LING5702: Lecture Notes 15
A Model of Memory Bounds as Interference

These notes describe results of simulations using a parser based on cued associations.

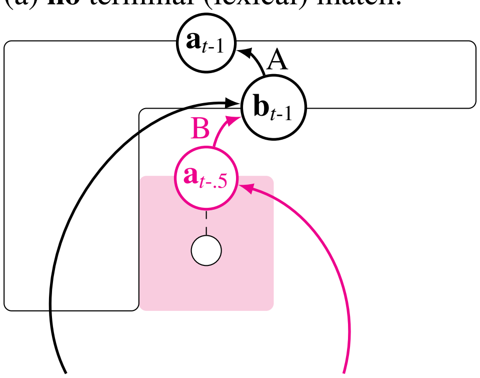
Contents

15.1 Review: parser operations using cued associations

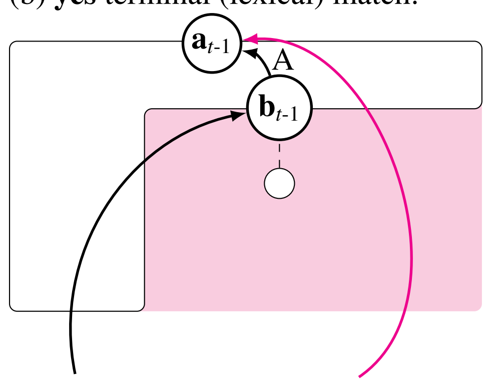
Recall we had defined the following parser operations:

1. a **terminal** decision is made about whether to **match** store elements at the next word, and

   (a) **no** terminal (lexical) match:

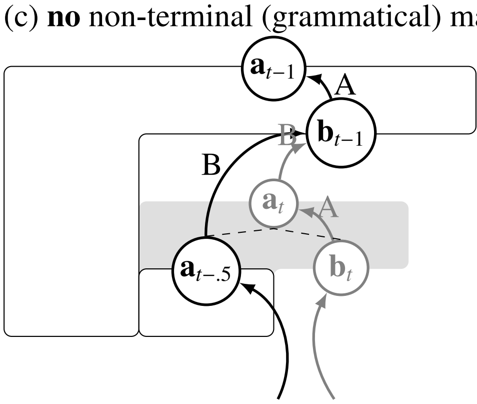


   (b) **yes** terminal (lexical) match:

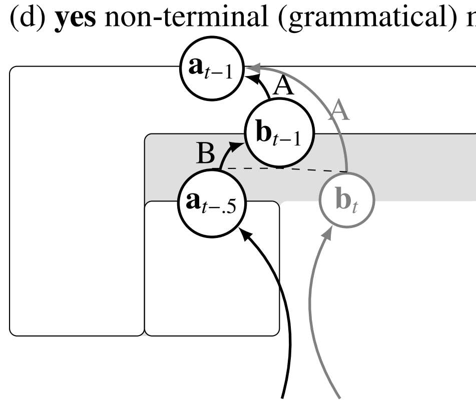


2. a **non-terminal** decision is made about whether to **match** store elements at the next rule,

   (c) **no** non-terminal (grammatical) match:



   (d) **yes** non-terminal (grammatical) match:



15.2 Simulation model [Rasmussen & Schuler, 2018, Schuler & Yue, 2024]

We can define equations for neural circuits that use these operations.

This model maintains a vector as focus of attention in each phase, first at $\mathbf{a}_{t\text{-}.5}$, then at $\mathbf{b}_t$.

(Time step $t - .5$ indicates the terminal phase; time step t indicates the non-terminal phase.)

Equations 'unify' association graphs by cueing two paths, storing (associating) each link as it goes.

Notation:

1. Diagonalization (a simplification to cover learning for now): $\mathrm{diag}(\mathbf{v})$
2. Renormalization (rescale **v** to have unit magnitude, e.g. by iterative search process): $\frac{\mathbf{v}}{\|\mathbf{v}\|}$
3. Kronecker product (implements 'tensor filtering' from lecture notes on ambiguity): $\mathbf{u} \otimes \mathbf{v}$

Initialization. Before processing, the simulation does these steps (this part isn't algorithmic-level):

1. **randomly generates** initial a top-level derivation fragment and category 'T':

$$\mathbf{a}_0 \in \mathbb{R}^d$$
$$\mathbf{b}_0 \in \mathbb{R}^d$$
$$\mathbf{c}_0 \in \mathbb{R}^d$$

2. **associates** the new signs and category in (time-subscripted) associative memory:

$$\mathbf{A}_0 = \mathbf{a}_0\, \mathbf{b}_0^\top \tag{1}$$
$$\mathbf{B}_0 = \mathbf{0}\, \mathbf{0}^\top \tag{2}$$
$$\mathbf{C}_0 = \mathbf{c}_0\, \mathbf{a}_0^\top + \mathbf{c}_0\, \mathbf{b}_0^\top \tag{3}$$

3. **associates** categories with m words and n grammar rules (as parent, left child, right child):

$$\mathbf{L} = \sum_{m=1}^{M} \mathbf{c}_m\, \mathbf{w}_m^\top\, \mathsf{P}(c_m \rightarrow w_m \mid c_m) \tag{4}$$

$$\mathbf{G}_{\mathrm{P}} = \sum_{n=1}^{N} \mathbf{r}_n\, \mathbf{c}_n^\top\, \mathsf{P}(c_n \rightarrow c'_n\, c''_n \mid c_n) \tag{5a}$$
$$\mathbf{G}_{\mathrm{L}} = \sum_{n=1}^{N} \mathbf{r}_n\, \mathbf{c}_n'^{\top} \tag{5b}$$
$$\mathbf{G}_{\mathrm{R}} = \sum_{n=1}^{N} \mathbf{r}_n\, \mathbf{c}_n''^{\top} \tag{5c}$$

4. **associates** categories with categories of left- and right-recursive descendants:

$$\mathbf{D}'_0 = \mathrm{diag}(\mathbf{1}) \tag{6a}$$
$$\mathbf{D}_0 = \mathrm{diag}(\mathbf{0}) \tag{6b}$$

$$\mathbf{D}'_k = \mathbf{G}_\mathrm{L}^\top \, \mathbf{G}_\mathrm{P} \, \mathbf{D}'_{k\text{-}1} \tag{6c}$$
$$\mathbf{D}_k = \mathbf{D}_{k\text{-}1} + \mathbf{D}'_k \tag{6d}$$

$$\mathbf{E}'_0 = \mathrm{diag}(\mathbf{1}) \tag{7a}$$
$$\mathbf{E}_0 = \mathrm{diag}(\mathbf{0}) \tag{7b}$$
$$\mathbf{E}'_k = \mathbf{G}_\mathrm{R}^\top \, \mathbf{G}_\mathrm{P} \, \mathbf{E}'_{k\text{-}1} \tag{7c}$$
$$\mathbf{E}_k = \mathbf{E}_{k\text{-}1} + \mathbf{E}'_k \tag{7d}$$

iterating to a maximum depth of $k = 20$, so $\mathbf{D} = \mathbf{D}_{20}$ and $\mathbf{E} = \mathbf{E}_{20}$.

5. **defines filters** for all category and grammar rule vectors:

$$\mathcal{W} = \sum_{m=1}^{M} \mathbf{w}_m \, (\mathbf{w}_m \otimes \mathbf{w}_m)^\top$$
$$\mathcal{C} = \sum_{m=1}^{M} \mathbf{c}_m \, (\mathbf{c}_m \otimes \mathbf{c}_m)^\top + \sum_{n=1}^{N} \mathbf{c}_n \, (\mathbf{c}_n \otimes \mathbf{c}_n)^\top$$
$$\mathcal{R} = \sum_{n=1}^{N} \mathbf{r}_n \, (\mathbf{r}_n \otimes \mathbf{r}_n)^\top$$

Terminal phase. At every word t, the model:

1. **cues** a new apex sign:

$$\mathbf{a}_{t-1} = \mathbf{A}_{t\text{-}1} \, \mathbf{b}_{t\text{-}1} \tag{8}$$

2. **randomly generates** new signs for yes-match and no-match results:

$$\mathbf{a}_{t-.5,\mathrm{yes}} \in \mathbb{R}^d$$
$$\mathbf{a}_{t-.5,\mathrm{no}} \in \mathbb{R}^d$$

3. **filters** a category label for each match result:

$$\mathbf{w}_{t,\mathrm{yes}} = \mathcal{W} \, (\mathbf{w}_t \otimes \mathbf{L}^\top \mathbf{C}_{t\text{-}1} \, \mathbf{b}_{t\text{-}1}) \tag{9a}$$
$$\mathbf{w}_{t,\mathrm{no}} = \mathcal{W} \, (\mathbf{w}_t \otimes \mathbf{L}^\top \mathbf{D} \, \mathbf{C}_{t\text{-}1} \, \mathbf{b}_{t\text{-}1}) \tag{9b}$$

4. **superposes** the possible signs in attentional focus, weighted by magnitudes of categories:

$$\mathbf{a}_{t-.5} = \frac{(||\mathbf{w}_{t,\mathrm{yes}}|| \, \mathbf{a}_{t-.5,\mathrm{yes}}) + (||\mathbf{w}_{t,\mathrm{no}}|| \, \mathbf{a}_{t-.5,\mathrm{no}})}{||\,(||\mathbf{w}_{t,\mathrm{yes}}|| \, \mathbf{a}_{t-.5,\mathrm{yes}}) + (||\mathbf{w}_{t,\mathrm{no}}|| \, \mathbf{a}_{t-.5,\mathrm{no}})\,||} \tag{10}$$

5. **associates** the new signs with categories and with the remainder of the analysis:

$$\begin{aligned}
\mathbf{C}_{t\text{-}.5} &= \mathbf{C}_{t\text{-}1} + \left(\frac{\mathbf{L}\,\mathbf{w}_{t,\text{no}}}{\|\,\mathbf{L}\,\mathbf{w}_{t,\text{no}}\,\|} - \mathbf{C}_{t\text{-}1}\,\mathbf{a}_{t-.5,\text{no}} \right) \mathbf{a}_{t-.5,\text{no}}{}^{\top} \\
&\quad + \left(\frac{\mathcal{C}\,(\mathbf{C}_{t\text{-}1}\,\mathbf{a}_{t-1} \otimes \mathbf{E}^{\top}\mathbf{L}\,\mathbf{w}_{t,\text{yes}})}{\|\,\mathcal{C}\,(\mathbf{C}_{t\text{-}1}\,\mathbf{a}_{t-1} \otimes \mathbf{E}^{\top}\mathbf{L}\,\mathbf{w}_{t,\text{yes}})\,\|} - \mathbf{C}_{t\text{-}1}\,\mathbf{a}_{t-.5,\text{yes}} \right) \mathbf{a}_{t-.5,\text{yes}}{}^{\top} && (11) \\
\mathbf{B}_{t\text{-}.5} &= \mathbf{B}_{t\text{-}1} + (\mathbf{b}_{t\text{-}1} - \mathbf{B}_{t\text{-}1}\,\mathbf{a}_{t-.5,\text{no}})\,\mathbf{a}_{t-.5,\text{no}}{}^{\top} + (\mathbf{B}_{t\text{-}1}\,\mathbf{a}_{t-1} - \mathbf{B}_{t\text{-}1}\,\mathbf{a}_{t-.5,\text{yes}})\,\mathbf{a}_{t-.5,\text{yes}}{}^{\top} && (12)
\end{aligned}$$

Non-terminal phase. Similarly, after each terminal phase, the model:

1. **cues** a new base sign:

$$\mathbf{b}_{t-.5} = \mathbf{B}_{t\text{-}.5}\,\mathbf{a}_{t-.5} \tag{13}$$

2. **randomly generates** a new sign for the no-match case ($\mathbf{a}_{t,\text{yes}}$ is just old apex), and new base:

$$\begin{aligned}
\mathbf{a}_{t,\text{no}} &\in \mathbb{R}^d \\
\mathbf{b}_t &\in \mathbb{R}^d
\end{aligned}$$

3. **filters** a grammar rule for each match result:

$$\begin{aligned}
\mathbf{r}_{t,\text{yes}} &= \mathcal{R}\,(\mathbf{G}_{\text{L}}\,\mathbf{C}_{t\text{-}.5}\,\mathbf{a}_{t-.5} \otimes \mathbf{G}_{\text{P}}\,\mathbf{C}_{t\text{-}.5}\,\mathbf{b}_{t-.5}) && (14a) \\
\mathbf{r}_{t,\text{no}} &= \mathcal{R}\,(\mathbf{G}_{\text{L}}\,\mathbf{C}_{t\text{-}.5}\,\mathbf{a}_{t-.5} \otimes \mathbf{G}_{\text{P}}\,\mathbf{D}\,\mathbf{C}_{t\text{-}.5}\,\mathbf{b}_{t-.5}) && (14b)
\end{aligned}$$

4. **superposes** the two possible signs as a new apex, weighted by magnitude of grammar rules:

$$\mathbf{a}_t = \frac{(\|\mathbf{r}_{t,\text{yes}}\|\,\mathbf{A}_{t-.5}\,\mathbf{b}_{t-.5}) + (\|\mathbf{r}_{t,\text{no}}\|\,\mathbf{a}_{t,\text{no}})}{\|\,(\|\mathbf{r}_{t,\text{yes}}\|\,\mathbf{A}_{t-.5}\,\mathbf{b}_{t-.5}) + (\|\mathbf{r}_{t,\text{no}}\|\,\mathbf{a}_{t,\text{no}})\,\|} \tag{15}$$

5. **associates** the possible signs with categories and the remainder of the analysis:

$$\begin{aligned}
\mathbf{A}_t &= \mathbf{A}_{t\text{-}1} + (\mathbf{a}_t - \mathbf{A}_{t\text{-}1}\,\mathbf{b}_t)\,\mathbf{b}_t^{\top} && (16) \\
\mathbf{B}_t &= \mathbf{B}_{t\text{-}.5} + (\mathbf{b}_{t-.5} - \mathbf{B}_{t\text{-}.5}\,\mathbf{a}_{t,\text{no}})\,\mathbf{a}_{t,\text{no}}{}^{\top} && (17) \\
\mathbf{C}_t &= \mathbf{C}_{t\text{-}.5} + \left(\frac{\mathbf{G}_{\text{P}}^{\top}\mathbf{r}_{t,\text{no}}}{\|\,\mathbf{G}_{\text{P}}^{\top}\mathbf{r}_{t,\text{no}}\,\|} - \mathbf{C}_{t\text{-}.5}\,\mathbf{a}_{t,\text{no}} \right) \mathbf{a}_{t,\text{no}}{}^{\top} + \left(\frac{\mathbf{G}_{\text{R}}^{\top}\mathbf{r}_{t,\text{yes}} + \mathbf{G}_{\text{R}}^{\top}\mathbf{r}_{t,\text{no}}}{\|\,\mathbf{G}_{\text{R}}^{\top}\mathbf{r}_{t,\text{yes}} + \mathbf{G}_{\text{R}}^{\top}\mathbf{r}_{t,\text{no}}\,\|} - \mathbf{C}_{t\text{-}.5}\,\mathbf{b}_t \right) \mathbf{b}_t^{\top} && (18)
\end{aligned}$$

15.3 Simulation results for expectation effect [Schuler & Yue, 2024]

Grammar aligned with intransitive *dressed*:

P (T → S T) = 1.0	P (D → the) = 1.0
P (S → PP S) = 0.5	P (N → baby) = 1.0
P (S → NP VP) = 0.5	P (PP → P S) = 1.0
P (NP → D N) = 0.5	P (VP → cried) = 0.5
P (NP → Susan) = 0.5	P (VP → dressed) = 0.5
P (P → as) = 1.0	

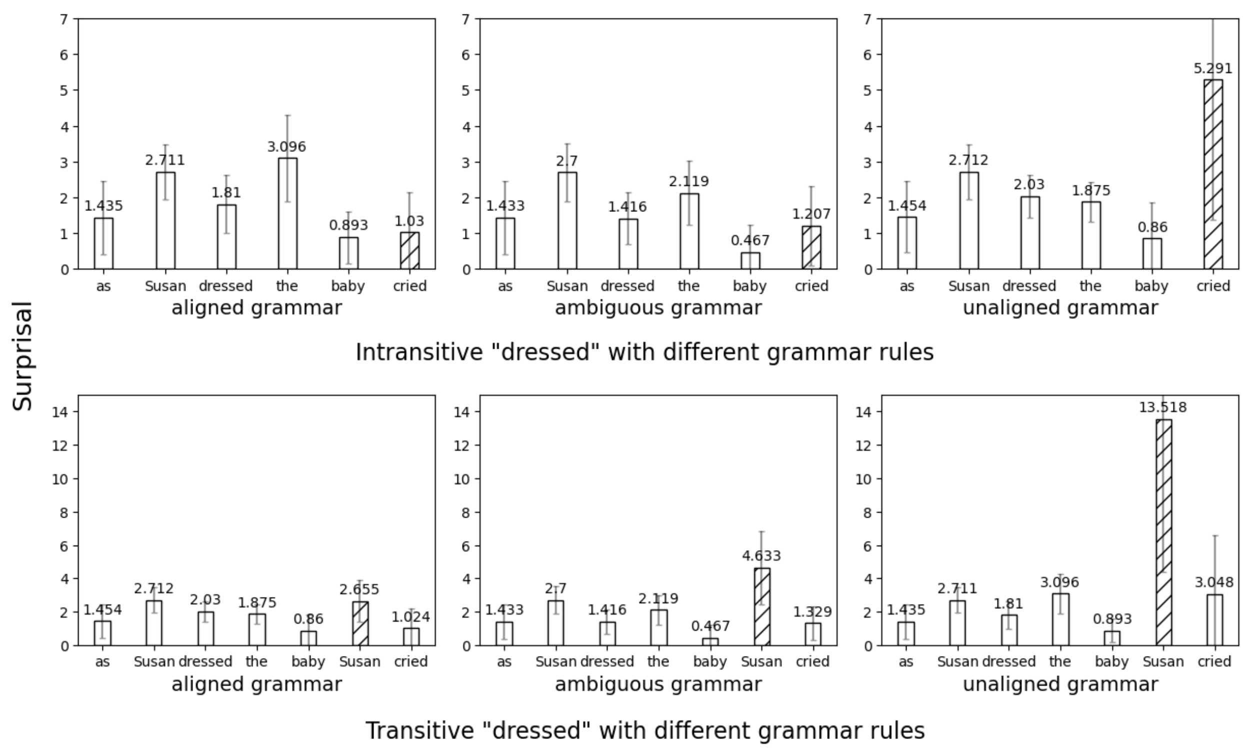
Grammar aligned with transitive *dressed*:

P (T → S T) = 1.0	P (D → the) = 1.0
P (S → PP S) = 0.5	P (N → baby) = 1.0
P (S → NP VP) = 0.5	P (PP → P S) = 1.0
P (NP → D N) = 0.5	P (VP → cried) = 0.5
P (NP → Susan) = 0.5	P (VP → VT NP) = 0.5
P (P → as) = 1.0	P (VT → dressed) = 1.0

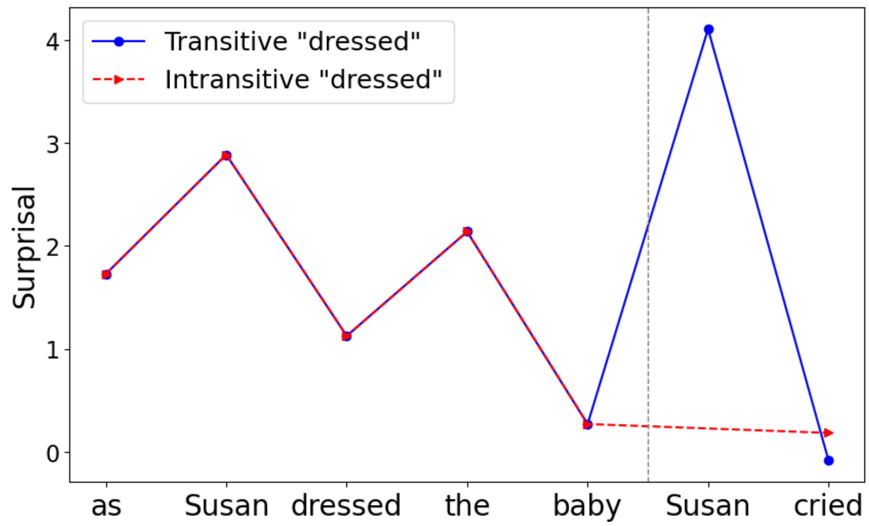
Ambiguous grammar:

P (T → S T) = 1.0	P (D → the) = 1.0
P (S → PP S) = 0.5	P (N → baby) = 1.0
P (S → NP VP) = 0.5	P (PP → P S) = 1.0
P (NP → D N) = 0.5	P (VP → cried) = 0.33
P (NP → Susan) = 0.5	P (VP → VT NP) = 0.33
P (P → as) = 1.0	P (VP → dressed) = 0.33
	P (VT → dressed) = 1.0

Surprisal predictions:



Surprisal in a run with shared prefix and different continuations using ambiguous grammar:

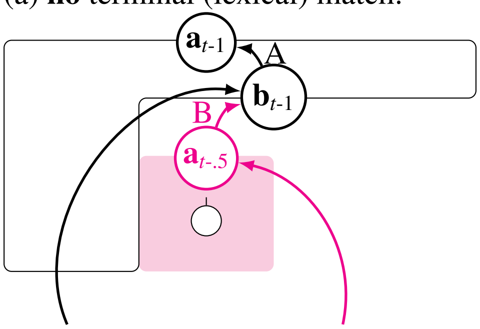


15.4 Review: parser operations use different amounts of cued associations

Comprehension proceeds as follows, using modified terminal and non-terminal decisions:

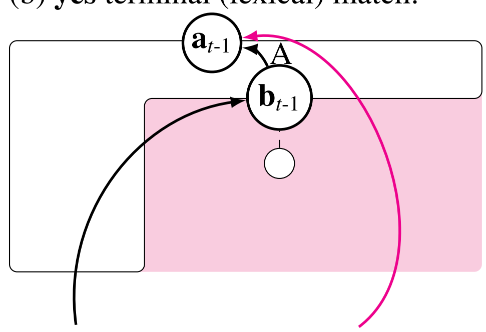
1. a **terminal** decision is made about whether to **match** store elements at the next word, and

   (a) **no** terminal (lexical) match:

   — **no** associations cued before any form

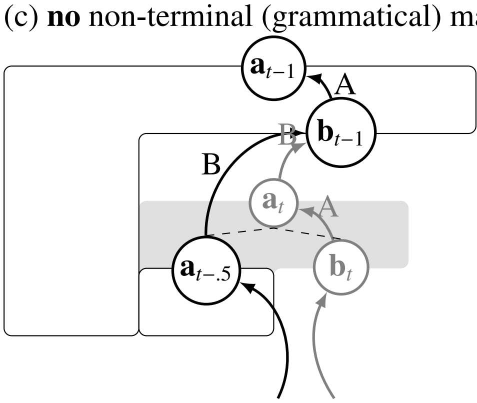
   (b) **yes** terminal (lexical) match:

   — **one** association cued before any form

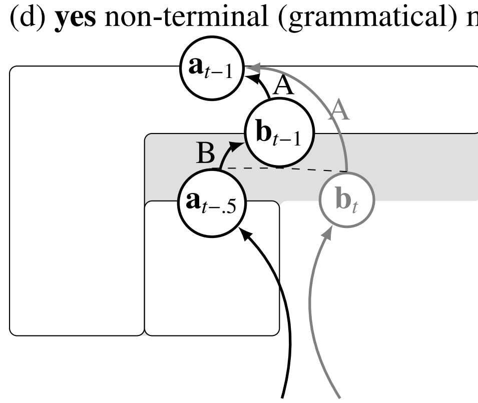
2. a **non-terminal** decision is made about whether to **match** store elements at the next rule,

   (c) **no** non-terminal (grammatical) match:

   — **one** association cued before any form

   (d) **yes** non-terminal (grammatical) match:

   — **two** associations cued before any form

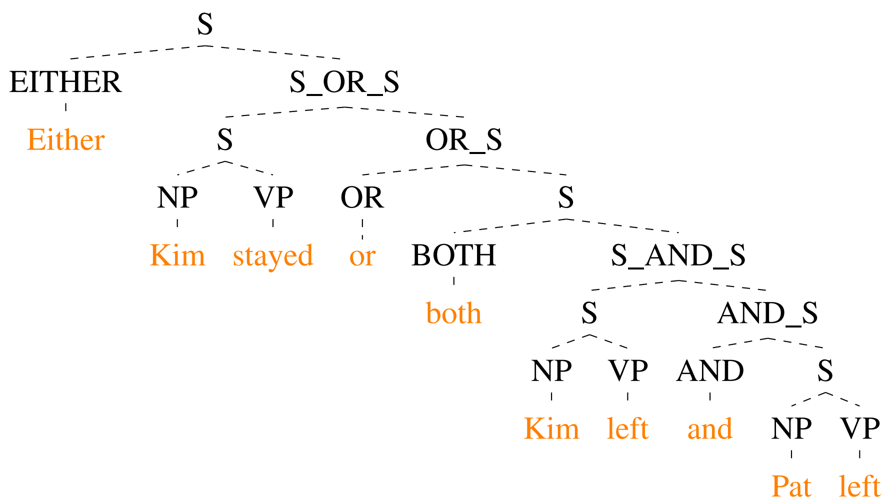
15.5 More cued associations mean more risk of interference

As cued associations for the same sentence are added, the risk of interference increases.

Perfect cueing of each target must avoid all other interfering cues.

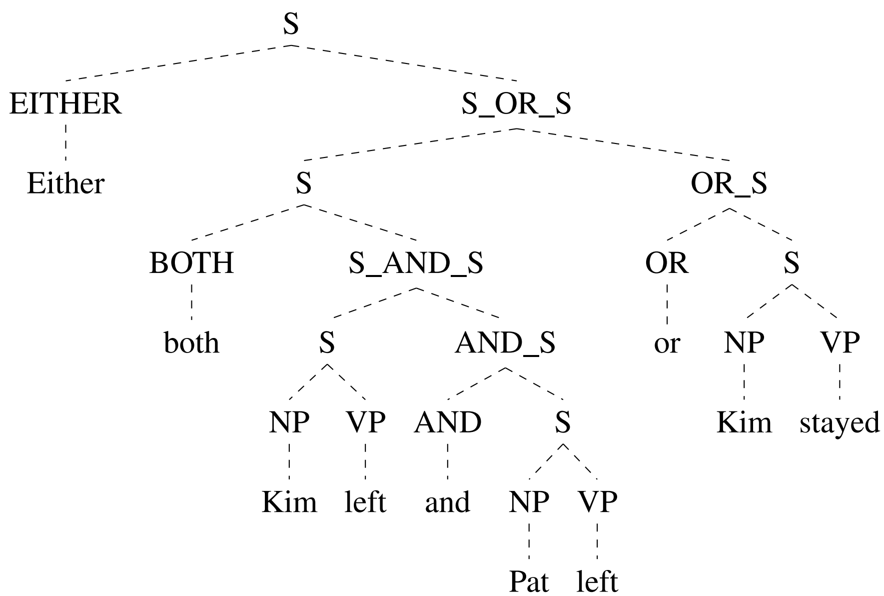
Operations that involve more cueing should happen earlier, to avoid interference.

1. For example, a right branching structure does cueing early, encounters less clutter:



	singly center-embedded sentence			
step	operations	avoidances	cumu	resulting store ; remaining input
0	(initial)			T/T ; *Either Kim stayed or ...*
1	T:no N:no	$0 \times 1 =$	0	T/T, S/S_OR_S ; *Kim ...*
2	T:no N:no	$+ 1 \times 1 =$	1	T/T, S/S_OR_S, S/VP ; *stayed ...*
3	T:yes N:yes	$+ 2 \times 3 =$	7	T/T, S/OR_S ; *or ...*
4	T:no N:yes	$+ 3 \times 2 =$	13	T/T, S/S ; *both ...*
5	T:no N:yes	$+ 4 \times 2 =$	21	T/T, S/S_AND_S ; *Kim ...*
6	T:no N:no	$+ 5 \times 1 =$	26	T/T, S/S_AND_S, S/VP ; *left ...*
7	T:yes N:yes	$+ 6 \times 3 =$	44	T/T, S/AND_S ; *and ...*
8	T:no N:yes	$+ 7 \times 2 =$	58	T/T, S/S ; *Pat ...*
9	T:no N:yes	$+ 8 \times 2 =$	74	T/T, S/VP ; *left*
10	T:yes N:yes	$+ 9 \times 3 =$	101	T/T ;

2. A center embedded structure does cueing later, encounters more clutter:



step	doubly center-embedded sentence operations	avoidances	cumu	resulting store ; remaining input
0	(initial)			T/T ; *Either both Kim left and . . .*
1	T:no N:no	$0 \times 1 =$	0	T/T, S/S_OR_S ; *both . . .*
2	T:no N:no	$+ 1 \times 1 =$	1	T/T, S/S_OR_S, S/S_AND_S ; *Kim . . .*
3	T:no N:no	$+ 2 \times 1 =$	3	T/T, S/S_OR_S, S/S_AND_S, S/VP ; *left . . .*
4	T:yes N:yes	$+ 3 \times 3 =$	12	T/T, S/S_OR_S, S/AND_S ; *and . . .*
5	T:no N:yes	$+ 4 \times 2 =$	20	T/T, S/S_OR_S, S/S ; *Pat . . .*
6	T:no N:yes	$+ 5 \times 2 =$	30	T/T, S/S_OR_S, S/VP ; *left . . .*
7	T:yes N:yes	$+ 6 \times 3 =$	48	T/T, S/OR_S ; *or . . .*
8	T:no N:yes	$+ 7 \times 2 =$	62	T/T, S/S ; *Kim . . .*
9	T:no N:yes	$+ 8 \times 2 =$	78	T/T, S/VP ; *stayed*
10	T:yes N:yes	$+ 9 \times 3 =$	105	T/T ;

15.6 Simulation results for memory effects [Rasmussen & Schuler, 2018]

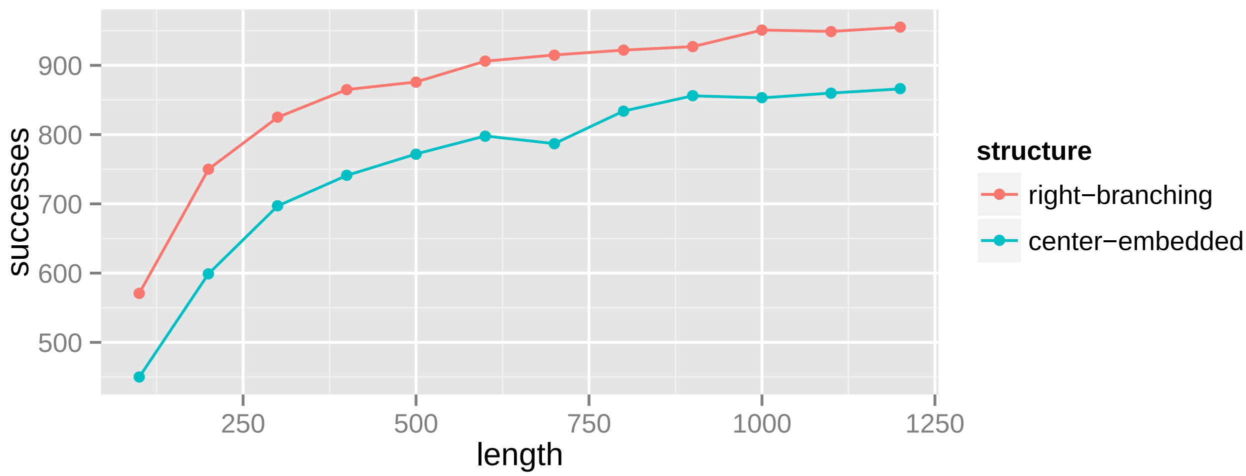
The model was run on this grammar, measuring the accuracy of retrieving the end category ‘T’:

$$\mathsf{P(S \rightarrow NP\ VP) = 0.5} \qquad \mathsf{P(NP \rightarrow kim) = 0.5}$$
$$\mathsf{P(S \rightarrow EITHER\ S\ OR\ S) = 0.25} \qquad \mathsf{P(NP \rightarrow pat) = 0.5}$$
$$\mathsf{P(S \rightarrow BOTH\ S\ AND\ S) = 0.25} \qquad \mathsf{P(BOTH \rightarrow both) = 1.0}$$
$$\mathsf{P(VP \rightarrow leaves) = 0.5} \qquad \mathsf{P(AND \rightarrow and) = 1.0}$$
$$\mathsf{P(VP \rightarrow stays) = 0.5} \qquad \mathsf{P(EITHER \rightarrow either) = 1.0}$$
$$\mathsf{P(OR \rightarrow or) = 1.0}$$

Like people, it shows higher difficulty for center embedding:

sentence	correct	incorrect
center-embedded	470	530
right-branching	555	445

The effect persists even as the vector size increases, suggesting it's not just due to capacity bounds:



References

[Rasmussen & Schuler, 2018] Rasmussen, N. E. & Schuler, W. (2018). Left-corner parsing with distributed associative memory produces surprisal and locality effects. *Cognitive Science*, 42(S4), 1009–1042.

[Schuler & Yue, 2024] Schuler, W. & Yue, S. (2024). Evaluation of an algorithmic-level left-corner parsing account of surprisal effects. *Cognitive Science*, 48(10), e13500.