Psycholinguistic Adequacy of Left-corner Parsing for Minimalist Grammars

Lei Liu

Institute of Linguistics
Leipzig University
lei.liu@uni-leipzig.de

UNIVERSITÄT LEIPZIG

Core Results

- Left-corner (LC) parsing for Minimalist Grammars (MGs) correctly models how humans parse multiple layers of left-, center-, and right-embedding.

Parser	Left	center	right
LC_{MG} (arc-standard)	$O(1)$	$O(n)$	$O(n)$
LC_{MG} (arc-eager)	$O(1)$	$O(n)$	$O(1)$
C.f. Human parser	$O(1)$	$O(n)$	$O(1)$

Table 1. Core results (format and human parser results from Resnik 1992)

- The results provide support for the psycholinguistic adequacy of LC parsing for MGs.
- A derivation tree indexing scheme is presented to help visualize parser items and calculate memory costs.

Left-, Center-, and Right-embeddings

- **Language facts**

(1) Left-embedding
  a. The rat's cheese is here.
  b. The rat's cheese's eyes are missing.

(2) Center-embedding
  a. The rat that the cat bit is here.
  b. # The cheese that the rat that the cat bit ate is here.

(3) Right embedding
  a. The rat that ate cheeses is here.
  b. The rat that ate the cheese that had eyes is here.

- Multiple left-, right- embedding: OK! constant memory space
- Multiple center-embedding: terrible! memory space $\propto$ tree height

- **Modeling attempts**

Parser	Left	Center	Right	Note
Top-down$_{CFG}$	$O(n)$	$O(n)$	$O(1)$	Resnik (1992)
Top-down$_{MG}$	$O(n)$	$O(n)$	$O(1)$	Kobele et al. (2013)
Left-corner$_{CFG}$	$O(1)$	$O(n)$	$O(1)$	Resnik (1992)
Left-corner$_{MG}$				Current study

Table 2. Reported modeling results and the current study

- **Current assumptions**
  - **Minimalist Grammars:** Stabler (1997)
    - lexicalized, context-sensitive, incorporating the Minimalist Program (Chomsky 2014)
  - **LC MG Parser:** Stanojević and Stabler (2018), Hunter et al. (2019)
    - arc-eager: possible to connect newly created item to existing item(s)
    - move-eager: LC prediction based on a movement licensor builds the landing site at the same step
  - **Complexity metric:** Tenure (Kobele et al. 2013)
    - the period of time parse items remain in memory

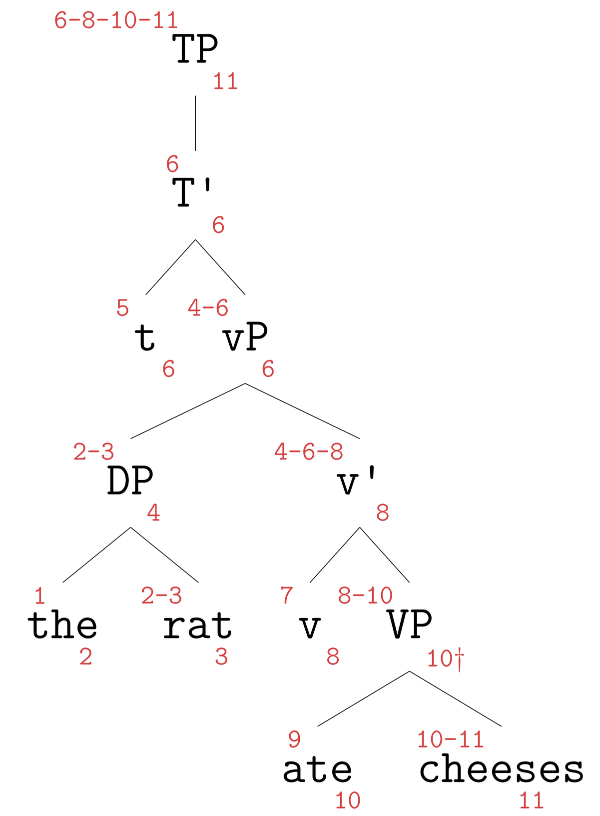
Left-corner Parsing for Minimalist Grammars

- **Parser operations:**
  - **Shift** read in the next word
  - **LC predict** when `c` is the left-corner in `B -> C A`, create and store `A => B`
  - **Complete** replace `A => B` with `B` when `A` is found
  - **Connect** (arc-eager) connect newly created item to existing item(s)
  - **Unmove** (move-eager) create landing site
- **Indexing scheme**
  - $^{index}\text{Node}_{outdex}$
  - Index:
    - initial prediction (e.g., shift, first LC prediction)
    - updated prediction (e.g., shift, further LC prediction, connect; join with dash "-")
  - Outdex:
    - consumption due to LC prediction, complete, unmove, connect

=> Annotated derivation trees: condensed yet complete representations of the parser's behaviors

- **Toy example:** *The rat t v ate cheeses.*

Step	parse item
1. `shift the::`	`the::`
2. `LC the::`	`NP => DP`
3. `shift rat:: + complete`	`DP:`
4. `LC the rat:`	`v' => vP`
5. `shift t::`	`t::` `v' => vP`
6. `LC t::+unmove +connect`	`v' => TP`
7. `shift v::`	`v::` `v' => TP`
8. `LC v:: + connect`	`VP => TP`
9. `shift ate:`	`ate::` `VP => TP`
10. `LC ate:: + connect`	`DP => TP`
11. `shift cheeses:: + complete`	`TP`



† Error in abstract

- **Complexity metric**
  - *Item Tenure*: the amount of steps a parse item remains in memory, i.e., the steps between two updates.
    - e.g., *Item Tenure*(`v'=>vP`) $= 2$
    - In the table, `v'=>vP` is stored from step 4 to 6
    - In the annotated tree, `v'` node has $4-6$ in its index, the same update sequence is found in `vP`. *Item Tenure* $= 6 - 4 = 2$
  - Maximal *Item Tenure* ($MaxT_{item}$): the maximal duration that any item remains in memory
    - e.g., $MaxT_{item}$ of the above tree is 2, found on multiple parse items.

=> Pairwise comparison: for two annotated derivation trees t_1 and t_2, if $MaxT_{item}(t_1) > MaxT_{item}(t_2)$, t_1 is more difficult to parse than t_2.

Comparisons and Results

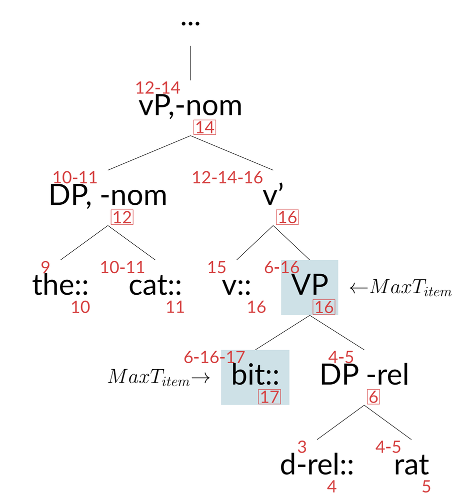
- Comparisons
  - 3 embedding directions
  - 2 layer conditions: 1-layer, 2-layer
  - 2 arc-strategies (not discussed here)

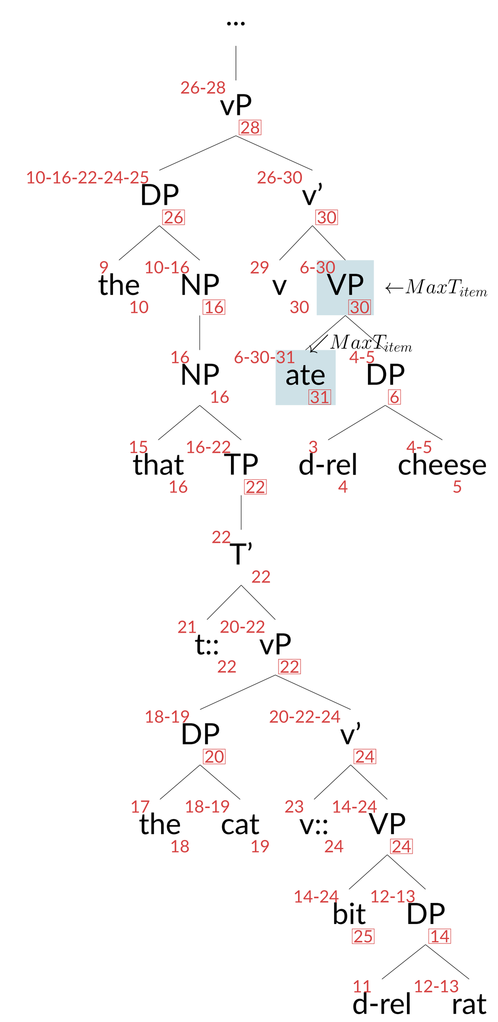
  => for each embedding direction, pairs of 1- and 2-layer sentences are compared.
- Results
  - Overall, for the arc-eager variant of LC parsing for MGs, as the number of layers increases, $MaxT_{item}$ remains the same for left- and right-embeddings, but grows as the number of layers grows in center-embeddings.

$MaxT_{item}$	Left	center	right
1-layer	2	10	6
2-layer	2	24	6

Table 3. Modeling results based on $MaxT_{item}$ (arc-eager)

- A closer look: annotated tree for Center-embeddings





(The annotated trees for left- and right-embeddings are found in the appendix to the abstract)

Conclusion

- Using $MaxT_{item}$ as a complexity metric, LC parsing for MGs derives human processing differences in left-, center-, and right-embeddings, suggesting its viability as a psycholinguistically adequate model for human sentence processing.
- The tree annotation scheme invites future research on the space of proper complexity metrics for LC parsing for MGs.

References

Chomsky, N. (2014). *The minimalist program.* MIT press.

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.

Kobele, G. M., Gerth, S., and Hale, J. (2013). Memory resource allocation in top-down minimalist parsing. In *Formal Grammar*, pages 32–51. Springer.

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

Stabler, E. (1997). Derivational minimalism. In *Logical Aspects of Computational Linguistics: First International Conference, LACL'96, Nancy, France, September 23-25, 1996. Selected Papers*, volume 1328, page 68. Springer Science & Business Media.

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.