Psycholinguistic adequacy of left-corner parsing for Minimalist Grammar

Lei Liu

November 30, 2023

Cyclop Retreat WS23

Previously...

Previously...

- Tenure:

Previously...

- Tenure:
  - how long nodes are retained in memory

Previously...

- Tenure:
  - how long nodes are retained in memory
  - reliable metric for processing difficulties

Previously...

- Tenure:
  - how long nodes are retained in memory
  - reliable metric for processing difficulties
    - (based on top-down MG parsing) 2022 Winter

Previously...

- Tenure:
  - how long nodes are retained in memory
  - reliable metric for processing difficulties
    - (based on top-down MG parsing) 2022 Winter
- Left-corner parsing:

Previously...

- Tenure:
  - how long nodes are retained in memory
  - reliable metric for processing difficulties
    - (based on top-down MG parsing)   2022 Winter
- Left-corner parsing:
  - predictive, incremental, input-driven

Previously...

- Tenure:
  - how long nodes are retained in memory
  - reliable metric for processing difficulties
    - (based on top-down MG parsing)

2022 Winter

- Left-corner parsing:
  - predictive, incremental, input-driven
  - shown to capture human parser

Previously...

- Tenure:
  - how long nodes are retained in memory
  - reliable metric for processing difficulties
    - (based on top-down MG parsing) 2022 Winter
- Left-corner parsing:
  - predictive, incremental, input-driven
  - shown to capture human parser
    - (based on Context Free Grammars) 2023 Summer

Previously...

- Tenure:
  - how long nodes are retained in memory
  - reliable metric for processing difficulties
    - (based on top-down MG parsing) 2022 Winter
- Left-corner parsing:
  - predictive, incremental, input-driven
  - shown to capture human parser
    - (based on Context Free Grammars) 2023 Summer
- The goal:

Previously...

- Tenure:
  - how long nodes are retained in memory
  - reliable metric for processing difficulties
    - (based on top-down MG parsing) 2022 Winter
- Left-corner parsing:
  - predictive, incremental, input-driven
  - shown to capture human parser
    - (based on Context Free Grammars) 2023 Summer
- The goal:
  - Tenure in Left-corner MG parsing

Previously...

- Tenure:
  - how long nodes are retained in memory
  - reliable metric for processing difficulties
    - (based on top-down MG parsing) 2022 Winter
- Left-corner parsing:
  - predictive, incremental, input-driven
  - shown to capture human parser
    - (based on Context Free Grammars) 2023 Summer
- The goal:
  - Tenure in Left-corner MG parsing

Previously...

- Tenure:
  - how long nodes are retained in memory
  - reliable metric for processing difficulties
    - (based on top-down MG parsing) 2022 Winter
- Left-corner parsing:
  - predictive, incremental, input-driven
  - shown to capture human parser
    - (based on Context Free Grammars) 2023 Summer
- The goal:
  - Tenure in Left-corner MG parsing

works as it should!

The presentation in bullet points:

The presentation in bullet points:

- Tenure (still) captures processing load in Left-corner parsing for Minimalist Grammars.

The presentation in bullet points:

- Tenure (still) captures processing load in Left-corner parsing for Minimalist Grammars.
- LC parsing for MG adequately models how humans process left-, right-, and center-embedding sentences.

Outline

Outline

1. Intro: embedding and psycholinguistic adequacy
2. Modeling embedding in LC parsing for MG
   - Tenure in LC parsing for MG
   - Modeling results
3. Conclusion

Intro: embedding and psycholinguistic adequacy

Embedding

(1) Left-embedding

a. the cat's rat

b. the cat's rat's cheese

c. the cat's rat's cheese's eyes

Embedding

(1) Left-embedding

a. the cat's rat

b. the cat's rat's cheese

c. the cat's rat's cheese's eyes

Embedding

(1) Left-embedding

  a. the cat's rat

  b. the cat's rat's cheese

  c. the cat's rat's cheese's eyes

(2) Center-embedding

  a. the rat that the cat bit

  b. the cheese that the rat that the cat bit ate

  c. the eyes that the cheese the rat that the cat bit ate had eyes

Embedding

(1) Left-embedding

a. the cat's rat

b. the cat's rat's cheese

c. the cat's rat's cheese's eyes

(2) Center-embedding

a. the rat that the cat bit

b. the cheese that the rat that the cat bit ate

c. the eyes that the cheese the rat that the cat bit ate had eyes

(3) Right-embedding

a. the cat that bit the rat

b. the cat that bit the rat that ate the cheese

c. the cat that bit the rat that ate the cheese that had eyes

Embedding

- Multiple left-, right- embedding: OK!

Embedding

- Multiple left-, right- embedding: OK!
- Multiple center-embedding: terrible!

Embedding

- Multiple left-, right- embedding: OK!
- Multiple center-embedding: terrible!

Embedding

- Multiple left-, right- embedding: OK!
- Multiple center-embedding: terrible!

(2) Center-embedding

a. the rat$_i$ that the cat bit t_i

b. the cheese$_i$ that the rat that the cat bit ate t_i

c. the eyes$_i$ that the cheese the rat that the cat bit ate had t_i

(3) Right-embedding

a. the cat$_i$ that t_i bit the rat

b. the cat$_i$ that t_i bit the rat that ate the cheese

c. the cat$_i$ that t_i bit the rat that ate the cheese that had eyes

Embedding

- Multiple left-, right- embedding: OK!
- Multiple center-embedding: terrible!

Embedding

- Multiple left-, right- embedding: OK!
- Multiple center-embedding: terrible!

Parser	Left	Center	Right
Top-down	O(n)	O(n)	O(n)
➞Left-corner	O(1)	O(n)	O(1)
➞Human	O(1)	O(n)	O(1)

Table 1: adapted from Resnik (1992)

Embedding

- Multiple left-, right- embedding: constant space
- Multiple center-embedding: space proportional to tree height

Parser	Left	Center	Right
Top-down	O(n)	O(n)	O(n)
➞Left-corner	O(1)	O(n)	O(1)
➞Human	O(1)	O(n)	O(1)

Table 1: adapted from Resnik (1992)

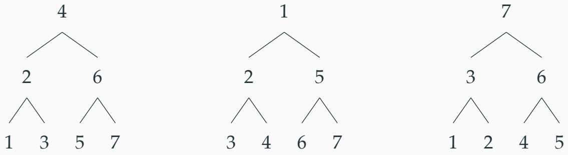
Modeling embedding in LC parsing for MG

Left-corner parsing (for MG)

- Parsing strategy as tree traversal: Resnik (1992)

Left-corner parsing (for MG)

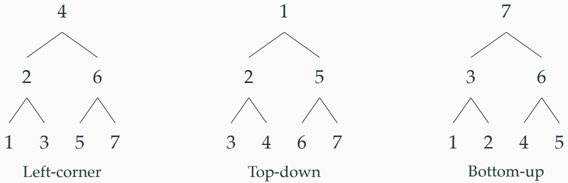
- Parsing strategy as tree traversal: Resnik (1992)



Left-corner Top-down Bottom-up

Left-corner parsing (for MG)

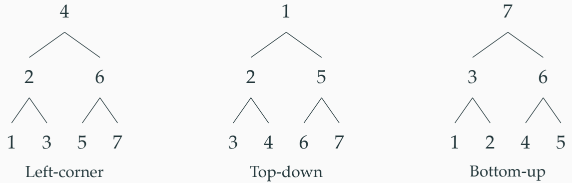
- Parsing strategy as tree traversal: Resnik (1992)



- Previous work

Left-corner parsing (for MG)

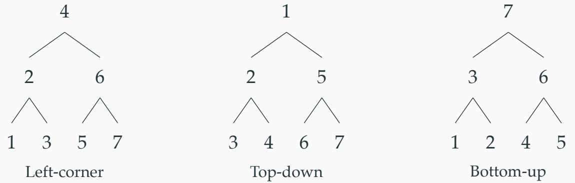
- Parsing strategy as tree traversal: Resnik (1992)



- Previous work
    - LC parser, arc-strategies Stanojević and Stabler (2018)

Left-corner parsing (for MG)

- Parsing strategy as tree traversal: Resnik (1992)

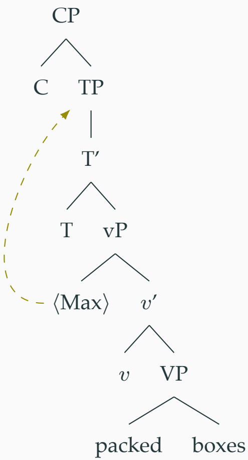


- Previous work
    - LC parser, arc-strategies Stanojević and Stabler (2018)
    - Move-strategies Hunter et al. (2019)

Tenure in LC parsing for MG

(4) • C Max T v packed boxes.

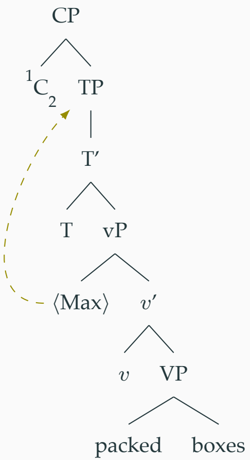
Step 1 read C
Step 2 predict CP, TP
Step 3 read Max
Step 4 predict vP, v'
Step 5 read T
Step 6 predict T′, vP
Step 7 connect down
Step 8 un-move
Step 9 connect up
Step 10 read v
Step 11 predict v', VP
Step 12 connect down
Step 13 read packed
Step 14 predict VP, boxes
Step 15 connect down
Step 16 read boxes and complete



Tenure in LC parsing for MG

(4) C • Max T *v* packed boxes.

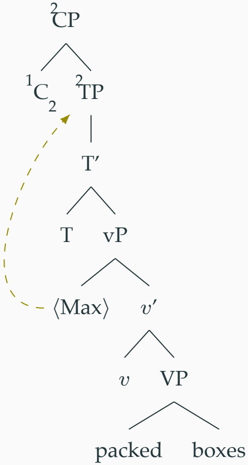
Step 1 read C
Step 2 predict CP, TP
Step 3 read Max
Step 4 predict *v*P, *v*′
Step 5 read T
Step 6 predict T′, *v*P
Step 7 connect down
Step 8 un-move
Step 9 connect up
Step 10 read *v*
Step 11 predict *v*′, VP
Step 12 connect down
Step 13 read packed
Step 14 predict VP, boxes
Step 15 connect down
Step 16 read boxes and complete



Tenure in LC parsing for MG

(4) C • Max T *v* packed boxes.

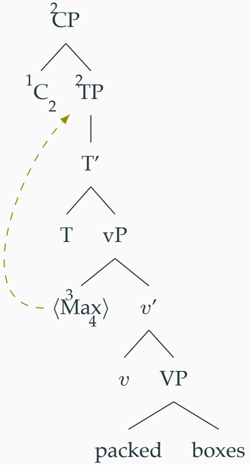
Step 1 read C
Step 2 predict CP, TP
Step 3 read Max
Step 4 predict *v*P, *v*′
Step 5 read T
Step 6 predict T′, *v*P
Step 7 connect down
Step 8 un-move
Step 9 connect up
Step 10 read *v*
Step 11 predict *v*′, VP
Step 12 connect down
Step 13 read packed
Step 14 predict VP, boxes
Step 15 connect down
Step 16 read boxes and complete



Tenure in LC parsing for MG

(4) C Max • T *v* packed boxes.

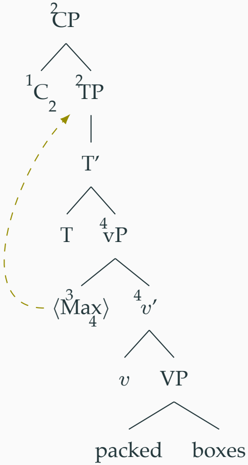
Step 1 read C
Step 2 predict CP, TP
Step 3 read Max
Step 4 predict *v*P, *v*′
Step 5 read T
Step 6 predict T′, *v*P
Step 7 connect down
Step 8 un-move
Step 9 connect up
Step 10 read *v*
Step 11 predict *v*′, VP
Step 12 connect down
Step 13 read packed
Step 14 predict VP, boxes
Step 15 connect down
Step 16 read boxes and complete



Tenure in LC parsing for MG

(4) C Max • T *v* packed boxes.

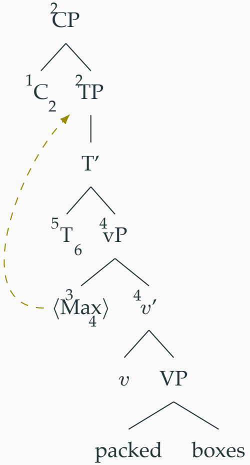
Step 1 read C
Step 2 predict CP, TP
Step 3 read Max
Step 4 predict *v*P, *v*′
Step 5 read T
Step 6 predict T′, *v*P
Step 7 connect down
Step 8 un-move
Step 9 connect up
Step 10 read *v*
Step 11 predict *v*′, VP
Step 12 connect down
Step 13 read packed
Step 14 predict VP, boxes
Step 15 connect down
Step 16 read boxes and complete



Tenure in LC parsing for MG

(4) C Max T • v packed boxes.

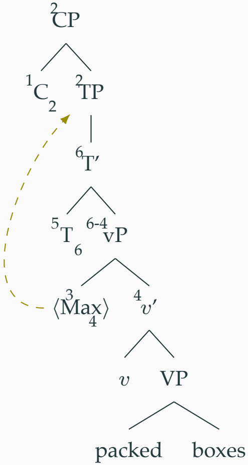
Step 1 read C
Step 2 predict CP, TP
Step 3 read Max
Step 4 predict vP, v'
Step 5 read T
Step 6 predict T′, vP
Step 7 connect down
Step 8 un-move
Step 9 connect up
Step 10 read v
Step 11 predict v', VP
Step 12 connect down
Step 13 read packed
Step 14 predict VP, boxes
Step 15 connect down
Step 16 read boxes and complete



Tenure in LC parsing for MG

(4) C Max T • *v* packed boxes.

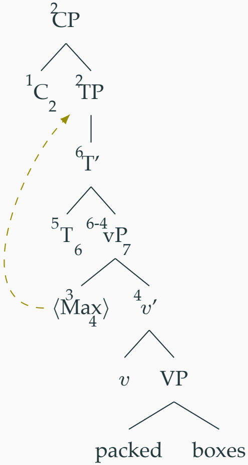
Step 1 read C
Step 2 predict CP, TP
Step 3 read Max
Step 4 predict *v*P, *v*′
Step 5 read T
Step 6 predict T′, *v*P
Step 7 connect down
Step 8 un-move
Step 9 connect up
Step 10 read *v*
Step 11 predict *v*′, VP
Step 12 connect down
Step 13 read packed
Step 14 predict VP, boxes
Step 15 connect down
Step 16 read boxes and complete



Tenure in LC parsing for MG

(4) C Max T • *v* packed boxes.

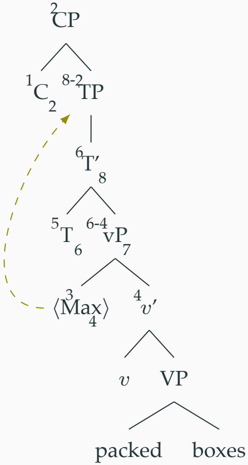
Step 1 read C
Step 2 predict CP, TP
Step 3 read Max
Step 4 predict *v*P, *v*′
Step 5 read T
Step 6 predict T′, *v*P
Step 7 connect down
Step 8 un-move
Step 9 connect up
Step 10 read *v*
Step 11 predict *v*′, VP
Step 12 connect down
Step 13 read packed
Step 14 predict VP, boxes
Step 15 connect down
Step 16 read boxes and complete



Tenure in LC parsing for MG

(4) C Max T • *v* packed boxes.

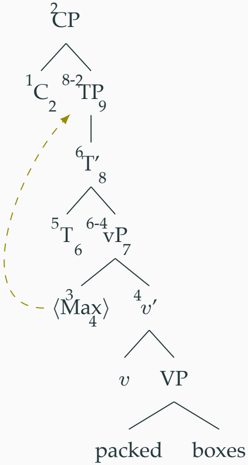
Step 1 read C
Step 2 predict CP, TP
Step 3 read Max
Step 4 predict *v*P, *v*′
Step 5 read T
Step 6 predict T′, *v*P
Step 7 connect down
Step 8 un-move
Step 9 connect up
Step 10 read *v*
Step 11 predict *v*′, VP
Step 12 connect down
Step 13 read packed
Step 14 predict VP, boxes
Step 15 connect down
Step 16 read boxes and complete



Tenure in LC parsing for MG

(4) C Max T • v packed boxes.

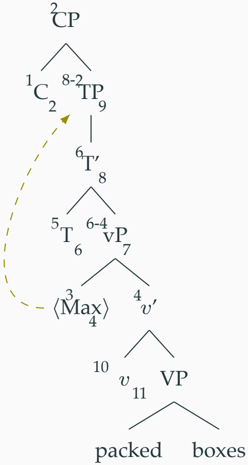
Step 1 read C
Step 2 predict CP, TP
Step 3 read Max
Step 4 predict vP, v'
Step 5 read T
Step 6 predict T′, vP
Step 7 connect down
Step 8 un-move
Step 9 connect up
Step 10 read v
Step 11 predict v', VP
Step 12 connect down
Step 13 read packed
Step 14 predict VP, boxes
Step 15 connect down
Step 16 read boxes and complete



Tenure in LC parsing for MG

(4) C Max T v • packed boxes.

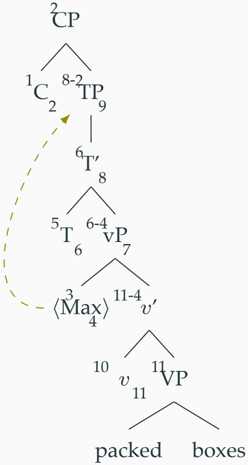
Step 1 read C
Step 2 predict CP, TP
Step 3 read Max
Step 4 predict vP, v'
Step 5 read T
Step 6 predict T′, vP
Step 7 connect down
Step 8 un-move
Step 9 connect up
Step 10 read v
Step 11 predict v', VP
Step 12 connect down
Step 13 read packed
Step 14 predict VP, boxes
Step 15 connect down
Step 16 read boxes and complete



Tenure in LC parsing for MG

(4) C Max T v • packed boxes.

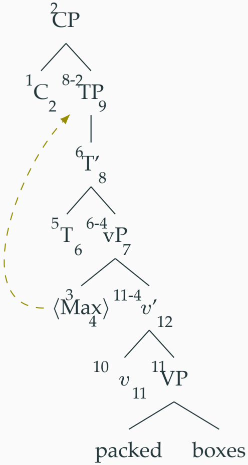
Step 1 read C
Step 2 predict CP, TP
Step 3 read Max
Step 4 predict vP, v'
Step 5 read T
Step 6 predict T′, vP
Step 7 connect down
Step 8 un-move
Step 9 connect up
Step 10 read v
Step 11 predict v', VP
Step 12 connect down
Step 13 read packed
Step 14 predict VP, boxes
Step 15 connect down
Step 16 read boxes and complete



Tenure in LC parsing for MG

(4) C Max T *v* • packed boxes.

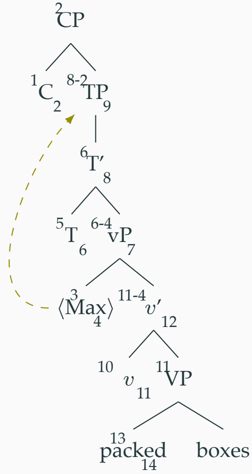
Step 1 read C
Step 2 predict CP, TP
Step 3 read Max
Step 4 predict *v*P, *v*′
Step 5 read T
Step 6 predict T′, *v*P
Step 7 connect down
Step 8 un-move
Step 9 connect up
Step 10 read *v*
Step 11 predict *v*′, VP
Step 12 connect down
Step 13 read packed
Step 14 predict VP, boxes
Step 15 connect down
Step 16 read boxes and complete



Tenure in LC parsing for MG

(4) C Max T *v* packed • boxes.

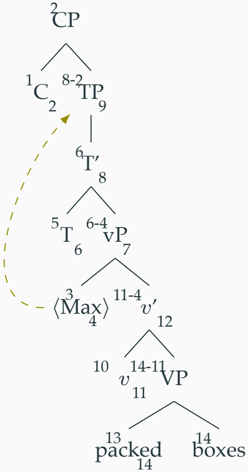
Step 1 read C
Step 2 predict CP, TP
Step 3 read Max
Step 4 predict *v*P, *v*′
Step 5 read T
Step 6 predict T′, *v*P
Step 7 connect down
Step 8 un-move
Step 9 connect up
Step 10 read *v*
Step 11 predict *v*′, VP
Step 12 connect down
Step 13 read packed
Step 14 predict VP, boxes
Step 15 connect down
Step 16 read boxes and complete



Tenure in LC parsing for MG

(4) C Max T *v* packed • boxes.

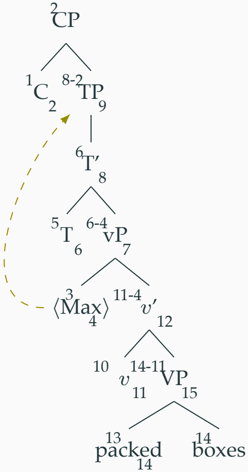
Step 1 read C
Step 2 predict CP, TP
Step 3 read Max
Step 4 predict *v*P, *v*′
Step 5 read T
Step 6 predict T′, *v*P
Step 7 connect down
Step 8 un-move
Step 9 connect up
Step 10 read *v*
Step 11 predict *v*′, VP
Step 12 connect down
Step 13 read packed
Step 14 predict VP, boxes
Step 15 connect down
Step 16 read boxes and complete



Tenure in LC parsing for MG

(4) C Max T v packed • boxes.

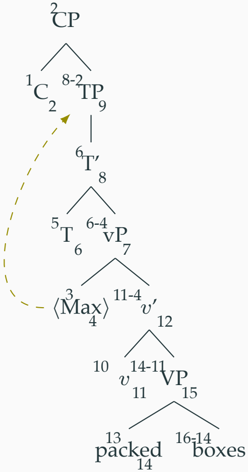
Step 1 read C
Step 2 predict CP, TP
Step 3 read Max
Step 4 predict vP, v'
Step 5 read T
Step 6 predict T′, vP
Step 7 connect down
Step 8 un-move
Step 9 connect up
Step 10 read v
Step 11 predict v', VP
Step 12 connect down
Step 13 read packed
Step 14 predict VP, boxes
Step 15 connect down
Step 16 read boxes and complete



Tenure in LC parsing for MG

(4) C Max T v packed • boxes.

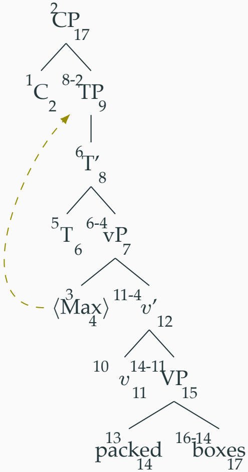
Step 1 read C
Step 2 predict CP, TP
Step 3 read Max
Step 4 predict vP, v'
Step 5 read T
Step 6 predict T′, vP
Step 7 connect down
Step 8 un-move
Step 9 connect up
Step 10 read v
Step 11 predict v', VP
Step 12 connect down
Step 13 read packed
Step 14 predict VP, boxes
Step 15 connect down
Step 16 read boxes and complete



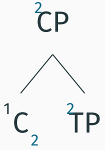
Tenure in LC parsing for MG

(4) C Max T v packed boxes. ⋉

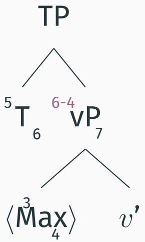
Step 1 read C
Step 2 predict CP, TP
Step 3 read Max
Step 4 predict vP, v'
Step 5 read T
Step 6 predict T′, vP
Step 7 connect down
Step 8 un-move
Step 9 connect up
Step 10 read v
Step 11 predict v', VP
Step 12 connect down
Step 13 read packed
Step 14 predict VP, boxes
Step 15 connect down
Step 16 read boxes and complete



Tenure in LC parsing for MG

- ${}^{2}\mathrm{CP}_{17}$
  - index: when the parser puts the node in the memory
  - outdex: when the parser throws the node out of the memory
- 

  - Same number: LC predictions
  - *If I find a TP, I will have a CP (since I have found its LC, C)*

- 

  - Dashed index: different instances a node is predicted
  - When $outdex = Max(dashed\_indexes) + 1$, arc-eager

Modeling results

- Target sentences

(5) right-embedding

a. the rat that ate the cheese

b. the rat that ate the cheese that had eyes

(6) center-embedding

a. the rat that the cat bit

b. the cheese that the rat that the cat bit ate

- Expected:
  - (5a) vs. (5b): constant memory load
  - (6a) vs. (6b): increased memory load in (6b) proportional to size

Modeling results: right-embedding

- (5a) vs. (5b): constant memory load ✓

Modeling results: right-embedding

- (5a) vs. (5b): constant memory load ✔
  - (5a)

Modeling results: right-embedding

- (5a) vs. (5b): constant memory load ✓
    - (5a)
        - `MaxT = 24 (root)`

Modeling results: right-embedding

- (5a) vs. (5b): constant memory load ✔
  - (5a)
    - `MaxT = 24 (root)`
    - $\texttt{MaxT}_{int}$ `= 12 (v')`

Modeling results: right-embedding

- (5a) vs. (5b): constant memory load ✔
  - (5a)
    - `MaxT = 24 (root)`
    - $\texttt{MaxT}_{int}$ `= 12 (v')`
    - `DashT = 8`

Modeling results: right-embedding

- (5a) vs. (5b): constant memory load ✔
  - (5a)
    - `MaxT = 24 (root)`
    - $\texttt{MaxT}_{int}$ `= 12 (v')`
    - `DashT = 8`
  - (5b)

Modeling results: right-embedding

- (5a) vs. (5b): constant memory load
  - (5a)
    - `MaxT = 24 (root)`
    - $\texttt{MaxT}_{int}$ `= 12 (v')`
    - `DashT = 8`
  - (5b)
    - `MaxT = 43 (root)`

Modeling results: right-embedding

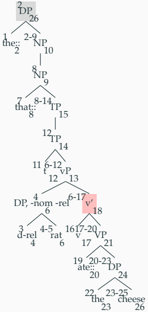
- (5a) vs. (5b): constant memory load ✔
  - (5a)
    - `MaxT = 24 (root)`
    - $\texttt{MaxT}_{int}$ `= 12 (v')`
    - `DashT = 8`
  - (5b)
    - `MaxT = 43 (root)`
    - $\texttt{MaxT}_{int}$ `= 12 (v' in two places)`

Modeling results: right-embedding

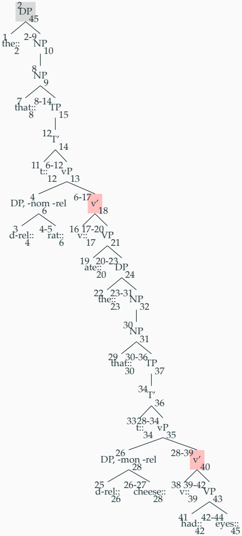
- (5a) vs. (5b): constant memory load ✔
  - (5a)
    - `MaxT = 24 (root)`
    - $\texttt{MaxT}_{int}$ `= 12 (v')`
    - `DashT = 8`
  - (5b)
    - `MaxT = 43 (root)`
    - $\texttt{MaxT}_{int}$ `= 12 (v' in two places)`
    - `DashT = 14`

Modeling results: right-embedding

(7) the rat that ate the cheese



(8) the rat that ate the cheese that had eyes



Modeling results: center-embedding

- (6a) vs. (6b): increased memory load in (6b) proportional to size ✓

Modeling results: center-embedding

- (6a) vs. (6b): increased memory load in (6b) proportional to size ✓
  - (6a)

Modeling results: center-embedding

- (6a) vs. (6b): increased memory load in (6b) proportional to size ✅
  - (6a)
    - `MaxT = 23 (root)`

Modeling results: center-embedding

- (6a) vs. (6b): increased memory load in (6b) proportional to size ✔
  - (6a)
    - `MaxT = 23 (root)`
    - $\texttt{MaxT}_{int}$ `= 19 (bit::)`

Modeling results: center-embedding

- (6a) vs. (6b): increased memory load in (6b) proportional to size ✓
  - (6a)
    - `MaxT = 23 (root)`
    - `MaxT`$_{int}$ `= 19 (bit::)`
    - `DashT = 8`

Modeling results: center-embedding

- (6a) vs. (6b): increased memory load in (6b) proportional to size ✓
  - (6a)
    - `MaxT = 23 (root)`
    - $\texttt{MaxT}_{int}$ `= 19 (bit::)`
    - `DashT = 8`
  - (6b)

Modeling results: center-embedding

- (6a) vs. (6b): increased memory load in (6b) proportional to size ✓
  - (6a)
    - `MaxT = 23 (root)`
    - $\texttt{MaxT}_{int}$ `= 19 (bit::)`
    - `DashT = 8`
  - (6b)
    - `MaxT = 44 (root)`

Modeling results: center-embedding

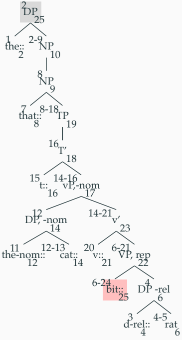
- (6a) vs. (6b): increased memory load in (6b) proportional to size ✔
  - (6a)
    - `MaxT = 23 (root)`
    - $\texttt{MaxT}_{int}$ `= 19 (bit::)`
    - `DashT = 8`
  - (6b)
    - `MaxT = 44 (root)`
    - $\texttt{MaxT}_{int}$ `= 40 (ate::)` `also 19 on (bit::)`

Modeling results: center-embedding

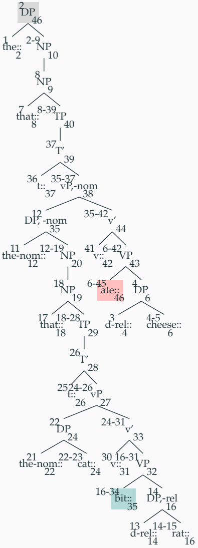
- (6a) vs. (6b): increased memory load in (6b) proportional to size ✓
  - (6a)
    - `MaxT = 23 (root)`
    - $\texttt{MaxT}_{int}$ `= 19 (bit::)`
    - `DashT = 8`
  - (6b)
    - `MaxT = 44 (root)`
    - $\texttt{MaxT}_{int}$ `= 40 (ate::)` `also 19 on (bit::)`
    - `DashT = 15`

Modeling results: center-embedding

(9) the rat that the cat bit



(10) the cheese that the rat that the cat bit ate



Conclusion

Conclusion

- In today's episode:

Conclusion

- In today's episode:
  - Tenure in left-corner MG parsing works (mostly) as it should.

Conclusion

- In today's episode:
    - Tenure in left-corner MG parsing works (mostly) as it should.
    - LC parsing for MG adequately modes how human process (left-,) right-, and center-embedding sentences.

Conclusion

- In today's episode:
  - Tenure in left-corner MG parsing works (mostly) as it should.
  - LC parsing for MG adequately modes how human process (left-,) right-, and center-embedding sentences.
- Next:

Conclusion

- In today's episode:
  - Tenure in left-corner MG parsing works (mostly) as it should.
  - LC parsing for MG adequately modes how human process (left-,) right-, and center-embedding sentences.
- Next:
  - status of new metrics ($\texttt{MaxT}_{int}$, $\texttt{DashT}$)

Conclusion

- In today's episode:
  - Tenure in left-corner MG parsing works (mostly) as it should.
  - LC parsing for MG adequately modes how human process (left-,) right-, and center-embedding sentences.
- Next:
  - status of new metrics ($\texttt{MaxT}_{int}$, `DashT`)
  - finding my nails (syntactic proposals)

Thank you!

References

Hunter, T., Stanojević, M., and Stabler, E. (2019). The active-filler strategy in a move-eager left-corner minimalist grammar parser. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 1–10.

Resnik, P. (1992). Left-corner parsing and psychological plausibility. In *COLING 1992 Volume 1: The 14th International Conference on Computational Linguistics*.

Stanojević, M. and Stabler, E. (2018). A sound and complete left-corner parsing for minimalist grammars. In *Proceedings of the Eight Workshop on Cognitive Aspects of Computational Language Learning and Processing*, pages 65–74.