LING3804: Lecture Notes 4
From Cued Associations to Syntax

We've seen that neurons can rapidly form durable cued associations to store complex relations.

This lecture will show show these cued associations can be used to recognize syntactic structures.

Contents

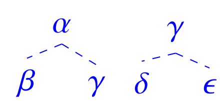
4.1 We compose meanings out of sub-parts [Frege, 1892]

We can assemble meanings by:

1. composing words into phrases and sentences using **grammar rules**; and then
2. associating these words and rules with meanings via function application in lambda calculus.

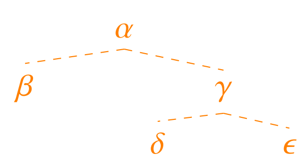
Specifically, we use **context-free grammar rules**, which assign categories to sequences of words.

They are 'context-free' [Chomsky, 1956] because they just have to match categories to apply.

These rules can be written like this: $\alpha \rightarrow \beta\ \gamma$, $\gamma \rightarrow \delta\ \epsilon$ or drawn like this:  ,

(here and subsequently I'll use Greek letters as schemata or 'meta-variables' over complex values).

These rules can be used to assemble **trees** by matching categories (the γ's in both rules match):



These trees will show how sentence meanings are assembled from bottom ('leaves') to top ('root').

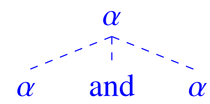
4.2 Human language seems to use syntactic categories [Sag et al., 1985]

Human languages seem to have rich category types we can see in conjunctions.

First, observe that natural languages use different argument structures for different verbs:

- They sleep. – one argument ahead (intransitive)

- They find things. – one argument ahead and one argument behind (transitive)
- They give people things. – one argument ahead, two arguments behind (ditransitive)

Next, observe natural languages *coordinate* conjunctions (combine like types): 

.

- $[_\beta$ $[_\beta$ They sleep] and $[_\beta$ they find things]]. – sounds ok (β is sentence)
- They find $[_\gamma$ $[_\gamma$ people] and $[_\gamma$ things]]. – sounds ok (γ is noun phrase)
- *They find $[_\gamma$ $[_\beta$ they sleep] and $[_\gamma$ things]]. – sounds wrong; conjuncts must match

Now, allowable conjunctions give us insight into the category structure of language:

- They $[_\delta$ $[_\delta$ sleep] and $[_\delta$ find things]]. – sounds ok (δ is verb phrase)
- They $[_\eta$ $[_\eta$ find] and $[_\eta$ give people]] things. – sounds ok (but what's η?)

Transitive verbs (find) match type with ditransitive verb + indirect object (give people)!

Both lack argument ahead and behind – it seems types are defined by missing arguments!

4.3 Categories compose by rules [Ajdukiewicz, 1935, Bar-Hillel, 1953]

Formalize set of categories C as follows – clauses with various unmet requirements:

1. every U is in C, for some set U of primitive categories;
2. every $C \times O \times C$ is in C, for some set O of type-combining operators;
3. nothing else is in C

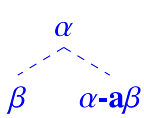
Define **primitive categories** $U = \{\mathbf{N}, \mathbf{V}\}$:

- **N**: noun-headed category with no missing arguments (noun phrase)
- **V**: verb-headed category with no missing arguments (sentence)

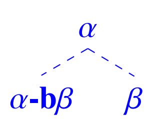
Define **type-combining operators** $O = \{\textbf{-a}, \textbf{-b}\}$:

- $\langle\alpha\textbf{-a}\beta\rangle$: α lacking β argument ahead (e.g. **V-aN** for intransitive δ above)
- $\langle\alpha\textbf{-b}\beta\rangle$: α lacking β argument behind (e.g. **V-aN-bN** for transitive η above)

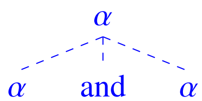
Now we can define **'context-free' rules** R over these categories:

(Aa) 

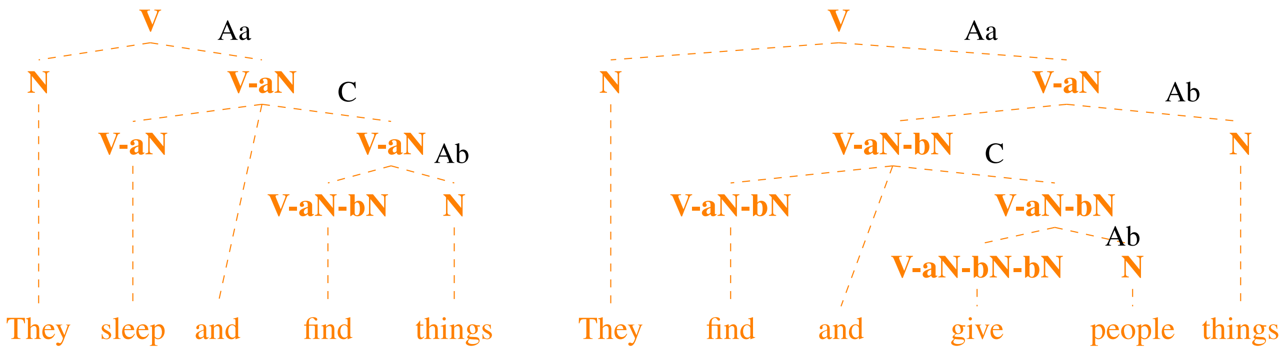
: argument attachment/elimination ahead

(Ab) 

: argument attachment/elimination behind

(C)  : conjunction

These three rules model all of the above sentences:



Also note that the parents in these rules all have simpler types than the children.

This means for any lexicon (constraining types at tree leaves), the set of categories *C* is finite.

Practice 4.1:

1. Draw an analysis tree for the sentence *Paris is in France* using the following categories:
   - **N** for *Paris* and *France*
   - **A-aN-bN** for *in* (**A-aN** is a predicative phrase that follows a form of *be*)
   - **V-aN-b(A-aN)** for *is* (a form of *be*)
2. Label the rules used in your analysis

4.4 Left-corner parsing [Johnson-Laird, 1983, Lewis & Vasishth, 2005]

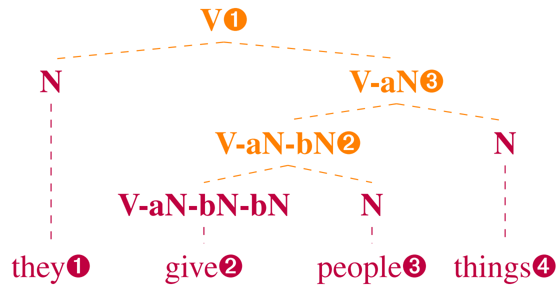
If we need grammar, how does it guide language comprehension?

Psycholinguists and cognitive scientists have proposed a simple two-step recognition process.

The two steps are:

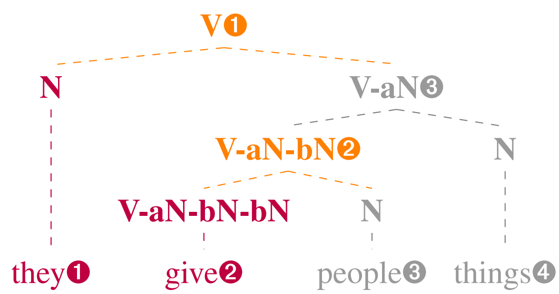
1. add a **lexical item** (word) – there are two different ways to do this to build any tree
2. add a **grammar rule** (tree branch) – there are two different ways to do this to build any tree

First, observe there are as many binary branches as words (minus one) in a binary-branching tree:

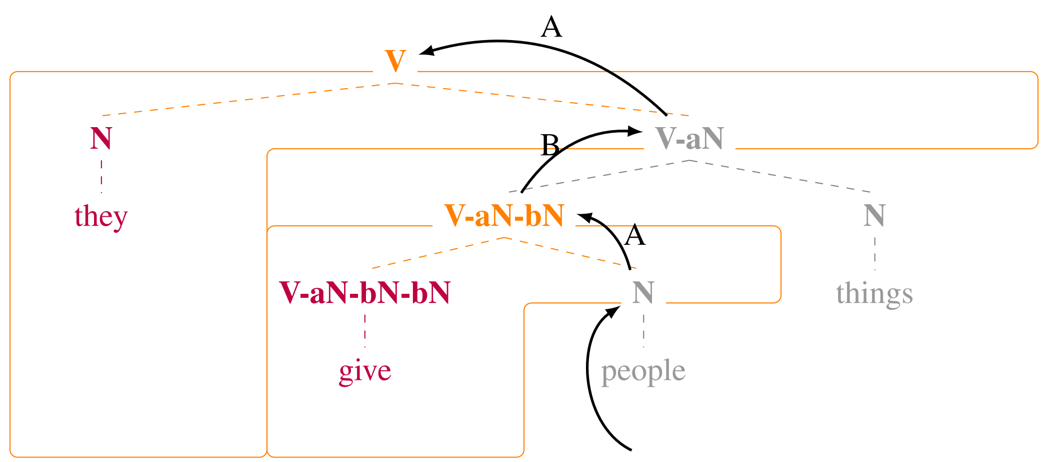


so we can efficiently build meanings by recognizing branches left to right as we recognize words. (We recognize words in time order and recognize branches in the time order of their parents.) This is called **left-corner parsing** [Rosenkrantz & Lewis, 1970].

This left-corner parsing sometimes leaves tree fragments disconnected, e.g. after word 2:



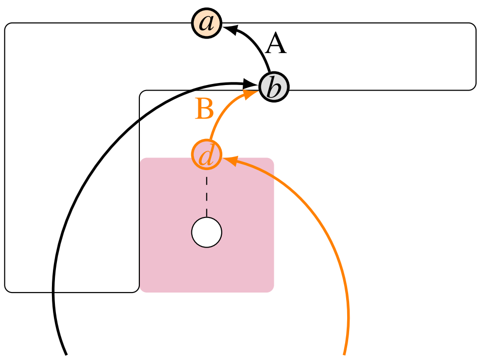
We can use cued associations to keep track of these disconnected fragments in associative memory:



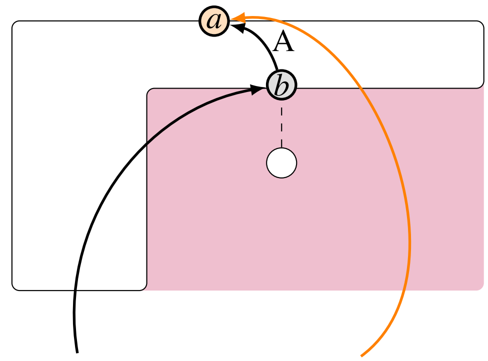
Comprehension proceeds as follows, using alternating lexical and grammatical decisions:

1. a **lexical** decision is made about whether to **match** (merge) fragments at the next word:

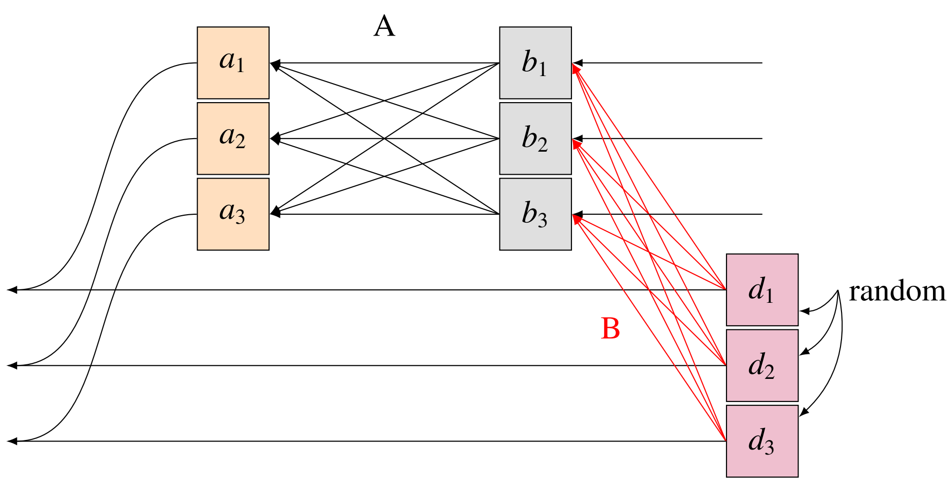
(a) no lexical match:



(b) yes lexical match:



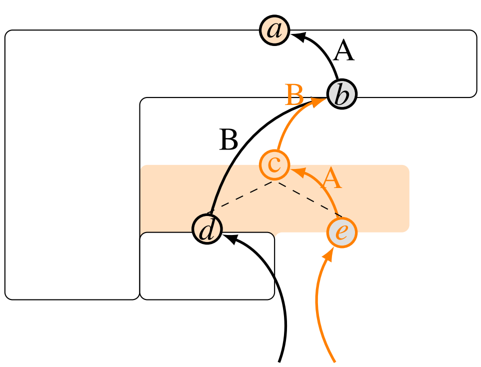
Here's what that looks like as a network:



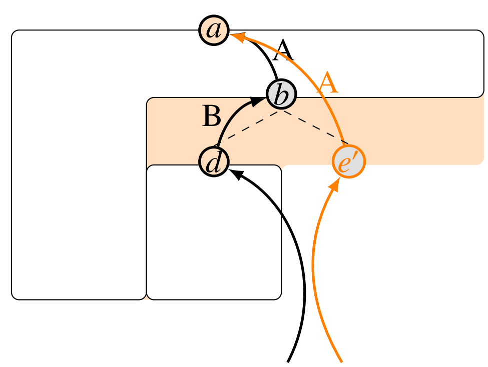
The two possible outcome states, to match with the fragment above or not, are superposed (they are also weighted by grammar rule probabilities, but this is not shown).

2. a **grammatical** decision is made about whether to **match** (merge) fragments at the next rule:

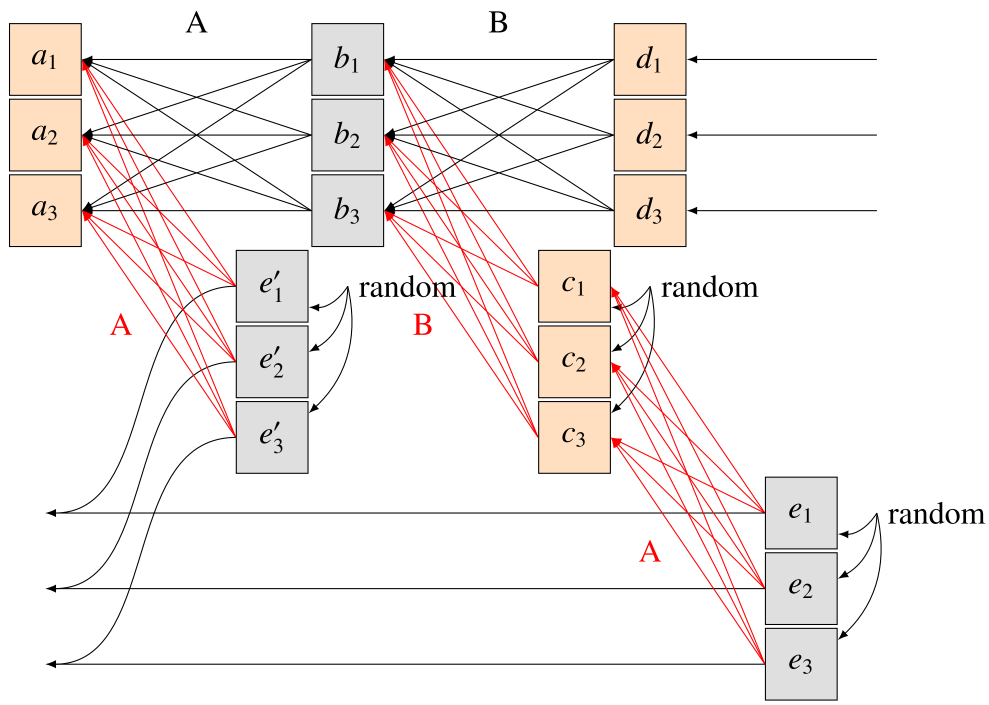
(c) no grammatical match:



(d) yes grammatical match:



Here’s what that looks like as a network:



Again, two possible outcome states, to match or not to match, are weighted and superposed.

So the first network turns gray states into colored ones, and the second turns colored states gray.

These two networks alternate as words are encountered, to build meanings according to grammar.

Think of this as a circuit that goes around and around building meaning using cued associations.

References

[Ajdukiewicz, 1935] Ajdukiewicz, K. (1935). Die syntaktische konnexitat. In S. McCall (Ed.), *Polish Logic 1920-1939* (pp. 207–231). Oxford University Press. Translated from Studia Philosophica 1: 1–27.

[Bar-Hillel, 1953] Bar-Hillel, Y. (1953). A quasi-arithmetical notation for syntactic description. *Language*, 29, 47–58.

[Chomsky, 1956] Chomsky, N. (1956). Three models for language. *IRE Transactions on Information Theory*, 2, 113–124.

[Frege, 1892] Frege, G. (1892). Uber sinn und bedeutung. *Zeitschrift fur Philosophie und Philosophischekritik*, 100, 25–50.

[Johnson-Laird, 1983] Johnson-Laird, P. N. (1983). *Mental models: Towards a cognitive science of language, inference, and consciousness*. Cambridge, MA, USA: Harvard University Press.

[Lewis & Vasishth, 2005] Lewis, R. L. & Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, 29(3), 375–419.

[Rosenkrantz & Lewis, 1970] Rosenkrantz, S. J. & Lewis, II, P. M. (1970). Deterministic left corner parsing. In *IEEE Conference Record of the 11th Annual Symposium on Switching and Automata* (pp. 139–152).

[Sag et al., 1985] Sag, I. A., Gazdar, G., Wasow, T., & Weisler, S. (1985). Coordination and how to distinguish categories. *Natural Language & Linguistic Theory*, 3, 117–171.