Title: Syntax through rapid synaptic changes

Authors: Lin Sun[1]*, Sanjay G. Manohar[2]

Affiliations:

[1]Queen Square Institute of Neurology, University College London; London, WC1N 3BG, United Kingdom.

[2]Nuffield Department of Clinical Neurosciences, University of Oxford; Oxford, OX3 9DU, United Kingdom.

*Corresponding author. Email: lin.sun1@nhs.net

Abstract: Syntax is a central organizing component of human language but few models explain how it may be implemented in neurons. We combined two rapid synaptic rules to demonstrate how neurons can implement a simple grammar. Words bind to syntactic roles (e.g. "dog" as subject or object) and the roles obey ordering rules (e.g. subject → verb → object), guided by predefined syntactic knowledge. We find that, like humans, the model recalls sentences better than shuffled word-lists, and it can serialize words to express an idea as a sentence. The model also supports order-free morphemic languages, exhibits syntactic priming and demonstrates typical patterns of aphasia when damaged. Crucially, it achieves these using an intuitive representation where words fill roles, allowing structured cognition.

Main Text: Why is it easier to remember "colorless green ideas sleep furiously", than "furiously sleep ideas green colorless"? Although the sequence of words is itself new, something about the order of word *types* is familiar. In a grammatically correct sentence, each word has a role, which makes it more likely to be followed by a word with another particular role. This underlying syntactic representation is key both in language comprehension (*1*, *2*), and production (*3*, *4*). However, we do not yet have an understanding of how the brain encodes syntax, which may itself be an important mechanism for reasoning and high-level cognition (*5*, *6*).

Currently, no dynamic neural model can capture the distinction between the *structure* and *content* of thoughts. Specifically, how do brain circuits capture the fact that when we think, our sequence of thoughts has a structure that obeys a set of rules, while the contents are highly flexible?

Sequences can be understood as forward associations between items (*7*), but in the case of syntax, it is the word-*roles*, not words themselves, that have sequential structure. We hypothesize that working memory links each word to its role in the sentence, and those roles follow probabilistic orderings. Although this cannot account for true grammar (*8*), since it does not support hierarchical structure (*9*), it provides a biologically plausible way to understand some aspects of syntax. In this paper we propose a neural architecture that stores and generates syntactic sequences.

Many neural models of working memory can hold sequential information, storing either timing (*10*), order (23), or production rules (*11–13*). Word sequences may form forward associations between words (*14*, *15*) with short-term synaptic plasticity (*16*) or by associating each word with a "temporal context" (neurons that represent the current time) in a "phonological loop" (*17*, *18*). But these models ignore the rules that assign words to their structural roles.

To account for syntax, sentences can be encoded using neural states (*19*, *20*) or neurons that code for word-and-role combinations (*21*). However these approaches suffer from combinatorial explosion, e.g. requiring duplicate neurons to allow encoding "dog" as either subject or object. One option is to interface neural networks with an external symbolic system (*22*). Fully neural systems include transformer-based large language models, which rely on operations that are difficult to implement using biological neurons. Recurrent neural networks may be more biologically plausible (*10*, *23*) and can represent some language structures (*24–26*), but still lack an intuitively symbol-like architecture (*27*) (Supplementary background).

We propose that words and their roles are coded by two populations of neurons. Each role within a sequence is associatively bound to a different word – or equivalently, each word is tagged with a different syntactic role. While neurons in superior temporal cortex are selective for the identity of auditory stimuli (*28*), neurons in prefrontal cortex encode abstract categorization of input (*29–32*) and might therefore represent word roles (*33*). We implement this role tagging with rapid synaptic changes that can store *bindings* between words ("fillers") and their roles. Since synapses vastly outnumber neurons, the combinatorial binding problem is solved.

We harness an idea first proposed for working memory, where contents are rapidly bound to 'slots' using a population of competing, flexible neurons (*34*). Here in the current model, words are bound to roles in a similar way. We use a highly simplified, hand-wired network to simulate hearing sentences, and also producing them from unordered words.

How our model works

Our approach is to tackle the minimal case, using just a handful of neurons, with hand-wired synapses and rapid plasticity. There is no training or training data, since the connections are so simple, and are pre-specified for a given language.

Word-selective neurons and role-selective neurons form a recurrent network (**Fig. 1A**). The two types of neuron are bidirectionally connected, and role neurons also have directed connections to each other (**fig. S1**). To flexibly encode a new sentence, word-selective neurons are activated in sequential order. These words drive role-selective neurons according to long-term knowledge of word classes encoded in synapses between word and role neurons. A role may be filled by one of several words (e.g. a subject noun could be "dog" or "cat"), and a given word could play different roles (e.g. "dog" could be a subject noun or object noun). In addition to this drive from words, role neurons drive each other. Long-term knowledge between role neurons provides a tendency to follow a familiar order (e.g. subject → verb → object) forming a Markov chain. Together these two constraints determine the role associated with each word.

Superimposed on the long-term synapses are short-term, rapidly changing weights. These weights between the word and its role, and between a role and its successor, are rapidly strengthened via a Hebbian rule, encoding the sequence of word-role pairs into working memory. The temporarily strengthened connections generate a *plastic attractor* – a partly-stable sequence of states of the network, that allows the subsequent re-activation (i.e. recall) of the same sequence of words and roles. Our model runs in continuous time and is robust to the timing of inputs, maintaining information across delays using these partly-stable states.

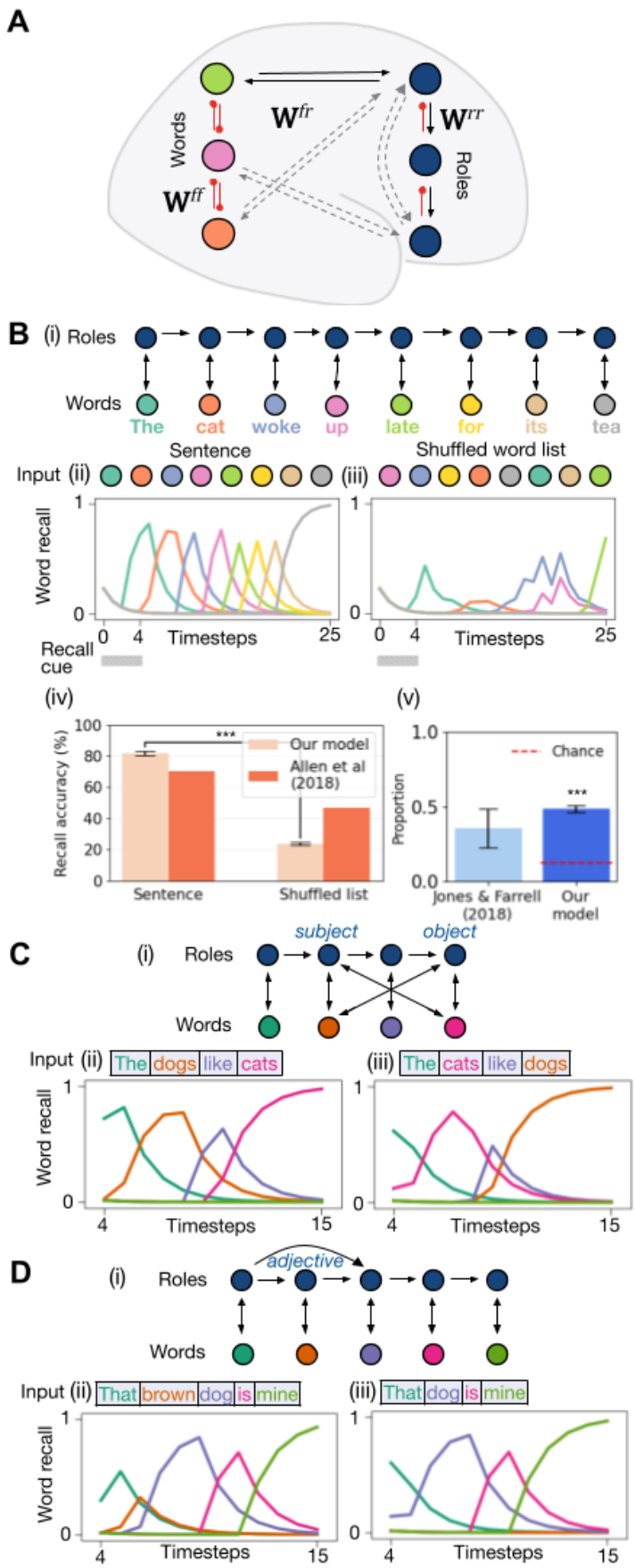

Fig. 1. Simulations of simple grammar with our model. (**A**) Architecture of our model comprising word and role-selective neurons in a rapidly reconfigurable neural network. The word neurons compete with each other via fixed interactions (W^{ff}), but connect reciprocally to and from the role neurons (W^{fr}). The role neurons also compete amongst themselves (W^{rr}), with some role sequences being prepotent. (**B**) Simulation 1: the trivial case of non-branching grammar (i). The model receives a recall cue to generate the input - either a (ii) syntactic sentence or a (iii) shuffled word-list. (iv) The model demonstrates better recall accuracy for a sentence compared to a shuffled word-list, comparable to human data from Allen et al (2018). (v) In fact, recall of the shuffled word-list tends to produce syntactic word pairs ("grammaticalization errors") above chance, comparable to human data from Jones & Farrell (2018). (**C**) Simulation 2: (i) The network can sequentially encode two sentences with the same syntax but different words. Both nouns "dogs" and "cats" are able to bind to either the subject or object role neurons. (ii) Recall of the first sentence input where "dogs" is the subject and "cats" is the object. (iii) Subsequent recall of the second sentence, where "dogs" and "cats" switched positions. (**D**) Simulation 3: (i) The network can encode sentences with different syntactic structures. (ii) The network encodes and recalls a first sentence with an adjective, and then (iii) a sentence without an adjective. *p<0.05; **p<0.01; ***p < 0.001, error bars are standard error of mean (SEM). Black arrows: connections with predefined long-term knowledge. Red rounded arrow: inhibitory connections.

Results

Simulating simple grammar in working memory

We begin with the simplest possible syntax, and extend it incrementally to evaluate the model's core capabilities.

First we simulate a grammar with only one possible syntax, and where only one word can fulfill each role (**Fig. 1B**). As expected, the model recalls syntactic sentences with greater accuracy than a shuffled word-list. The pre-existing long-term syntactic knowledge means that role neurons are activated in order. However with a shuffled list, the role neurons receive two conflicting drives: the word neurons activate their corresponding roles, but this does not match the natural tendency of the role-to-role synapses to generate a grammatical sequence. Consequently, errors occur, resembling human data (*35*). Furthermore, when humans recall shuffled-order sentences, they make "grammaticalization" errors where adjacent words are more likely to fit with grammar than chance (*1*). Our model gives rise to this because role neurons tend to activate in grammatical sequences (supplementary text).

Next we show that the model can store which word is bound to each role (**Fig. 1C**). Two word neurons can compete to drive one role neuron, for example either "dogs" or "cats" could be the subject. Similarly, a single word could fill one of two possible roles, e.g. "dogs" may play either the subject or object role in the sentence. To remember which word fills which role, the word-to-role synapses rapidly strengthen to create a new attractor state, potentially overwriting other connections of those units.

Finally, we demonstrate that the network can remember which of two syntactic structures was presented, by rapid plasticity between the role neurons (**Fig. 1D**). For example, the chain of role neurons can branch as adjectives can be skipped. The network remembers which particular sentence structure was used by strengthening the role-to-role synapses. This channels the sequence of role neurons during recall. If motifs from both simulations 2 and 3 are combined, the network simultaneously remembers both the words' roles and the branches taken (**fig. S2A**).

Since the model allows multiple role sequences and multiple roles for a word, ambiguity naturally arises. The long-term strengths of word-to-role synapses determine the probability of a word being used in a particular role, and simultaneously, the long-term strengths of role-to-role synapses determine the transition probabilities to the next role, given the history. The competition results in parsing of sentences with soft constraint satisfaction (*36*) (**fig. S2B**). The disambiguation of word class based on role-sequence is also supported by evidence from magneto-encephalography studies (*37*) (**fig. S2C**).

Morphemic languages with flexible word order

In many languages, word order is not fixed. Instead, morphemes (parts of words) provide cues to a word's role. It turns out that no extra machinery is needed for our network to model this. The word neurons function as roots (e.g. "lik-" in liking) allowing them to fill a range of roles depending on affixes (e.g. "-ing", "-able") (*38*). We add morpheme neurons that function like word neurons, but are distinct from words as they need to be concurrently active without mutually inhibiting word neurons. Each morpheme neuron is paired with a role neuron via a long-term, fixed connection (**Fig. 2A**, bold solid arrows). During encoding, if a word requires a morpheme to determine its role, we activate both the word *and* morpheme

neurons simultaneously. The morpheme dictates the syntactic roles attached to each word (overriding word order effects) by directly driving the corresponding role neuron.

In simulations 4-6, the network encodes and recalls sentences in a language with no constraint on word order (**Fig. 2B-D**). Due to rapid changes in the role-to-role synapses, the network recalls words in the same order they were encoded, even without fixed role-ordering. Across languages, this architecture can flexibly accommodate a wide range of possible structures, from morphemic to order-based, including situations where morphemes are paired only to some words.

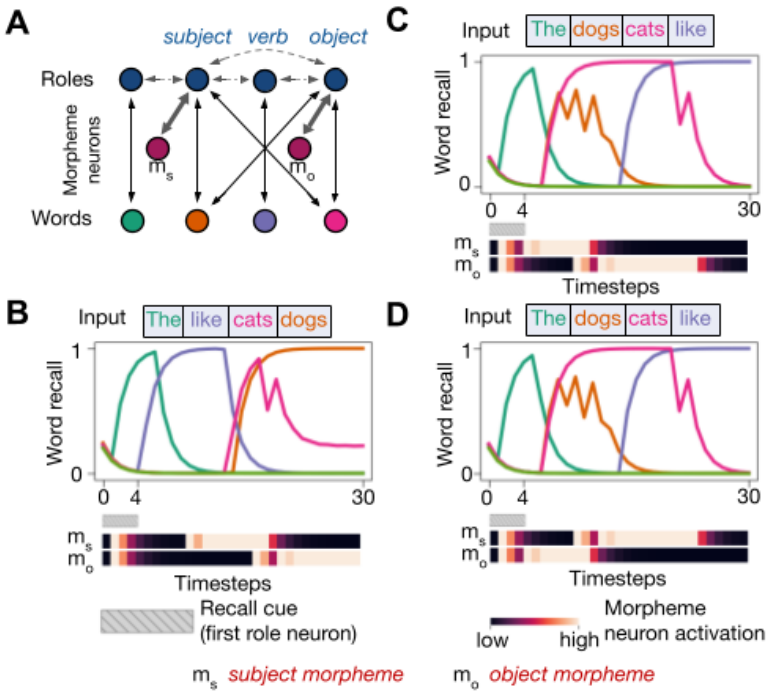

Fig. 2. Morphemes cue word roles. (**A**) In this network, morphemes accompany word input in a language with no long-term knowledge of syntactic structure. Each role neuron has no bias to follow any other role neuron (broken arrows, not all shown). Therefore, the order in which the words are presented during the encoding phase will determine their order during recall. Bold solid arrows: fixed strong connection between pairs of morpheme and role neurons. (**B**) Simulation 4: Encoding and recall of an input sentence with the structure "verb-subject-object". (**C**) Simulation 5: Same for "subject-object-verb". (**D**) Simulation 6: Same for "object-subject-verb" but with the nouns in the same serial position as **C**.

Sentence generation: syntactic serialization

A central feature of human cognition is that we can express ideas. In particular, we arrange words into ordered sentences, to express an idea held in an internal semantic representation. We demonstrate that our model is able to serialize a "bag of words" that are fed simultaneously to the network (**Fig. 3A,B**). Simultaneously activated words compete with each other, and the order in which they become active during recall is constrained by role-to-role synapses. This leads to the idea being expressed as a grammatical sequence of words. The model even inserts function words that were not present in the input (**Fig. 3C**). To achieve this, a 'conceptual' input must be added to the word neurons (equation S1), and role-to-role weights must be stronger to promote automatic sequential word activation.

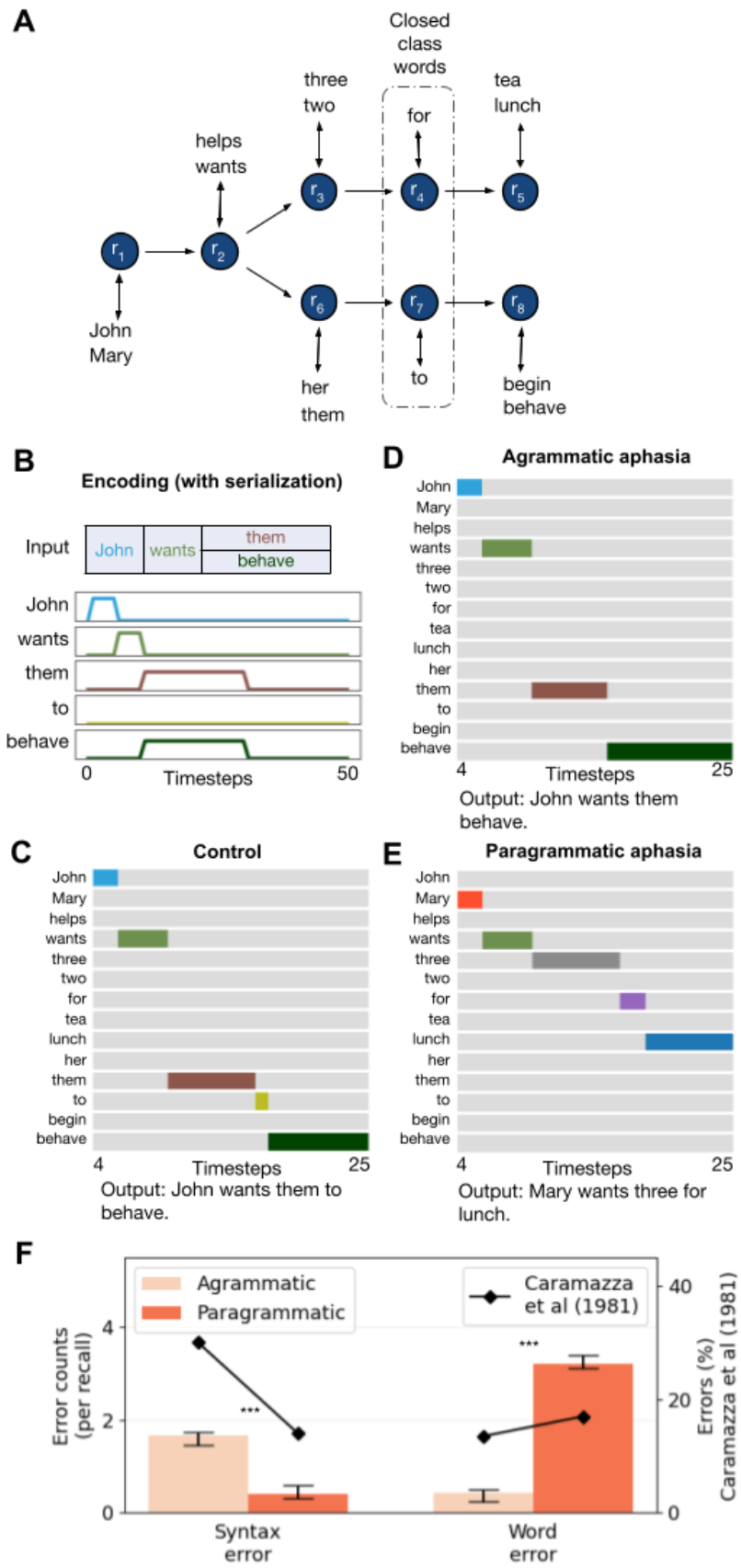

Fig. 3. Sentence generation and aphasia. (**A**) Network architecture for simulating syntactic serialization and agrammatic and paragrammatic aphasias. (**B**) During the encoding phase, "them" and "behave" are fed into the network simultaneously without the closed-class word "to". (**C**) Simulation 7: The network serializes the word inputs, adding the missing closed-class word, to form a syntactic sentence. (**D**) Simulation 8: The agrammatic lesion results in omission of the closed-class word while the message is preserved. Coloured maps of word activation represent the strongest activated word of each timestep. (**E**) Simulation 9: In paragrammatic aphasia, grammatical structure is preserved but there are word substitution errors. (**F**) Frequency of syntax errors versus word errors in agrammatic and paragrammatic aphasias produced by the model (simulations 8 and 9), compared to data from humans. Agrammatic and paragrammatic simulations include noise, therefore more example outputs are presented in **fig. S3**.

Neurological disorders can disrupt language production. Strikingly, some patients lose grammar without losing content (agrammatic aphasia) whereas others lose contents of speech with preserved grammar (paragrammatic aphasia). A key characteristic of agrammatism, often seen in Broca's aphasia, is a breakdown in sentence structure despite the correct words being generated (*39*). Since we model only one neuron per role, to simulate lesions we added noise, corresponding to weaker representational strength. To model agrammatic aphasia, we introduced noise into rapid synaptic plasticity between role neurons. We further assumed that certain roles can only be filled by a limited set of words (closed-class words, e.g. prepositions). The model produces errors characteristic of agrammatic aphasia (**Fig. 3D**) (*40*, *41*).

In a complementary syndrome of jargon or paragrammatic aphasia, sometimes seen in Wernicke's aphasia, speech is fluent and often grammatical but is semantically incoherent with word substitution errors. If noise is added to the long-term word-to-role knowledge with reduced rapid synaptic plasticity between word and role neurons, we find that the model produces these kinds of errors (**Fig. 3E**). These results qualitatively match the archetypal neuropsychological distinctions (**Fig. 3F**) (*42*).

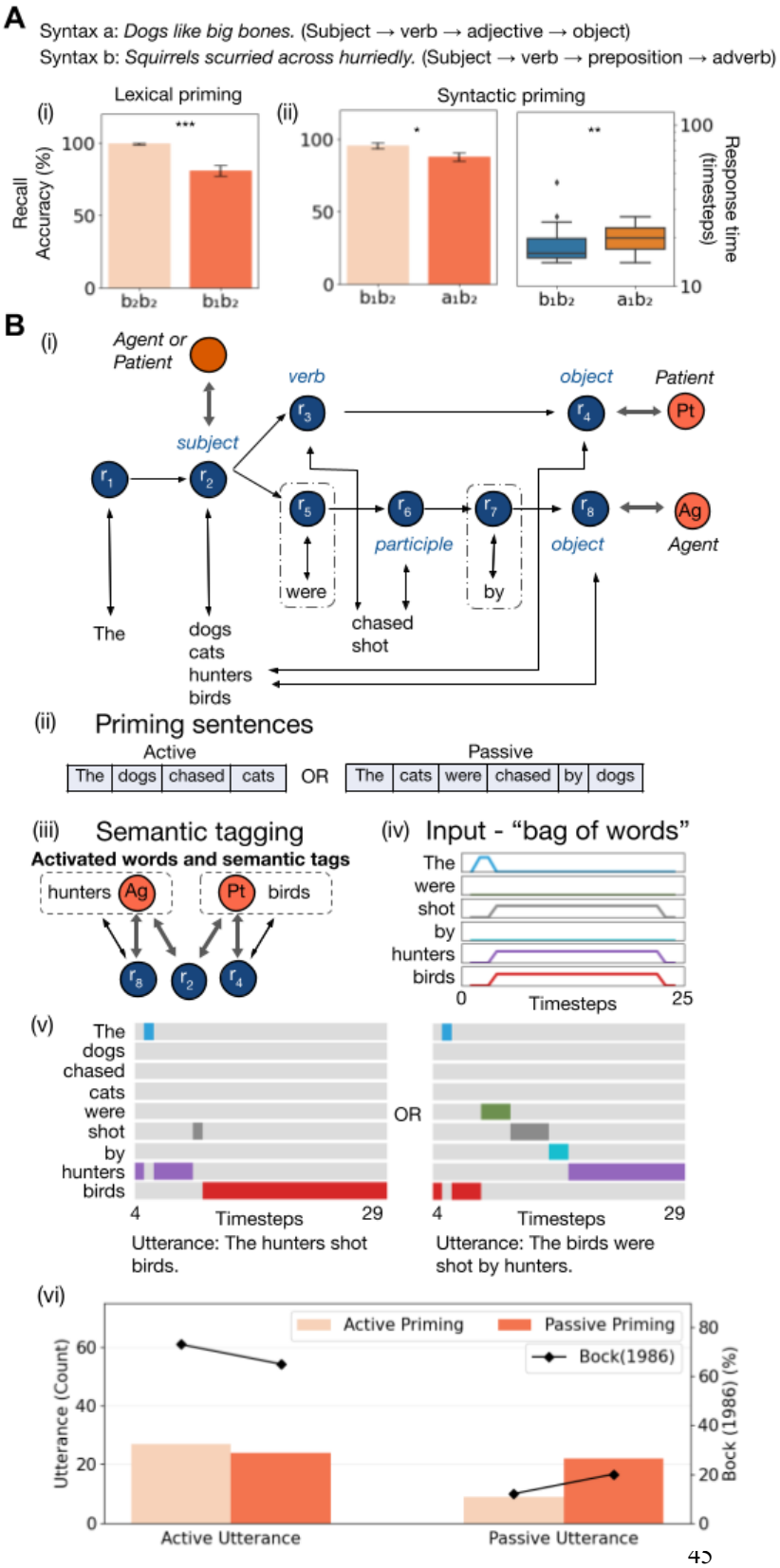

Fig. 4. Synaptic changes predict priming. (**A**) Lexical and syntactic priming during comprehension. (i) Simulation 10: Lexical priming - recall accuracy improves on repeated encoding of the same sentence. (ii) Simulation 11: Syntactic priming - recall accuracy is reduced following encoding of a syntactically distinct sentence. Response time is also increased. Different syntactic structures are coded with letters, a or b. Numerical subscripts (a_1, a_2 etc) represent different sentences with non-overlapping words. More lexical and syntactic priming results are presented in **fig. S4**. ns non-significant; *P-value<0.05; **P-value<0.01; ***P-value < 0.001, error bars are SEM. Box plot shows interquartile range. (**B**) Simulation 12: Syntactic priming during production. (i) To encode active and passive sentences, the network has two branching pathways through the role neurons. There are two object role neurons, each with a semantic tag (one for agent, and the other, patient). The subject can either be an agent or patient. (ii) The network is first presented with an active or passive priming sentence. (iii) Then on encountering a mental idea, both nouns are semantically associated with the relevant agent or patient tag. (iv) "Bag of words" without closed-class words (i.e. "were" and "by") were fed simultaneously to the network to allow syntactic serialization. (v) The network generates either the active or passive voice. (vi) The syntactic priming effect qualitatively matches human data.

Lexical and syntactic priming

Synaptic changes also provide immediate explanations for both lexical and syntactic priming. In the model, lexical priming occurs when exposure to the same item improves retrieval and syntactic priming improves recall accuracy when target and priming sentences have the same syntax (**Fig. 4A**), similar to human priming (*3*, *4*, *43–45*). The network also takes longer to repeat a sentence when the syntax changes (*46–49*). Intuitively, this is because encoding a sentence creates stable attractor states, which make the network less effective at shifting to new states, yielding priming for both words in their roles, and also sequences of roles.

Syntactic priming can also bias language production (**Fig. 4B**) (*3*). The network is first primed with a sentence in either the active or passive voice. Then, the content elements in a scenario are pre-associated to their appropriate semantic roles (i.e. nouns to 'agent' or 'patient') via semantic tags. The semantic tags behave just like the morpheme neurons, preferentially driving different role neurons. Finally, the words necessary for forming sentences are simultaneously fed to the model as a “bag of words”. The roles activate sequentially with a preference for the primed order, generating words in the order appropriate for that syntactic construction. This reproduces the empirical finding that a syntactic prime biases language production.

Semantic tags can also guide the flow of role neurons, for example allowing the phenomenon of “affix hopping” where semantic tags can control morphemes after the word to which they apply (**fig. S5**).

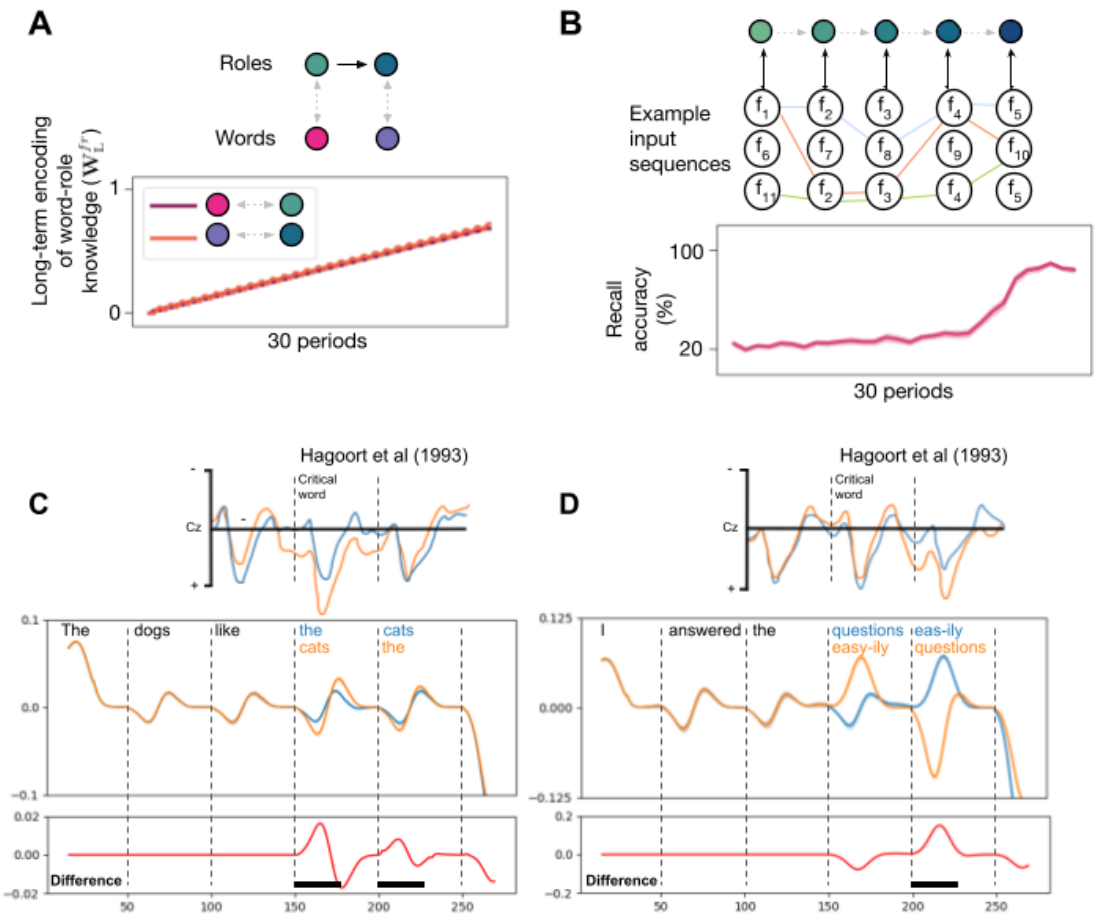

Fig. 5. Long-term acquisition of word-class and syntactic knowledge. (**A**) Simulation 13: Long-term acquisition of word class with pivot grammar. The role neurons parse only two-word sentences. The role neurons are activated sequentially, in tandem with the word input. This results in gradual learning between pairs of word and role neurons, representing long-term knowledge that some words appear in the first role, and others in the second. (**B**) Simulation 14: Long-term acquisition of syntactic structure. The network receives randomly generated sentences with the same syntax over 30 periods (three possible words per role), and long-term knowledge between role neurons is learnt from the resulting order of activation of the role neurons. After each learning period recall accuracy is tested. Gray broken arrows indicate lack of predefined long-term knowledge. (**C**) Simulation 15: Simulated event-related potentials (ERPs) from a syntactic vs non-syntactic sentence. Upper panel shows data from an ERP study, and the lower panel is the total neural activity in the model, passed through a delay, time derivative and smoothing filter. Larger neural responses seen after a syntactic violation resembles positive shifts in human ERPs (black bar). (**D**) Simulation 16: Simulated ERP from correct vs incorrect morpheme conditions. Positive shift in total neural activity in the model matches positive shift observed in ERP study (black bar). The networks are presented in **fig. S7**. Error band is SEM.

Learning a language

So far, we assumed that long-term word-class and role-to-role knowledge are already pre-learned and that plastic attractors are created *within* the existing knowledge structure. Next we challenge the model to acquire syntactic knowledge using slow plasticity, by simply adding long term plasticity that mirrors short term changes.

First we ask whether the model can learn the potential roles of each word. To demonstrate this, we simplify the grammar with just two roles (**Fig. 5A**). This could correspond to a "pivot grammar" (primitive two-element utterances in early childhood) (*50*) or an innate syntactic structure present in all children (*51*). Words that tend to come first in this two-word phrase bind preferentially to the first role neuron. A separate set of words tend to follow, and associate with the second role neuron, thus establishing two primitive word classes.

Second we challenge the model to learn connections between role neurons. After word-to-role mappings have been established, new orderings of these words drive new role sequences (**Fig. 5B**). Frequent role orders become embedded in the long-term role-to-role weights. Even branched role-to-role connections can be learned (**fig. S6**). This aligns well with theories that propose both learning and priming might arise from a single plasticity rule (*52*).

While word classes and role sequences can be learned independently, it remains unclear how they organically emerge together in a developing child, with overlapping time frames during language acquisition.

Simulating evoked potentials

The model's response to incoming words varies dynamically, and could therefore generate physiological predictions. To simulate EEG data from a single trial, we applied a derivative, smoothing and delay to the total neural activity in the model (**Fig. 5C-D**). Both violations in phrase structure and permissible morphemes produce ERPs resembling human EEG studies (*53*).

Discussion

We describe a neural model that encodes words in a sentence together with their syntactic roles, separating their structure from content. It binds words to their respective roles via a rapid Hebbian rule. The model predicts sentence superiority, grammaticalization errors, types of aphasia, and priming. It allows sentence generation, and morphemic languages. The model relies on two core principles. First, winner-takes-all dynamics allows symbol-like discrete representations to activate in structured sequences (**fig. S8)**. Second, Markov-like transitions between role neurons maintain structure over the words. This new architecture solves an old problem: How does the brain encode words as well as their function in a sentence?

Unlike large language models, our architecture is simple and interpretable. It may also be more biologically plausible, using only synapse-local rules, and may be realized in cortical synapses (*54*). Few neural models can *generate* as well as encode syntax – transformer-based neural networks achieve this using biologically implausible artifices, such as scrolling buffers and positional encoding. Our model also explains why violations of syntax lead to a larger neural response in evoked potentials, a key neural signature of syntax (*55*).

The approach allows us to model errors and response times. Lesions produce separable agrammatism and paragrammatism, often seen in stroke patients – but which are unlikely to occur in other language models. The explicit symbol-like behavior, where structural rules are

separated from content, could allow this architecture to be applied to other sequential cognitive processes, such as logical reasoning or deduction. In contrast with approaches that link neural networks to pure symbolic systems (*56*, *57*) a sentence is encoded *in synapses*, rather than as a pattern of neural activity. One alternative solution has used oscillatory synchrony to bind words to roles, but no readout or manipulation mechanisms have been proposed yet for such systems (*58*).

Our directional plasticity requires that the role-to-role facilitation must depend on the postsynaptic role neuron also receiving inputs from word neurons. Is this plausible? We speculate that potentiation only arises if the postsynaptic action potential travels retrogradely up dendrites, but the dendrites must be concurrently activated by word axons that arrive proximally on the dendritic tree. Such 3-way gating of facilitation is a strong prediction of our model.

The simple model presented has a number of important limitations. Presently, it does not reflect the hierarchical nature of sentence structures. This means that, to encode sentences like "the dog that chases the cat runs", the current model needs to duplicate all its syntactic rules for the subordinate clause (*8*), and the model's linear order will not explain rules of agreement (*9*). Nor is the model able to simultaneously assign the same word to two different roles, in the example "the little star is beside a big star". A possible mechanism for this may be to include dynamic splitting of a population of word or role neurons (*59*) whereby new subgroups might correspond to clauses which reuse instances of the same syntactic rule. This splitting could create hierarchy, characteristic of human language. We also do not model word recognition, effects of semantic context (*60*), and do not address general variable binding that would allow semantic inference (*61*).

The strengths of our approach are its breadth, accounting for cognitive, psychophysical, neurophysiological and neurological findings, while remaining so simple as to be fully transparent.

References and Notes

1. T. Jones, S. Farrell, Does syntax bias serial order reconstruction of verbal short-term memory? *J. Mem. Lang.* **100**, 98–122 (2018).

2. N. Perham, J. E. Marsh, D. M. Jones, Syntax and serial recall: How language supports short-term memory for order. *Q. J. Exp. Psychol.* . **62**, 1285–1291 (2009).

3. J. Kathryn Bock, Syntactic persistence in language production. *Cogn. Psychol.* **18**, 355–387 (1986).

4. M. J. Pickering, V. S. Ferreira, Structural priming: a critical review. *Psychol. Bull.* **134**, 427–459 (2008).

5. T. Verguts, P. De Boeck, On the correlation between working memory capacity and performance on intelligence tests. *Learn. Individ. Differ.* **13**, 37–55 (2002).

6. H.-M. Süß, K. Oberauer, W. W. Wittmann, O. Wilhelm, R. Schulze, Working-memory capacity explains reasoning ability—and a little bit more. *Intelligence*. **30**, 261–288 (2002).

7. T. Wennekers, M. Garagnani, F. Pulvermüller, Language models based on Hebbian cell assemblies. *J. Physiol. Paris*. **100**, 16–30 (2006).

8. F. H. Poletiek, P. Monaghan, M. van de Velde, B. R. Bocanegra, The semantics-syntax interface: Learning grammatical categories and hierarchical syntactic structure through semantics. *J. Exp. Psychol. Learn. Mem. Cogn.* **47**, 1141–1155 (2021).

9. M. B. H. Everaert, M. A. C. Huybregts, N. Chomsky, R. C. Berwick, J. J. Bolhuis, Structures, Not Strings: Linguistics as Part of the Cognitive Sciences. *Trends Cogn. Sci.* **19**, 729–743 (2015).

10. H. Tokuhara, K. Fujita, Y. Kashimori, Neural Mechanisms of Maintenance and Manipulation of Information of Temporal Sequences in Working Memory. *Cognit. Comput.* **13**, 1085–1098 (2021).

11. R. C. O'Reilly, M. J. Frank, Making working memory work: a computational model of learning in the prefrontal cortex and basal ganglia. *Neural Comput.* **18**, 283–328 (2006).

12. W. Kruijne, S. M. Bohte, P. R. Roelfsema, C. N. L. Olivers, Flexible Working Memory Through Selective Gating and Attentional Tagging. *Neural Comput.* **33**, 1–40 (2021).

13. D. M. Cer, R. C. O'Reilly, "Neural mechanisms of binding in the hippocampus and neocortex: insights from computational models" in *Handbook of Binding and Memory: Perspectives from Cognitive Neuroscience* (Oxford University Press, 2006), pp. 192–220.

14. H. Zhang, Y. Zhen, S. Yu, T. Long, B. Zhang, X. Jiang, J. Li, W. Fang, M. Sigman, S. Dehaene, L. Wang, Working Memory for Spatial Sequences: Developmental and Evolutionary Factors in Encoding Ordinal and Relational Structures. *J. Neurosci.* **42**, 850–864 (2022).

15. D. Rezende, D. Wierstra, W. Gerstner, "Variational learning for recurrent spiking networks" in *Adv. Neural Inf. Process. Syst.* (2011), vol. 24.

16. D. Kappel, B. Nessler, W. Maass, STDP installs in Winner-Take-All circuits an online approximation to hidden Markov model learning. *PLoS Comput. Biol.* **10**, e1003511 (2014).

17. O'Reilly, Soto, "A model of the phonological loop: Generalization and binding" in *Adv. Neural Inf. Process. Syst.*, vol. 14.

18. M. W. Howard, M. J. Kahana, A Distributed Representation of Temporal Context. *J. Math. Psychol.* **46**, 269–299 (2002).

19. H. Markert, A. Knoblauch, G. Palm, "Detecting Sequences and Understanding Language with Neural Associative Memories and Cell Assemblies" in *Biomimetic Neural Learning for Intelligent Robots: Intelligent Systems, Cognitive Robotics, and Neuroscience*, S. Wermter, G. Palm, M. Elshaw, Eds. (Springer Berlin Heidelberg, Berlin, Heidelberg, 2005), pp. 107–117.

20. T. Kriete, D. C. Noelle, J. D. Cohen, R. C. O'Reilly, Indirection and symbol-like processing in the prefrontal cortex and basal ganglia. *Proc. Natl. Acad. Sci. U. S. A.* **110**, 16390–16395 (2013).

21. E. T. Rolls, G. Deco, Networks for memory, perception, and decision-making, and beyond to how the syntax for language might be implemented in the brain. *Brain Res.* **1621**, 316–334 (2015).

22. K. Hammond, D. Leake, "Large Language Models Need Symbolic AI" in *17th International Workshop on Neural-Symbolic Learning and Reasoning* (CEUR Workshop Proceedings, 2023; https://ceur-ws.org/Vol-3432/paper17.pdf).

23. J. Gu, S. Lim, Unsupervised learning for robust working memory. *PLoS Comput. Biol.* **18**, e1009083 (2022).

24. M. Suzgun, S. Gehrmann, Y. Belinkov, S. M. Shieber, "LSTM Networks Can Perform Dynamic Counting" in *Proceedings of the Workshop on Deep Learning and Formal Languages: Building Bridges* (Association for Computational Linguistics, 2019), pp. 44–54.

25. R. Marvin, T. Linzen, "Targeted Syntactic Evaluation of Language Models" in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, 2018), pp. 1192–1202.

26. M. Schrimpf, I. A. Blank, G. Tuckute, C. Kauf, E. A. Hosseini, N. Kanwisher, J. B. Tenenbaum, E. Fedorenko, The neural architecture of language: Integrative modeling converges on predictive processing. *Proc. Natl. Acad. Sci. U. S. A.* **118** (2021), doi:10.1073/pnas.2105646118.

27. S. Dehaene, F. Al Roumi, Y. Lakretz, S. Planton, M. Sablé-Meyer, Symbols and mental programs: a hypothesis about human singularity. *Trends Cogn. Sci.* **26**, 751–766 (2022).

28. P. Belin, R. J. Zatorre, P. Lafaille, P. Ahad, B. Pike, Voice-selective areas in human auditory cortex. *Nature*. **403**, 309–312 (2000).

29. D. J. Freedman, M. Riesenhuber, T. Poggio, E. K. Miller, A comparison of primate prefrontal and inferior temporal cortices during visual categorization. *J. Neurosci.* **23**, 5235–5246 (2003).

30. J. Tanji, K. Shima, H. Mushiake, Concept-based behavioral planning and the lateral prefrontal cortex. *Trends Cogn. Sci.* **11**, 528–534 (2007).

31. K. Shima, M. Isoda, H. Mushiake, J. Tanji, Categorization of behavioural sequences in the prefrontal cortex. *Nature*. **445**, 315–318 (2007).

32. E. G. Antzoulatos, E. K. Miller, Increases in functional connectivity between prefrontal cortex and striatum during category learning. *Neuron*. **83**, 216–225 (2014).

33. L. Gwilliams, J.-R. King, A. Marantz, D. Poeppel, Neural dynamics of phoneme sequences reveal position-invariant code for content and order. *Nat. Commun.* **13**, 6606 (2022).

34. S. G. Manohar, N. Zokaei, S. J. Fallon, T. P. Vogels, M. Husain, Neural mechanisms of attending to items in working memory. *Neurosci. Biobehav. Rev.* **101**, 1–12 (2019).

35. R. J. Allen, G. J. Hitch, A. D. Baddeley, Exploring the sentence advantage in working memory: Insights from serial recall and recognition. *Q. J. Exp. Psychol.* **71**, 2571–2585 (2018).

36. P. W. Cho, M. Goldrick, P. Smolensky, Incremental parsing in a continuous dynamical system: sentence processing in Gradient Symbolic Computation. *Linguistics Vanguard*. **3** (2017), doi:10.1515/lingvan-2016-0105.

37. L. Gwilliams, A. Marantz, D. Poeppel, J.-R. King, Top-down information shapes lexical processing when listening to continuous speech. *Language, Cognition and Neuroscience* (2023), doi:10.1080/23273798.2023.2171072.

38. S. R. Anderson, M. Aronoff, M. Baerman, A. Carstairs-Mccarthy, J. Mugdan, "The Morpheme: Its Nature and Use" in *The Oxford Handbook of Inflection* (Oxford University Press, 2016), pp. 11–34.

39. E. A. Bates, A. D. Friederici, B. B. Wulfeck, L. A. Juarez, On the preservation of word order in aphasia: cross-linguistic evidence. *Brain Lang.* **33**, 323–364 (1988).

40. S. Martínez-Ferreiro, B. Ishkhanyan, V. Rosell-Clarí, K. Boye, Prepositions and pronouns in connected discourse of individuals with aphasia. *Clin. Linguist. Phon.* **33**, 497–517 (2019).

41. A. D. Friederici, Syntactic and semantic processes in aphasic deficits: the availability of prepositions. *Brain Lang.* **15**, 249–258 (1982).

42. A. Caramazza, R. S. Berndt, A. G. Basili, J. J. Koller, Syntactic Processing Deficits in Aphasia. *Cortex*. **17**, 333–347 (1981).

43. E. Tulving, D. L. Schacter, Priming and human memory systems. *Science*. **247**, 301–306 (1990).

44. K. Bock, H. Loebell, R. Morey, From conceptual roles to structural relations: bridging the syntactic cleft. *Psychol. Rev.* **99**, 150–171 (1992).

45. K. Bock, Z. M. Griffin, The persistence of structural priming: transient activation or implicit learning? *J. Exp. Psychol. Gen.* **129**, 177–192 (2000).

46. F. Chang, G. S. Dell, K. Bock, Becoming syntactic. *Psychol. Rev.* **113**, 234–272 (2006).

47. G. Malhotra, thesis, The University of Edinburgh (2009).

48. T. F. Jaeger, N. E. Snider, Alignment as a consequence of expectation adaptation: syntactic priming is affected by the prime's prediction error given both prior and recent experience. *Cognition*. **127**, 57–83 (2013).

49. D. Reitter, F. Keller, J. D. Moore, A computational cognitive model of syntactic priming. *Cogn. Sci.* **35**, 587–637 (2011).

50. M. D. S. Braine, The Ontogeny of English Phrase Structure: The First Phase. *Language*. **39**, 1–13 (1963).

51. G. Giusti, R. Gozzi, "The acquisition of determiners: Evidence for the Full Competence Hypothesis" in *Language Acquisition and Development: Proceedings of GALA2005*, A. Belletti, E. Bennati, C. Chesi, E. Di Domenico, I. Ferrari, Eds. (Cambridge Scholars Publishing, 2006), pp. 232–237.

52. F. Chang, G. S. Dell, K. Bock, Z. M. Griffin, Structural priming as implicit learning: a comparison of models of sentence production. *J. Psycholinguist. Res.* **29**, 217–229 (2000).

53. P. Hagoort, C. Brown, J. Groothusen, The syntactic positive shift (sps) as an erp measure of syntactic processing. *Lang. Cogn. Process.* **8**, 439–483 (1993).

54. F. Fiebig, A. Lansner, A Spiking Working Memory Model Based on Hebbian Short-Term Potentiation. *J. Neurosci.* **37**, 83–96 (2017).

55. S. M. Clardy, M. E. Hayden, S. Singh, E. L. Reilly, Speech evoked potentials in response to grammatical structures. *J. Acoust. Soc. Am.* **65** (1979), doi:10.1121/1.2016949.

56. J. Mao, H. Shi, W. Jiajun, R. P. Levy, J. B. Tenenbaum, "Grammar-Based Grounded Lexicon Learning" in *35th Conference on Neural Information Processing Systems* (2021).

57. P. Smolensky, Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artif. Intell.* **46**, 159–216 (1990).

58. K. Segaert, A. Mazaheri, P. Hagoort, Binding language: structuring sentences through precisely timed oscillatory mechanisms. *Eur. J. Neurosci.* **48**, 2651–2662 (2018).

59. K. J. Hayworth, Dynamically partitionable autoassociative networks as a solution to the neural binding problem. *Front. Comput. Neurosci.* **6**, 73 (2012).

60. M. A. L. Ralph, E. Jefferies, K. Patterson, T. T. Rogers, The neural and computational bases of semantic cognition. *Nat. Rev. Neurosci.* **18**, 42–55 (2017).

61. J. Feldman, The neural binding problem(s). *Cogn. Neurodyn.* **7**, 1–11 (2013).

62. N. M K, K. S. Prema Rao, Processing syntax: perspectives on language specificity. *Int. J. Neurosci.* **130**, 841–851 (2020).

63. A. Glushko, D. Poeppel, K. Steinhauer, Overt and implicit prosody contribute to neurophysiological responses previously attributed to grammatical processing. *Sci. Rep.* **12**, 14759 (2022).

64. P. Hagoort, How the brain solves the binding problem for language: a neurocomputational model of syntactic processing. *Neuroimage*. **20 Suppl 1**, S18–29 (2003).

65. A. Warstadt, Y. Cao, I. Grosu, W. Peng, H. Blix, Y. Nie, A. Alsop, S. Bordia, H. Liu, A. Parrish, S.-F. Wang, J. Phang, A. Mohananey, P. M. Htut, P. Jeretič, S. R. Bowman, "Investigating BERT's Knowledge of Language: Five Analysis Methods with NPIs" in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (Association for Computational Lingusitics, 2019), pp. 2877–2887.

66. L. Clouatre, P. Parthasarathi, A. Zouaq, S. Chandar, "Local Structure Matters Most: Perturbation Study in NLU" in *Findings of the Association for Computational Linguistics: ACL 2022* (Association for Computational Linguistics, 2022), pp. 3712–3731.

67. T. Linzen, E. Dupoux, Y. Goldberg, Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Transactions of the Association for Computational Linguistics*. **4**, 521–535 (2016).

68. G. Bhatt, H. Bansal, R. Singh, S. Agarwal, "How much complexity does an RNN architecture need to learn syntax-sensitive dependencies?" in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop* (Association for Computational Linguistics, Stroudsburg, PA, USA, 2020; https://www.aclweb.org/anthology/2020.acl-srw.33).

69. E. Wilcox, P. Qian, R. Futrell, M. Ballesteros, R. Levy, "Structural Supervision Improves Learning of Non-Local Grammatical Dependencies" in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (Association for Computational Linguistics, 2019), pp. 3302–3312.

70. P. Qian, T. Naseem, R. Levy, R. F. Astudillo, "Structural Guidance for Transformer Language Models" in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (Association for Computational Linguistics, 2021), pp. 3735–3745.

71. J. R. Brennan, C. Dyer, A. Kuncoro, J. T. Hale, Localizing syntactic predictions using recurrent neural network grammars. *Neuropsychologia*. **146**, 107479 (2020).

72. A. Kuncoro, M. Ballesteros, L. Kong, C. Dyer, G. Neubig, N. A. Smith, "What Do Recurrent Neural Network Grammars Learn About Syntax?" in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers* (2017), pp. 1249–1258.

73. J. Russin, J. Jo, R. C. O'Reilly, Y. Bengio, Compositional generalization in a deep seq2seq model by separating syntax and semantics. *arXiv [cs.LG]* (2019), (available at http://arxiv.org/abs/1904.09708).

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

75. Y. Grodzinsky, The neurology of syntax: Language use without Broca's area. *Behav. Brain Sci.* **23**, 1–21 (2000).

76. M. Hoffmann, R. Chen, The Spectrum of Aphasia Subtypes and Etiology in Subacute Stroke. *J. Stroke Cerebrovasc. Dis.* **22**, 1385–1392 (2013).

77. A. E. Hillis, D. Boatman, J. Hart, B. Gordon, Making sense out of jargon: A neurolinguistic and computational account of jargon aphasia: Neurology: Vol 53, No 8. *Neurology*. **53**, 1813–1824.

78. A. Roelofs, A neurocognitive computational account of word production, comprehension, and repetition in primary progressive aphasia. *Brain Lang.* **227**, 105094 (2022).

79. Y.-N. Chang, M. A. Lambon Ralph, A unified neurocomputational bilateral model of spoken language production in healthy participants and recovery in poststroke aphasia. *Proc. Natl. Acad. Sci. U. S. A.* **117**, 32779–32790 (2020).

80. J. M. Rodd, S. Vitello, A. M. Woollams, P. Adank, Localising semantic and syntactic processing in spoken and written language comprehension: an Activation Likelihood Estimation meta-analysis. *Brain Lang.* **141**, 89–102 (2015).

81. M. Vigneau, V. Beaucousin, P. Y. Hervé, H. Duffau, F. Crivello, O. Houdé, B. Mazoyer, N. Tzourio-Mazoyer, Meta-analyzing left hemisphere language areas: phonology, semantics, and sentence processing. *Neuroimage*. **30**, 1414–1432 (2006).

Acknowledgments: We thank Ray Jackendoff and Matthew Husband for invaluable discussions that shaped the manuscript.

Funding:

National Institute for Healthcare Research (NIHR) Oxford Biomedical Research Centre (BRC) (SGM)

James S McDonnell Foundation grant (SGM)

ESRC research grant ES/S015477/1 (SGM)

Author contributions:

Conceptualization: LS, SGM

Methodology: LS, SGM

Investigation: LS, SGM

Visualization: LS, SGM

Funding acquisition: SGM

Project administration: LS, SGM

Supervision: SGM

Writing – original draft: LS, SGM

Writing – review & editing: LS, SGM

Competing interests: Authors declare that they have no competing interests.

Data and materials availability: All data are available in the main text or the supplementary materials.

Supplementary Materials

Materials and Methods

Neurons in our network follow simple firing-rate dynamics. The two main neuron classes are words, i.e. fillers (**f**), and roles (**r**) (Fig. 1A). In the simplest case, long-term background weights encode which roles a word can fill ($\mathbf{W}_{\mathrm{L}}^{fr}$), and which roles can follow which others ($\mathbf{W}_{\mathrm{L}}^{rr}$). To encode a sentence, words **f** are sequentially activated by auditory inputs, and drive role neurons **r** (fig. S1A). The word-role synapses ($\mathbf{W}_{\mathrm{S}}^{fr}$) are strengthened bidirectionally. At the same time, the activation of a new role suppresses other roles, by winner-takes-all dynamics, but also rapidly strengthens synapses from the previously active role to the newly active role ($\mathbf{W}_{\mathrm{S}}^{rr}$) (fig. S1B). These two kinds of rapid plasticity respectively retain a working memory of the words that fill each role, and which roles follow which. So, when the sequence is later triggered by driving the first role neuron, the corresponding word neurons are re-activated in sequence (fig. S1C).

We term this encoding and retrieval behavior a 'plastic attractor', since the input words create a set of partly-stable states, each state corresponding to a role paired with a word. Syntax constrains the role dynamics, so the sequence of states tends to align with a grammatical sequence of words.

In some simulations, a third neuron type, "morphemes" (**m**), follow similar dynamics to feature neurons, but with constant non-encodable synapses to roles.

<u>Connectivity</u>

Neurons have three main sources of input: external, feedback from neurons of other types, and dynamics among neurons of the same type (equation S1).

$$\frac{1}{\tau}\begin{bmatrix}\Delta\mathbf{f}\\ \Delta\mathbf{r}\\ \Delta\mathbf{m}\end{bmatrix} = -\begin{bmatrix}\mathbf{f}\\ \mathbf{r}\\ \mathbf{m}\end{bmatrix} + \sigma\left(\begin{bmatrix}\mathbf{W}^{ff} & \mathbf{W}^{rf} & 0\\ \mathbf{W}^{fr} & \mathbf{W}^{rr} & \mathbf{W}^{mr}\\ 0 & \mathbf{W}^{rm} & \mathbf{W}^{mm}\end{bmatrix}\begin{bmatrix}\mathbf{f}\\ \mathbf{r}\\ \mathbf{m}\end{bmatrix}\right) + \boldsymbol{\alpha} \tag{S1}$$

Neurons of three classes, with firing rates represented by **f**, **r**, and **m**, receive external activation α through sensory input. Word neurons **f** can drive role neurons **r** via $\mathbf{W}^{fr} = (\mathbf{W}^{fr})^{\top}$. Words mutually inhibit each other by a fixed amount $\mathbf{W}^{ff} = -1.5$, and role neurons are connected to each other with a structured sequential pattern given by $\mathbf{W}^{rr}$. Activity is limited by a sigmoid activation function σ described below in equation S8. Activity decays over time with rate -1. The constant $\tau = 0.5$ determines the rate of updates.

Sensory input during memory encoding comprises a one-hot sequence of words, and then at the very start of recall, one role neuron receives a short burst of direct input as trigger. Morpheme neurons (**m**), where present, also receives and returns activation to its paired role neurons. $\mathbf{W}^{mm} = -10$ (mutual inhibition between morphemes) and $\mathbf{W}^{rm} = \mathbf{W}^{mr}$ (direct activation between morpheme and respective role neuron) are also fixed non-encodable constants.

Firing rates **f**, **r**, **m** are truncated to lie between 0 and 1, $n \Leftarrow [n + \Delta n]_0^1$.

Note that α is included within the sigmoid function in simulations 7, 8, 9 and 12 to perform syntactic serialization to non-auditory input. Weights $\mathbf{W}^{fr}$ and $\mathbf{W}^{rr}$ are the sum of a stable long-term knowledge component $\mathbf{W}_{\mathrm{L}}$ and rapidly plastic short-term component $\mathbf{W}_{\mathrm{S}}$, in addition to noise (e) (uniform distribution) where applicable (equation S2).

$$\mathbf{W} = k_{\mathrm{L}}\mathbf{W}_{\mathrm{L}} + \mathbf{W}_{\mathrm{S}} + e \tag{S2}$$

Long-term weights $\mathbf{W}_{\mathrm{L}}$ are scaled from 0 to 1, so that k_{L} is the maximum strength of prior knowledge. The rapidly changing components of the weights, $\mathbf{W}_{\mathrm{S}}$, is updated based on Hebbian rule (equation S3).

$$\mathbf{W}_{\mathrm{S}} \Leftarrow [\mathbf{W}_{\mathrm{S}} + \Delta\mathbf{W}_{\mathrm{S}}]_k^{\epsilon} \tag{S3}$$

The square bracket indicates the value is truncated to values between k and ϵ (equation S4).

$$[x]_k^{\epsilon} = \begin{cases} -k & \text{if } x < -k \\ x & \text{if } -k \leq x \leq \epsilon \\ \epsilon & \text{if } x > \epsilon \end{cases} \tag{S4}$$

Overall this means that $\mathbf{W}_{\mathrm{S}}$ varies from $-k$ to $+\epsilon$. $\mathbf{W}_{\mathrm{L}}$ is fixed as a pattern of either 0 or 1 and pre-specified in all simulations to reflect prior understanding of word classes and role order, with the exception of long-term knowledge acquisition (simulation 13 and 14) which is updated with identical rules but much more slowly (equation S5).

$$\mathbf{W}_{\mathrm{L}} \Leftarrow [\mathbf{W}_{\mathrm{L}} + \upsilon\Delta\mathbf{W}_{\mathrm{S}}]_0^1 \tag{S5}$$

Here, $\upsilon = 0.001$ for long-term word class knowledge acquisition and $\upsilon = 0.025$ for long-term role order knowledge acquisition. These determine the size of update at each timestep.

<u>Plasticity rules</u>

There are key differences between Hebbian rules in word-to-role connections and in role-to-role connections. $\mathbf{W}_{\mathrm{S}}^{fr}$ is strengthened when a role neuron and word neuron are simultaneously active, but weakened if only one is active (equation S6). h is a constant that controls the balance between strengthening and weakening, and $\lambda^{fr} = 1$ is the learning rate.

$$\frac{h\Delta\mathbf{W}_{\mathrm{S}}^{fr}}{\lambda^{fr}(2+h)} = \left(\mathbf{r} - \frac{1}{(2+h)}\sum\mathbf{r}\right)\left(\mathbf{f} - \frac{1}{(2+h)}\sum\mathbf{f}\right)^{\top} - \frac{1}{(2+h)^2}\sum\mathbf{r}\sum\mathbf{f} \tag{S6}$$

For $\mathbf{W}_{\mathrm{S}}^{rr}$, the connection weight from role neurons r_1 to r_2 is strengthened during rapid encoding if both role neurons are simultaneously active and paired with a high level of input from the word neurons to r_2 (equation S7). In other words, if a role neuron is being driven by an external word, then it will learn to receive stronger input from other roles that were also recently active, but those synapses will weaken otherwise. The diagonal entries of $\mathbf{W}^{rr}$ is kept constant at zero and λ^{rr} is the learning rate. $\odot$ is element wise multiplication.

$$\frac{2\Delta\mathbf{W}_{\mathrm{S}}^{rr}}{\lambda^{rr}} = \left(3\mathbf{r}' - \sum\mathbf{r}'\right)\mathbf{r}^{\top} - 2\mathbf{r}\mathbf{r}'^{\top} \quad \text{where, } \mathbf{r}' = \mathbf{r} \odot [\mathbf{W}^{fr}\mathbf{f}]_0^1 \tag{S7}$$

Words are presented either sequentially or simultaneously (i.e. during syntactic serialization) and with no gaps. Each period consists of one run of the complete input (i.e. a sentence or a word-list). Recall phase is prompted by activating the first role neuron for five timesteps, followed by no further direct input. Analyzed result begins from the fifth timestep onwards. The neurons are expected to spontaneously retrieve the input with help from a leftward

shift of the sigmoid function to increase all neuronal tendency to self-activate. The equations of the recall phase are identical to the encoding phase except that $\mathbf{W}_{\mathrm{S}}$ is not updated.

Other than these Hebbian updates, there is no built in weight decay.

Equation S8 is a sigmoid softmax function with slope 10, where θ is 0.5 during all encoding and zero during all recall.

$$\sigma(x) = \frac{1}{1 + e^{-10(x-\theta)}} \tag{S8}$$

Table S1 summarizes constants unspecified above. Graphical representations of equations S6 and S7 are presented in fig. S8. The pseudocode for the algorithm of each timestep is presented in fig. S9. All statistical analyses were performed with Scipy 1.5.0. Python code and random initializations were created with random seeds of Python 3.8.5 and NumPy 1.19.1. Jupyter Notebooks for our experiments are available at https://doi.org/10.5281/zenodo.10372949.

Simulating simple grammar in working memory (**Simulations 1, 2, 3, S1, S2, S3**)

In simple recall comparing a syntactic sentence with a shuffled word-list, the model either receives a syntactic sentence or a shuffled word-list during the encoding phase. Each word is activated for 50 timesteps. Each repeat of a sentence or a word-list defines a single period. Then the model is prompted (i.e. with 5 steps of activation of the first role neuron) to recall the encoded sentence. In recall of sentences where there is switching, the model receives the first sentence for encoding and then is prompted to recall the first encoded sentence. Following which and without resetting, the model receives the second sentence for encoding (i.e. the switch) and it is then prompted to recall the second encoded sentence.

Recall accuracy is the proportion of input words that appear during the recall phase. A word is considered to have "appeared" when it has the strongest activation amongst all words of a timestep (*Argmax*). Recall accuracies are averaged over 200 trials with ±0.5 uniform noise in all connection weights. Human data on recall accuracy is adapted from Allen et al (2018) figure 1(a) by averaging the recall accuracy at each serial position. Statistical significance is calculated using unpaired, two-tailed Student's *t*-test.

Grammaticalization errors made by our model are measured as the proportion of recalled word pairs that were syntactic. The 8-word long sentence is shuffled randomly 30 times. Each of these 30 shuffled word-lists is encoded and recalled by the model over 200 trials with different random seed initialization and ±0.5 uniform noise to all connections. One such word-list has been included in Fig. 1B(iii) ("Up woke for cat tea the its late.") Statistical significance is performed using Pearson's Chi-square test on summed total syntactic and summed total non-syntactic word pairs in recall over the 200 trials of all 30 word-lists. The null hypothesis is that there is no difference between the observed count of syntactic word pairs versus the expected count of syntactic word pairs, which is the chance level seen in Fig. 1B(v) (red dashed line). The expected count is one-eighth (7 syntactic word pairs out of all 56 possible pairs in a 8-word sentence) of total number of word pair appearances during recall. The proportion of grammaticalization errors in Jones and Farrell 2018 is adapted from their figure 3. It is calculated as the average proportion of improvement in 6-token parsing over all 20 participants.

Simulation of soft constraint is achieved by varying k_{L} of $\mathbf{W}_{\mathrm{L}}^{cc}$ and $\mathbf{W}_{\mathrm{L}}^{fc}$ for connections "determiner" to "adjective" ($j1$), "determiner" to "subject-noun" ($j2$), "rich" to "subject-noun" ($k1$) and "rich" to "adjective" ($k2$). Heatmap is generated by repeating 2601 trials with ±0.25 uniform noise in all connections. Each trial involves defining the values of $j1$, $j2$, $k1$ and $k2$.

Then, the network is encoded with the sentence "The rich men have money" and prompted to recall (i.e. parse). The network is then reset and encoded with the sentence "The rich have money" and similarly prompted to recall (i.e. parse). Each square in the heatmap is a single trial. Neighboring trials in the vertical direction differs by 0.01 for $j1$ and $j2$, with $j1$ increasing and $j2$ decreasing from top to bottom. Neighboring trials in the horizontal direction differs by 0.01 for $k1$ and $k2$, with $k1$ decreasing and $k2$ increasing from left to right.

Simulation of word class disambiguation uses the same constants as simulation 1.

Morphemic languages with flexible word order (**Simulations 4, 5, 6**)

The model receives input sentences which are either words or pairs of words and morphemes (50 timesteps for each word or pair) and is then prompted to recall. The lack of long-term role-to-role knowledge results in the need for stronger inhibition and activation between all neurons. Therefore deactivation of the preceding role neuron with an accompanying morpheme will require stronger inhibition between the role neurons, which has to be coupled with higher ceiling of activation to progress firing to the following role neuron. Likewise stronger activation between word and role neurons is needed to overcome the stable activation of the preceding role neuron accompanied with a morpheme. This too has to be balanced with stronger inhibition between word and role neurons. These parameter changes are reflected in table S1.

Syntactic serialization, agrammatic and paragrammatic aphasias (**Simulations 7, 8, 9**)

The model receives the sentence (with each word taking 5 timesteps) or concurrent word inputs (over 20 timesteps) to be encoded and is then prompted to recall the syntactically structured sentence, leading to either control or aphasic output. During serialization, grammaticalization is achieved by including α inside the σ function. Closed-class words have lower k_{L} in $\mathbf{W}_{\mathrm{L}}^{fr}$, to ensure these words have connections more sensitive to noise in the network. The agrammatic lesion adds ±0.25 uniform noise to the rapid synaptic plasticity between role neurons $\mathbf{W}_{\mathrm{S}}^{rr}$. Intuitively, this makes the network less sure of the role of the next word in the sequence. The paragrammatic lesion adds ±0.005 uniform noise to long-term connections between word and role neurons, and with $\lambda^{fr} = 0$. Intuitively, this means that while the role sequence is grammatical and grammatically appropriate words will be selected, the word will not be the encoded one.

Syntax errors were the number of non-syntactic word pairs in the output, including situations where more than one word had the highest activation at a timestep. Word errors were the number of words appearing in recall but were not in the input. Both agrammatic and paragrammatic aphasias produced by the model were repeated 50 trials each, with different random initializations. Human data is taken from table 1 of Caramazza et al (1981) with the agrammatic error counts being the mean of the Broca's aphasia cases. Statistical significance for the model is calculated using unpaired two-tailed Student's *t*-test.

Errors in the agrammatic simulation included dropping the closed-class word, stopping the sentence early, and skipping intermediate words.

Affix hopping (**Simulation S4**)

Affix hopping is the phenomenon where a morpheme that belongs early in a sequence of words occurs later in the sequence. For example, in the English sentence "the dogs were eating", the participle ending "-ing" indicates continuous tense, but does not attach to the active verb "were", which should indicate tense but here is modal.

Affix hopping can arise in the model because once a particular syntactic branch is chosen, there is no requirement for morphemes to connect to only the first role neuron in the branch. To

demonstrate this, simulation S4 adopts all parameters from simulation 7 with the exception that concurrent word and morpheme (or semantic tag) inputs are run over 30 timesteps. The model receives either a sentence with simple active verb (e.g. "the dogs eat" with semantic tag for the role neuron of the active verb) or a sentence with a past participle (e.g. "the dogs were eat-ing" with a semantic tag for the auxiliary role neuron with strong connection to "were". The morpheme neuron "-ing" is not activated during the encoding phase and is expected to spontaneously activate during the recall phase. "eat" in both sentences belongs to the same single word neuron (fig. S5).

Lexical and syntactic priming (**Simulations 10, 11, 12, S5, S6, S7, S8**)

In simulations of lexical and syntactic priming effects on comprehension, the model receives a pair of sentences to encode with no reset in between. For instance in a_1b_1, the model is encoded with priming sentence a_1 and then target sentence b_1, each over 50 timesteps. The model is then prompted to recall the target sentence. Recall accuracy follows protocol set out in simulation 1. Each simulation is repeated over 50 trials with ±0.25 uniform noise for simulations 10, S5 and S7, ±0.15 uniform noise for simulations 11 and S8, and ±0.1 uniform noise for simulation S6 in all connection weights. Response time is measured by the timesteps elapsed to reach the final word as the strongest item (*Argmax*). Response times are calculated only if 3 or more words in the target sentence are successfully retrieved. Unpaired, two-tailed Student's *t*-test is used for comparing recall accuracy and Mann-Whitney U test for comparing response time.

To simulate Bock (1986)'s syntactic priming experiment in language production, the network needs to include neurons for agent and patient, which we term "semantic tags". They behave just like morphemes, in that they drive a given role neuron, and can be co-active with words. The agent and patient semantic tags drive the role neurons for objects, but in different syntactic branches – one corresponding to a sentence where the object is an agent, and another where the object is the patient. The subject role is not constrained to either role because, in English, the subject can be parsed without knowing which semantic role (agent or patient) it will take. Therefore the subject role neuron has a semantic tag representing either agent or patient.

First, the priming sentence is encoded with no semantic tag activation. Each word is presented in sequence for 10 timesteps, driving the role neurons down one of the two syntactic branches, and strengthening those role-to-role weights.

Second, to simulate a mental idea, the network has to first refresh its semantic representation as the subject and object in this mental idea is different from the nouns in the priming sentence. Therefore, we activate each noun in the mental idea simultaneously with its appropriate semantic tag (agent or patient). This results in each noun being associated to one of two possible object role neurons; thereby one of the nouns will be assigned to the object role in the primed syntactic pathway. In addition, a subject semantic tag is also activated with both nouns. We associate *both* nouns to the subject role as we want the network to choose the subject noun in the next step based on the priming sentence. Therefore three neurons are activated at a time as a triad (e.g. triad containing "birds" + patient semantic tag + subject semantic tag then triad containing "hunters" + agent semantic tag + subject semantic tag). The action word in the mental idea is also activated, in order to associate it with both the verb and participle roles. Through rapid synaptic plasticity, association of this new action word helps overwrite the network's association to the previous action word in the priming sentence. Likewise, the network will also spontaneously choose whether the action word in the mental idea is a finite verb (active) or participle (passive) during the next step, based on the priming sentence. The action word does not require any semantic tag and therefore only one neuron is activated (i.e. "shot"). Each triad or neuron is activated for two timesteps, separated by 20 timesteps between triads or

neuron to avoid cross-association. This cycle is repeated 15 times. The reason for using short repeated encoding of the semantic tags, is because the role neurons would otherwise tend to progress to the subsequent words. From this stage onwards, the rapid synaptic plasticity between role neurons is switched off ($\lambda^{rr} = 0$), just as for recall, but word-to-role plasticity (λ^{fr}) is still needed to facilitate selection of the appropriate word in the sequence.

Third, the network has to form the sentence before utterance in the next step. It therefore receives a "bag of words" consisting of semantic contents ("birds", "shot" and "hunters") and the network spontaneously serializes the word content into a grammatical sentence in either the active or passive voice depending on initial priming sentence. The network receives two timesteps of the first word "the" and then 20 timesteps of simultaneous input of words "birds", "shot" and "hunters" for syntactic serialization. This allows the network to encode a grammatical sentence from just semantic representations of the mental idea. A crucial dynamical process is also taking place during this step. Recall that we mentioned in the second step that both "birds" and "hunters" were activated with the subject semantic tag, but clearly only one can be the subject of the target sentence. Therefore at this stage, the network dynamics weakens the connection between the subject role neuron and the noun attached to the object role in the primed pathway. This is possible as the noun attached to the object role has its weight to the object role neuron strengthened from increased activation of the object role neuron in the destined branch as determined by the initial priming sentence. The Hebbian dynamics thus results in the weakening of the connection between this noun and the subject role neuron. For example, in the passive priming case, "hunters" is the object of the sentence; thus its weight to role neuron "r_8" is strengthened while the network dynamics weakens its weight to the subject role neuron "r_2".

Fourth, during the utterance stage, the network is prompted to produce the target sentence describing the mental idea by activating the first role neuron for 5 timesteps, similar to a recall phase. This results in the words being activated in sequential order, either corresponding to the active or passive voice.

Fifth, the whole process is repeated 1000 trials for each priming condition (i.e. active or passive voice) with ±1.5 uniform noise in all Hebbian connection weights. The resultant utterance is counted as an active voice if the words appear in the order "agent, verb then patient" regardless of presence or absence of other words. Likewise, passive voice is counted if the order is "patient, participle then agent". If both sequences are present, the first sequence to appear is counted in keeping with Bock's original experimental protocol. Human subject data is obtained from table 1 of Bock (1986).

Long-term knowledge acquisition (**Simulations 13, 14, S9**)

In long-term knowledge acquisition, the model is given either partial or complete lack of long-term knowledge (i.e. weights in $\mathbf{W}_{\mathrm{L}}^{rr}$ and $\mathbf{W}_{\mathrm{L}}^{fr}$ are zero) to simulate the naive learner. The network undergoes 30 learning periods and as before, during each period a sentence is encoded. In the case of long-term acquisition of word class knowledge using pivot grammar (simulation 13), $\mathbf{W}_{\mathrm{S}}$ is re-initialized to 0 at the end of every timestep to simulate a naive learner with no inhibitory connections at rest. In the case of long-term acquisition of role-role knowledge, $\mathbf{W}_{\mathrm{S}}$ is re-initialized to $-k$ at the end of every timestep. The learning rates for the long-term weights are much lower than for the short-term weights, but still follow the same Hebbian rules. Each neuronal activation lasts 50 timesteps. All other constants are unchanged from simulation 1.

During long-term acquisition of word-to-role knowledge, the role neurons receive simultaneous activation with the word input to simulate an already-existing pivot grammar. Sequential activation of the two role neurons, as opposed to spontaneous activation of the second

role neuron from the first role neuron, is required due to the naive network's lack of inhibitory dynamics resulting in early activation of the second role neuron.

During long-term acquisition of role-to-role structure, the network receives a different randomly generated sentence for each period, but with the same syntax. Each role neuron has three possible words. Recall accuracy after each period is determined by running simulation 1 (50 trials and ±0.5 uniform noise in all connection weights) on the network with frozen long-term weights and allowing for rapid synaptic plasticity.

Evoked potential (ERP) simulation (**Simulation 15, 16**)

Evoked potentials from the electroencephalogram (EEG) have provided neural signatures of syntax for over 40 years (*56*) in many languages (*63*), and remain a subject of active research (*64*). To simulate an EEG for one trial, we apply a temporal derivative to the sum of squared activations of all the neurons. We then apply 30 timestep gaussian smoothing and a 15 timestep delay.

In simulation 15, 120 trials were simulated. A sentence was presented to the network in the standard way. Half were syntactic (Determiner → Subject → Verb → Determiner → Object) and half were violations (Determiner → Subject → Verb → Object → Determiner). We used all combinations of 3 possible nouns and ±0.5 uniform noise was applied to all connections to generate within-condition variability. The EEG was averaged for each condition to obtain an ERP. Constants were kept the same as simulation 1 with each word taking 50 timesteps. Human ERP data was adapted from figure 7 of Hagoort et al (1993). Here, phrase structure violation coactivates multiple role neurons, pushing the neurons into a higher energy state, further from an attractor, increasing firing rate. This results in the positive ERPs at the labeled points.

In simulation 16, 60 trials were simulated. A sentence was presented to the network with either the correct morpheme placement (Determiner → subject → auxiliary → verb → adverb with morpheme "-ily") or incorrect morpheme placement (Determiner → subject → auxiliary → adjective with morpheme "-ily" → object). All combinations of 3 adjectives and adverbs were used and ±2 uniform noise was applied to all connections. Likewise, EEG was averaged for each condition to generate ERP. Constants were kept the same as simulation 4. Human ERP data was adapted from figure 1 of Hagoort et al (1993). Activation of the morpheme neuron "-ily" in the correct agreement condition results in increased total firing rates amongst all neurons. Consequently there is a positive ERP. The preceding negative ERP is also explained by the activation of "-ily" in the incorrect agreement condition.

The model does not include semantics or agreement rules, and so does not generate ERP components associated with violations of agreement.

Supplementary Text

Supplementary background

Neural networks to model language have a long history, and have previously adopted a range of methods to simulate syntax and structure within sequences. Many neural models of ***working memory*** have been proposed that hold sequential information, for example through timing (*10*), order (23), or production rules (*11–13*). Word sequences may form forward associations between words (*14*, *15*) with short-term synaptic plasticity (*16*) or by associating each word with a "temporal context" (neurons that represent the current time) to create a "phonological loop" (*17*, *18*). But these models do *not* account for the effect of syntactic rules, e.g. that word lists are recalled better if they are syntactic than non-syntactic.

To account for syntax, sentences can be encoded using states (*19*, *20*) or neurons that code for word-and-role combinations (*21*). However this quickly leads to a combinatoric problem with

longer sentences. A more linguistically inspired strategy is to build a syntactic tree as words are encoded (*49*, *64*), however these models do not specify how words are stored together with the structure.

More recent approaches use backpropagation to train weights in a ***deep neural network***. Transformer-based large language models can generate and understand syntax (*65*) but may encounter difficulties when role-position information is critical (*66*), and rely on shifting buffers which are difficult to implement using biological neurons. Recurrent neural networks (RNN) trained on next word prediction can account for long-range-dependencies such as agreement (*67*, *68*) and can represent rich language structures (*10*, *23–25*), but still lack an intuitively symbol-like architecture (*27*) and require structural information in training (*69*, *70*).

To address this, alternative approaches have used RNNs to predict syntax by training them to generate grammar parsing *actions* (e.g. "open new noun phrase", "close phrase") (*71*, *72*). While these RNN grammars represent syntactic information, they do not store the resulting structured sentence. Other RNNs have incorporated separate attention mechanisms for syntactic and semantic information (*73*). This mechanism can generalize grammatical rules to new content words, but requires tricks like reading sentences both forward and backward in parallel, and it is unclear whether the RNN keeps a representation of the whole sentence.

In summary, although models of language can clearly form deep representations of grammatical structure, they suffer from various shortcomings related to explainability and biological plausibility, and are usually only compared against a narrow set of biological findings.

Arguably the most dramatic insights into the biology of language have come from aphasia. Lesion effects can be modeled in terms of stepwise syntactic operations (*74*, *75*). These have met with variable success, since a wide range of language deficits are possible after brain lesions (*76*). Artificial neural networks are good at simulating a spectrum of deficits, but the classic dissociation between ***agrammatic*** and ***paragrammatic*** aphasia has been the most challenging (*77–79*).

Agrammatism is characterized by non-fluent speech with pauses, and telegraphic speech with missing closed-class words (*41*). Despite this, content words are relatively preserved. In contrast, patients with paragrammatic or jargon aphasia have fluent, grammatical speech, but with word substitutions. The incorrect words may be semantically related (verbal paraphasias), or phonologically related (phonemic paraphasias).

It is unclear how the semantic-syntactic distinction is embodied in the brain. While neurons in superior temporal cortex are selective for the identity of auditory stimuli (*28*), neurons in prefrontal cortex encode abstract categorization of input (*29–32*) and might therefore represent word roles (*33*). However, the dissociation is unclear in functional MRI, where both syntax and semantics activate a distributed network of dominant hemisphere areas (*80*, *81*). One possible reason for this is that during any kind of syntactic and semantic language processing, both kinds of neural representation are requisite.

<u>Grammaticalization errors</u>

How do role neurons in our model result in the grammaticalization of a shuffled sentence? Consider a sentence "the cat woke up late" that is carefully shuffled to place "cat" in a later position (i.e. "the woke up cat late"). Recall of the shuffled list with noise will demonstrate frequent recall of "up late" even though the input for encoding was "... up cat late …". The sequence of events leading to this is as follows – often, the role neuron for "the" remains activated even after "woke" takes over as the input word. The reason for this is that the role neuron for "woke" requires both the activation of an appropriate preceding role neuron (which is not the role neuron of "the") and the input from "woke" to overcome strong activation from the

role neuron of “the”. Since it is missing one key input, the role neuron of “the” may remain activated and *slips* to bind to the next word input (i.e. “woke”). However, the word input following “woke” is up. The role neuron for “up” does not syntactically follow the role neuron for “the” as well. Therefore, the role neuron for “the” may continue to *slip* forward again. Finally, on encountering “cat”, the role neuron of “cat” receives input from the appropriate role neuron (i.e. syntactic flow from “the” to “cat”) and hence the role neuron of “the” successfully deactivates. Since the last word that activated the role neuron of “the” was “up”, there is now a strong rapid synaptic connection between role neuron of “the” and the word “up”. Following the activation of the role neuron of “cat”, “late” is activated. However, “cat late” is not syntactic and therefore the role neuron of “late” is not successfully activated. The role neuron of “cat” therefore *slips* forward to bind to “late”. The same *slipping* process may continue on until the end of the shuffled list. If the process stops at “late”, then the end state is that the role neuron of “the” is strongly connected to “up” and the role neuron of “cat” is strongly connected to “late”, resulting in recall of “up late”.

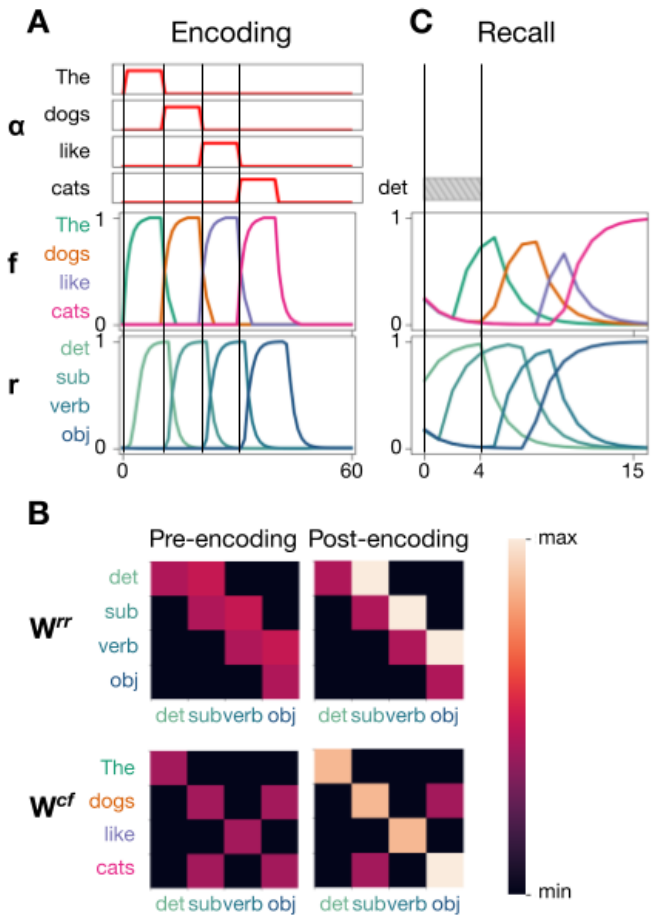

Fig. S1. Simple walkthrough of encoding and recall of one sentence.
(**A**) Sensory input (α) results in activation of word neurons (**f**) and role neurons (**r**). Consequently, there is encoding through rapid synaptic plasticity of connections between neurons. This then allows recall of the sentence after brief activation of the first role neuron, the determiner. (**B**) The sum of long-term and short-term synaptic weights before and after rapid synaptic plasticity. Role neurons: det=determiner, sub=subject, obj=object.

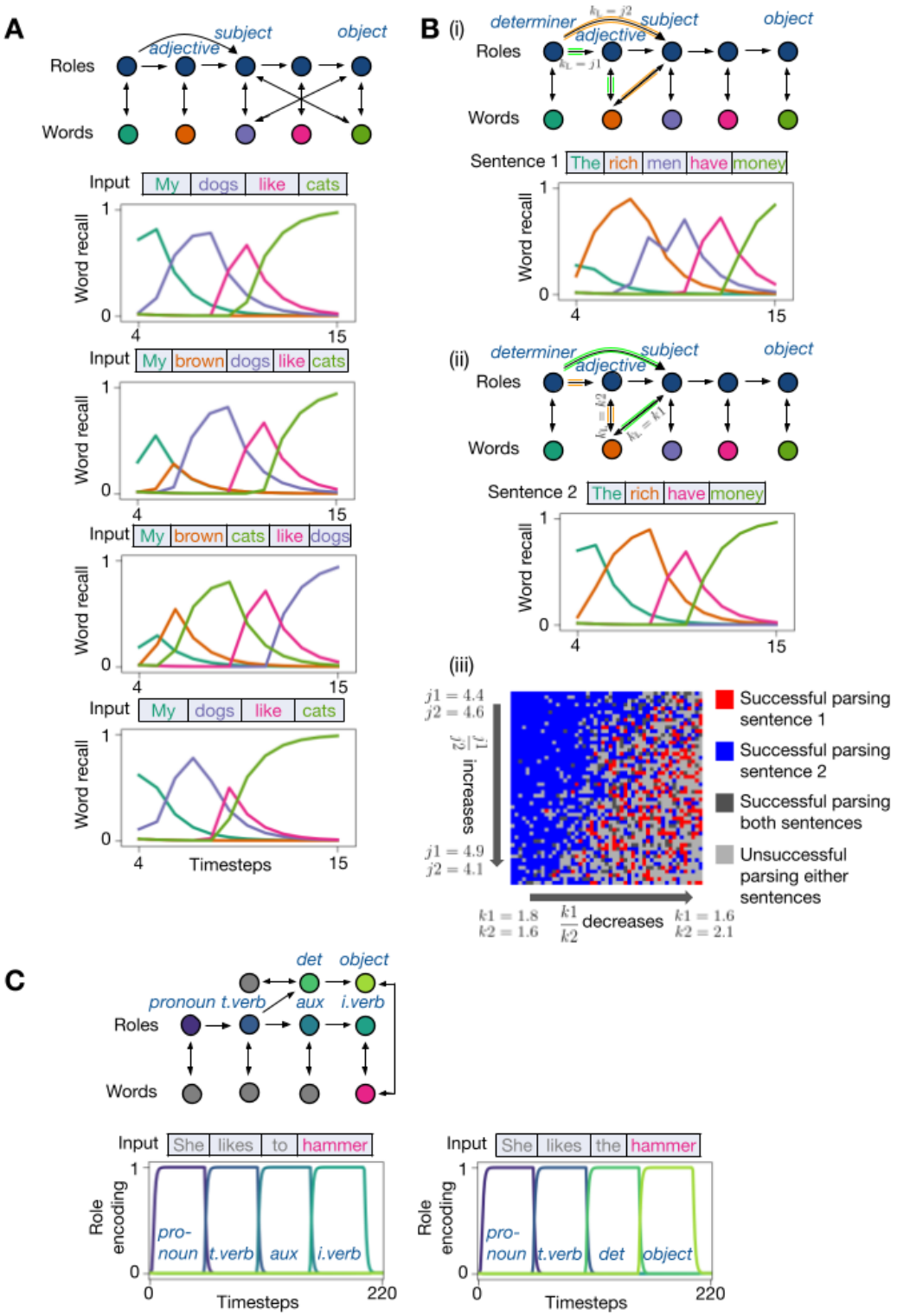

Fig. S2. Additional simulations with simple grammar.
(**A**) Simulation S1: Both motifs from simulations 2 and 3. Sequential rapid plasticity encoding and recall demonstrates the model's ability to simultaneously handle both motifs. Notably in the final step from "My brown cats like dogs." to "My dogs like cats." there are both a switch from presence to absence of adjective and a switch between subject and object roles of the nouns. (**B**) Simulation S2 of a garden-path sentence where "rich" could be an adjective or a subject. (i)

“rich” preferentially binds to the adjective role neuron ($k2 > k1$), which is stronger than the syntactic preference to skip the adjective role neuron ($j2 > j1$). This results in parsing “rich” as an adjective. (ii) The word “rich" could also be parsed as a subject, but this leads to a garden path effect (failure to parse the sentence), unless the “rich”-“subject” synapse ($k1$) is strengthened relative to “rich”-“adjective” ($k2$), or the “determiner”-“subject” synapse ($j1$) is strengthened relative to “determiner”- “adjective” ($j2$). (iii) Varying the strengths of the syntactic and word-class constraints leads to different success rates in parsing the sentence. (**C**) Simulation S3 of disambiguation of word class based on role sequence. Role neuron encoding is the activation of role neurons during the input of respective sentences. In “she likes to hammer”, activation of word “to” strongly activates the determiner role neuron which then activates the intransitive verb role therefore binding “hammer” to the intransitive verb role neuron. In “she likes the hammer”, the activation of “the” activates the determiner role neuron which prompts activation of the object role, therefore binding “hammer” to the object role neuron. This agrees with the “top-down” word class activation seen in MEG – that is the word role activated is one which fits the context.

Fig. S3. Network output with different random initializations during simulation of agrammatic and paragrammatic aphasias.
All network outputs are a result of the same hyperparameters but with different random initialization seeds. (**A**) Agrammatic outputs that are non-syntactic. '...' represents more than one word having the same *Argmax* value. (**B**) Paragrammatic outputs with characteristically well formed sentences but missing the underlying encoded message.

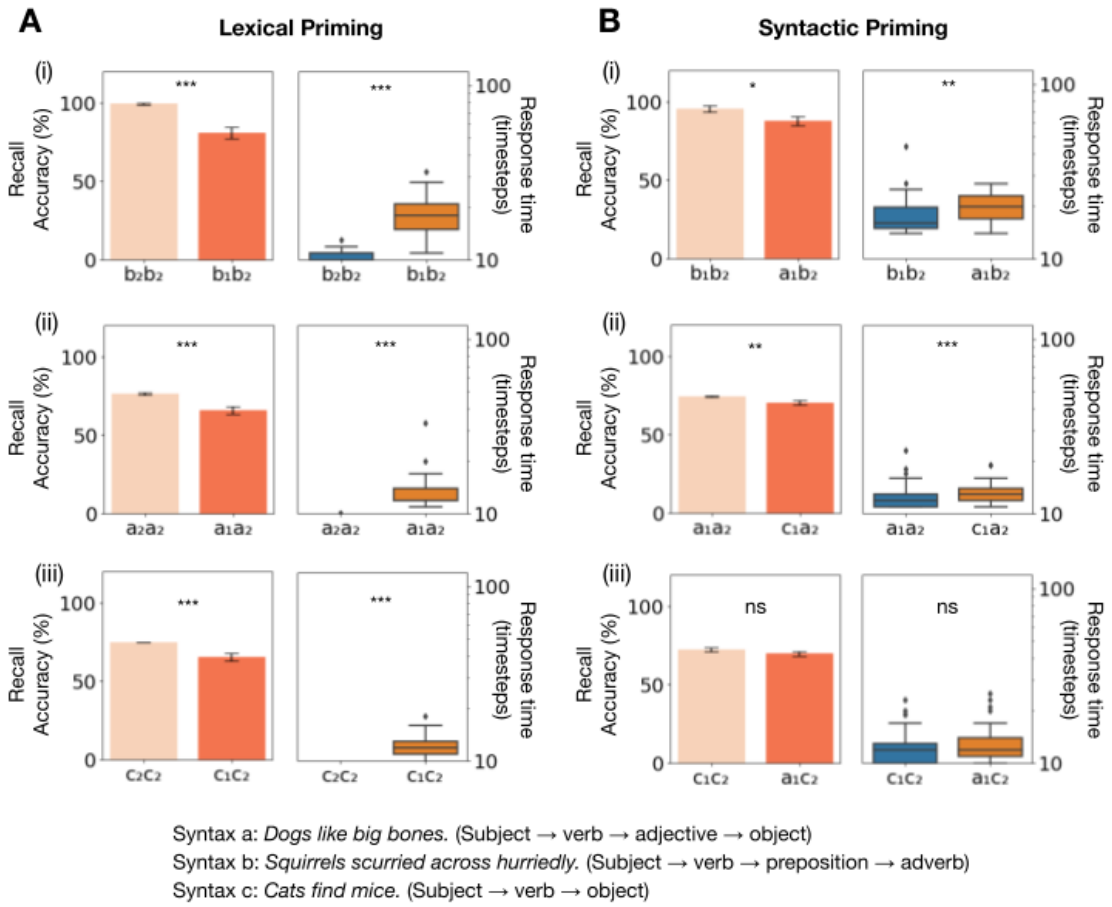

Fig. S4. Results of lexical and syntactic priming during language comprehension. (**A**) Recall accuracies and response times after lexical priming. (i) Simulation 10: Lexical priming of syntax b. (ii) Simulation S5: Lexical priming of syntax a. (iii) Simulation S7: Lexical priming of syntax c. (**B**) Recall accuracies and response times following syntactic priming. (i) Simulation 11: Priming with either syntax a or b for target sentence of syntax b. (ii) Simulation S6: Priming with either syntax a or c for target sentence of syntax a. (iii) Simulation S8: Priming with either syntax a or c for target sentence of syntax c. Recall accuracies are with respect to the target sentence (i.e. second sentence in each pair). ns non-significant; *P-value<0.05; **P-value<0.01; ***P-value < 0.001, error bars are SEM. Box plot showing interquartile range.

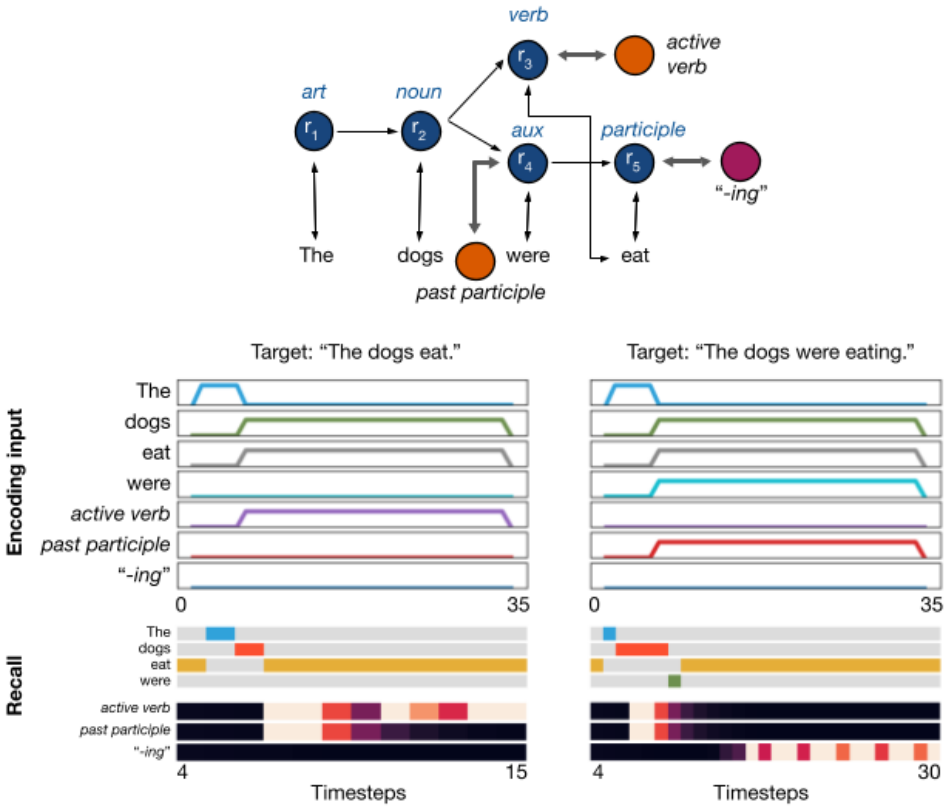

Fig. S5. Affix hopping.
In simulation S4, the network is initiated with two possible tenses – simple present and past participle, each controlled by a semantic tag neuron which is functionally the same as a morpheme neuron. A morpheme neuron "-ing" is also tagged to the participle role neuron. The encoding input is a concurrent "bag of words" with respective semantic tags. Recall of the sentence in the past participle tense includes "-ing" activation with "eat", simulating affix hopping.

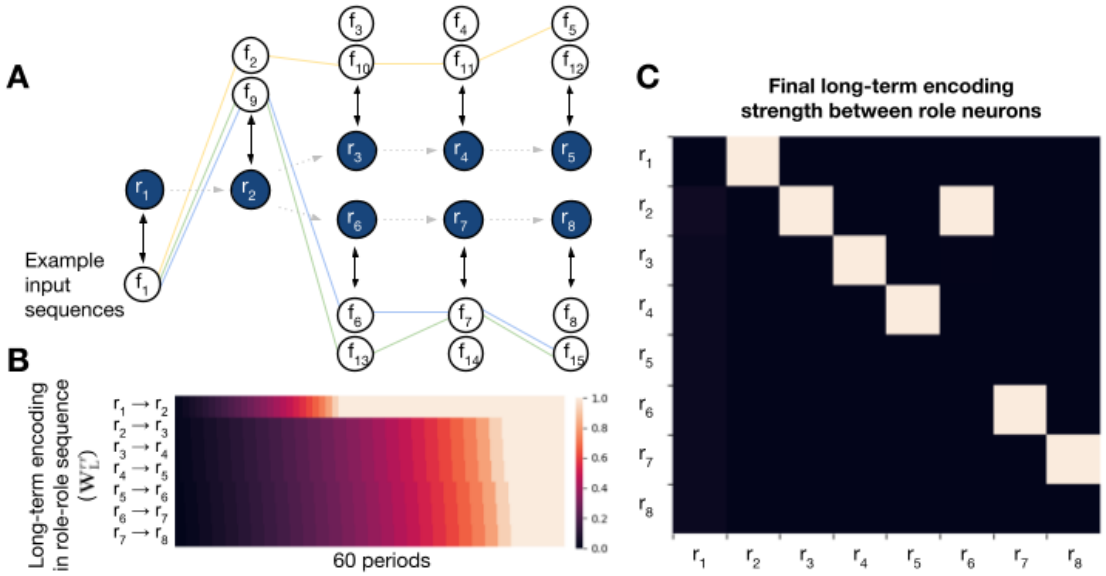

Fig. S6. Long-term encoding of role-to-role knowledge with branching.
(**A**) Simulation S9: two possible syntactic structures amongst the role neurons can be learnt from the 15 available words. The model is fed randomly generated sentences from these 15 words. (**B**) Strengthening of long-term role-to-role knowledge over 60 periods of interleaved sentences from each role sequence. (**C**) The final long-term encoding weights of all role-to-role connections after the last period.

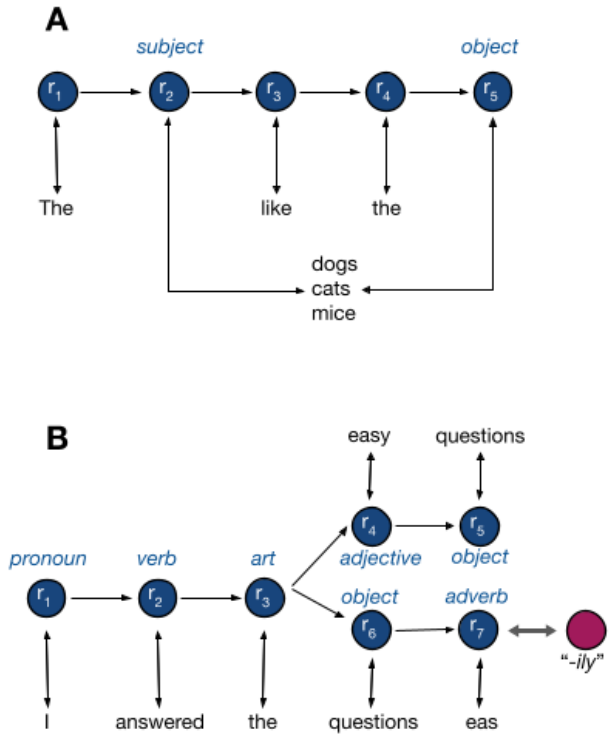

Fig. S7. Networks of Simulations 15 and 16.
(**A**) A simple grammatical word order in the role neurons to demonstrate ERP during break in syntax. (**B**) A morpheme neuron for "-ily" is bound to the adverb role neuron. In incorrect morpheme placement, "-ily" is activated with the adjective "easy".

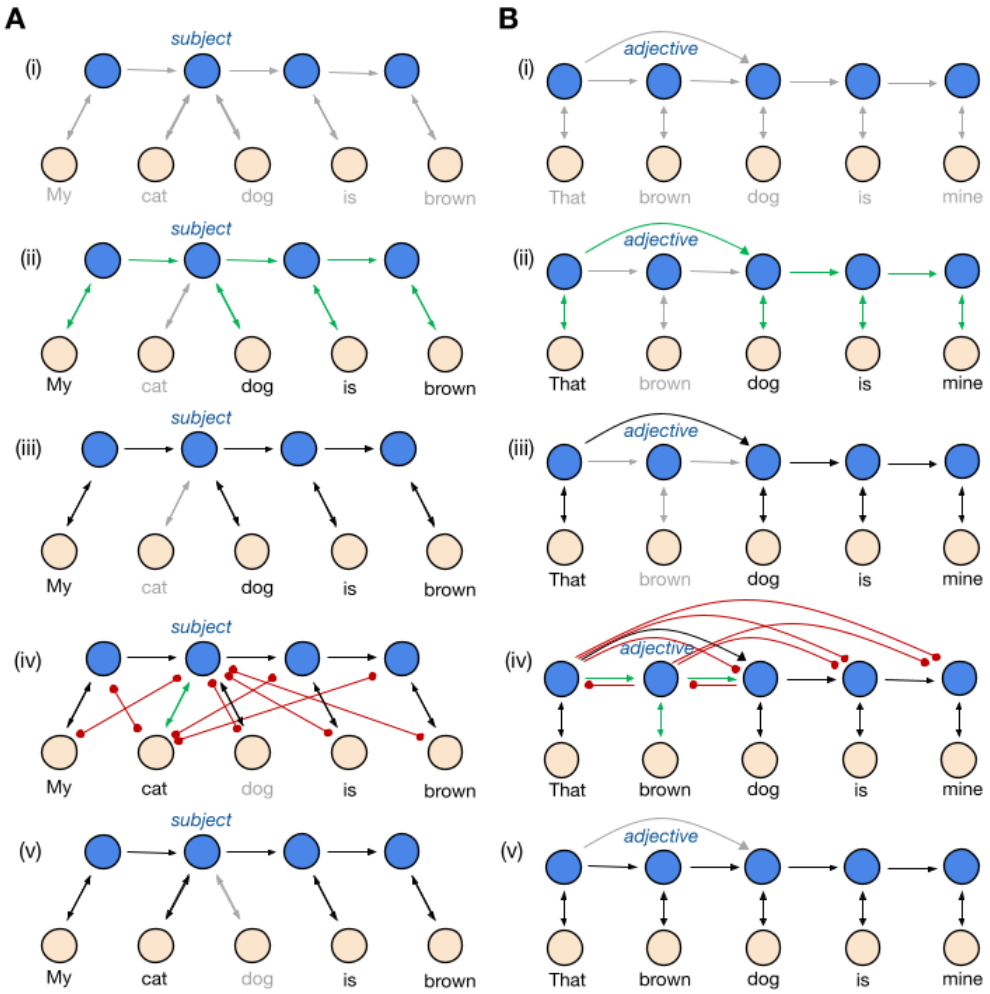

Fig. S8. Graphical visualization of equations S6 and S7.
(**A**) Different words of the same class – e.g. both "cat" and "dog" – can be the subject of a sentence and therefore be connected to the subject role neuron. (i) Initial state of connection weights before any input. Gray arrows: long-term knowledge with no short-term rapid plasticity encoding. (ii) Reinforcement of some connections when presented with the first sentence "My dog is brown". Green arrow: strengthening of connection. (iii) The resultant state after the first sentence, with "dog" associated with the subject role neuron. Black arrows: short-term rapid plasticity encoded. (iv) The concurrent reinforcement and weakening of connections when presented with the second sentence "My cat is brown". (v) Final state of connection weights, with cat associated with the subject role. (**B**) Distinct but overlapping sentence structures – the adjective "brown" may or may not be present in a sentence and therefore the adjective role neuron is skippable. (i) Initial state of connection weights before any input. (ii) Reinforcement of some connections when presented with the first sentence "That dog is mine", with no adjective. (iii) The resultant state after the first sentence. (iv) The concurrent reinforcement and inhibition of connections when presented with the second sentence "That brown dog is mine", recruiting the adjective role neuron. Red circle-ended arrow: weakening of connection. (v) Final state of connection weights, with the adjective neuron in the sequence.

```
Algorithm: at each time step
FOR each timestep:
        IF encoding phase:
                activation input f and m
        ELSE IF recall phase and time step 1-5:
                activation input r
        END IF
        CALCULATE and STORE:
                Temporary store W^fr *f
                Temporary store W^rr * r
                Temporary store W^rf *r
                Temporary store W^ff *f
                Temporary store W^rm *r
                Temporary store W^mr *m
        UPDATE W^rr:
                STRENGTHEN W_ij ∝ r_i * r_j * W^fr_j *f
                WEAKEN W_ij ∝ r_j * r_i * W^fr_i *f
                FOR k = 1 ... :
                        WEAKEN W_ij ∝ r_i * r_k≠j * W^fr_(k≠j) *f
        UPDATE W^fr:
                STRENGTHEN W_ij ∝ r_i * f_j
                FOR k = 1 ... :
                        WEAKEN W_ij ∝ r_k≠i * f_j
                FOR k = 1 ... :
                        WEAKEN W_ij ∝ r_i * f_k≠j
        UPDATE decay of each neuron
        UPDATE temporary store into neurons:
        RESET temporary stores to zero
END time step
```

Fig. S9. High-level pseudocode of the model at each timestep.
The model receives sensory input to its neurons depending on if it is in the encoding or recall phase. The model then stores the output of all the neurons in a temporary memory. Rapid plasticity occurs as the short-term connection weights are updated. Each neuron then undergoes decay and their firing rate is then updated from values in the temporary memory store. The temporary memory store is finally reset to zero.

			Morphemes	Simple grammar	Priming comprehension	Priming production	Serialization aphasias
Simulations			4,5,6	1,2,3,S1,S2,S3	10,11,S5,S6,S7,S8	12	7,8,9,S4
f & **m** Word and morpheme neurons	α	Input strength	1 (During encoding and with input*)				
			-1 (During encoding and without input†)				
			0 (during recall)				
r Role neurons	α	Input strength	0 (at encoding)				
			1 to first role neuron on first 5 timesteps, then 0. (at recall)				
$\mathbf{W}_{\mathrm{L}}^{fr}$ Long-term word-role knowledge	k_{L} ‡	Weight scaling	3.7	1.7			
$\mathbf{W}_{\mathrm{S}}^{fr}$ Short-term word-role memory	h	Hebbian baseline	2			6§	
	k	Minimum	2	1			
	ϵ	Scale	4	1.5			
$\mathbf{W}_{\mathrm{L}}^{rr}$ Long-term role-sequence knowledge	k_{L} ¶	Scale	0	4.5			8.25
$\mathbf{W}_{\mathrm{S}}^{rr}$ Short-term role-sequence memory	λ^{rr}	Learning rate	50#				
	k	Minimum	7.5	4			
	ϵ	Scale	4	0.75			
$\mathbf{W}^{mr}$ Morpheme-role weights		Scale	6			2	

Table S1. Summary table of constants.

*During serialization, $\alpha = 0.8$ for words and morphemes receiving input.

†During serialization, $\alpha = -10$ for words and morphemes not receiving input.

‡The closed-class words have $k_{\mathrm{L}} = 1.45$ for $\mathbf{W}_{\mathrm{L}}^{fr}$.

§The normalization term for simulations 7, 8, 9 and 12 are higher due to more words being concurrently activated.

¶There is no long-term role-to-role knowledge during Simulations 4, 5 and 6. Simulations 7, 8 and 9 require stronger role-to-role long-term bias due to serialization, while simulation 12 is exempt (which also has serialization) due to its priming phase.

#Reduced to 1.25 during simulations 10, 11, S5, S6, S7 and S8 to accentuate priming during comprehension. Switched off to 0 during semantic binding, syntactic serialization and utterance in simulation 12.

5