

A Neural Model of Compositional Sentence Structures

Frank van der Velde

Cognitive Psychology, Leiden University
Wassenaarseweg 52
2333 AK Leiden, The Netherlands
vdvelde@fsw.leidenuniv.nl

Abstract

A neural architecture for compositional sentence structures is presented. The architecture solves the ‘four challenges for cognitive neuroscience’ described by Jackendoff (2002). Sentence structures are encoded in this neural architecture by temporarily binding word representations with structure representations in a manner that preserves sentence structure. The architecture can store different sentence structures simultaneously. Answers to specific ‘who does what to whom’ questions can be produced by means of a selective activation process within the architecture. The architecture can account for effects of sentence complexity.

Introduction

Jackendoff (2002) analyzed the difficulties given by the neural instantiation of linguistic structures (‘four challenges for cognitive neuroscience’), in particular the binding problem in language and the problem of multiple instantiation (the ‘problem of 2’).

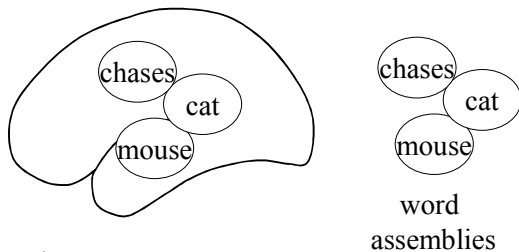


Figure 1

Figure 1 provides an illustration. A word like *cat* will activate the same neural structure (or ‘word assembly’) in *The cat chases the mouse* and *The mouse chases the cat*. This raises the question of how the multiple instantiations of *cat* (and *mouse*) are distinguished in both sentences (the problem of 2) and how *cat* and *mouse* are bound correctly to *chases*, without creating the incorrect bindings of *cat chases cat* and *mouse chases mouse* (the binding problem).

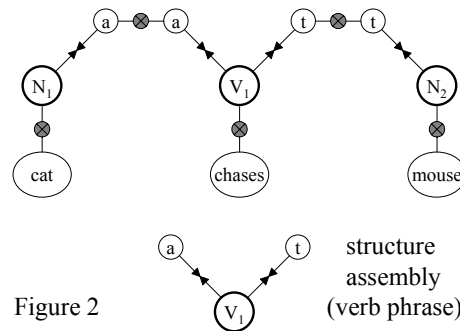


Figure 2

Figure 2 illustrates a solution of these problems, in which neural ‘word assemblies’ are embedded in a neural architecture consisting of neural ‘structure assemblies’ and neural ‘memory circuits’ used for binding. A structure assembly consists of a ‘main assembly’ and a number of ‘subassemblies’. Two kinds of structure assemblies are shown: verb-phrases (VPs) and noun-phrases (NPs), with agent (a) and theme (t) subassemblies. The subassemblies are connected to the main assembly by gating circuits, which can be activated when certain structural control conditions are met. During syntactic processing, word and structure assemblies are bound to one another by activating memory circuits that connect the assemblies.

Figure 3 illustrates a gating circuit, consisting of a ‘disinhibition’ circuit that controls the activation flow between two assemblies (X,Y). X activates an inhibitory neuron i_x , which blocks Y’s activation. When an external control circuit activates I_x to inhibit i_x , X can activate Y. A similar circuit controls Y’s activation of X.

Memory circuits temporarily bind assemblies. Figure 3 shows how X binds to Y (a similar circuit binds Y to X). Memory circuits are gating circuits with control given by a ‘delay’ assembly, which is activated when X and Y are simultaneously active (see Van der Velde 2003). As long as the delay assembly is active, due to self-sustained (reverberating) activation, activation can flow between X and Y, which temporarily binds (merges) these assemblies.

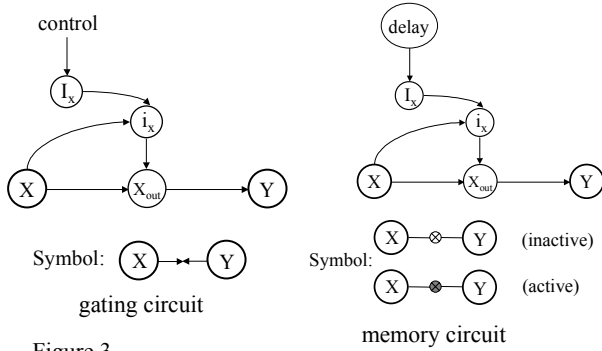


Figure 3

Figure 2 illustrates the neural structure of *cat chases mouse* in the architecture described here. The assembly for *cat* is bound by an active memory circuit to an arbitrary NP assembly N_1 . Likewise, *mouse* is bound to N_2 , and *chases* is bound to a VP assembly V_1 . N_1 is then bound to V_1 by means of their agent subassemblies, and N_2 is bound to V_1 by their theme subassemblies. The binding process is controlled by neural circuits (not presented here), that interpret *cat* as the agent of *chases* (activating the gating circuits for ‘agent’), and *mouse* as the theme of *chases* (activating the gating circuits for theme).

Overview of the Architecture

Figure 4 presents an overview of a neural architecture for neural sentence structure (in particular for verb-argument binding). Each word assembly that encodes a noun is connected to the main assembly of each NP assembly by means of a memory circuit, which is initially inactive. In the same manner, each word assembly that encodes a verb is connected to the main assembly of each VP assembly by means of an (initially inactive) memory circuit. The main assembly of each NP or VP assembly is connected to subassemblies by means of gating circuits (i.e., each NP or VP assembly has its own set of subassemblies, as illustrated with V_1 in figure 2). Main assemblies and subassemblies are also delay assemblies, in the sense that they can remain active on their own. Subassemblies are used to represent thematic roles, such as agent or theme as illustrated in figures 2 and 4. They can also be used to represent syntactic structures such as complements or relative clauses (as discussed later on). Subassemblies can be used to represent thematic roles or syntactic structures, because they are used to connect the NP and VP assemblies. Thus, all agent subassemblies of the NP assemblies are connected to all agent subassemblies of the VP assemblies, by means of (initially inactive) memory circuits. Likewise for the other kinds of subassemblies.

There is also an inhibitory interaction between the VP main assemblies, as illustrated with the horizontal bar in Figure 4. This results in a competition between VP main assemblies. Likewise for NP main assemblies.

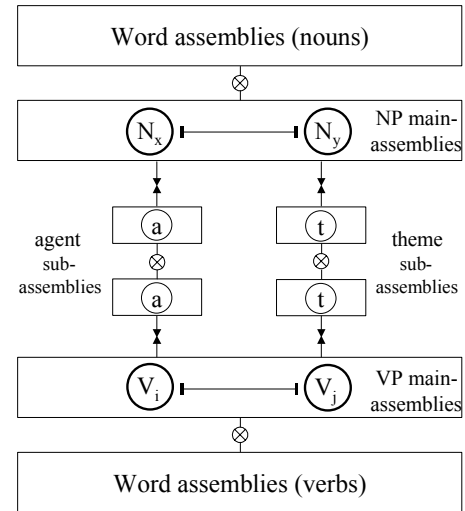


Figure 4

The ability to retrieve information (i.e., to answer ‘who does what to whom’ questions) in this architecture critically depends on the competition between main assemblies.

Multiple Instantiation

Figure 5 shows the simultaneous encoding of the sentences *The cat chases the mouse*, *The mouse chases the cat* and *The cat bites the dog* in the architecture presented in figure 4. The neural structure of *The mouse chases the cat* in figure 5 is the same as in figure 2.

The words *cat*, *mouse* and *chases* occur in more than one sentence in figure 5, and, in the case of *cat* and *mouse*, in more than one thematic role. This creates the problem of the multiple instantiations of the words *cat*, *mouse* and *chases* (the problem of 2). Figure 5 illustrates how the problem of multiple instantiation is solved in the architecture presented in figure 4. Each word in a sentence is encoded by binding its word assembly to a unique structure assembly. For instance, the word assembly for *cat* is bound the NP assemblies N_1 , N_4 and N_5 in figure 5. These different NP assemblies bind *cat* in the different sentences involved. In this way, *cat* can be bound as the agent in one sentence (by N_1 or N_5) and as theme in another (by N_4). Thus, the different NP assemblies encode *cat* as different tokens of the same type. Similarly, the different VP assemblies (V_1 and V_2) encode *chases* as different tokens of the same type.

As illustrated in figure 1, the sentences presented in figure 5 cannot be encoded in terms of direct associations between word (noun and verb) assemblies. For instance, the association of *cat-chase-mouse* does not distinguish between the sentences *The cat chases the mouse* and *The mouse chases the cat*, because *cat* and *mouse* are not encoded as agent or theme in these associations.

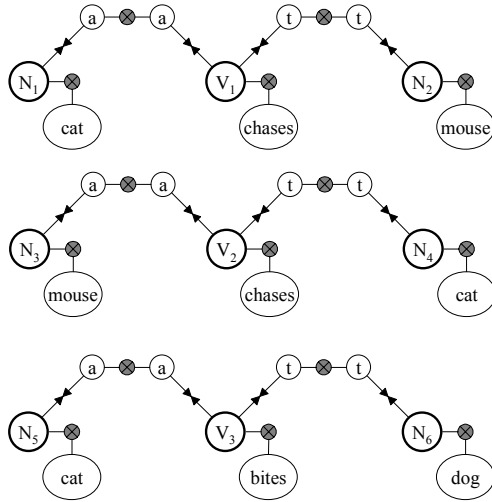


Figure 5

The internal structure of the NP and VP assemblies, given by the gating circuits, is of crucial importance in this respect. Without this internal structure, the neural sentence structures presented in figure 5 would also consist of direct associations between neural assemblies, which would create the same problems as described above, such as the failure to distinguish between *The cat chases the mouse* and *The mouse chases the cat*.

Answering Binding Questions

With the control of activation provided by gating circuits, the neural structures of encoded sentences can be selectively (re)activated. Selective reactivation of a sentence structure in figure 5 is necessary to retrieve information from the architecture. In particular, to answer specific binding questions, such as the question “Whom does the cat chase?”. The question provides the information that *cat* is the agent of *chases* and it asks for the theme of *chases* in that sentence (i.e., it asks for *x* in the sentence *cat chases x*). The production of the answer consists of the selective activation of the word assembly for *mouse*. This can be achieved through a competition process between the VP assemblies, in which V_1 emerges as the winner. After that, the activation of the gating circuits for theme will result in the activation of N_2 by V_1 , and thus in the activation of *mouse* as the answer to the question.

The competition between the VP assemblies is determined by the information provided by the question, which results in the activation of the assemblies for *cat* and *chases*, and the gating circuits for agent. Inspection of figure 5 shows that V_1 receives the most activation in this case, because it is activated by *chases* and N_1 (through *cat*), whereas V_2 is only activated by *chases* and V_3 is only activated by N_5 (through *cat*). The mutual inhibition between VP main assemblies (figure 4) will then result in

V_1 as the winner of the VP competition. This analysis has been confirmed by pilot simulations, using populations of spiking neurons to simulate the assemblies (Van der Velde and de Kamps 2003).

Extension of the Architecture

The architecture solves the problem of 2 because it can, e.g., bind *cat* to different NP assemblies (N_1 , N_4 and N_5 in figure 5) in different sentence contexts. It solves the binding problem, because NP and VP assemblies can bind through different subassemblies. In this way, *cat* can be the agent of *chases* in one sentence and the theme of *chases* in another. Through selective activation, resulting from gating and memory circuits, questions like “Whom does the cat chase?” or “Who chases the mouse?” can be answered correctly. It can be shown that the architecture presented here provides a solution to all of Jackendoff’s challenges (see Van der Velde 2003).

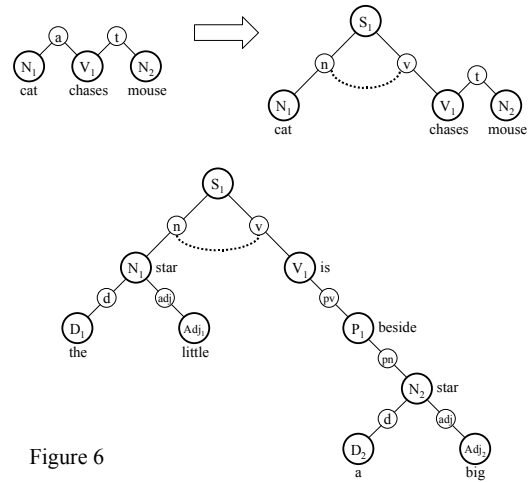


Figure 6

Extension of the architecture is introduced in figure 6. Figure 6 (top-left) shows a shorthand presentation of *cat chases mouse* (omitting gating and memory circuits, writing words below their structure assemblies). However, the full structure (figure 2) is always implied: main assemblies are connected to subassemblies by gating circuits; subassemblies are bound (merged) by active memory circuits.

Figure 6 (top-right) shows a more elaborate (shorthand) structure of *cat chases mouse*, using a sentence structure assembly (S_1) and noun and verb subassemblies. N_1 (*cat*) is bound to S_1 by noun subassemblies, and V_1 (*chases*) is bound to S_1 by verb subassemblies. Interaction between noun and verb subassemblies (dotted line) controls agreement between noun and verb (e.g., *cat-chases* vs. *cats-chase*).

Figure 6 (bottom) shows the (shorthand) structure of *The little star is beside a big star* (Jackendoff 2002), with structure assemblies and subassemblies for adjective,

preposition, and determiner. Different structure assemblies and subassemblies can be introduced for other syntactic features (Van der Velde 2003), as illustrated below.

Embedded Clauses and Complexity

Figure 7a presents the structure of the sentence *The cat that bites the dog chases the mouse*, with 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 figure 7a). The word assemblies of complementizers can bind with C assemblies, as illustrated with *that* in figure 7a. Like S assemblies, C assemblies can be used to encode agreement between subject and verb (here, between the subject of the main sentence and the verb of the relative clause).

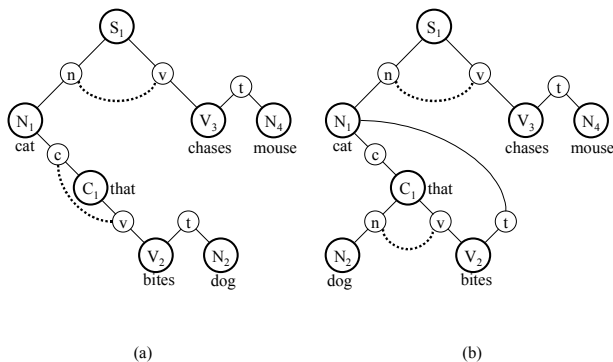


Figure 7

Figure 7b presents the structure of the sentence *The cat that the dog bites chases the mouse*, with the object-relative clause *that the dog bites*. Here, *dog* is the subject of *bites*, so it is bound to the noun subassembly of C_1 , and agreement between *dog* and *bites* is encoded by agreement between the noun and verb subassemblies of C_1 . In an object-relative sentence like *The cat that the dog bites chases the mouse*, the subject of the main sentence is the theme of the verb in the relative clause. In the structure presented in figure 7b, this is instantiated by the binding of *cat* (N_1) with *bites* (V_2) by means of their theme subassemblies. This 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 theme subassembly of N_1 has to be activated before the activation of N_2 . Thus, the gating circuits for theme have to be activated before the activation of the main assembly of N_2 .

Figure 8a presents the structure of *The cat that the dog that the boy likes bites chases the mouse* (1), with the double center-embedded object-relative clause *that the dog*

that the boy likes bites. Sentences of this type are notoriously hard to process. The phrase *the dog that the boy* is structurally similar to the phrase *the cat that the dog*, so that the theme subassembly of N_2 (*dog*) will be activated as well to bind with the theme subassembly of the next verb. Thus, when the first verb (*likes*) appears, there are two subassemblies that can bind with the theme subassembly of this verb, whereas the verb should bind with *dog* (N_2) as its theme argument. The situation is similar with the second verb (*bites*). The two problematic bindings are indicated with the dashed lines in figure 8a.

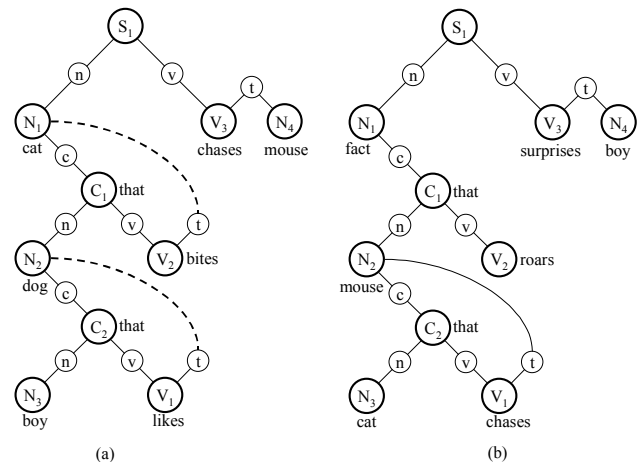


Figure 8

Figure 8b shows the structure of the sentence *The fact that the mouse that the cat chases roars surprises the boy* (2). Here, *roars* (V_2) does not have a theme argument, and *mouse* (N_2) will bind with the first verb (*chases*) as its theme (see Van der Velde 2003). 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 that the ambiguities in (1) do not arise in (2). This example shows that the architecture can account for effects of sentence complexity.

References

- Jackendoff, R. (2002). *Foundations of language*. OUP
- Van der Velde, F (2003). Neural blackboard architectures of combinatorial structures in cognition. http://cogprints.ecs.soton.ac.uk/secure/00003196/01/neural_archs.pdf
- Van der Velde, F. and de Kamps, M. (2003). Neural assembly binding in linguistic representation. *European Symposium on Artificial Neural Networks*, Evre: d-side.