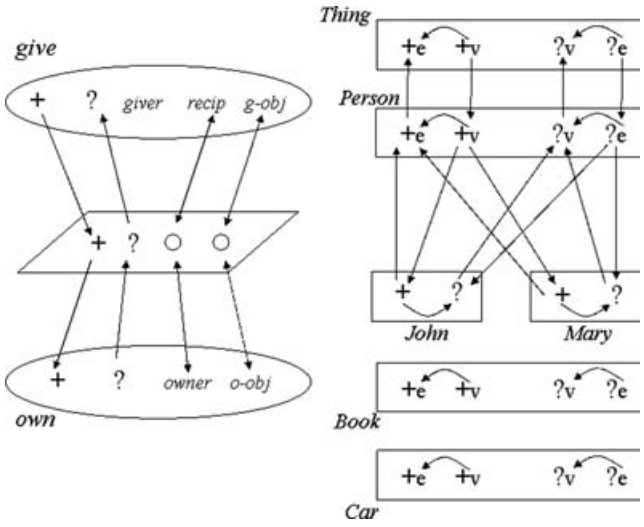


Figure 1 (Shastri). A simplified depiction of how relations, entities, types, and rules are represented in SHRUTI. Labels such as +, ?, and *giver* are nodes. Each node corresponds to a small ensemble of cells. A bidirectional link is a shorthand notation for two links, one in either direction. Only some of the nodes and links are shown (e.g., like Person, Book and Car are also subtypes of Things).

Learning the link between all pairs of *give* and *own* focal-clusters would require either a mechanism to perform automatically the requisite weight changes across all pairs of focal-clusters *or* a large number of occurrences (sooner or later each pair of *own* and *give* focal-clusters would participate in a relevant situation and learn the link).

Unfortunately, *the solution proposed by the authors suffers from exactly the same drawback as that suffered by the solution developed for SHRUTI.* Instead of using multiple copies of relational focal-clusters, the neural blackboard (NBB) architecture uses multiple copies of assemblies for each linguistic constituent (e.g., S, NP, and VP). The use of multiple copies renders the learning of structural dependencies and syntactic constraints between constituents very difficult, because *any regularity learned from a sentence will be recorded only in the copy of a constituent that was used to encode the sentence*; *it will not generalize to all the other copies of the constituent*. Hence, although the noun subassembly of N_1 (or S_1 or C_1) and the verb subassembly of V_1 (or S_1 or C_1) may learn the correct constraints/rules about noun-verb agreement, the noun and verb subassemblies of other copies of NP, S, and C assemblies will not. The lack of generalization over multiple copies will also manifest itself in the learning of (i) interactions between constituents and control circuits governing sentence parsing; and (ii) gating circuits for controlling the flow of activity between subassemblies.

Problem of variables (reasoning with abstract rules). The NBB architecture can look up and extract bindings from compositional structures, *but contrary to what is said in Section 6.6, it cannot perform reasoning with such structures*. The authors' state that "information of the form *own(X, ?)* can be transformed into information of the form *give(-, X, ?)* on the basis of a long-term association between *own-agent* and *give-recipient* (as in the model by Shastri & Ajjanagadde 1993[b])." Indeed, SHRUTI can make such transformations rapidly. If the state of the network shown in Figure 1, is initialized such that ?:*own* fires and *owner* and *own-object* fire in synchrony with ?:*Tom* and ?:e:*Thing*, respectively, the resulting network state represents the *active query*: "Does Tom own some thing?" Spreading activation within the type hierarchy and rules would transform this query into a large number of queries, including "Did a person give Tom something?"

But SHRUTI's ability to make such transformations has no bearing on the ability of the NBB architecture to perform reasoning. SHRUTI can make such transformations rapidly because it explicitly encodes (i) each semantic role of relations such as *give* and *own* (e.g., *give-recipient* and *own-agent*) and (ii) the systematic associations between these roles (see the encoding of $give(x, y, z) \Rightarrow own(y, z)$ in Fig. 1). In contrast, NBB captures the compositional structure of sentences involving *give* and *own* by binding the role-fillers of give and own in a given sentence to generic linguistic constituents and generic thematic roles (fig. 14 of the target article). NBB, however, does not explicitly encode semantic roles such as *give-recipient* and *own-agent* – neither in its activity-based representation nor in its long-term memory – and it does not have any representational machinery for capturing the associations between such semantic roles. Consequently, it cannot readily transform *own(X,?)* into *give(-, X, ?)*.

Interaction between activity-based (dynamic) bindings and long-term bindings. The authors' proposal about encoding constituent binding in long-term memory suffers from two problems. First, the authors entertain a variant of the long-term learning problem that is of limited relevance from the standpoint of cognition. Second, they present a solution that is unlikely to work.

The idea of storing sentence structure in long-term memory seems ill motivated. It is widely believed that we remember the semantic content of what we hear and read and *not* the actual linguistic input used to convey the content. At the sentence level, this semantic content corresponds, typically, to events and situations that can be viewed as instantiations of multi-dimensional relations (e.g., semantic frames). Language, then, is a means of conveying the bindings of roles and parameters in a *multidimensional* relational instance using a *one-dimensional* stream of words, and parsing is the process by which the appropriate bindings of a semantic frame are extracted from a stream of words. The sentence structure (or parse tree) is only a means to this end. Hence what need to be memorized are events and situations conveyed by sentences, not sentence structures.

The authors' solution to the problem of one-trial learning of sentence structure using the hippocampal complex (HC) (sect. 6.5) rests on untested assumptions and is unlikely to work. The solution requires HC to create a conjunctive encoding of the activity of the delay assemblies of all the memory circuits involved in representing the sentence structure (sect. 6.5.1). The number of such assemblies – for even moderately complex sentences – is likely to be large (>10), and it is not clear whether the requisite conjunctive representations can be recruited in the HC while keeping interference within reasonable bounds. It is also not clear how many distinct sentence structures can be memorized. The answers to these questions depend on, among other things, the anatomy of the HC, the number of cells in different subregions of the HC, the density of projections between subregions, and the physiological parameters governing LTP (e.g., how many concurrent inputs are required to induce LTP). An analysis of some of the available data about the human HC suggests that if one wants to maintain a low level of cross-talk, the maximum number of bindings that can be grouped together in a conjunct is likely to be small (~ 7) (Shastri 2001b; in preparation). In view of this, it would seem difficult to memorize any but the simplest of sentence structures.

Fortunately, the problem of one-shot memorization of events and situations *described by* sentences seems more amenable to a biologically plausible solution. A computational model of cortico-hippocampal interactions (Shastri 2001a; 2001b; 2002; in preparation) demonstrates that a cortically expressed transient pattern of activity representing any arbitrary event

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

can be transformed rapidly into a persistent and robust memory trace in the HC as long as the number of distinct role-fillers specified in the event remains less than approximately seven.

Nested structure. As discussed by Shastri and Ajjanagadde (1993a), a possible way of increasing nesting levels is to use a richer temporal structure whereby bindings at a deeper level of nesting are represented by synchronization at faster frequencies (e.g., gamma band) and bindings at a shallow level of nesting by slower frequencies (e.g., theta band). Moreover, as discussed by Shastri and Ajjanagadde (1993a, sect. R2.5), many problems that seem to require deeper levels of nesting can be reformulated so as to require shallow nesting.

Parsing. The NBB architecture parses English sentences; it encodes rules and constraints pertaining to grammatical knowledge and performs inferences required for parsing sentences (c.f. target article sect. 6.8.4). Given the focus of the NBB architecture on parsing, it is odd that the authors did not compare their approach to that of Henderson (1994), who developed an online, incremental parser motivated by the temporal synchrony framework. Henderson showed that many limitations of human parsing can be explained by constraints on working memory imposed by the use of temporal synchrony to solve the binding problem. Henderson's parser can recover the structure of arbitrarily long sentences as long as the dynamic state required to parse the sentence does not exceed the bounds of working memory. In doing so the parser explains a range of linguistic phenomena pertaining to long-distance dependencies, local ambiguity, and center-embedding.

Can neural models of cognition benefit from the advantages of connectionism?

Friedrich T. Sommer[a,b] and Pentti Kanerva[a]

[a]*Redwood Neuroscience Institute, Menlo Park, CA 94025;* [b]*Redwood Center of Theoretical Neuroscience, University of California at Berkeley. Berkeley, CA 94720-3190* ***fsommer@berkeley.edu***
http://www.rni.org/ftsommer/FS/FTSommer.html
pkanerva@rni.org http://www.rni.org/kanerva/homepg.html

Abstract: Cognitive function certainly poses the biggest challenge for computational neuroscience. As we argue, past efforts to build neural models of cognition (the target article included) had too narrow a focus on implementing rule-based language processing. The problem with these models is that they sacrifice the advantages of connectionism rather than building on them. Recent and more promising approaches for modeling cognition build on the mathematical properties of distributed neural representations. These approaches truly exploit the key advantages of connectionism, that is, the high representational power of distributed neural codes and similarity-based pattern recognition. The architectures for cognitive computing that emerge from these approaches are neural associative memories endowed with additional mapping operations to handle invariances and to form reduced representations of combinatorial structures.

Introduction. There is an abundance of cognitive phenomena that our existing models are unable to explain. Jackendoff (2002) has singled out four from language, and the target article proposes to solve them with a neural blackboard architecture. How likely are we to come up with the right cognitive architecture by starting with language? Although language is a cognitive function without an equal, it is only a fraction of what makes up human cognition. It is also evolution's latest invention. In keeping with the principles of biological design, it rests fundamentally on earlier inventions that are found in mammals and many other animals. These include rich sensor integration, complex learned motor-action sequences, memory for places and things, learned judgment, memory for learned actions, mental imagery (imagination), learning from example, and signaling. In other words, neural mechanisms that predate language are capable of incredibly rich adaptive behavior. It therefore seems prudent to seek neural-based solutions to these more basic cognitive functions first and then to model language by elaborating on the mechanisms for the basic functions. However, that is not the tradition in cognitive modeling (Feigenbaum & Feldman 1963; Newell & Simon 1972).

Local versus distributed representation. Language is the part of cognition that we are the most cognizant of. This has led us to modeling the rest of cognition with language and language-like systems such as logic, rather than the other way around. The target article exhibits the typical hallmark of these approaches, local representation, as van der Velde and de Kamps (vdV&dK) represent words by nonoverlapping cell populations. In nature, this single-neuron–single-function design is found only in relatively primitive animals, in the most basic life-sustaining functions of higher animals, and at the outer sensor and motor periphery – that is, far from where cognition is assumed to take place.

Local representations make rigid assignments of what part in a connectionist architecture represents what entity. The more flexible and more promising alternative of distributed coding goes beyond simply replacing single cells by cell populations. A code is distributed if the neurons truly collaborate. In other words, not any single neuron in the network but only a distributed pattern of neural activities can fully specify the meaning of a representation. Cell assemblies, by Hebb's (1949) definition, fulfill this property. Experimental neuroscience offers ample evidence for distributed codes. Braitenberg (1978) has pointed out that results of single-cell physiology strongly suggest distributed representations. Many neurons respond to generic features (such as edges in visual input) that are shared by a large class of stimuli rather than responding to one specific stimulus. Recent techniques for recording the activity in many cells at the same time prove direct evidence for distributed patterns (Harris et al. 2003; Ikegaya et al. 2004). Quite confusingly, vdV&dK use local representations but describe their model in the language of (distributed) cell assemblies and populations.

Advantages of distributed representation have been emphasized in early artificial neural-network models, such as neural associative memories. Distributed representation yields drastically increased representational power: A network with n neurons can represent n/k localist population codes with k neurons, but (n choose k) $\approx (n/k)^k$ distributed representations. This means that a connectionist architecture with distributed representation can process more items than it has neurons, which might be crucial for many brain regions. Second, distributed representations form metric spaces where the metric is given by pattern similarity, such as pattern overlap or the inner product. In high-dimensional representational spaces, these metrics behave entirely differently from the Euclidean distance in two- or three-dimensional space. For example, a pair of random vectors is in general almost maximally dissimilar. The properties of these metrics are the mathematical basis for the pattern-recognition properties of artificial neural networks.

Neural architectures for processing distributed representations. Neural associative memories (Hopfield 1982; Kohonen 1977; Palm 1982; Willshaw et al. 1969) are essentially the artificial neural-net implementation of cell assemblies (Sommer & Wennekers 2003). In these models, cell assemblies are distributed memory patterns that can be formed by learning and can later be recalled. A cell assembly is formed by incrementing the synaptic strengths between all its active neurons and it can be recalled by stimulating a large enough fraction of its cells. The recall is a pattern-recognition process based on the previously described pattern similarity: a noisy input pattern will evoke this most-similar cell assembly. Associative memories also provide a form of short-term

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

memory: once a cell assembly is active, the recurrent excitation keeps it persistent (Amit 1989).

However, the above-described models of associative memory cannot perform invariant pattern recognition nor can they represent composite or combinatorial structures. The main problem is that memory recall hinges exclusively on overlaps with stored patterns. Input patterns that do not have meaningful overlaps with stored patterns cannot be recognized. Architectures that overcome this limitation have been proposed (Anderson & van Essen 1987; Arathorn 2002; Dotsenko 1988; Olshausen et al. 1993), and we refer to them as feature-mapping memories. In these models, the input is transformed by a parameterized mapping before it enters the associative memory. By tuning the mapping parameters while searching for a recognizable stored pattern, these systems can cope with invariance that is not reflected in the direct overlap-similarity of the input. The result of memory retrieval is then not only a stored pattern but also the appropriate mapping parameters for mapping the current input to the memory. These architectures have also been extended to more general graph-matching operations (Bienenstock & von der Malsburg 1987; Kree & Zippelius 1988).

The above feature-mapping memories can perform invariant pattern recognition but they do not provide for a more sophisticated working memory that would allow combinatorial structures to be composed and decomposed on the fly. This requires the introduction of an (invertible) mapping operation for binding (and decomposition) of representations. Smolensky (1990) proposes a map into higher-dimensional space, the tensor product of the representational spaces of the constituents. Such a dimension-expanding binding operation can keep the full structure of the constituents. But this comes at a high price. The problem is that the depth of nested structure is severely limited by affordable dimension and that atoms cannot be substituted for nested structures. Because of the downsides of dimension-expanding binding, Hinton (1990) proposes reduced representation of nested structures in the original space of the constituents. Different binding operations have been proposed for generating reduced representations, in particular, convolutions (Plate 1994; 2003) elementwise multiplication (Gayler 1998; Kanerva 1994; 1997), and permutation-based thinning (Rachkovskij & Kussul 2001).

Gayler (2003) details how the Jackendoff problems can be solved with reduced representations. For example, he uses binding of a representation with a permuted version of this representation to generate frames (for keeping the little star and the big star apart). It is important to note that systems of reduced representations have to be embedded in networks of cell-assembly memory, as we describe above. This is because the reduction in the binding process produces a noisy result upon decomposition. The patterns resulting from decomposition have to be passed through a similarity-based filtering process, a clean-up process. In this regard the cognitive operations with reduced representations have close similarity to the recognition of a pattern in feature-mapping memories that we describe above.

Conclusions. We believe that connectionist models for cognitive processing should exploit the strengths of connectionism, that is, high representational power and the ability for pattern recognition. These strengths rely fundamentally on distributed representation and, as we explain, are not realized in the blackboard architecture proposed in the target article. Neural associative memories in their original form incorporated the described strengths of connectionism but were too limited in function. Recently, however, memory-based architectures have been developed that can perform invariant pattern recognition in difficult perceptual problems (Arathorn 2002). Hence it seems promising to realize neural cognitive architectures by adding operations for processing reduced representations in similar memory systems. In general, we believe that serious progress in cognitive modeling will be based on understanding the general mathematical properties of high-dimensional representational spaces rather than on a specific solution to a relatively narrow set of challenge problems.

An alternative model of sentence parsing explains complexity phenomena more comprehensively without problems of localist encoding

Carol Whitney

Department of Linguistics, University of Maryland, College Park, MD 20742-7505.
cwhitney@cs.umd.edu ***http://www.cs.umd.edu/~shankar/cwhitney***

Abstract: I discuss weaknesses of the proposed model related to reinstantiation of encodings recorded by the hippocampal complex and to the inability of the model to explain complexity phenomena. An alternative model that also addresses the formation of hierarchical representations of sentences in working memory is outlined, and the ability of this model to account for complexity phenomena is briefly reviewed.

In this target article, the van der Velde & de Kamps (vdV&dK) address the difficult question of how the hierarchical structure of a sentence could be neurally represented in working memory (WM). Their solution essentially assigns a unique index to each instance of a word within a grammatical class by activating one of the N instances of each word. There is an $N \times N$ arrangement of gating nodes between any two grammatical classes that can be associated with each other. Activation of gating node *ij* encodes an association between items *i* and *j* from those grammatical classes. In this commentary, I point out some difficulties with their proposal and describe an alternative model (Whitney 2004).

One of the goals of the present model is to allow the representation of multiple sentences in WM, including reactivation of representations recorded in the hippocampal complex (HC). When a representation is constructed from the incoming input, each index is dynamically assigned, and the uniqueness constraint can be directly imposed. However, when a representation is reactivated from the HC, the indices have already been chosen. These indices might conflict with ones already active in WM. For example, assume that $N = 100$, and consider a sentence with four nouns encoded in WM, which has triggered the recall of another sentence with four nouns recorded by the HC. Assuming random initial choice of indices (under the no-duplication constraint), the probability that the two sentences will share at least one noun index is quite high – approximately 15% (i.e., 1.0 - 96/100*95/99*94/98*93/97). Increasing N by a factor of ten reduces the probability of a conflict to a more manageable level, but at the cost of an exponential explosion in the number of nodes – 1,000,000 gating nodes are then required for every pair of associable grammatical classes. Hence the model in its current form does not actually allow the reinstantiation of HC encodings under reasonable parameters.

van der Velde & de Kamps also claim that the proposed mechanisms for activating mappings between grammatical classes can explain complexity phenomena. They argue that the reason that a relative clause within a relative clause (RC/RC) is more difficult than a relative clause within a noun complement (NC/RC) (Gibson 1998) is because of the availability of two potential binders to the gap position in the innermost clause. However, a noun complement within a relative clause (RC/NC) is as difficult as an RC/RC (Gibson 1998), even when the NC's verb is intransitive. Because there is no ambiguity in this case, their account would incorrectly predict that an RC/NC should be as easy as an NC/RC. Of course, it would be logically possible to explain the RC/NC difficulty via an entirely different mechanism, but such a lack of parsimony is undesirable. There is a range of other complexity issues that the model does not explain, such as effects of the type of the innermost noun phrase (Gibson 1998; Kaan & Vasic 2004), the relative ease of cross-serial dependencies in Dutch

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

(Bach et al. 1986), and the perceived grammaticality of an RC/RC in which the second verb has been omitted (Gibson & Thomas 1999).

In contrast, the TPARRSE model of sentence parsing addresses these same issues (Whitney 2004), but does not have these shortcomings. Each word is represented by a vector, and grammatical relationships are encoded by combinatory vector operators. Because words and syntactic relationships are represented as distributed patterns, multiple instances of words are not required, and there is no need to choose indices. Such a representation is also more robust than the localist encoding proposed by the authors.

The construction of this distributed encoding is supported by a temporal representation of the relationships between the phrases of incomplete clauses. This temporal representation essentially encodes a pushdown stack via firing order. However, unlike an abstract stack, there is no direct way to remove (pop) the top item. Rather, the syntactic features of an item are recorded, and when it is time to pop that item, the "stack" is sequentially "searched" until an item containing all the required syntactic features becomes activated. That item and all successive items are then inhibited. This mechanism provides the basis for a unified explanation of sentence-complexity phenomena, which I briefly outline.

A noun phrase (NP) is indicated by the *Np* feature, and additional features are activated to record restrictions on the function of the NP. The *Pr* feature indicates that the NP is a pronoun; *Cl* denotes the start of an embedded clause, and *GapR* indicates that the embedded clause requires a gap (i.e., it must be a relative clause). For example, in a RC/RC in which all subjects are full noun phrases, the first NP (NP1 – subject of main clause) has the *Np* feature, and the second NP (NP2 – subject of outer RC) has the *Np, Cl*, and *GapR* features, as does the third NP (NP3 – subject of inner RC). During processing of the inner RC, the representations of these three subjects fire in order, repeating over time. When the inner RC is complete, its encoding should be deleted from this temporal list. As the list is sequentially read out, the NPs are checked for the syntactic features recorded from the current subject (i.e., *Np, Cl, GapR* from NP3). NP1 does not contain these features, but NP2 does, and inhibition is initiated. However, this was the wrong NP at which to begin inhibition; the subject of the outer RC is incorrectly deleted. Therefore, processing of the outer RC cannot be reinitiated, and the main clause is reinstantiated instead. Hence the sentence does not make sense, and an ungrammatical version of the sentence in which the second verb is omitted can seem preferable (Gibson & Thomas 1999).

If the sentence is an RC/NC, such an incorrect deletion will also occur, as all of NP3's features (*Np, Cl*) are also activated for NP2 (*Np, Cl, GapR*). Incorrect deletion is prevented if NP2's features do not include all of NP3's. This will be the case if NP3 is a pronoun or if the sentence is an NC/RC, explaining the relative ease of such constructions (Gibson 1998). For a more in depth discussion of these issues, including the processing of cross-serial dependencies and the proposed neural underpinnings of the distributed and temporal encodings, see Whitney (2004).

Authors' Response

From neural dynamics to true combinatorial structures

Frank van der Velde[a] and Marc de Kamps[b]

[a]*Cognitive Psychology, Leiden University, Wassenaarseweg 52, 2333 AK Leiden, The Netherlands;* [b]*Robotics and Embedded Systems, Institut für Informatik, Technische Universität München, Boltzmannstrasse 3, D-85748 Garching bei München, Germany.* ***vdvelde@fsw.leidenuniv.nl kamps@in.tum.de***

Abstract: Various issues concerning the neural blackboard architectures for combinatorial structures are discussed and clarified. They range from issues related to neural dynamics, the structure of the architectures for language and vision, and alternative architectures, to linguistic issues concerning the language architecture. Particular attention is given to the nature of true combinatorial structures and the way in which information can be retrieved from them in a productive and systematic manner.

R1. Introduction

To begin, we would like to express our appreciation for the work done by the commentators. They have presented us with a wide array of comments on the target article, ranging from dynamics and neural structure to intricate linguistic issues. The topic of the target article was to solve the four problems described by Jackendoff (2002) and to illustrate that the solutions offered have the potential for further development. The problems described by Jackendoff concern the neural instantiation of combinatorial structures. Although it is true that language provides the most complex and hierarchical examples of combinatorial structures, these structures are also found in other domains of cognition, such as vision. Therefore we discussed sentence structure and visual binding as examples of combinatorial structures. We argued and illustrated that these structures can be instantiated in neural terms by means of neural blackboard architectures. We aimed to discuss as many topics as possible within the framework of a target article. As a result, we had to ignore many details and related issues. Our response to the commentaries offers us the opportunity to discuss some of these in more detail and to rectify some of the misunderstandings about the nature of the architectures that might have resulted from this approach.

R2. Goal

The goal of our article as outlined above was perhaps misunderstood by **Bod, Fitz & Zuidema** (**Bod et al.**), because they complain that we have not solved "the binding problem." From the remainder of their commentary we get the impression that "the binding problem," in the view of Bod et al., concerns the issue of how all the potential representations can be derived from a single sentence input. In the case of a multidisciplinary discussion it is always important to clarify terms because they could have a different meaning in different fields. For this reason, we introduced the goal of our article with a rather detailed discussion of the problems addressed by Jackendoff. The binding problem in this context refers to the notion of binding in neurocomputational terms. Jackendoff introduces it as follows (2002, p. 59):

> The need for combining independent bits into a single coherent percept has been recognized in the theory of vision under the name of the *binding problem* (not to be confused with linguists' Binding Theory . . .). In the discussions I have encountered, the binding problem is usually stated in this way: we have found that the shape and the color of an object are encoded in different regions of the brain, and they can be differentially impaired by brain damage. How is it, then, that we sense a particular shape and color as attributes of the same object?

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

Jackendoff goes on to explain that this binding problem also occurs in language, and even more massively than in vision. So the issue is how information encoded in different regions in the brain can be combined (bound) in a coherent representation, both in language and in vision. We argued that this can be done with blackboard architectures. Furthermore, we discussed the relation between the blackboard architectures in language and vision to emphasize the relation between the binding problems in both domains.

Bod et al. also complain that we have concentrated on the issue of representing combinatorial structures in neural terms. This remark ignores the difference between symbolic computation and neural computation. Briefly, what is easy in symbolic computation is difficult in neural computation, and vice versa. Representation of symbolic structures is easy in symbolic computation. Basically, anything goes. Hence there are no restrictions on the kind of structures and processes one can have in symbolic computation. This is one of the reasons why we have so many different theories in theoretical linguistics today. (For example, Bod et al. urge us to take more note of Chomsky 1957. But why this Chomsky? Why not the Chomsky of Government and Binding or the Chomsky of the Minimalist Program?)

The importance of the issue of representation can be seen very clearly in the need to copy symbols in the symbolic approach. Newell et al. (1989, p. 105) defend copying symbols as follows: "The need for symbols arises because it is not possible for all of the structure involved in a computation to be assembled ahead of time at the physical site of the computation. Thus it is necessary to travel out to other (distal) parts of the memory to obtain the additional structure." Hence it is necessary to copy the information over there and to bring it over here. But nodes in a connectionist network seem to be fixed in their web of connections. This was a main argument used by Fodor & Pylyshyn (1988) to dismiss connectionism as a basis for human cognition. In the words of Dennett (1991, p. 270): "This problem of fixed versus movable semantic elements is one way of looking at a fundamental unsolved problem of cognitive science." In our target article, we have provided a solution for this problem. Of course, this solution does not directly result in a complete theory of language and the brain. But, with it "the dialogue between linguistic theory and neural network modeling will begin to be more productive" (Jackendoff 2002, p. 65).

Bod et al. dismiss the architecture we present because it uses prewired connections. This remark again ignores the specific nature of neural computation. For example, to establish a relation between two words, there has to be a connection path between the neural structures that represent these words in the brain. Dramatic evidence comes from split-brain patients. In these patients, both hemispheres operate separately because the connection between the two is missing. For example, when the word *hammer* is presented in the left visual field, a split-brain patient will perceive the word with the right hemisphere. So the patient can use the left arm to point at an object representing a hammer. But he cannot name the word, because verbalization is (usually) produced by the left hemisphere, and this hemisphere is oblivious to the word perceived by the right hemisphere. Similarly, the left hemisphere can name the word *nail* when it is presented in the right visual field, but the right hemisphere is unaware of this word. So when both *hammer* and *nail* are presented together in this way, the split-brain patient is unaware of the relation between these two words, precisely because a connection path between their neural instantiations in both hemispheres is missing. Therefore, unless you want to assume that connections can grow on the fly, a prewired connection path has to be available to relate any two words with each other. The architecture we present provides a very parsimonious solution for this problem (see R4).

Bod et al. argue that we should have concentrated on the issue of finding the appropriate neural representations of all the syntactic structures that can be assigned to a given sentence. This remark illustrates that they are out of touch with the literature on human parsing. The human parser does not operate like a theoretical linguist. That is, the human parser does not wait until the sentence has been presented and then tries to make as many representations as possible. Instead, the parsing process starts straightaway and operates in an incremental manner, almost word by word. The circuit we illustrated in Figure 20 operates in such an incremental manner. Humans are quite often oblivious to the ambiguities that can be found in a sentence (e.g., Pinker 1994). Evidence comes from garden-path sentences (e.g., Meng and Bader 2000) such as *The man who hunts ducks out on weekends* (Pinker 1994). Typically, humans interpret *duck* as a noun instead of a verb, which results in a misrepresentation. Furthermore, recent research is beginning to show that human sentence comprehension can produce underspecified representations of a given sentence (Sanford & Sturt 2002), which is quite the opposite of analyzing all possible interpretations of a sentence.

Bod et al. also argue that connectionism should be about learning. This idea is wrong for several reasons. First, the notion of connectionism derives from Cajal, who described the basic structure of the brain in terms of neurons as the functional units, which are related with connections. Nothing in this description associates connectionism exclusively with learning. Second, even in the Parallel Distributed Processing (PDP) tradition one can find influential models that are not about learning. An example is the interactive activation model of letter and word recognition by McClelland & Rumelhart (1981). Third, it is a hazardous idea. There is no guarantee that connectionism, based on an initially unstructured architecture and learning principles, will eventually result in a cognitive system comparable to that of humans (or even primates). By far most species that developed a neural system in the course of evolution did not develop a neural system on the level of that of humans or primates. The lesson from this is that simple architectures with simple adaptive processes do not in general result in human-like cognition. In this respect, we fully accept the possibility that an architecture like ours will not develop on the basis of simple learning principles and an initially unstructured architecture. But that does not mean that it is incorrect. Fourth, it does not agree with brain development. A good example is given by the orientation maps in the primary visual cortex. The basis structure of these maps is available at birth, including individual cells that respond to specific line orientations (e.g., Chapman et al.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

1996). So here one can already find an example that contradicts the notion that connectionism should be only about learning.

Other remarks of **Bod et al.** can be easily countered as well. For example, they argue that the architecture cannot detect the similarity between *The man frightens the child* and *The child fears the man*. We were surprised to read this statement, because we discussed this issue in section 6.6, where we examined how the architecture can relate *John gives Mary a book* to *Mary owns a book*. We did this in terms of the long-term memory structures of these sentences, but the same mechanism works for structures in the blackboard, because the relation between these sentences depends on the associations between the word assemblies for *give* and *own*. These assemblies are the same for structures in the blackboard and for structures in long-term memory. We discussed this issue in terms of a single example because we wanted to present as many different examples as possible, trusting the reader to understand that if the architecture can instantiate the relation between *gives(John, Mary, book)* and *own(Mary, book)*, the architecture can also instantiate the relation between *frighten(man, child)* and *fear(child, man)* or any other example of this kind. Similar remarks can be made for issues like agreement and long-distance binding. Examples of these are found in the article as well.

R3. Related work

Baars argues that one can find many other examples of blackboard architectures in cognition. We agree with that observation. For example, Jackendoff (2002) describes symbolic blackboard architectures for phonological structures, sentence structures, and semantic structures. Each of these blackboards has its own specific structure, dependent on the nature of the combinatorial structures they are involved in. In fact, we took the importance of blackboard architectures for cognition as granted and concentrated on the issue of their neural instantiation.

We took language as our prime target because of the obvious combinatorial structure of language. But we decided to include vision as well, because combinatorial structures are also found in vision, and because they seem to be different from language. In this way, we could investigate how neural blackboard architectures can be described for different cognitive domains. Furthermore, we could investigate what makes them different, what makes them similar, and how they can be related (e.g., how they can provide grounding of language in the visual domain). The differences between the neural architectures we discussed follow from the differences between their cognitive domains (simply stated, sequential for language and spatial for vision). Yet there are also similarities. In particular, they both support the process view of binding, that is, binding within the frame of reference of the system itself. Furthermore, they are related in terms of the grounding of language in perception, which could be the basis for a further development of semantics.

Similar questions can be discussed in case of the blackboard architectures mentioned by **Baars**. A neural instantiation of these architectures would have to be developed, with an emphasis, in our view, on the ability to answer "binding questions" in these architectures without relying on conjunctive forms of representation. These binding questions can be explicitly stated, but they could also be implicit. In fact, it would be interesting to explore whether conscious cognition and voluntary control are related to the process of answering self-generated binding questions. For example, are we conscious of the color of an object when (or because?) we (implicitly) answer the question: "What is the color of this object?"

Clancey describes an interesting relation between our blackboard architecture for language and his Conceptual Coordination (CC). The relation between our blackboard architecture and architectures such as CC indeed indicates that the integration between neural and symbolic analyses of cognition is progressing.

As **Clancey** notes, comprehension is a sequential process and not some kind of state that is held or "moved" in memory. This is what we have tried to illustrate in terms of the process of answering binding questions, both in the sentence blackboard and in the visual blackboard. It contrasts with the notion, as expressed by **Doumas, Holyoak & Hummel** (**Doumas et al.**), that "downstream" neurons should have an explicit representation of the binding structure instantiated with "upstream" neurons.

Clancey discusses some refinements, related to binding problems involved in sentence comprehension, that could be introduced into the architecture. The examples given are interesting and need to be investigated in more detail. For the moment, we note that some of these refinements are already implemented in the architecture. They concern the structure assemblies that are simultaneously active in memory. Their number is indeed related to memory span. Clancey suggests that the memory span is not given by the multiple activations that constitute the structure assemblies themselves. Instead, the memory span depends on the categorizations given by the different delay assemblies that interconnect the structure assemblies. Indeed, this is what happens in the blackboard. The activation of the structure assemblies guides the binding process, but once binding has been achieved, the memory is given by the active delay assemblies. The idea that only one delay assembly can be active at a time at a specific binding site is reflected in the connection structure illustrated in Figure 5. The competition within the "rows" of columns and within the "columns" of columns also ensures that a delay assembly can only be active in one way at a time.

R4. Structure of the architecture

Hadley argues that we have presented a conjunctive form of binding, which would require in the order of 60 million distinct binding circuits to succeed. If this were true, the architecture we presented would hardly be worth a discussion. But Hadley's point and his calculation are based on a misinterpretation of our architecture. It turns out that Hadley confuses words (e.g., nouns and verbs) with structure assemblies (e.g., NP and VP assemblies). With 5,000 nouns and 3,000 verbs, Hadley calculates that the "agent matrix" (Fig. 5) would consist of 15 million memory circuits. However, the agent matrix does not represent binding between nouns and verbs but between NP

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

assemblies and VP assemblies. The binding of nouns and verbs is indirect in the architecture. First nouns bind with NP assemblies, and verbs bind with VP assemblies. Then the NP and VP assemblies bind with each other. This intermediary layer of binding makes all the difference. This is what makes the architecture a blackboard architecture. NP and VP assemblies are needed only for the temporal and online representation and processing of a sentence structure.

We noted in the target article that in the order of 100 NP assemblies and 100 VP assemblies would suffice for this purpose. In that case, one could have in the order of 100 nouns and 100 verbs involved in online representation and processing, which is more than sufficient to account for working-memory capacity in language comprehension (e.g., Just & Carpenter 1992), that is, before transfer to the hippocampus complex (HC). With 100 NP and 100 VP assemblies, the agent matrix consists of 10,000 memory circuits, a far cry from the 15 million memory circuits calculated by **Hadley**. But, of course, we also have to count the bindings between words and structure assemblies. Using Hadley's example, we would have 500,000 memory circuits to bind the nouns with the NP assemblies and 300,000 memory circuits to bind the verbs with the VP assemblies, that is, of the order of 10^6 bindings between words and structure assemblies. Again, this is significantly fewer than the number calculated by Hadley, but it is a substantial number. **Choe, Durstewitz**, and **Grüning & Treves** also raised doubts about this number and about the fact that novel words could hardly bind with structure assemblies in this way.

In section 6.9 we noted that in a future development, the sentence blackboard architecture should be combined with a phonological blackboard architecture (as in Jackendoff 2002). We will illustrate here how such a phonological blackboard architecture will significantly reduce the number of bindings between words and structure assemblies and how it can account for the binding of novel words. Figure R1 illustrates a quasi–"phonological blackboard" that connects with the sentence blackboard (e.g., to the NP and VP assemblies). The structures in the "phonological blackboard" are not intended as genuine phonological structures, but they do illustrate the point that a word is also a combinatorial structure, based on phonological constituents. These constituents will bind with W_x assemblies similar to the way in which words (structure assemblies) bind with the S_i assemblies in the sentence blackboard. The binding circuits will be the same as in the sentence blackboard. The word assemblies, representing the long-term information related to words, are located outside the phonological blackboard but they are connected to it in an interactive way. The word assemblies can be as distributed as they have to be, but each word assembly will be connected to specific phonological constituents in the phonological blackboard. So the word assembly for *cat* is connected to the quasi-phonological constituents /c/, /a/, and /t/ of the word *cat*.

So when a word is presented, it will activate its word assembly and its phonological constituents in the phonological blackboard. The phonological constituents, in turn, will bind with an arbitrary (but "free") W_x assembly. Like the structure assemblies, the W_x assemblies are involved only in online representation and processing. So one can assume that the number of W_x assemblies will

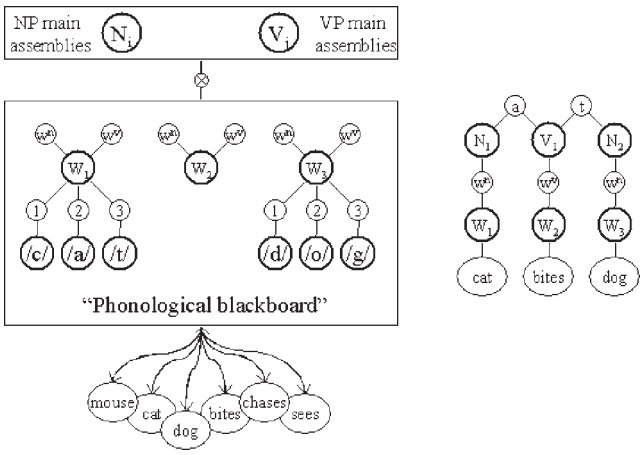


Figure R1. Illustration of how a "phonological blackboard" would bind words to structure assemblies in the sentence blackboard.

be of the order of 100 as well. Because the W_x assemblies are arbitrary, one needs a connection matrix to bind W_x assemblies with structure (e.g., NP and VP) assemblies. In turn, this means that the NP and VP assemblies have to have subassemblies for this binding, as illustrated in Figure R1 (which makes all bindings of structure assemblies of the same kind, i.e., via subassemblies). The W_x assemblies have to have subassemblies as well. We assume that they have specific subassemblies that allow selective binding with NP assemblies (with w^n subassemblies) and VP assemblies (with w^v subassemblies). For each of these, one needs a connection matrix of the order of 10,000 memory circuits (like the agent matrix in fig. 5). So the 10^6 bindings between words and structure assemblies reduce to something of the order of 10^4 bindings between the phonological blackboard and the sentence blackboard. Hence all bindings in the architecture (i.e., between subassemblies) are of the order of 10^4. The number of different subassemblies, reflecting different categories (e.g., "agent," "theme," "clause," "preposition") will be between 10 and 100. Therefore the overall number of bindings (memory circuits) in the architecture will be between 10^5 and 10^6. This number is certainly more feasible than the 60 million calculated by **Hadley**.

The use of a phonological blackboard also explains how novel words can bind with the sentence structure. A novel word is a new combination of familiar phonological constituents, which can bind with a W_x assembly. The phonological nature of the new word could be used to decide if the word is a noun or verb, so that it can bind directly in the available sentence structure.

A similar confusion between words (e.g., nouns, verbs) and structure assemblies (e.g., NP and VP) seems to be the basis of **Dyer's** comment. The connection matrix in Figure 5 refers to binding between the NP and VP structure assemblies, not to the binding between words. As we argue above (sect. R2), some kind of connection path between any pair of words (i.e., their neural instantiation) is necessary to account for the fact that these words can be related to one another. As we calculate here (sect. R4), the binding structure in the blackboard is a very parsimonious way to provide for all possible connection paths that language would require.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

The connection matrix between structure assemblies also accounts for the binding process itself. Binding occurs between active structure assemblies (i.e., their corresponding subassemblies). Usually only one NP subassembly and one VP subassembly are simultaneously active. So binding in the connection matrix will be immediate, and no winner-take-all process is needed. In fact, the conflicts in binding that do sometimes occur are a major source of complexity in this architecture, as illustrated in Figure 18 and Figure R2 (sect. R9.1).

The connection matrix also accounts for the way in which the binding process is initiated. When sentence information indicates that a noun should bind with a verb as its "agent," the gating circuits between the main assemblies and the subassemblies are opened. Indeed, this occurs for all NP and VP assemblies (but not for all nouns and verbs). But only one NP (main) assembly and only one VP (main) assembly are active at a given moment. So only one subassembly for NP assemblies and only one subassembly for VP assemblies will be activated, which will result in the binding process described in the previous paragraph.

Dyer raises an interesting point about whether the architecture would scale up for semantics. We used the labels "agent" and "theme" to describe the binding structure in the architecture. In passing, we noted that we took these labels as semantic macroroles (van Valin 2001, see note 6). We could also have used the notation of Pinker (1989), who refers to the argument slots of verbs in terms of *X*, *Y*, and Z. So *X chases Y*, *X bites Y*, and *X gives Y to* Z.

A substantial number of different semantic roles can be distinguished, but nevertheless verbs can have one, two, or three arguments, and these argument positions seem to have something in common. The connection matrices we described would be used for these argument positions, that is, for the *X*, *Y*, and Z arguments of verbs. The more precise classification of the semantic roles will depend on the (lexical) information associated with a particular verb. There is no need to instantiate all these different semantic roles directly within the architecture. Notice that there is only one instance of any given predicate in the architecture (e.g., *give* or *own*), that is, there is only one word assembly for *give* or *own*. Long-term associations can be formed between these predicates, that is, between their neural assemblies. In the same way, information can be associated with a word assembly of a verb to express the fact that, say, the *X* argument of that verb has a particular semantic role (e.g., agent, experiencer, cognizer, believer, etc.).

R4.1. Holistic representation

In their commentary, **Dresp & Barthaud** argue for holistic representations instead of combinatorial ones that can be implemented in a blackboard architecture. In particular, holistic representations would fit better with the need for generating behavioral success. We agree that if a problem can be handled with conjunctive or holistic forms of representation, it pays to do so. Processes that use these forms of representation typically operate very fast, and speed of processing can increase behavioral success. However, the success of this approach depends on the size of the problem space. When the problem space expands, the success of this approach declines rapidly.

Furthermore, we do not accept the argument that every process in human cognition must be understood in terms of immediate behavioral success. For example, human language provides the ability to go beyond the here and now that is typical for behavioral adaptation. The argument that one does not find these processes in animals is of little value. There is much in human cognition that cannot be compared directly with animal cognition. Human cognition has produced an environment that is substantially different from the environments that animals live in. Just think of all the aspects of cognition that are involved in creating and sustaining a research institute or university or in organizing a discussion like this one. One cannot explain that away by placing human cognition on the procrustean bed of animal cognition and immediate behavioral success. Something has to be different to account for the uniqueness of human cognition.

So the need for behavioral success is primarily related to the visual blackboard architecture we presented. We do believe that our blackboard architecture for vision is related to behavioral success and that the relation between the language blackboard and the vision blackboard we proposed (the "grounding" of language) is sound. **Dresp & Barthaud** argue that the adaptive resonance theory (ART) theory of Grossberg (1999) is an alternative theory that fits with their holistic view of cognition. Binding between "what" and "where" would correspond with resonant activation states in ART, formed by the activation patterns in the "what" and "where" processing steams. So binding could be achieved without any language-like combinatorial process.

However, the binding process in the visual architecture is combinatorial but not language-like. The reason we discussed the blackboards for language and vision is to show that one can have different blackboard architectures that are tailor-made for the modality in which they operate. Contrary to the claim of **Dresp & Barthaud**, they are not at all general-purpose machines. For example, the purpose of discussing language complexity was to show that the language architecture is selective to the kind of sentence structures it can encode, contrary what one finds in a general-purpose machine.

Furthermore, the idea that resonant activation states in ART bind activation patterns in the "what" and "where" processing steams is based on a misinterpretation of Grossberg (1999). Grossberg argues that ART concerns the process of learning and categorization in the "what" (ventral) stream and that an entirely different (in fact, complementary) model accounts for learning and processing in the "where" (dorsal) stream. The difference between these two complementary models is a crucial point in Grossberg (1999). In short, ART is related to conscious memories, and the "dorsal" model is related to procedural (unconscious) memory.

ART is about learning and categorization in the "what" stream, whereas our vision architecture is related to the interaction between the "what" and "where" stream. There are a number of animal experiments that are in agreement with the interaction in the vision blackboard illustrated in Figures 22 to 24. A detailed description can be found in van der Velde and de Kamps (2001). Furthermore, the blackboard architecture is in agreement with

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

aspects of visual agnosia. Farah (1990) describes a dissociation between dorsal simultanagnosia and ventral simultanagnosia. The first results from damage to the dorsal pathway. Object recognition is still possible with these patients, but they cannot integrate that information with spatial information. Ventral simultanagnosia results from a damage to the ventral pathway and it consists of a loss of identity information about objects. The double dissociation between these two forms of agnosia (the first can occur without the second, and vice versa) is a classical argument against a holistic form of representation. Damaged holistic representations are always damaged holistically. On the basis of these and other forms of visual agnosia, Farah (1990, p. 151) presents a global model that is in agreement with the visual architecture we presented.

The visual architecture we presented here is, in fact, motivated by speed of processing. Initial processing in the visual cortex proceeds with feed-forward networks (e.g., Oram & Perrett 1994), because processing in feed-forward networks is very fast. But feed-forward networks are prone to catastrophic interference (e.g., McCloskey & Cohen 1989). This occurs when the same feed-forward network is used to learn different tasks. The best way to avoid catastrophic interference with feed-forward networks is to use different feed-forward networks for different tasks. This solution seems to be the basis for the structure of the visual cortex, which consists of different pathways that initially operate in a feed-forward manner. But with different feed-forward networks the integration of the information they process becomes a problem. The solution to that problem is the basis for the visual blackboard we have presented here. So our visual blackboard architecture directly derives from issues related to speed of processing. Therefore the architecture provides the ability for fast behavioral adaptation to changes occurring in the environment. Moreover, these fast-occurring changes are often related to new compositions of familiar entities (trees in a forest, houses and roads in a city). Handling novel compositions of familiar entities is the prime target of the visual blackboard architecture. It can do it straightaway. No new learning, as in forming new holistic representations for these new combinations, is required. So even in this respect, the architecture is faster than models based on holistic representation.

The grounding of language in the visual blackboard concerns the interaction between these two different types of blackboards. The fact that they interact is obvious. Visual information can be transformed into linguistic information, and linguistic information can be used to guide the visual system. The latter is obvious, given the ability to direct visual attention by means of verbal instructions. This is not a form of priming, as **Dresp & Barthaud** mistakenly assume. Instead, it concerns the selection of visual information on the basis of linguistic information. The interaction between the vision and language blackboards derives from the fact that word assemblies are not copied or moved around, as in a symbolic architecture, but always remain *"in situ."* In this way, associations can develop between the word assemblies and other representations in the brain, such as representations in the visual cortex.

Powers also seems to argue that the cognitive abilities provided by language are not very different from those provided by vision. However, this discussion itself illustrates the specific importance of language in cognition. We can have a discussion on a topic such as the importance of language without any direct (visual) contact. Visual perception is always bound to the here and now, but language can make us go beyond the here and now. This is what makes language unique: People can exchange information without direct visual contact, without living in the same place, or even without living in the same time. In this way, knowledge can accumulate in a manner that is simply not possible if knowledge depended exclusively on direct visual contact with the world. Language can inform you about events that occurred thousands of years ago or thousands of miles away. It can do so precisely because it can capture the actor-patient situation of a given event in the structure of a sentence. Readers can then understand the actor-patient situation of that event because they can understand the "Who does what to whom" relation expressed in the sentence. In other words, because they can answer (implicit or explicit) binding questions in the manner we analyzed in our target article.

R4.2. Architecture versus association

In their commentary, **Doumas et al.** argue that von der Malsburg (1986) shows that connections between assemblies cannot carry information about binding relations. This analysis of processing in cell assemblies does not relate to the architecture we have presented. Cell assemblies are associative, but our architecture is not. Due to the effect of the gating circuits in the architecture, the flow of activation can be controlled in a manner that is not possible in an associative structure like a cell assembly. We illustrated this difference in Figures 6 and 15.

Furthermore, **Doumas et al.** note that the architecture can answer a question like "Who is the cat chasing?" in the case of the proposition *chase*(*cat, mouse*), but that downstream neurons "see" only the neural activation of the assemblies involved, but not their binding relations. But why would these binding relations have to be represented explicitly by the downstream neurons, that is, why would they have to be burdened with representing that information themselves? There is a limit on the information that can be stored at a given site. So when the need to store more information arises, for example, when cognitive capabilities increase, information has to be stored in different sites. This could be the reason why the cortex of humans is much larger than the cortex of apes and monkeys (Calvin 1995). Distribution of information over different sites is not a problem as long as relevant information can be retrieved from distant sites. So if the downstream neurons of Doumas et al. want to "know" whether *chase*(*cat, mouse*) or *chase*(*mouse*, *cat*) is stored by the neurons upstream, they can initiate a sequence of questions like "Who is chasing?" and "Who is being chased?" If the sequence of answers is *cat*, *mouse* (instead of *mouse*, *cat*), the downstream neurons "know" that *chase*(*cat, mouse*) is stored by the neurons upstream, instead of *chase*(*mouse*, *cat*). It is necessary that downstream and upstream neurons can interact with each other. It is not necessary for each of them to do the task of the other. It would, in fact, be highly detrimental for the system as a whole if such a duplication of tasks were always needed.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

For example, in the visual cortex, the neurons in the "What" stream do not have to know where the object is and the neurons in the "Where" stream do not have to know what the object is. In this way, each stream can restrict learning to the specific task it has to perform. Learning would be much more difficult if each stream had to learn all tasks simultaneously.

Yet, because the streams influence each other, the system as a whole knows where the object is and what the object is, as the interaction process in our visual architecture illustrates.

R5. Binding with synchrony

A number of commentators—**Dyer, Doumas et al., Durstewitz, Müller, Shastri,** and **Powers**—have argued against our dismissal of synchrony of activation as a binding mechanism in combinatorial structures. It is important to understand the topic of this debate. The topic is not so much whether one will find cross-talk in hierarchical representations. We are willing to grant that this can be avoided to a certain point (see van der Velde & de Kamps 2002). The crucial issue concerns the "read-out" (i.e., answering binding questions) in the case of true forms of combinatorial structures, that is, structures for which no conjunctive (coincidence) representations exist. The commentaries that advocate synchrony as a binding mechanism do not really address this issue. Two questions need to be answered on synchrony as a binding mechanism in combinatorial structures (e.g., as found in language). How can binding questions be answered without relying on conjunction (coincidence) detectors needed to detect synchronous activation? And if this is not possible, how can one have conjunction (coincidence) detectors for all possible binding relations that could exist in language? We are not aware of a model that has answered these questions.

R5.1. SHRUTI

In his commentary, **Shastri** provides details of the reasoning process in SHRUTI. These details are much appreciated, but they do not touch upon the topic of the discussion we have with this model. The topic is this: Human cognition, in particular on higher levels, is characterized by productivity, compositionality, and systematicity. These characteristics are related in the sense that one goes with the others (Fodor & Pylyshyn 1988). For example, the productivity of language requires linguistic structures to be compositional. If the "performance set" of natural language (i.e., the set of sentences that human language users can comprehend) is at least of the order of 10^{20}, it is difficult to see how one could have specific conjunctive forms of representation for each of these sentences. For example, the time to develop or learn these representations seems not available. So neural models of these forms of human cognition have to have the ability to represent and process cognitive structures in a productive, compositional, and systematic manner. This, in short, is the topic of our argument with models like SHRUTI.

SHRUTI is a model about reasoning in long-term memory. For instance, it can relate the proposition *John gives Mary a book*, stored in long-term memory, to the proposition *Mary owns a book*. However, the systematicity of reasoning entails that the relation between *give(X, Y, Z)* and *own(Y, Z)* does not hold for just the specific tokens of this relation stored in long-term memory but holds for all possible tokens of this relation. Viewed in this way, it does not make much sense to have a model that deals only with reasoning in long-term memory. It should be a model that can handle all tokens of the same kind, regardless of whether these tokens are stored in long-term memory or are represented on the fly (i.e., novel tokens of the same relation).

With this in mind we can have a closer look at SHRUTI. Figure 12 in Shastri & Ajjanagadde (1993b) illustrates how SHRUTI can relate the proposition *John gives Mary a book* to the proposition *Mary owns a book*. This figure shows that the proposition *John gives Mary a book* is represented with a specific "fact node" (or, indeed, a "fact circuit"). On its own, it is acceptable to represent a proposition in long-term memory with a designated form of representation like a fact node or fact circuit. Long-term memory consists only of those propositions that we have actually memorized, so the problem of productivity that one finds with the performance set of language does not occur here. However, figure 12 in Shastri & Ajjanagadde (1993b) shows that the fact node for *John gives Mary a book* plays a crucial role in the inference process that establishes the relation between *John gives Mary a book* and the proposition *Mary owns a book*. For example, activation of the fact node is needed to activate the "collector node" of the predicate *give*. In turn, the activation of a collector node is needed to answer the queries in SHRUTI. So, if *John gives Mary a book* is stored in long-term memory, the collector node of *give* has to be activated to answer the query "Does Mary own a book?" In turn, the collector node of *give* is activated by the fact node for *John gives Mary a book*, which shows that fact nodes play a crucial role in the reasoning process itself instead of just representing a specific proposition.

The role of fact nodes in the reasoning process in SHRUTI raises the issue of the true systematicity of this model. Consider a novel proposition *John gives Dumbledore a broom* and the query "Does Dumbledore own a broom?" **Shastri** asserts that SHRUTI can handle a novel event when it is told that "Dumbledore is an instance of an existing type." We do not quite understand what this means and how it works, but we can argue that reasoning on the basis of type information in the case of specific propositions is a hazardous affair. *Dumbledore* is of the type *person* or *noun*, so it would seem that this solution entails that one can answer the query "Does Dumbledore own a book?" on the basis of the representation *John gives noun/person a book.* However, it is quite possible that *Dumbledore owns a book* is not true, even though *John gives noun/person a book* is true, for example, because *John gives Mary a book*.

The reasoning process we discussed in section 6.6 does not depend on specific tokens of the kind *John gives Mary a book*. Instead, it depends only on the relations between the predicates *give* and *own* and between the argument slots of these predicates. We used a long-term memory encoding of *John gives Mary a book* to illustrate how combinatorial structures can be encoded in a more explicit form in long-term memory.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

But the reasoning process described in section 6.6 works in the same way for a novel proposition like *Dumbledore gives Harry a broom* that is temporarily stored in the blackboard. The only difference between a long-term memory encoding and a temporal encoding of a proposition is the nature of binding of the arguments. In the temporal encoding, binding results from the (temporal) activity in the memory circuits, whereas, in long-term memory, binding results from direct associations between arguments and predicate slots. This difference in binding does not affect the reasoning process, because that depends only on the associations between the predicates.

Furthermore, the reasoning process described in section 6.6 can answer specific binding questions like "Who owns the book?" instead of just yes/no queries like "Does Mary own the book?" in SHRUTI. This is a consequence of the fact that SHRUTI answers a query on the basis of the activation of a collector node, which is a yes or no event. In turn, the yes or no activation of a collector node reflects the yes or no activation of a fact node, needed to detect the respective synchrony between arguments and predicate slots.

The lack of systematicity in SHRUTI is also reflected in the problem of 2. **Shastri** acknowledges that the solution in SHRUTI for this problem, that is, the duplication (multiplication) of predicate nodes, results in a loss of systematicity. Instead of having one relation between the predicates *give* and *own*, SHRUTI has distinct relations between $give_1$ and own_1 and between $give_2$ and own_2. Shastri asserts that our blackboard architecture is faced with a similar problem, because different tokens of a proposition are represented with different structure assemblies. This is correct, but that has no consequence for learning the relations that produce systematic reasoning, because these relations depend only on the associations between predicates, for example, between the word assemblies for *give* and *own*, and predicates (word assemblies) are not duplicated in the blackboard architecture. The predicate *give* is always represented with one and the same word assembly in any proposition in which *give* occurs. Indeed, the very reason for the existence of structure assemblies next to word assemblies is that structure assemblies can be used to handle binding problems and the problem of 2, whereas the (single) word assemblies can be used to account for relationships between predicates and issues such as grounding, as illustrated in figure 27.

Finally, **Shastri** notes that we should have discussed the neural parsing model of Henderson (1994), based on SHRUTI, as an alternative to our own. There is a simple reason why we did not do this: There is no neural parsing model to be found in Henderson (1994). Henderson introduces a parser in which computational constraints derived from SHRUTI are used. But the parser itself operates in an entirely symbolic way. For example, Henderson (1994, p. 367) describes how syntactic nodes can be closed for further parsing: "Closed nodes can be removed from the phrase structure representation, thereby reducing the number of nodes [the parser] needs to store information about." Removing nodes from a structure is one of those things that one can do easily in a symbolic implementation, but it turns out to be very hard (if not impossible) in a genuine neural implementation. None of these issues about real neural implementation is discussed, let alone answered, in this parsing model. None of the problems that derive from these implementation issues, such as answering specific binding questions without using conjunctive representations, is discussed, let alone answered, in this parsing model.

R5.2. LISA

The fundamental problem with any model using synchrony of activation, as we stated in section 3.2, is the dependence on conjunctive forms of encoding to get things done. Dependence on conjunctive encoding destroys the productivity and systematicity of a model. We analyzed this in case of SHRUTI, but we can give the same analysis in the case of the model known as LISA, advocated by **Doumas et al.** For example, LISA encodes *Bill loves Mary* by having conjunctive (coincidence) nodes for the entire proposition, for *Bill-lover* and for *Mary-beloved* (see Hummel & Holyoak 2003, fig. 3). So the question is: How can LISA represent and operate on a novel proposition like *Dumbledore loves Harry*? How could one have a conjunctive node for this proposition if one has never seen or heard it before? By the same token, how could one have a conjunctive representation of *Dumbledore-lover* and *Harry-beloved*? Furthermore, it is possible to have the conjunctive representations for one relation but not for the reverse relation. So one could have conjunctive nodes for *Harry loves Dumbledore*, *Harry-lover*, and *Dumbledore-beloved*, but not for *Dumbledore loves Harry*, *Dumbledore-lover*, and *Harry-beloved*. Consequently, LISA could make inferences about *Harry loves Dumbledore* but not about *Dumbledore loves Harry*. This kind of behavior is characteristic of a system that lacks systematicity.

The inference process in LISA critically depends on its conjunctive form of representation. An example can be found in figure 4 in Hummel and Holyoak (1997). The figure illustrates a mapping from *John loves Mary* onto *Bill likes Susan* versus *Peter fears Beth* (i.e., the inference that *John loves Mary* is more related to *Bill likes Susan* than to *Peter fears Beth*). The process starts with the activation of the node for *John loves Mary*. This node in turn activates conjunction nodes for *John-lover* and *Mary-beloved*, which then activate a set of nodes for semantic primitives, which in turn activate conjunction nodes for conjunctions such as *Bill-liker* and *Bill likes Susan*. The need for synchrony is obvious: There are so many conjunctive nodes, both for entire propositions and for partial propositions, that without synchrony, a superposition catastrophe would easily occur. But this is not what systematic reasoning is about. The purpose of systematic reasoning is not to avoid a superposition catastrophe in a selected set of conjunctive representations but to establish relations between arbitrary tokens of familiar predicates.

This is where LISA fails. LISA cannot conclude that a novel proposition like *Dumbledore loves Harry* is more related to *Bill likes Susan* than to *Peter fears Beth*, because the conjunctive representation of *Dumbledore loves Harry* and its relations with representations for *Dumbledore-lover* and *Harry-beloved* are missing in LISA. Yet the resemblance between *Dumbledore loves Harry* and *Bill likes Susan* is just as great as the resemblance between *John loves Mary* and *Bill likes Susan*.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

LISA, however, is completely blind to the systematicity between these two tokens of the same resemblance. This is why synchrony of activation in a model like LISA fails to encode productive and systematic cognitive structures and relations.

R5.3. Role of synchrony

In his commentary, **Durstewitz** argues that synchronization could be mechanism for binding without the use of specific coincidence detectors, because our architecture also operates with assemblies and subassemblies that are synchronously active. However, the word "synchronous" is used with two different meanings here: simultaneous and true synchrony. The members of a band can play in the same period (simultaneous), even if they do not play in (true) synchrony. The first meaning of "synchronous" as used by Durstewitz refers to simultaneous activation in our architecture. The populations in the architecture are simultaneously active in a given time period in the sense that they generate enough activity within that period to influence each other (just as band members can produce sound in the same period). The second meaning of the word "synchronous" as used by Durstewitz refers to phase coherence. This requires that populations should not just be simultaneously active in a given period, but that they also play in the same rhythm, so to speak. It is not required that the activations of the populations in our architecture show any phase coherence. In contrast, phase coherence is essential in models that use synchrony as a binding mechanism, such as SHRUTI and LISA. If one were to take away the phase coherence in these models, the units/neurons in these models would still be simultaneously active. But the ability for binding is lost in these models, because that ability depends critically on phase coherence, which is detected by the coincidence detectors in these models.

It is important to note, however, that our rejection of synchrony as a binding mechanism in combinatorial structures does not entail a rejection of any role of synchronous activation in the brain. Synchrony of activation can play a role in processes that depend on conjunctive forms of representation. As we argue in section R4.1, if an organism can solve a problem with conjunctive representations, it should do so. As **Durstewitz** notes, time is a serious constraint in biology, and conjunctive forms of processing are much faster than combinatorial forms of processing. However, for this solution to succeed, the problem space at hand should not be too large. Otherwise, the conjunctive representations that the organism can learn in its lifetime will not sufficiently cover the problem space. In that case, there is a reasonable chance that the organism can be confronted with a problem for which it has not (yet) developed a conjunctive representation.

The basis for combinatorial structures in human cognition is the observation that problem spaces arise in human cognition that are too large to be covered in any reasonable way with conjunctive forms of representation. Here, the need for combinatorial processing arises. We used the example of the performance set of natural language, which consists of 10^{20} sentences or more. In any lifetime, one can form only a minute set of conjunctive representations for this set (of the order of one sentence for every 10^{10} sentences or more). In visual cognition, an example would be the number of different patterns that can be created on a chessboard.

But for problem spaces that are not too large, conjunctive representation would be the preferred choice. Many examples can be found in the visual cortex (e.g., the orientation columns in the primary visual cortex). In the case of conjunctive representation, synchrony can help to avoid the superposition catastrophe that can easily occur with conjunctive representations, as suggested by von der Malsburg (1987). So, historically, the use of synchrony arose out of a problem with conjunctive representations. Synchrony solved the superposition catastrophe with these representations by using coincidence detectors as conjunctive units. It is important to realize that this is a refinement of the use of conjunctive representation, not an introduction of combinatorial representation.

The empirical evidence for synchrony of activation is in line with this interpretation. In a characteristic experiment (e.g., Singer & Gray 1995), one or more light bars were moved across the receptive fields of two groups of neurons. A synchrony of activation was found between both groups of neurons when a single light bar (i.e., a coherent object) was moved across the receptive fields of both groups of neurons. But when two light bars were moved in opposite directions, each one across the receptive field of one group of neurons, synchrony of activation did not occur. As we argued in section 2.1, problems such as motion detection are most likely solved in conjunctive manner in the visual cortex. The same could be true for the detection of motion coherence of a moving object, as in the experiment by Singer & Gray. Given the occurrence of conjunctive coding in the visual cortex, it is not a surprise that most of the empirical evidence related to synchrony of activation has been observed in the visual cortex.

Besides a mechanism for binding in these forms of conjunctive processing, synchrony of activation as observed in the brain could also be a "dynamical fingerprint" of an interaction that occurs between brain areas. When brain areas exchange information in a functional manner in the context of the cognitive process they instantiate, they will also interact as coupled dynamical systems. Coupled dynamical systems have a tendency to synchronize their behavior because of the (oscillating) force they exert on each other. In the same way, brain areas that interact in a cognitive process could have a tendency to synchronize their dynamics as well. This form of synchronization does not have to play a functional role in the cognitive process, as in answering binding questions. Instead, it could just be a dynamical by-product of the interaction between brain areas. But it provides important information about which areas interact (communicate) in a given cognitive process. That is, it serves as a "dynamical fingerprint" of an interaction between brain areas. The fact that these synchronization effects are sometimes detected only with sophisticated mathematical techniques, of the kind that intelligence services use to eavesdrop through windows or thin walls, corroborates this view. It is difficult to see how such a faint effect could play a functional role in a cognitive process. But the effect does not have to be strong to be a dynamical fingerprint of an interaction between brain areas.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

R6. Recurrent networks and language

Dyer, Durstewitz, and **Müller** argue that we have discarded recurrent neural networks (RNNs) of the kind used by Elman (1991) too easily as potential neural models of language processing. For example, Durstewitz notes that the fact that RNNs could not (in our simulations) learn generative properties of language does not entail that they could not represent these properties. In a similar vein, Müller notes that RNNs with more than one hidden layer (e.g., 100) could represent much more than the RNNs we have tested. Indeed, it would not be justified for us to criticize RNNs on their learning (in)abilities, considering the fact that we have not yet shown how learning could take place in our architecture.

However, that is not the basis of our argument. Instead, we do indeed seriously question the abilities of RNNs (Elman-style) to represent the generative capabilities of language, whether they have one or 100 hidden layers. To understand why we make this claim, it is important to realize that these models arose out of an attempt to model language performance without using the distinction between rules and a lexicon. Linguistics has (typically) assumed that the distinction between rules and a lexicon is necessary to account for the productivity of language. For example, with the rules one could create a sentence (syntactic) "skeleton," which in combination with the lexicon could generate a huge number of sentences. In contrast with this approach, RNNs operate directly on word strings and they learn the words and the sentence structures at the same time. In other words, the aim is to represent the generative capabilities of natural language by deriving representations of all possible sentences on the basis of a set of sentences actually learned by the network.

The problem with this approach does not reside in the number of hidden layers or in the specific learning algorithm used. The problem resides in the contrast between the number of sentences that can be learned by the network (in any reasonable amount of time) and the number of sentences that can be formed on the basis of the combinatorial productivity of language. This performance set is of the order of 10^{20} or more. In contrast, even if a child learns a sentence every second for the first 20 years of life, the set of learned sentences is only in the order of 10^9. This is just one sentence for every 10^{11} sentences in the performance set. In other words, between the sentences learned by the network there are gaps of 10^{11} sentences (or more) about which the network has had no direct information.

The significance of these gaps can be appreciated by looking at the mathematical background of these RNNs (van der Velde 2001). In performance and learning, the RNNs operate as feed-forward networks (the feedback connections are frozen in these moments). In fact, the feedback connections determine only the input given to the network in performance or learning (which consists of a word in the sentence and the activation state in the hidden layer produced with the previous input). The reason it is possible to process a sequence of words with RNNs results from the input presentation: The activation of the hidden layer with input n is part of the input $n + 1$. In this way (and only in this way) does the previous processing in the network influence the current processing.

In mathematical terms, a feed-forward network implements a function between an input vector space and an output vector space. Hornik et al. (1989) show that feed-forward networks are "universal approximators" for a wide class of functions. This means that for any function in this class, one can find a feed-forward network that approximates that function on a subset of the input space within any desired level of accuracy. The proof by Hornik et al. is based on the Stone-Weierstrass theorem, which is itself a generalization of a theorem originally discovered by Weierstrass (see, e.g., Rudin 1976). The Weierstrass theorem states that a continuous function f(x) can be approximated on a finite interval of its input domain by a sequence of polynomials. This theorem is the mathematical foundation of important approximation methods such as Fourier series and Legendre polynomials (see, e.g., Byron & Fuller 1992).

The limitation of the approximation to a finite interval is the basis for the fact that feed-forward networks can approximate functions only on limited subsets of the domain. Furthermore, the ability to approximate a function derives from the use of a set of input values and their corresponding function values to determine the coefficients of the polynomial or the weights of the feed-forward network (e.g., by using the back-propagation algorithm). For this to succeed, the Weierstrass theorem demands that the function is known sufficiently ("globally") over the entire interval on which the function is approximated. That is, the gaps between the input values used to set the coefficients or weights cannot be too large (Byron & Fuller 1992).

As stated, RNNs are feed-forward networks during learning, that is, when the weights of the network are set to determine the approximation of the function that the network computes. Thus, as demanded by the Weierstrass theorem, the gaps between the input values used to set the coefficients or weights cannot be too large. This restriction is clearly violated in the case of language, because the set of sentences that can be learned in a lifetime is minute compared to the performance set of language. As we illustrate above, gaps of 10^{11} sentences or more inevitably occur.

Furthermore, it should also be mentioned that the tasks that RNNs perform are only remotely related to language performance. All they can do is predict the lexical category of a word that can follow after a given input string. They cannot, however, answer specific binding questions ("Who does what to whom") for the sentences they process. As Jackendoff (2002, p 163) noted about language performance with RNNs: "virtually everything of linguistic relevance has been bled out of the task the network is designed to perform."

The reference by **Durstewitz** to Hanson & Negishi (2002) is not relevant here. First of all, this paper is about learning finite-state languages with neural networks. It is well known that neural networks can handle languages of this kind. Furthermore, the language is an artificial language, that is, the set of words in these languages is very small. The crucial problem for natural language is in dealing with languages that are more complex than a finite-state language and that have a very large lexicon.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

R7. Reduced vector coding

Gayler and **Sommer & Kanerva** argue for an alternative solution of combinatorial structures in terms of reduced representations in high-dimensional vector spaces. In the target article, we briefly referred to problems associated with tensor networks as models for combinatorial structures, in particular the problem that tensor networks increase in size with levels of binding. Both Gayler and Sommer & Kanerva acknowledge this problem, but they note that the problem can be solved with reduced forms of representation, as given by vector symbolic architectures (VSAs). In a VSA, the binding between two (or more) vectors does not result in an increase in size. Instead, the constituent vectors and the combinatorial vectors have the same dimensionality. An example of a VSA is the Holographic Reduced Representation (HRR) proposed by Plate (1997; 2003). So with VSAs, the problem of tensor networks is avoided, and one would have distributed vector processing as the basis of neurocognition.

It is true that VSAs such as HRR solve the dimensionality problem of tensor networks. However, the solution itself prevents their use in truly combinatorial (novel) structures. We will use HRR to illustrate why this is the case. Consider the proposition *John eats fish*. A HRR representation of this proposition proceeds as follows (e.g., Plate 1997; 2003). First, *John*, *eats*, and *fish* are represented as *n*-dimensional vectors. The exact value of *n* is not crucial, but it is crucial that all vectors have the same dimensionality. So the *n*-dimensional vectors John, eat, and fish represent the words *John*, *eat*, and *fish*. Furthermore, the argument roles of *eat* are also represented with *n*-dimensional vectors, giving $\text{eat}_{\text{agent}}$ and $\text{eat}_{\text{theme}}$. The binding of John with $\text{eat}_{\text{agent}}$ is achieved with an intricate mathematical operation called circular convolution. Fortunately, we do not have to consider the details of this operation. It suffices to know that it produces an *n*-dimensional vector $\text{eat}_{\text{agent}} \otimes \text{John}$ that represents the binding in *John eats* ($\otimes$ represents the binding operation). The fact that $\text{eat}_{\text{agent}} \otimes \text{John}$ has the same dimensionality as John, eat and $\text{eat}_{\text{agent}}$ shows that the size problem of binding in tensor networks is solved. In the same way, $\text{eat}_{\text{theme}} \otimes \text{fish}$ represents the binding in *eats fish*.

The entire proposition *John eats fish* is represented with the vector $P_1 = [\text{eat} + \text{eat}_{\text{agent}} \otimes \text{John} + \text{eat}_{\text{theme}} \otimes \text{fish}]$. P_1 is just an arbitrary label for this vector. It is also an *n*-dimensional vector. The symbol + denotes the operation of superposition, which basically consists of adding the component values of the vectors involved. The brackets [ ] denote that P_1 is normalized, that is, its magnitude (Euclidean length) is equal to 1. The proposition *fish eats John* can also be represented in this way, with $P_2 = \text{eat} + \text{eat}_{\text{agent}} \otimes \text{fish} + \text{eat}_{\text{theme}} \otimes \text{John}$. The vectors P_1 and P_2 are different, so the role reversal of *John* and *fish* is adequately represented with HRR. Because a proposition is itself an *n*-dimensional vector, it can be used as an argument in another proposition. For example, consider *Susan knows John eats fish*. Here, P_1 is the "theme" argument of know. The entire vector for this proposition is $P_3 = [\text{know} + \text{know}_{\text{agent}} \otimes \text{Susan} + \text{know}_{\text{theme}} \otimes P_1]$.

Because vectors are normalized, some loss of information occurs. However, it is remarkable to see how much information can be encoded in this kind of representation (see, e.g., Plate 2003). Information retrieval with HRR proceeds by decoding bindings. For example, *John* as the argument of *John eats* can be found by convolving the vector $\text{eat}_{\text{agent}} \otimes \text{John}$ with the "approximate inverse" $\text{eat}^*_{\text{agent}}$ of $\text{eat}_{\text{agent}}$. The result $\text{eat}^*_{\text{agent}} \otimes \text{eat}_{\text{agent}} \otimes \text{John}$ is a noisy version of John. This noisy version of John has to be "cleaned up" by finding its closest match in a cleanup memory. This process is a form of pattern recognition that produces John as its outcome when the input is $\text{eat}^*_{\text{agent}} \otimes \text{eat}_{\text{agent}} \otimes \text{John}$. The cleanup memory must contain all patterns of both objects and structures that can result from decoding a HHR in the system.

These examples illustrate that HRR (and VSAs) can encode combinatorial structures. However, encoding is just one part of the story. It is also important that information can be retrieved in a combinatorial manner, that is, without relying on conjunctive forms of representation. Moreover, the system itself has to be able to retrieve that information. With this in mind, we can take a closer look at the retrieval process in HRR.

Consider the propositions *John eats fish* and *Susan knows John eats fish*. Suppose we ask the binding question "What does John eat?" to an HRR system. When the proposition *John eats fish* is stored in its memory, the system could answer the question by first decomposing the vector $P_1 = [\text{eat} + \text{eat}_{\text{agent}} \otimes \text{John} + \text{eat}_{\text{theme}} \otimes \text{fish}]$ into eat, $\text{eat}_{\text{agent}} \otimes \text{John}$, and $\text{eat}_{\text{theme}} \otimes \text{fish}$. Then it would have to produce $\text{eat}^*_{\text{theme}} \otimes \text{eat}_{\text{theme}} \otimes \text{fish}$, and present this vector to the cleanup memory to produce fish as the answer. Or the system could calculate $\text{eat}^*_{\text{theme}} \otimes P_1$ directly and present the result to the cleanup memory (Plate 2003). However, when the proposition *Susan knows John eats fish* is stored in its memory, the system would first have to decompose the vector $P_3 = [\text{know} + \text{know}_{\text{agent}} \otimes \text{Susan} + \text{know}_{\text{theme}} \otimes P_1]$ into know, $\text{know}_{\text{agent}} \otimes \text{Susan}$, and $\text{know}_{\text{theme}} \otimes P_1$. Then it would have to compute $\text{know}^*_{\text{theme}} \otimes \text{know}_{\text{theme}} \otimes P_1$, retrieve P_1 from the cleanup memory, and execute the entire sequence of operations described for the proposition *John eats fish*. Or the system could calculate $\text{eat}^*_{\text{theme}} \otimes \text{know}_{\text{theme}} \otimes P_3$ directly and present the result to the cleanup memory (Plate 2003).

So, to answer the question "What does John eat?" different operations have to be carried out by the system, depending on the vector (e.g., P_1 or P_3) stored in its memory. To make a selection of the operations to be carried out, the system has to know which of these two vectors is stored in its memory. However, both P_1 and P_3 are *n*-dimensional numerical vectors. On the surface they look alike. How is the system to know which of these is stored in its memory and so which operations it has to carry out to produce the answer to the binding question? For example, assume that *John eats fish* and *Susan knows John eats fish* are both novel propositions (of familiar constituents) never seen before by the system. In that case, P_1 and P_3 are both novel *n*-dimensional numerical vectors never seen before by the system. So on what kind of information can the system decide that, for example, P_3 is stored in its memory, so that it has to execute the operations needed for P_3? To argue that there will be some kind of representation of the proposition in the system that could guide the decoding processing is, of course, begging the question. The *n*-dimensional vector is the representation of the proposition in the system, and nothing else.

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

This problem illustrates an important aspect of combinatorial or compositional structures. The constituents of a combinatorial structure have to be recognizable within the structure itself. This is where reduced vector coding fails. As a result of the reduced representation, the constituents are no longer recognizable within the combinatorial structure. Instead, they are encapsulated within the combinatorial structure. Because the *n*-dimensional vector of a familiar constituent is not recognizable within the *n*-dimensional vector of a novel combinatorial structure, it cannot be used to guide the process of answering binding questions (as it does in the architecture we present here, e.g., see Fig. 7a).

Sommer & Kanerva argue that serious progress in cognitive modeling will be based on the understanding of the mathematical properties of high-dimensional representation spaces rather than on a specific solution to a relatively narrow set of challenge problems, that is, the ones we discuss here. Of course, more knowledge of mathematics is always useful, but cognitive modeling is not just an exercise in mathematics. The aim is to capture cognitive and brain processing in a cognitive model in such a manner that "boundary conditions" related with cognition and the brain are incorporated in the model. One such boundary condition is the ability to answer binding questions related to novel combinatorial structures. The benefit of finding specific solutions to a relatively narrow set of challenge problems is that these boundary conditions are incorporated from the beginning.

R8. Neural basis

Choe addresses the issue of the neural instantiation of our blackboard architecture for sentence structure. It is interesting to see that a gating circuit can be identified in a corticothalamic loop. However, like Choe, we do not believe that the thalamus would be the site of the neural blackboard proposed here. In our view, the neural blackboard is most likely to be found in the cortex. This is one of the reasons we included a discussion of the visual blackboard—to illustrate that processes in the blackboard (e.g., gating, memory by delay activity) can be found in the cortex. Choe describes a gating circuit, based on disinhibition, in the corticothalamic circuit. Gonchar & Burkhalter (1999) describe a disinhibition circuit in the visual cortex of the rat. In terms of evolution, the visual cortex is one of the oldest parts of the cortex. So if disinhibition circuits are found in the visual cortex and in the cortex of animals like rats, it could be that they belong to the "building blocks" that determine the structure of the cortex (also given the structural similarity found within the cortex). Delay activity is know to be found in the cortex as well (e.g., Fuster 1995), so the building blocks for the gating circuits in our architecture could exist within the cortex.

Furthermore, the binding structure we illustrated in Figure 5 should be seen as a structure of cortical columns. Each column could be a microcolumn or perhaps hypercolumn in the cortex. They would all consist of the same kind of circuit, that is, a gating circuit based on disinhibition, and the same kind of delay assembly. Regular column structures can be found in the cortex. An example is the "ice-cube" model of the primary visual cortex (e.g., Coren, Ward & Enns 1999) that consists of a very regular pattern of columns each dedicated to a specific task, such as orientation detection of edges in a particular direction. In fact, each of these columns is a local representation of an elementary shape on a given retinal coordinate. So the brain can develop regular cortical structures with local representation when it needs them for specific purposes. We do not see why such regular and even local structures could not have evolved for language processing as well.

Examples of regular structures and more or less local representation can be found in the language areas. For example, some stroke patients can name tools but not animals. Other specific naming disabilities include plants, body parts, and verbs (Calvin & Bickerton 2000; Pinker 1994). The temporal cortex seems to be organized in terms of categories of concepts, with different sites for different concepts. The temporal cortex, in turn, is connected with the prefrontal language areas with an important fiber bundle (the arcuate fasciculus), with axon branches that produce a very regular pattern of connectivity (Calvin & Bickerton 2000). In fact, the cortex has a much more regular structure than it is sometimes credited for (Hubel 1995).

A regular structure of columns also answers the question about the duplication of structure assemblies. They are not duplicated on the fly, but they form an existing structure (produced in a period of development). Again, the ice-cube model in the visual cortex comes to mind. Here one also finds a regular structure with repetitions (duplications) of the same representations (e.g., orientation columns).

R8.1. Dynamics

In their commentary, **Grüning & Treves** argue that we should combine our architecture with an effective use of cortical attractor dynamics. It is indeed our intention to do so. In fact, there is more attractor dynamics implied in the architecture than we were able to illustrate and describe. The word assemblies are indeed assumed to be attractor-like networks and not just symbolic representations in neural terms. This impression may follow from the fact that we have connected word assemblies directly to the structure assemblies in the blackboard. However, as we discuss in section R4, word assemblies will interact with the sentence blackboard through a phonological blackboard. The word assemblies are located outside the phonological (and sentence) blackboard. They would consist of attractors that are more or less distributed.

In terms of attractor dynamics, one could think of word assemblies as different attractors in the same network. That is, each word assembly would be an attractor state of the same set of neurons. This is most likely not the way in which word structures are instantiated in the brain. It does not agree, for example, with the fact that very specific loss of word use can be found in patients. Among anomic patients (i.e., patients who have trouble using nouns) one can find those who have difficulties in using concrete nouns but not abstract nouns, or vice versa. Similarly, one can find patients who have difficulties in using nouns for living things but not for nonliving things, or vice versa. Other examples include specific

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

deficits for naming animals, body parts, fruits and vegetables, colors, proper names, or objects that are typically found indoors (see, e.g., Pinker 1994). If all these words were instantiated as attractors in the same network, that would be hard to explain.

Therefore we assume that word assemblies are partly overlapping, sharing network structures with some words in one domain and with other words in other domains. In each of these domains one will find attractor dynamics that selects an attractor state in these domains. Together, the domains interact to select the overall word assembly at a given moment. The dynamics that select a word assembly will consist of a combination of attractor dynamics and interactive activation. The word assemblies we have shown are just those parts of the overall word assemblies that are connected to the sentence blackboard (or better, to the phonological blackboard).

R8.2. Hippocampus

The role of the hippocampus in our architecture is discussed by **Shastri** and **Whitney**. The hippocampus (HC) model we used to explain one-trial learning is the Hebb-Marr model. The neuroscientific evidence for this model is well documented (e.g., Rolls & Treves 1998). This model, as explained in section 6.5, provides a "snapshot" memory of about a second of an ongoing event. We showed how a sentence structure can be encoded in this way. Contrary to Shastri's assumption, longer sentences can also be encoded in HC in terms of separate but partly overlapping events, as illustrated in Figure 12. The fact that a sentence is temporarily encoded in HC does not mean that every sentence will eventually be encoded in long-term memory nor that a sentence will always be encoded in the way it was presented. In fact, the role of the HC in memory encoding is to form an initial storage of information that can be used in an elaborate process that incorporates aspects of that information into existing cortical memory structures. A discussion of this process and of the role of the HC can be found in O'Reilly & Rudy (2001).

Whitney notes that the sentence structure in the HC could interfere with the sentence structure in the blackboard because of an overlap in the structure assemblies used. This is not a real issue. First of all, the HC does not play a direct role in sentence processing in our model. The role of the HC concerns the transfer of information to long-term memory. But the issue of using the same structure assemblies would not really be important even if the HC was involved (e.g., in longer sentences). It is likely that a kind of "inhibition of return" occurs that prevents recently used structure assemblies being used again. So the chances of an overlap, even for longer sentences, are small.

R8.3. Central pattern generator

The central pattern generator (CPG) does not regulate the flow of activation in a gating circuit, as assumed by **Choe**. It regulates the onset and offset of different stages of competition in the process of answering a binding question. In this way, it resembles motor control, in which the onset and offset of muscle activation has to be regulated. Motor control consists of controlling the interaction between a specific set of muscles. In a similar way, the competition in the blackboard controls the interaction between a specific set of structure assemblies. The control of the competition process in the blackboard is not programmed, as assumed by **Durstewitz**. Instead, as suggested by **Grüning & Treves**, it will have developed on the basis of self-organization during language development. Because the control of the blackboard as given by a CPG does not depend on the specific content in the blackboard, it is possible to learn over time how control of specific sentence types (or clause types) proceeds in the blackboard. Further understanding of how motor control proceeds in the brain will be useful for understanding dynamic control in the blackboard as well. But in addition to motor control, linguistic processing is faced with the combinatorial productivity of language. The blackboard architecture was introduced to solve this issue.

R8.4. Development and evolution

The issue of how an architecture like the one we have presented can develop and evolve is raised by **Durstewitz, Grüning & Treves, Hadley,** and **Müller**. This is an important topic for further research. To study these issues, we have chosen a "backtrack" approach (van der Velde 2005). The aim of this approach is first to develop an architecture that can handle a number of linguistic issues (including human processing) reasonably well. Then this architecture can be used as a "target" architecture to investigate how this target architecture can develop on the basis of a more simplified version of it in combination with learning and development procedures. The benefit of this approach is that the information available on language and processing can be used to guide the process of development. Furthermore, the functionality of the target architecture is known, so there is no risk of getting stuck halfway. That risk is a serious problem of the procedure in which language development and evolution is modeled on the basis of arbitrary (i.e., unstructured) architectures and learning procedures. There is no guarantee that language can develop on the basis of any given neural structure. In fact, most species have not developed brains that can handle cognition of the complexity comparable to language. With this procedure, one could get stuck halfway, that is, even after some initial success, the initial architecture might not develop further to a more complete architecture for language (van der Velde 2005).

The fact that, for example, orientation maps and single-cell selectivity to orientation are already available at birth (e.g., Chapman et al. 1996) is in line with this approach. Apparently, basic elements of structure in the brain are available before learning. The disinhibition circuits in the cortex (e.g., Gonchar & Burkhalter 1999), the circuits for delay activity in the cortex (e.g., Fuster 1995), and the regular pattern of connectivity in the superficial layers of the cortex (e.g., Calvin 1995) could also belong to the building blocks of brain structure that are available before learning occurs. In this way, the development of the blackboard could indeed be the result of a process of self-organization, as suggested by **Grüning & Treves.**

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

R9. Linguistic issues

Several commentators have raised interesting linguistic issues concerning the blackboard for sentence structure and the processes related with it. Further development of the language blackboard is a topic for future research. But as Jackendoff (2002) noted, a solution of the problems he describes will produce a more productive dialogue between linguistic theory and neural network modeling. The issues raised by the commentators illustrate the beginning of such a dialogue.

R9.1. Complexity

We have argued that aspects of the neural dynamics in the architecture (e.g., dynamic interactions) can account for some complexity effects observed in linguistic processing. A more substantial account of these effects will depend on a further development of the architecture. However, the effect of dynamics can be used to illustrate the complexity difference between complement clause within relative clause (RC/CC) versus relative clause within complement clause (CC/RC) referred to by **Whitney.** The examples provided by Gibson (1998) are:

CC/RC: *The fact that the employee who the manager hired stole office supplies worried the executive.*

RC/CC: *The executive who the fact that the employee stole office supplies worried hired the manager.*

The RC/CC sentence is more complex than the CC/RC sentence. Figure R2 provides the basic structure of the sentences in terms of the blackboard architecture (ignoring *the* and *office*). On the face of it, they are similar and of equal complexity. However, in terms of the dynamics in the architecture, the situation is different. Delay activity, as used in our model, is based on stochastic neural activity (e.g., Amit 1989). This will have at least two effects on complexity. First, the longer it takes for a binding to occur, the greater the chance that delay activity (of a subassembly) is disrupted before a binding needs to be produced. Secondly, any binding conflict between subassemblies (even of an unequal nature) will affect the complexity of a sentence.

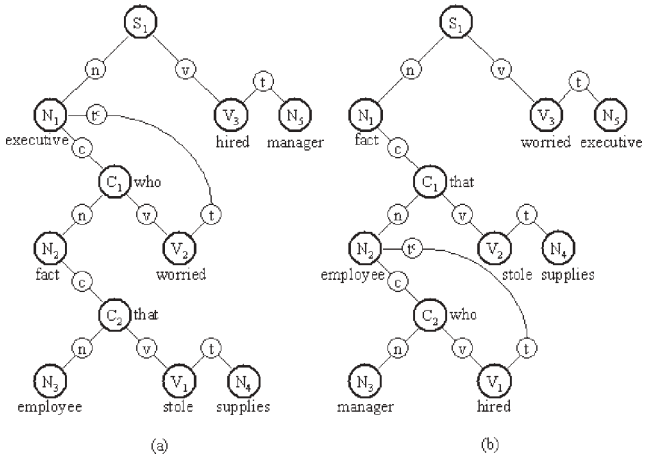


Figure R2. (a) Illustration of the structure of *The executive who the fact that the employee stole office supplies worried hired the manager* (without *the* and *office*). (b) Illustration of the structure of *The fact that the employee who the manager hired stole office supplies worried the executive* (without *the* and *office*).

In this respect, the sentences in Figure R2 are different. In the RC/CC sentence, the binding of *executive* (with *worried*) spans a greater interval of time compared to the CC/RC sentence. Furthermore, there is a conflict with the binding of *stole* with *supplies*, which occurs within this interval. In the CC/RC sentence, the interval of binding *employee* with *hired* is shorter, and there is no conflict in between. An additional aspect of complexity may result from the ambiguity associated with the word *fact*. This word can introduce a complement clause, but it can also be a theme of a next verb. The human language processor, operating in an incremental manner, often makes a choice between these ambiguities. It could be that when *fact* is found at the beginning of sentence, it is interpreted more often as a noun that introduces a complement clause (based on a familiarity with this sentence type). Instead, when *fact* occurs within the sentence, it could also be interpreted as a noun that can be the theme of the next verb. In that case, the RC/CC sentence in Figure R2 is even more complex than the RC/RC sentence in Figure 18.

The fact that delay activity, in particular in binding conflicts, can be disrupted (i.e., disappear) could account for the observation that RC/RC sentences can be perceived as grammatical when the second verb has been omitted, as noted by **Whitney**. For example, if the activity of the theme subassembly of *cat* in Figure 18a is disrupted (because of the long binding interval and binding conflict), the sentence without *bites* would indeed be perceived as grammatical (no open bindings or conflicts remain with this sentence).

Whitney argues that her TPARRSE model could provide an alternative to our architecture. Unfortunately, we are not familiar with the details of this model, so a more detailed comparison with our architecture will have to wait. However, there are a few aspects about the model referred to by Whitney that raise doubts about it. The first is the aspect of vector coding and vector operations for binding in this model. In section R7 we discuss vector coding and binding in more detail. It seems that there are two possibilities. The binding operations could be like those found in tensor models. In that case, the need for an increase of structure with depth of binding will be a serious problem for Whitney's model, as it is for tensor networks. The second option is the use of reduced vector representation to account for binding. However, we argue in section R7 that there is a fundamental flaw with this form of binding. Constituent representations are encapsulated in reduced vector coding, so they cannot be used to guide the process of answering binding questions. This means that, certainly in novel binding structures, the system has no information in order to decide on its own what sequence of operations it has to execute to answer a given binding question.

The second aspect of **Whitney**'s model that raises doubts is the use of "firing order" to implement a stack memory. How can firing order distinguish between different occurrences of the same word? Consider, for example, *A spy to spy on our spy who searches for their spy* (van der Velde 1999). The noun *spy* occurs in three positions in this

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

sentence: beginning, middle, and end. So it has to have three firing orders at the same time. It would seem, though, that any new occurrence of a word will eliminate its previous firing order, leaving only the last firing order for *spy* in this sentence.

R9.2. Cross-serial dependency

In her commentary, **Whitney** also refers to the relative ease of cross-serial dependencies in Dutch. As an illustration, consider the difference between the following clauses (see, e.g., Steedman 2000):

. . . omdat Jan Hans Peter de paarden zag helpen voeren.

. . . because Jan Hans Peter the horses saw help feed.

. . . because Jan saw Hans help Peter feed the horses.

The clause *Jan Hans Peter de paarden zag helpen voeren* is an example of cross-serial dependency in Dutch. The English clause *Jan saw Hans help Peter feed the horses* illustrates the binding relations in the Dutch clause. The binding relations are cross-serial because the first "agent" (*Jan*) binds with the first verb (*zag*/*saw*). The binding relations between the other "agents" and verbs proceed in the same way. Figure R3a illustrates the bare encoding of the English clause *Jan saw Hans help Peter feed the horses* in the blackboard architecture, using the W_x assemblies of the "phonological" blackboard illustrated in Figure R1. The temporal order in which the words bind to their W_x assemblies is the same as the temporal order in which the W_x assemblies bind to the structure assemblies (NP and VP) and the temporal order in which the structure assemblies bind to each other.

Figure R3b illustrates the bare encoding of the Dutch structure *Jan Hans Peter de paarden zag helpen voeren*. The structure is the same as the English structure in Figure R3a. The difference is in the temporal order of binding. Because word assemblies now bind to W_x assemblies, VP assemblies can be activated ahead of verbs. So V_1 and V_2 can be activated in anticipation of a verb. In the case of V_1, its "word" subassembly will be activated, and in the case of V_2, its main assembly. In this way, W_5 (*zag*) will bind with V_1; W_6 (*helpen*) can then bind with V_2. After that, W_7 (*voeren*) can bind to V_3, which can bind to the "agent" subassembly of N_3 and to N_4 (with their "theme" subassemblies). Figure R3b illustrates that this binding sequence is possible in the architecture, although it is of a more elaborate nature than the binding sequence for the equivalent English structure (Fig. R3a). This might account for the fact that cross-serial dependencies do not occur very often in languages. The fact that the English and Dutch binding structures in the architecture are similar may relate to the fact that both languages are grammatically related, despite the difference in the surface structure of the clauses involved (see, e.g., Steedman 2000). The binding process in Figure R3b is easier than the binding process with center embedding illustrated in Figure 18a.

R9.3. Processing

Several questions are raised concerning processing in the model. **Powers** argues that we have not shown how words are recognized, including their semantics. Word recognition was not a topic in our article because it has been a

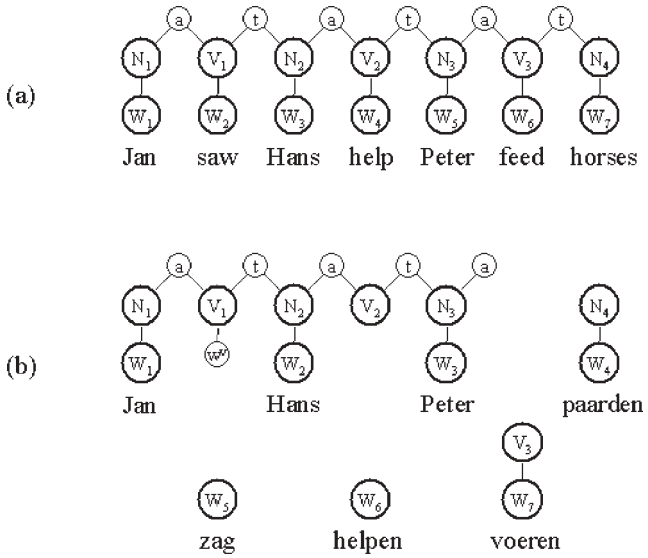


Figure R3. (a) Illustration of the clause *Jan saw Hans help Peter feed the horses*. (b) Illustration of the Dutch clause *Jan Hans Peter de paarden zag helpen voeren*.

topic in neural network research for many years. Feed-forward networks, interactive activation networks (see, e.g., McClelland & Rumelhart 1981), or ART (see, e.g., Grossberg 1999) could, in our view, be used for this purpose. Our aim was to discuss how words, once recognized, can be combined in combinatorial structures. Figure R1 illustrates that the word assemblies interact with the sentence blackboard through the phonological blackboard. As a result, the word assemblies can be as distributed as they need to be. Semantics will derive from the web of relations between the word assemblies and between word assemblies and other structures in the brain, such as the motor cortex or the visual cortex, as illustrated in Figure 27.

One aspect of word recognition is that it will provide information about the lexical nature of a word. In turn, this information will be used in the binding process in the architecture, as we illustrated in Figure 20. This is an aspect of the architecture that needs to be developed much further. Nevertheless, the circuit in Figure 20 illustrates some basic aspects of this process, in particular its incremental and interactive nature. What the circuit in Figure 20 does not show is the stochastic nature of this process. Consider a sentence like *The man who hunts ducks out on weekend* (Pinker 1994). The word *duck* is ambiguous because it can be a noun and a verb. We assume that the neural structure of the word *duck* (e.g., its word assembly) contains both kinds of information, which can be activated in the course of word processing. However, the incremental nature of human processing suggest that a competition between these aspects occurs that results in one interpretation of *duck* (i.e., noun or verb), which is then used in a circuit for binding, as illustrated in Figure 20. The garden-path nature of this sentence suggest that *duck* is interpreted initially as a noun.

Several factors could influence the competition between the ambiguous information that is related to a word, for example, the content of the sentence already processed, familiarity, or syntactic priming, that is, sentence

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

structures of previously processed sentences (e.g., Pickering & Branigan 1999). Aspects of prosody could also be of influence. For example, the sentence *The man who hunts ducks . . . out on weekend* would be interpreted differently from the sentence *The man who hunts . . . ducks out on weekend* because of the difference in the moment of the pause.

In terms of neural networks, these competition processes could be instantiated in interactive activation networks or attractor neural networks, as suggested by **Grüning & Treves**. In fact, we have already used attractor dynamics in the production of a context-free language with a production system (van der Velde 1995). This is, of course, a toy model of language production, but it illustrates that attractor networks can be used to implement control-like structures, as presented in Figure 20. Because the attractor dynamics used in the model was stochastic, the model already shows the noise tolerance, context-sensitivity, and analogue nature of processing referred to by **Durstewitz** and Grüning & Treves. It could also result in underspecified representations of a given sentence (Sanford & Sturt 2002).

So the fact that the rules would operate in a stochastic manner, as noted by **Powers**, is not a real issue. Having statistical rules is not the same as having no rules. The difference is that the application of a rule is not deterministic but depends on a stochastic process that can be influenced by a number of factors (e.g., context, syntactic priming). This kind of processing fits very well with the abilities of interactive activation networks or attractor neural networks. The interaction between these networks and the blackboard for sentence structure can account for the resolution of ambiguity, which, as **Baars** notes, will consist of an expectation-driven process that selects the most plausible interpretation, disregarding the multitude of other options.

R9.4. Challenges

In their commentary, **Phillips & Wagers** raise a number of very interesting challenges for the blackboard architecture and suggestions for its further development. This is an illustration of how the interaction between linguistics and neural network modeling can begin to be more productive.

Phillips & Wagers argue that hierarchical constituent structure is missing in the architecture. In the target article we noted that binding in this architecture is not a state but a process (see also sect. R3). The same is true for constituents. Assemblies such as S_i or C_i do not themselves represent constituents, as correctly noted by Phillips & Wagers. But they are crucial for the process in which constituents can be retrieved. For example, in Figure 16a, *cat that bites the dog* is a hierarchical constituent because it can be reactivated (retrieved) from S_1 downward. That is, by opening all "downward" gating circuits in a hierarchical manner, the reactivation process will eventually reach N_2. But it will also stop there, which indicates that a constituent boundary has been reached. This illustrates that hierarchical constituent structure is given by a process in this architecture (i.e., the process that reactivates the constituent in a hierarchical manner). Issues related to constituents, as discussed by Phillips & Wagers, should be handled in the same way in the further development of the architecture.

Coordination rules, addressed by **Phillips & Wagers**, are an important topic for further development. One suggestion we have been exploring is the introduction of specific structure assemblies for coordination (e.g., "and" assemblies). This solution is in line with the suggestion made by Phillips & Wagers that NP and VP assemblies should bind recursively. (Furthermore, Figure R1 illustrates that they do not directly bind to words in the present version of the architecture.) A question that we explore is whether these coordination assemblies should be general (i.e., used for all coordination) or specific (e.g., specific "and" assemblies for NPs). Coordination assemblies could also be used for structures such as *six big red India rubber*, which could be seen as an elliptic version of *six (and) big (and) red (and) India (and) rubber*. The word *and* is dropped, but the "and" assemblies could still be used.

Anaphoric relations are not yet included in the blackboard architecture. An interesting aspect of anaphoric relations is that they can reach beyond the boundaries of a sentence. For example, in *The boy sees ice cream. He wants it*, the anaphoric relations bind information across sentences. In Figure R1 we illustrate a (quasi-)architecture for structure within words. In the target article we discussed an architecture for structure within sentences and clauses. A direct extension would be a specific blackboard architecture for structure beyond sentence boundaries, as given by anaphoric relations.

R10. Final remarks

The purpose of the target article was to show how true combinatorial structures can be instantiated in neural terms. True combinatorial structures depend on constituent structures that can be combined on the fly. Furthermore, information can be retrieved from the combinatorial structures on the basis of their constituents. In this way (and, as we argue, only in this way) can information be processed and retrieved in a manner that is productive and systematic. Neural structure and dynamics seem to prevent constituents being copied and moved elsewhere. Therefore we suggest that neural representations of constituents remain "*in situ*." Combinatorial structures can be instantiated on the basis of these constituents by dynamic binding processes and structures (e.g., structure assemblies) of the kind illustrated in the target article.

However, the success of a particular approach cannot be determined on the basis of a single article. This is true for the architecture we present here as well as for the alternative approaches discussed here or presented elsewhere in the literature. In the end, the success of any approach is determined by the research program it initiates and the progress of that program. We are fully aware that we are just at the beginning of such a program and that it is far too soon to determine whether it will be successful. The commentators have been very helpful in pointing out and discussing the problems that have to be solved and the difficulties that have to be surmounted. Most likely, more difficulties will be lurking in the background, waiting to pop up at the right (or wrong) moment. However, we have not yet seen a problem that seems to be unsolvable nor any

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

difficulty that seems to be insurmountable. This raises our confidence in the further development of the architectures we present here.

References

Letters "a" and "r" appearing before authors' initials refer to target article and response, respectively.

Amit, D. (1989) *Modeling brain function.* Cambridge University Press. [FTS, arFVDV]

Amunts, K., Schleicher, A., Burgel, U., Mohlberg, H., Uylings, H. B. & Zilles, K. (1999) Broca's region revisited: Cytoarchitecture and intersubject variability. *Journal of Comparative Neurology* 412:319–41. [R-AM]

Amunts, K., Schleicher, A., Ditterich, A. & Zilles, K. (2003) Broca's region: Cytoarchitectonic asymmetry and developmental changes. *Journal of Comparative Neurology* 465:72–89. [R-AM]

Anderson, C. H. & van Essen, D. C. B. (1987) Shifter circuits: A computational strategy for directed visual attention and other dynamic processes. *Proceedings of the National Academy of Sciences of the United States of America* 84:6297–301. [FTS]

Anderson, J. R. (1976) *Language, memory and thought.* Erlbaum. [DMWP]
(1983) *The architecture of cognition.* Harvard University Press. [aFVDV]

Anstis, S. (2005) Holistic word processing. *Perception* 34:237–40. [BD]

Arathorn, D. (2002) *Map-seeking circuits in visual cognition.* Stanford University Press. [FTS]

Baars, B. J. (1988) *A cognitive theory of consciousness*. Cambridge University Press. [BJB]
(1997) *In the theater of consciousness: The workspace of the mind*. Oxford University Press. [BJB]
(2002) The conscious access hypothesis: Origins and recent evidence. *Trends in Cognitive Science* 6:47–52. [BJB]

Baars, B. J. & Franklin, S. (2003) How conscious experience and working memory interact. *Trends in Cognitive Science* 7:166–72. [BJB]

Baars, B. J., Laureys, S. & Ramsoy, T. Z. (2004) Brain, conscious experience and the observing self. *Trends in Neurosciences* 26:671–75. [BJB]

Bach, E., Brown, C. & Marslen-Wilson, W. (1986) Crossed and nested dependencies in German and Dutch: A psycholinguistic study. *Language and Cognitive Processes* 1:249–62. [CW]

Barsalou, L. W. (1999) Perceptual symbol systems. *Behavioral and Brain Sciences* 22(4):577–610. [aFVDV]

Barsalou, L. W., Simmons, W. K., Barbey, A. K. & Wilson, C. D. (2003) Grounding conceptual knowledge in modality-specific systems. *Trends in Cognitive Sciences* 7:84–91. [aFVDV]

Barthaud, J. C., Koenig, O. & Dresp, B. (2003) Effects of explicit versus implicit instructions on visual repetition priming by shape and colour. *Perception* 32(Supp.):52–53. [BD]

Bastiaansen, M. C., Posthuma, D., Groot, P. F. & de Geus, E. J. (2002) Event-related alpha and theta responses in a visuo-spatial working memory task. *Clinical Neurophysiology* 113:1882–93. [R-AM]

Batista, A. P., Buneo, C. A., Snyder, L. H. & Andersen, R. A. (1999) Reach plans in eye-centered coordinates. *Science* 285:257–60. [aFVDV]

Bechtel, W., & Abrahamsen, A. (2002) *Connectionism and the mind.* Blackwell. [aFVDV]

Ben-Sachar, M., Hendler, T., Kahn, I., Ben-Bashat, D. & Grodzinsky, Y. (2003) The neural reality of syntactic transformations: Evidence from functional magnetic resonance imaging. *Psychological Science* 14:433–40. [aFVDV]

Biederman, I. (1987) Recognition by components: A theory of human image understanding. *Psychological Review* 94:115–47. [aFVDV]

Bienenstock, E. & von der Malsburg, C. (1987) A neural network for invariant pattern recognition. *Europhysics Letters* 4:121–26. [FTS]

Bierwisch, M. (1999) Words in the brain are not just labelled concepts. *Behavioral and Brain Sciences* 22(2):280–82. commentary on Pulvermüller, F. Words in the brain's language. *Behavioral and Brain Sciences* 22(2): 253–79. [aFVDV]

Bloom, P. (2000) *How children learn the meanings of words.* MIT Press. [aFVDV]

Bloom, P., Peterson, M. A., Nadel, L & Garrett, M. F., eds. (1996) *Language and space.* MIT Press. [aFVDV]

Bod, R. (1995) Enriching linguistics with statistics: Performance models of Natural Language. Doctoral dissertation, University of Amsterdam. [DMWP]
(1998) *Beyond grammar*. CSLI. [RB]

Braitenberg, V. (1978) Cell assemblies in the cerebral cortex. In: *Theoretical Approaches to Complex Systems*, ed. R. Heim & G. Palm. Springer. [FTS]

Bresnan, J. (2001) *Lexical-functional syntax.* Blackwell. [CP]

Brody, C. D. & Hopfield, J. J. (2003) Simple networks for spike-timing-based computation, with application to olfactory processing. *Neuron* 37:843–52. [DD]

Brunel, N. (1996) Hebbian learning of context in recurrent neural networks. *Neural Computation* 8:1677–710. [aFVDV]

Bulier J. (2001) Feedback connections and conscious vision. *Trends in Cognitive Sciences* 5:369–70. [aFVDV]

Byron, F. W. & Fuller, R. W. (1992) *Mathematics of classical and quantum physics.* Dover. [rFVDV]

Calvin, W. H. (1995) Cortical columns, modules, and Hebbian assemblies. In: *The handbook of brain theory and neural networks*, ed. M. A. Arbib. MIT Press. [rFVDV]

Calvin, W. H. & Bickerton, D. (2000) *Lingua ex machina: Reconciling Darwin and Chomsky with the human brain.* MIT Press. [arFVDV]

Caplan, D. (1992) *Language: Structure, processing and disorders.* MIT Press. [aFVDV]

Caplan, D., Alpert, N. & Waters, G. (1998) Effects of syntactic structure and propositional number on patterns of regional cerebral blood flow. *Journal of Cognitive Neuroscience* 10:541–52. [R-AM]

Chapman, B., Stryker, M. P. & Bonhoeffer, T. (1996) Development of orientation preference maps in ferret primary visual cortex. *Journal of Neuroscience* 16:6443–53. [rFVDV]

Charniak, E. (1993) *Statistical language learning*. MIT Press. [DMWP]
(1997) Statistical techniques for natural language parsing. *AI Magazine* Winter. [RB]

Chelazzi, L., Miller, E. K., Duncan, J. & Desimone, R. (1993) A neural basis for visual search in inferior temporal cortex. *Nature* 363:345–47. [aFVDV]

Choe, Y. (2004) The role of temporal parameters in a thalamocortical model of analogy. *IEEE Transactions on Neural Networks* 15:1071–82. [YC]

Chomsky, N. (1957) *Syntactic structures.* Mouton. [RB, rFVDV]
(1965) *Aspects of the theory of syntax.* MIT Press. [CP]
(2000) *New horizons in the study of language and mind.* Cambridge University Press. [aFVDV]

Churchland, P. M. (1995) *The engine of reason, the seat of the soul.* MIT Press. [aFVDV]

Clancey, W. J. (1999) *Conceptual coordination: How the mind orders experience in time.* Erlbaum. [WJC]
(2000) Conceptual coordination bridges information processing and neuropsychology. *Behavioral and Brain Sciences* 23(6):919–22. [WJC]
(2003) The Newell Test should commit to diagnosing dysfunctions. *Behavioral and Brain Sciences* 26(5):604–605. Commentary on Anderson, J. R. & Lebiere, C. The Newell test for a theory of cognition. *Behavioral and Brain Sciences* 26(5):587–601. [WJC]

Clark, A. (2001) Unsupervised language acquisition: Theory and practice. Doctoral dissertation, COGS, University of Sussex. [DMWP]

Colby, C. L., Duhamel, J-R. & Goldberg, M. E. (1996) Visual, presaccadic, and cognitive activation of single neurons in monkey lateral intraparietal area. *Journal of Neurophysiology* 76:2841–52. [aFVDV]

Coren, S., Ward, L. M. & Enns, J. T. (1999) *Sensation and perception*. Harcourt Brace. [rFVDV]

Crabtree, J. W., Collingridge, G. L. & Isaac, J. T. (1998) A new intrathalamic pathway linking modality-related nuclei in the dorsal thalamus. *Nature Neuroscience* 1:389–94. [YC]

Damasio, A. R. (1983) Pure alexia. *Trends in Neurosciences* 6:93–96. [BD]

de Kamps, M. & van der Velde, F. (2001a) From artificial neural networks to spiking neuron populations and back again. *Neural Networks* 14:941–53. [aFVDV]
(2001b) Using a recurrent network to bind form, color and position into a unified percept. *Neurocomputing* 38–40:523–28. [aFVDV]

Deane, P. (1992) *Grammar in mind and brain.* Mouton de Gruyter. [DMWP]

Dehaene, S., Sergent, C. & Changeux, J. P. (2003) A neuronal network model linking subjective reports and objective physiological data during conscious perception. *Proceedings of the National Academy of Sciences of the United States of America* 100:8520–25. [BJB]

Delorme, A., Rousselet, G. A., Mace, M. J. & Fabre-Thorpe, M. (2004) Interaction of top-down and bottom-up processing in the fast visual analysis of natural scenes. *Cognitive Brain Research* 19(2):103–13. [BD]

Dennett, D. C. (1991) *Consciousness explained.* Allen Lane. [arFVDV]

Desimone, R. & Duncan, J. (1995) Neural mechanisms of selective visual attention. *Annual Review of Neuroscience* 18:193–222. [aFVDV]

D'Esposito, M. (2001) Functional neuroimaging of working memory. In: *Handbook of functional neuroimaging of cognition*, ed. R. Cabeza & A. Kingstone. MIT Press. [aFVDV]

Dominey, P. F. (1997) An anatomically structured sensory-motor sequence learning system displays some general linguistic capacities. *Brain and Language* 59:50–75. [aFVDV]

Dotsenko, V. S. (1988) Neural networks: Translation-, rotation-, and scale-invariant pattern recognition. *Journal of Physics A* 21:L783–87. [FTS]

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

Doumas, L. A. A. & Hummel, J. E. (2005) Modeling human mental representations: What works, what doesn't, and why. In: *The Cambridge handbook of thinking and reasoning*, ed. K. J. Holyoak & R. G. Morrison. Cambridge University Press. [LAAD]

Duncan, J. (2001) An adaptive coding model of neural function in prefrontal cortex. *Nature Reviews Neuroscience* 2:820–29. [aFVDV]

Durstewitz, D., Seamans, J. K. & Sejnowski, T. J. (2000) Neurocomputational models of working memory. *Nature Neuroscience* 3(Supp.):1184–91. [aFVDV]

Edelman, S. & Intrator, N. (2003) Towards structural systematicity in distributed, statically bound visual representations. *Cognitive Science* 27:73–109. [aFVDV]

Elman, J. L. (1991) Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning* 7:195–225. [arFVDV]

Embick, D., Marantz, A., Miyashita, Y., O'Neil, W. & Sakai, K. L. (2000) A syntactic specialization for Broca's area. *Proceedings of the National Academy of Sciences of the United States of America* 97:6150–54. [R-AM]

Farah, M., Humphries, G. W. & Rodman, H. R. (1999) Object and face recognition. In: *Fundamental neuroscience*, ed. M. J. Zigmond, F. E. Bloom, S. C. Landis, J. L. Roberts & L. R. Squire. Academic Press. [aFVDV]

Farah, M. J. (1990) *Visual agnosia*. MIT Press. [rFVDV]

Feigenbaum, E. A. & Feldman, J., eds. (1963) *Computers and thought*. McGraw-Hill. [FTS]

Felleman, D. J. & van Essen, D. C. (1991) Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex* 1:1–47. [aFVDV]

Fellous, J. M., Tiesinga, P. H., Thomas, P. J. & Sejnowski, T. J. (2004) Discovering spike patterns in neuronal responses. *Journal of Neuroscience* 24:2989–3001. [DD]

Fodor, J. A. (2000) *The mind doesn't work that way: The scope and limits of computational psychology*. MIT Press. [aFVDV]

Fodor, J. A. & McLaughlin, B. P. (1990) Connectionism and the problem of systematicity: Why Smolensky's solution doesn't work. *Cognition* 35:183–204. [RWG, aFVDV]

Fodor, J. A., & Pylyshyn, Z. W. (1988) Connectionism and cognitive architecture: A critical analysis. *Cognition* 28:3–71. Available at: http://citeseer.ist.psu.edu/446398.html. [RGW, arFVDV]

Frank, R. (2002) *Phrase structure composition and syntactic dependencies*. MIT Press. [CP]

Franklin, S. (2003) IDA: A conscious artifact? *Journal of Consciousness Studies* 10:47–66. [BJB]

Fries, P., Neuenschwander, S., Engel, A. K., Goebel, R. & Singer, W. (2001) Rapid feature selective neuronal synchronization through correlated latency shifting. *Nature Neuroscience* 4:194–200. [R-AM]

Friston, K. J., Harrison, L. & Penny, W. (2003) Dynamic causal modelling. *NeuroImage* 19:1273–302. [aFVDV]

Fulvi Mari, C. & Treves, A. (1998) Modeling neurocortical areas with a modular neural network. *Biosystems* 48:47–55. [AG]

Fuster, J. M. (1995) *Memory in the cerebral cortex*. MIT Press. [arFVDV]

(2001) The prefrontal cortex—An update: Time is of the essence. *Neuron* 30:319–33. [aFVDV]

Fuster, J. M., Bauer, R. H. & Jervey, J. P. (1985) Functional interactions between inferotemporal and prefrontal cortex in a cognitive task. *Brain Research* 330:299–307. [aFVDV]

Gansner, E. R., Koutsofios, E., North, S. C. & Vo, K. P. (1993) A technique for drawing directed graphs. *IEEE Transactions on Software Engineering* 19:214–30. [aFVDV]

Gayler, R. W. (1998) Multiplicative binding, representation operators, and analogy [Abstract of poster]. In: *Advances in analogy research: Integration of theory and data from the cognitive, computational, and neural sciences*, ed. K. Holyoak, D. Gentner & B. Kokinov. New Bulgarian University. Available at: http://cogprints.org/502/. [RWG, FTS]

(2003) Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience. In: *Proceedings of the Joint International Conference on Cognitive Science*, ed. P. P. Slezak. University of New South Wales. Available at: http://cogprints.org/3983/. [RWG, FTS]

Gayler, R. W. & Wales, R. (1998) Connections, binding, unification, and analogical promiscuity. In: *Advances in analogy research: Integration of theory and data from the cognitive, computational, and neural sciences*, ed. K. Holyoak, D. Gentner & B. Kokinov. New Bulgarian University. Available at: http://cogprints.org/500/. [RWG]

Gazzaniga, M. S. (1995) Preface. In: *The cognitive neurosciences*, ed. M. S. Gazzaniga. MIT Press. [aFVDV]

Gers, F. A. & Schmidhuber, J. (2001) LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks* 12:1333–40. [aFVDV]

Gerstner, W. (2000) Population dynamics of spiking neurons: Fast transients, asynchronous states, and locking. *Neural Computation* 12:43–89. [aFVDV]

Gibson, E. (1998) Linguistic complexity: Locality of syntactic dependencies. *Cognition* 68:1–76. [arFVDV, CW]

Gibson, E. & Thomas, J. (1999) Memory limitations and structural forgetting: The perception of complex ungrammatical sentences as grammatical. *Language and Cognitive Processes* 14:225–48. [CW]

Goldberg, A. (1995) *Constructions: A construction grammar approach to argument structure*. University of Chicago Press. [CP]

Gonchar, Y. & Burkhalter, A. (1999) Connectivity of GABAergic calretinin-immunoreactive neurons in rat primary visual cortex. *Cerebral Cortex* 9:683–96. [arFVDV]

Gottlieb, J. P., Kusunoki, M. & Goldberg, M. E. (1998) The representation of visual salience in monkey parietal cortex. *Nature* 391:481–84. [aFVDV]

Grodzinsky, Y. (2000) The neurology of syntax: Language use without Broca's area. *Behavioral and Brain Sciences* 23(1):1–21. [aFVDV]

Grossberg, S. (1999) The link between brain learning, attention, and consciousness. *Consciousness & Cognition* 8(1):1–44. [BD, rFVDV]

Grüning, A. (in press) Stack- and queue-like dynamics in recurrent neural networks. *Connection Science*. [AG]

Hadley, R. F. & Cardei, V. C. (1999) Language acquisition from sparse input with no error feedback. *Neural Networks* 12:217–35. [RFH]

Hadley, R. F. & Hayward, M. B. (1997) Strong semantic systematicity from Hebbian connectionist learning. *Minds and Machines* 7:1–37. [RFH]

Halford, G. S., Wilson, W. H. & Phillips S. (1998) Processing capacity defined by relational complexity: Implications for comparative, developmental, and cognitive psychology. *Behavioral and Brain Sciences* 21(6):803–31. Available at: http://www.bbsonline.org/documents/a/00/00/05/50/index.html. [RWG]

Hanson, S. J. & Negishi, M. (2002) On the emergence of rules in neural networks. *Neural Computation* 14:2245–68. [DD, rFVDV]

Harnad, S. (1991) The symbol grounding problem. In: *Emergent computation: Self-organizing, collective, and cooperative phenomena in natural and artificial computing networks*, ed. S. Forrest. MIT Press. [aFVDV]

Harris, K. D., Csicsvari, J., Buhl, D. & Buzsaki, G. (2003) Organization of cell assemblies in the hippocampus. *Nature* 424:552–56. [FTS]

Harth, E., Unnikrishnan, K. P. & Pandaya, A. S. (1987) The inversion of sensory processing by feedback pathways: A model of visual cognitive functions. *Science* 237:184–87. [YC]

Hauser, M. D., Chomsky, N. & Fitch, W. T. (2002) The faculty of language: What is it, who has it, and how did it evolve. *Science* 298:1569–79. [AG]

Hawkins, J. A. (1994) *A performance theory of order and constituency*. Cambridge University Press. [aFVDV]

Hebb, D. O. (1949) *The organization of behavior: A neuropsychological theory*. Wiley. [DMWP, FTS, aFVDV]

Henderson, J. (1994) Connectionist syntactic parsing using temporal variable binding. *Journal of Psycholinguistic Research* 23:353–79. [LS, rFVDV]

Hinton, G. E. (1988) Representing part-whole hierarchies in connectionist networks. In: *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*, pp. 48–54. Erlbaum. [RWG]

(1990) Mapping part-whole hierarchies into connectionist networks. *Artificial Intelligence* 46:47–75. Available at: http://dx.doi.org/10.1016/0004-3702(90)90004-J [RWG, FTS]

Holyoak, K. J. & Hummel, J. E. (2000) The proper treatment of symbols in a connectionist architecture. In: *Cognitive dynamics: Conceptual change in humans and machines*, ed. E. Dietrich & A. Markman. Erlbaum. [LAAD]

(2001) Toward an understanding of analogy within a biological symbol system. In: *The analogical mind: Perspectives from cognitive science*, ed. D. Gentner, K. J. Holyoak, & B. N. Kokinov. MIT Press. [LAAD]

Hopfield, J. J. (1982) Neural networks and physical systems with emerging collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America* 79:2554–58. [FTS]

(1995) Pattern recognition computation using action potential timing for stimulus representation. *Nature* 376:33–36. [DD]

Hornik, K., Stinchcombe, M. & White, H. (1989) Multilayer feedforward networks are universal approximators. *Neural Networks* 2:359–66. [rFVDV]

Hubel, D. (1995) *Eye, brain, and vision*. Scientific American Library. [DMWP, arFVDV]

Hummel, J. E. (2000) Where view-based theories break down: The role of structure in shape perception and object recognition. In: *Cognitive dynamics: Conceptual change in humans and machines*, ed. E. Dietrich & A. Markman. Erlbaum. [LAAD]

Hummel, J. E. & Biederman, I. (1992) Dynamic binding in a neural network for shape recognition. *Psychological Review* 99:480–517. [LAAD]

Hummel, J. E. & Holyoak, K. J. (1992) Indirect analogical mapping. *Proceedings of the 14th Annual Conference of the Cognitive Science Society* 516–21. [LAAD]

(1993) Distributing structure over time. *Behavioral and Brain Sciences* 16(3):464. commentary on Shastri L. & Ajjanagadde, V. (1993) From simple associations

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences* 16(3):417–51. [aFVDV]

(1996) LISA: A computational model of analogical inference and schema induction. *Proceedings of the 18th Annual Conference of the Cognitive Science Society*, La Jolla, CA. Lawrence Erlbaum Associates. 352–57. [LAAD]

(1997) Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review* 104:427–66. [LAAD, rFVDV]

(2003) A symbolic-connectionist theory of relational inference and generalization. *Psychological Review* 110(2):220–64. [LAAD, rFVDV]

Hummel, J. E., Holyoak, K. J., Green, C., Doumas, L. A. A., Devnich, D., Kittur, A. & Kalar, D. J. (2004) A solution to the binding problem for compositional connectionism. In: *Compositional connectionism in cognitive science: Papers from the AAAI fall symposium*, ed. S. D. Levy & R. Gayler, Technical Report FS-04-03. AAAI Press. [LAAD]

Hutchens, J. A. (1995) Natural language grammatical inference. Doctoral dissertation, University of Western Australia, Department of Information Technology. [DMWP]

Ikegaya, Y., Aaron, G., Cossart, R., Aronov, D., Lampl, I., Ferster, D. & Yuste, R. (2004) Synfire chains and cortical songs: Temporal modules of cortical activity. *Science* 304:559–64. [DD, FTS]

Jackendoff, R. (2002) *Foundations of language: Brain, meaning, grammar, evolution.* Oxford University Press. [AG, RWG, CP, FTS, arFVDV]

(2003) Précis to *Foundations of language: Brain, meaning, grammar, evolution*. *Behavioral and Brain Sciences* 26(6):651–65. [aFVDV]

Jacobs, B., Schall, M., Prather, M., Kapler, E., Driscoll, L., Baca, S., Jacobs, J., Ford, K., Wainwright, M. & Treml, M. (2001) Regional dendritic and spine variation in human cerebral cortex: A quantitative Golgi study. *Cerebral Cortex* 11(6):558–71. [R-AM]

Joshi, A., Vijay-Shanker, K. & Weir, D. (1991) The convergence of mildly context-sensitive grammar formalisms. In: *Foundational issues in natural language processing*, ed. P. Sells, S. Shieber & T. Wasow. MIT Press. [RB]

Just, M. A. & Carpenter, P. A. (1992) A capacity theory of comprehension: Individual differences in working memory. *Psychological Review* 99:122–49. [rFVDV]

Just, M. A., Carpenter, P. A., Keller, T. A., Eddy, W. F. & Thulborn, K. R. (1996) Brain activation modulated by sentence comprehension. *Science* 274:114–16. [R-AM]

Kaan, E. & Vasic, N (2004) Cross-serial dependencies in Dutch: Testing the influence of NP type on processing load. *Memory and Cognition* 32:175–84 [CW]

Kanerva, P. (1994) The spatter code for encoding concepts at many levels. In: *ICANN'94, Proceedings of International Conference on Artificial Neural Networks*, ed. M. Marinaro and P. G. Morasso, pp. 226–29. Springer-Verlag. Available at: http://citeseer.ist.psu.edu/666731.html. [RWG, FTS]

(1997) Fully distributed representation. In: *Proceedings of the 1997 Real World Computing Symposium (RWC'97, Tokyo).* Real World Computing Partnership. Available at: http://citeseer.ist.psu.edu/247558.html. [RWG, FTS]

Kling, U. & Szekély, G. (1968) Simulation of rhythmic activities. I. Function of network with cyclic inhibitions. *Kybernetics* 5:89–103. [aFVDV]

Koch, C. (1999) *Biophysics of computation: Information processing in single neurons.* Oxford University Press. [aFVDV]

Kohonen, T. (1977) *Associative memory: A system theoretic approach.* Springer. [FTS]

Kree, R. & Zippelius, A. (1988) Recognition of topological features of graphs and images in neural networks. *Journal of Physics A* 21:L813–18. [FTS]

Lange, T. & Wharton, C. (1995) A structured connectionist approach to inferencing and retrieval. In: *Computational architectures integrating neural and symbolic processes*, ed. R. Sun & L. A. Bookman. Kluwer. [MGD]

Lange, T. E. & Dyer, M. G. (1989) high-level inferencing in a connectionist network. *Connection Science* 1:181–217. [MGD]

Lewis, R. L. (1996) Inference in short-term memory: The magical number two (or three) in sentence processing. *Journal of Psycholinguistic Research* 25:93–115. [WJC]

(1999) Accounting for the fine structure of syntactic working memory: Similarity-based interference as a unifying principle. *Behavioral and Brain Sciences* 22(1):105–106. commentary on Caplan, D. & Waters, G. S. Verbal working memory and sentence. *Behavioral and Brain Sciences* 22(1): 77–94. [aFVDV]

Livingstone, M. S. & Hubel, D. H. (1988) Segregation of form, colour, movement and depth: Anatomy, physiology, and perception. *Science* 240:740–49. [aFVDV]

Luck, S. J. & Vogel, E. K. (1997) The capacity of visual working memory for features and conjunctions. *Nature* 390:279–81. [aFVDV]

Macchi, G., Bentivoglio, M., Molinari, M. & Minciacchi, D. (1984) The thalamo-caudal versus thalamo-cortical projections as studied in the cat with fluorescent retrograde double labeling. *Experimental Brain Research* 54: 225–39. [YC]

Mace, M. J., Richard, G., Delorme, A. & Fabre-Thorpe, M. (2005) Rapid categorization of natural scenes in monkeys: Target predictability and processing speed. *Neuroreport* 16:349–54. [BD]

Mani, D. & Shastri, L. (1993) Reflexive reasoning with multiple-instantiation in a connectionist reasoning system with a type hierarchy. *Connection Science* 5:205–42. [LS]

Marcus, G. F. (2001) *The algebraic mind.* MIT Press. [aFVDV]

Marcus, M. (1991) The automatic acquisition of linguistic structure from large corpora. In: *AAAI Spring Symposium on Machine Learning of Natural Language and Ontology*, ed. D. Powers & L. Reeker. Deutches Forchungszentrum für Künstliche Intelligenz GmbH. [DMWP]

McClelland, J. L. & Kawamoto, A. H. (1986) Mechanisms of sentence processing: Assigning roles to constituents. In: *Parallel distributed processing: Explorations in the microstructure of cognition, vol. 2: Psychological and biological models*, ed. J. L. McClelland, D. E. Rumelhart & PDP Research Group. MIT Press. [aFVDV]

McClelland, J. L. & Rumelhart, D. E. (1981) An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review* 88:375–407. [rFVDV]

McCloskey, M. & Cohen, N. J. (1989) Catastrophic interference in connectionist networks: The sequential learning problem. In: *The Psychology of Learning and Motivation, vol. 24*, ed. G. H. Bower. Academic Press. [rFVDV]

McNaughton, B. L. & Nadel, L. (1990) Hebb-Marr networks and the neurobiological representation of action in space. In: *Neuroscience and connectionist theory*, ed. M. A. Gluck & D. E. Rumelhart. Erlbaum. [aFVDV]

Meng, M. & Bader, M. (2000) Ungrammaticality detection and garden path strength: Evidence for serial parsing. *Language and Cognitive Processes* 15:615–66. [rFVDV]

Miikkulainen, R. (1993) *Subsymbolic natural language processing.* MIT Press. [MGD]

(1996) Subsymbolic case-role analysis of sentences with embedded clauses. *Cognitive Science* 20:47–73. [aFVDV]

Miikkulainen, R. & Dyer, M. G. (1991) Natural language processing with modular PDP networks and distributed lexicon. *Cognitive Science* 15:343–99. [MGD]

Miller, G. (1956) The magical number seven plus or minus two. *Psychological Review* 63:81–97 [DMWP]

Miller, G. A. (1967) *The psychology of communication.* Penguin. [aFVDV]

Moscovitch, M., Winocur, G. & Behrmann, M. (1997) What is special about face recognition? Nineteen experiments on a person with visual object agnosia and dyslexia but normal face recognition. *Journal of Cognitive Neuroscience* 9:555–64. [BD]

Motter, B. C. (1994) Neural correlates of attentive selection for color or luminance in extrastriate area V4. *Journal of Neuroscience* 14:2178–89. [aFVDV]

Müller, R.-A. (2000) A big "housing" problem and a trace of neuroimaging: Broca's area is more than a transformation center. *Behavioral and Brain Sciences* 23(1):42. commentary on Grodzinsky, Y. (2000) The neurology of syntax: Language use without Broca's area. *Behavioral and Brain Sciences* 23(1):1–21. [R-AM]

Müller, R.-A. & Basho, S. (2004) Are nonlinguistic functions in "Broca's area" prerequisites for language acquisition? FMRI findings from an ontogenetic viewpoint. *Brain and Language* 89:329–36. [R-AM]

Mumford, D. (1991) On the computational architecture of the neocortex, Part I, the role of the thalamo-cortical loop. *Biological Cybernetics* 65:135–45. [YC]

(1995) Thalamus. In: *The handbook of brain theory and neural networks*, ed. M. A. Arbib. MIT Press. [YC]

Nadel, L. & Moscovitch, M. (2001) The hippocampal complex and long-term memory revisited. *Trends in Cognitive Sciences* 5:228–30. [aFVDV]

Nase, G., Singer, W., Monyer, H. & Engel, A. K. (2003) Features of neuronal synchrony in mouse visual cortex. *Journal of Neurophysiology* 90,1115–23. [R-AM]

Nenov, V. I. & Dyer, M. G. (1993) Perceptually grounded language learning: Part 1—A neural network architecture for robust sequential association. *Connection Science* 5:115–38. [MGD]

(1994) Perceptually grounded language learning: Part 2—DETE: A neural/procedural model. *Connection Science* 6:3–41. [MGD]

Newell, A. (1990/1994) *Unified theories of cognition: The William James lectures.* Harvard University Press. [BJB, aFVDV]

Newell, A., Rosenbloom, P. S. & Laird, J. E. (1989) Symbolic architectures for cognition. In: *Foundations of cognitive science*, ed. M. Posner. MIT Press. [rFVDV]

Newell, A. & Simon, H. A. (1972) *Human problem solving.* Prentice-Hall. [FTS]

Newman, J., Baars, B. J. & Cho, S.-B. (1997) A neural global workspace model for conscious attention. *Neural Networks* 10:1195–206. [YC, aFVDV]

O'Kane, D. & Treves, A. (1992) Why the simplest notion of neocortex as an auto-associative memory would not work. *Network* 3:379–84. [AG]

Olshausen, B. A., Anderson, C. H. & Van Essen, D. C. B. (1993) A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *Journal of Neuroscience* 13:4700–19. [FTS]

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

Oram, M. W. & Perrett, D. I. (1994) Modeling visual recognition from neurobiological constraints. *Neural Networks* 7:945–72. [arFVDV]
O'Reilly, R. C. & Rudy, J. W. (2001) Conjunctive representations in learning and memory: Principles of cortical and hippocampal function. *Psychological Review* 108:311–45. [arFVDV]
Palm, G. (1982) *Neural assemblies.* Springer. [FTS]
Palmer-Brown, D., Tepper, J. A. & Powell, H. M. (2002) Connectionist natural language parsing. *Trends in Cognitive Sciences* 6:437–42. [aFVDV]
Piaget, J. (1954) *The construction of reality in the child.* Basic Books. [DMWP]
Piattelli-Palmarini, M. (2002) The barest essentials. *Nature* 416:129. [aFVDV]
Pickering, M. J. & Branigan, H. P. (1999) Syntactic priming in language production. *Trends in Cognitive Sciences* 3:136–41. [rFVDV]
Pike, K. & Pike, E. (1977) *Grammatical analysis.* Summer Institute of Linguistics and the University of Texas, Austin. [DMWP]
Pinker, S. (1989) *Learnability and cognition*. MIT Press. [rFVDV]
(1994) *The language instinct.* Penguin. [arFVDV]
(1998) *How the mind works.* Penguin. [aFVDV]
Plate, T. A. (1991) Holographic reduced representations: Convolution algebra for compositional distributed representations. In: *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, ed. J. Mylopoulos & R. Reiter, Morgan Kaufmann. Available at: http://www.d-reps.org/tplate/papers/plate.ijcai91.pdf. [RWG]
(1994) *Distributed representations and nested compositional structure.* Doctoral dissertation. Department of Computer Science, University of Toronto, Toronto, Canada. Available at: http://citeseer.ist.psu.edu/90353.html. [RWG, FTS]
(1997) A common framework for distributed representation schemes for compositional structure. In: *Connectionist systems for knowledge representation and deduction*, ed. F. Maire, R. Hayward & J. Diederich. Queensland University of Technology Press. Available at: http://citeseer.ist.psu.edu/32374.html. [RWG, rFVDV]
(2000) Randomly connected sigma-pi neurons can form associator networks. *Network: Computation in Neural Systems* 11:321–32. Available at: http://www.d-reps.org/tplate/papers/network00.pdf. [RWG]
(2003) *Holographic reduced representation: Distributed representation for cognitive structures*. CSLI. [RWG, FTS, rFVDV]
Pollard, C. & Sag, I. A. (1994) *Head-driven phrase structure grammar.* CSLI. [CP]
Port, R. F. & van Gelder, T., eds. (1995) *Mind as motion: Explorations in the dynamics of cognition.* MIT Press. [aFVDV]
Posner, M. I. & Petersen, S. E. (1990) The attention system of the human brain. *Annual Review of Neuroscience* 13:25–42. [aFVDV]
Powers, D. M. W. (1983) Neurolinguistics and psycholinguistics as a basis for computer acquisition of natural language. *SIGART* 84:29–34. [DMWP]
(1986) A psycholinguistic approach to the acquisition of natural language by computer. Doctoral dissertation, University of New South Wales. [DMWP]
(1992) On the significance of closed classes and boundary conditions: Experiments in lexical and syntactic learning. In: *Background and experiments in machine learning of natural language: First SHOE workshop*, ed. W. Daelemans & D. M. W. Powers, pp. 245–66. ITK Proceedings 92/1, Tilburg University. [DMWP]
(1997) Perceptual foundations for cognitive linguistics. Paper presented at the International Conference on Cognitive Linguistics, Amsterdam, July 18, 1997. [DMWP]
Powers, D. M. W. & Turk, C. C. R. (1989). *Machine learning of natural language.* Springer-Verlag. [DMWP]
Prabhakaran, V., Narayanan, K., Zhao, Z. & Gabrieli. J. D. E. (2000) Integration of diverse information in working memory within the frontal lobe. *Nature Neuroscience* 3:85–90. [aFVDV]
Pullum, G. K. & Scholz, B. C. (2001) More than words. *Nature* 413:367. [aFVDV]
Pulvermüller, F. (1999) Words in the brain's language. *Behavioral and Brain Sciences* 22(2):253–79. [AG, aFVDV]
(2002) A brain perspective on language mechanisms: From discrete neuronal ensembles to serial order. *Progress in Neurobiology* 67:85–111. [DMWP]
Pylyshyn, Z. W. (2002) Mental imagery: In search of a theory. *Behavioral and Brain Sciences* 25(2):157–83. [aFVDV]
Rachkovskij, D. A. & Kussul, E. M. (2001) Binding and normalization of binary sparse distributed representations by context-dependent thinning. *Neural Computation* 13:411–52. Available at: http://cogprints.org/1240/. [RWG, FTS]
Raffone, A. & van Leeuwen, C. (2001) Activation and coherence in memory processes. *Connection Science* 13:349–82. [MGD]
Raffone, A. & Wolters, G. (2001) A cortical mechanism for binding in visual working memory. *Journal of Cognitive Neuroscience* 13:766–85. [aFVDV]
Rainer, G., Assad, W. F. & Miller, E. K. (1998) Memory fields of neurons in the primate prefrontal cortex. *Proceedings of the National Academy of Science USA* 95:15008–13. [aFVDV]
Rao, S. C., Rainer, G. & Miller, E. K. (1997) Integration of what and where in the primate prefrontal cortex. *Science* 276:821–24. [aFVDV]
Riehle, A., Grun, S., Diesmann, M. & Aertsen, A. (1997) Spike synchronization and rate modulation differentially involved in motor cortical function. *Science* 278:1950–53. [DD]
Roelfsema, P. R., Engel, A. K., König, P. & Singer, W. (1997) Visuomotor integration is associated with zero time-lag synchronization among cortical areas. *Nature* 385:157–61. [DMWP]
Rolls, E. T. & Treves, A. (1998) *Neural networks and Brain Function.* Oxford University Press. [arFVDV]
Ruchkin, D. S., Grafman, J., Cameron, K. & Berndt, R. S. (2003) Working memory retention systems: A state of activated long-term memory. *Behavioral and Brain Sciences* 26(6):709–28. [aFVDV]
Rudin, W. (1976) *Principles of mathematical analysis.* McGraw-Hill. [rFVDV]
Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986) Learning representations by back-propagating errors. *Nature* 323:533–36. [aFVDV]
Sag, I. A. & Wasow, T. (1999) *Syntactic theory.* CSLI Publications. [aFVDV]
Sanford, A. J. & Sturt, P. (2002) Depth of processing in language comprehension: Not noticing the evidence. *Trends in Cognitive Sciences* 6:382–86. [rFVDV]
Sarnthein, J., Petsche, H., Rappelsberger, P., Shaw, G. L. & von Stein, A. (1999) Synchronization between prefrontal and posterior association cortext during human working memory. *Proceedings of the National Academy of Sciences of the United States of America* 95:7092–96. [DMWP]
Selfridge, O. G. (1959) Pandemonium: A paradigm for learning. In: *Proceedings of the symposium on the mechanisation of thought processes*, ed. D. V. Blake & A. M. Uttley. Her Majesty's Stationary Office. [aFVDV]
Shanahan, M. & Baars, B. J. (in press) Applying global workspace theory to the frame problem. *Cognition.* [BJB]
Shastri, L. (1999) Advances in SHRUTI – A neurally motivated model of relational knowledge representation and rapid inferencing using temporal synchrony. *Applied Intelligence* 11:79–108. [MGD, LS]
(2001a) A computational model of episodic memory formation in the hippocampal system. *Neurocomputing* 38–40:889–97. [LS]
(2001b) Episodic memory trace formation in the hippocampal system: A model of cortico-hippocampal interaction. ICSI Technical Report tr-01-00, Berkeley, CA. Available at: ftp://ftp.icsi.berkeley.edu/pub/techreports/2001/tr-01-004.pdf [LS]
(2002) Episodic memory formation and cortico-hippocampal interactions. *Trends in Cognitive Science* 6:162–68. [LS]
(in preparation) From transient patterns to persistent structures. Available at: www.icsi.berkeley.edu/~shastri/psfiles/shastri_em.pdf [LS]
Shastri L. & Ajjanagadde, V. (1993a) A step toward reflexive reasoning. *Behavioral and Brain Sciences* 16(3):477–94. Response to commentaries on Shastri L. & Ajjanagadde, V. (1993) From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences* 16(3):417–51. [LS]
(1993b) From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences* 16(3):417–51. [R-AM, DMWP, LS, arFVDV]
Shastri, L. & Wendelken, C. (2000) Seeking coherent explanations—A fusion of structured connectionism, temporal synchrony, and evidential reasoning. In: *Proceedings of the Twenty-Second Conference of the Cognitive Science Society, Philadelphia, PA*, ed. L. R. Gleitman & A. K. Joshi, pp. 453–58. Erlbaum. [LS]
Sherman, S. M. & Guillery, R. W. (2001) *Exploring the thalamus*. Academic Press. [YC]
Simonds, R. J. & Scheibel, A. B. (1989) The postnatal development of the motor speech area: A preliminary study. *Brain and Language* 37:42–58. [R-AM]
Singer, W. (2001) Consciousness and the binding problem. *Annals of the New York Academy of Science* 929:123–46. [R-AM]
Singer, W. & Gray, C. M. (1995) Visual feature integration and the temporal correlation hypothesis. *Annual Review of Neuroscience* 18:555–86. [rFVDV]
Smolensky, P. (1990) Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence* 46:159–216. Available at: http://dx.doi.org/10.1016/0004-3702(90)90007-M. [RWG, FTS]
Sommer, F. T. & Wennekers, T. (2003) Models of distributed associative memory networks in the brain. *Theory in Biosciences* 122:55–69. [FTS]
Sougne, J. P. (2001) Binding and multiple instantiation in a distributed network of spiking neurons. *Connection Science* 13:99–126. [MGD]
St. John, M. F. & McClelland, J. L. (1990) Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence* 46:217–57. [aFVDV]
Steedman, M. (2000) *The syntactic process*. MIT Press. [CP, rFVDV]
Stevenson, S. (1994) A unified model of preference and recovery mechanisms in human parsing. *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society.* 16:824–29. Erlbaum. [RFH]
Stroop, J. R. (1935) Studies of interference in serial verbal reactions. *Journal of Experimental Psychology* 18:643–62. [BD]

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press

Sumida, R. A. & Dyer, M. G. (1992) Propagation filters in PDS networks for sequencing and ambiguity resolution. In: *Advances in neural information processing systems, vol. 4*, ed. J. E. Moody, S. J. Hanson & R. P. Lippmann. Morgan Kaufmann. [MGD]

Treves, A. (2005) Frontal latching networks: A possible neural basis for infinite recursion. *Cognitive Neuropsychology* 21:276–91. [AG]

Ungerleider, L. G. & Mishkin, M. (1982) Two cortical visual systems. In: *Analysis of Visual Behavior*, ed. D. J. Ingle, M. A. Goodale, R. J. W. Mansfield, pp. 549–86. MIT Press. [BD]

van der Heijden, A. H. C., Kurvink, A. G., de Lange, L., de Leeuw, F. & van der Geest, J. N. (1996) Attending to color with proper fixation. *Perception & Psychophysics* 58:1224–37. [aFVDV]

van der Heijden A. H. C. & van der Velde, F. (1999) Cognition for selection. *Visual Cognition* 6:83–87. [aFVDV]

van der Velde, F. (1995) Symbol manipulation with neural networks: Production of a context-free language using a modifiable working memory. *Connection Science* 7:247–80. [arFVDV]

(1997) On the use of computation in modelling behavior. *Network: Computation in Neural Systems* 8:1–32. [aFVDV]

(1999) A spy to spy on a spy: From type to token representation with cell assemblies. *Behavioral and Brain Sciences* 22(2):306–307. commentary on Pulvermüller, F. Words in the brain's language. *Behavioral and Brain Sciences* 1999 22(2):253–79. [arFVDV]

(2001) The study of brain function: Methodological considerations. In: *Frontiers of life, vol. III, part two, Building blocks for intelligence systems*, ed. D. Amit & G. Parisi. Academic Press. [arFVDV]

(2005) Modelling language development and evolution with the benefit of hindsight. *Connection Science* 17:361–79. [rFVDV]

van der Velde, F. & de Kamps, M. (2001) From knowing what to knowing where: Modeling object-based attention with feedback disinhibition of activation. *Journal of Cognitive Neuroscience* 13:479–91. [arFVDV]

(2002a) Involvement of a visual blackboard architecture in imagery. *Behavioral and Brain Sciences* 25(2):213–14. commentary on Pylyshyn, Z. W. (2002) Mental imagery: In search of a theory. *Behavioral and Brain Sciences* 25(2):157–83. [aFVDV]

(2002b) Synchrony in the eye of the beholder: An analysis of the role of neural synchronization in cognitive processes. *Brain and Mind* 3:291–312. [arFVDV]

(2003) A model of visual working memory in PFC. *Neurocomputing* 52–54:419–24. [aFVDV]

van der Velde, F., de Kamps, M. & van der Voort van der Kleij, G. T. (2004a) CLAM: Closed-loop attention model for visual search. *Neurocomputing* 58–60:607–12. [aFVDV]

van der Velde, F., van der Voort van der Kleij, G. T. & de Kamps, M. (2004b) Lack of combinatorial productivity in language processing with simple recurrent networks. *Connection Science* 16:21–46. [aFVDV]

van der Voort van der Kleij, G. T., de Kamps, M. & van der Velde, F. (2003) A neural model of binding and capacity in visual working memory. *Lecture Notes in Computer Science* 2714:771–78. [aFVDV]

van Gelder, T. (1990) Compositionality: A connectionist variation on a classical theme. *Cognitive Science* 14:355–84. [aFVDV]

van Valin, R. D., Jr. (2001) *An introduction to syntax.* Cambridge University Press. [arFVDV]

von der Malsburg, C. (1981/1994) The correlation theory of brain function (1994 reprint of a report originally published in 1981). In: *Models of neural networks II*, ed. E. Domany, J. L. van Hemmen & K. Schulten. Springer. [LAAD]

(1986) Am I thinking in assemblies? In: *Brain theory*, ed. G. Palm & A Aertsen. Springer. [LAAD, rFVDV]

(1987) Synaptic plasticity as basis of brain organization. In: *The neural and molecular bases of learning*, ed. J.P. Changeux & M. Konishi. Wiley. [arFVDV]

von Stein, A., Rappelsberger, P., Sarnthein, J. & Petsche, H. (1999) Synchronization between temporal and parietal cortex during multimodal object processing in man. *Cerebral Cortex* 9:150. [DMWP]

Webelhuth, G., ed. (1995) *Government and binding theory and the minimalist program.* Blackwell. [aFVDV]

Wehr, M. & Laurent, G. (1996) Odour encoding by temporal sequences of firing in oscillating neural assemblies. *Nature* 384:162–66. [DD]

Weiss, S., Muller, H. M. & Rappelsberger, P (2001) Theta sychronization predicts efficient memory encoding of concrete and abstract nouns. *NeuroReport* 11:2357–61. [DMWP]

Wendelken, C. & Shastri, L. (2004) Multiple instantiation and rule mediation in SHRUTI *Connection Science* 16:211–17. [LS]

Whitney, C. (2004) Investigations into the neural basis of structured representations. Doctoral dissertation, University of Maryland, College Park, MD. Available at: http://www.cs.umd.edu/~shankar/cwhitney/. [CW]

Willshaw, D. J., Buneman, O. P. & Longuet-Higging, H. C. (1969) Non-holographic associative memory. *Nature* 222:960–62. [FTS]

Wilson, F. A. W., Ó Scalaidhe, S. P. & Goldman-Rakic, P. S. (1993) Dissociation of object and spatial processing domains in primate prefrontal cortex. *Science* 260:1955–58. [aFVDV]

Wilson, H. R. & Cowan, J. D. (1972) Excitatory an inhibitory interactions in localized populations of model neurons. *Biophysical Journal* 12:1–24. [aFVDV]

Wilson, W. H. & Halford, G. S. (1994) Robustness of tensor product networks using distributed representations. In: *Proceedings of the Fifth Australian Conference on Neural Networks (ACNN'94), Brisbane, Australia*, ed. A. C. Tsoi & T. Downs, pp. 258–61, University of Queensland. Available at: http://www.cse.unsw.edu.au/~billw/reprints/wilson.lobotomy.pdf. [RWG]

Yakovlev, V., Fusi, S., Berman, E. & Zohary, E. (1998) Inter-trial neuronal activity in inferior temporal cortex: A putative vehicle to generate long-term associations. *Nature Neuroscience* 1:310–17. [aFVDV]

https://doi.org/10.1017/S0140525X06329020 Published online by Cambridge University Press