Memory Usage Predicts Relative Difficulty in Human Sentence Processing

Thomas Graf • **James Monette** • **Robert Pasternak** • **Chong Zhang**

Department of Linguistics

Stony Brook University

What is Sentence Processing?

- Linguistic fact: Sentences have hidden structure.
  1. I saw [the man with the telescope].
  2. I saw [the man] [with the telescope].
- Finding structures is very hard computationally. (massive non-determinism ⇒ combinatorial explosion)
- But: humans easily infer the correct structure very quickly.

Big Goal Improve computational parsing models by copying techniques from human processing.
First Step Parsing model that replicates which sentences are easy/hard for humans.

Building Sentence Structure

- Chemistry Metaphor
  words : sentences ≡ atoms : molecules
- Two operations for building "sentence molecules"

Merge combine two pieces
$[_{V}$ likes$]$ + $[_{DP}$ this girl$]$ = $[_{VP}[_{V}$ likes$]$ $[_{DP}$ this girl$]]$

Move pronounce a substructure in a different position
$[_{TP}$ $[_{DP}$ John$]$ $[_{VP}[_{V}$ likes$]$ $[_{DP}$ this girl$]]]$ ⇒
$[_{TP}$ $[_{DP}$ this girl$]$ $[_{TP}$ $[_{DP}$ John$]$ $[_{VP}[_{V}$ likes$]]]]$

Incremental Top-Down Parsing (Stabler 2013)

Input sentence represented as string of words
Output tree encoding of sentence structure

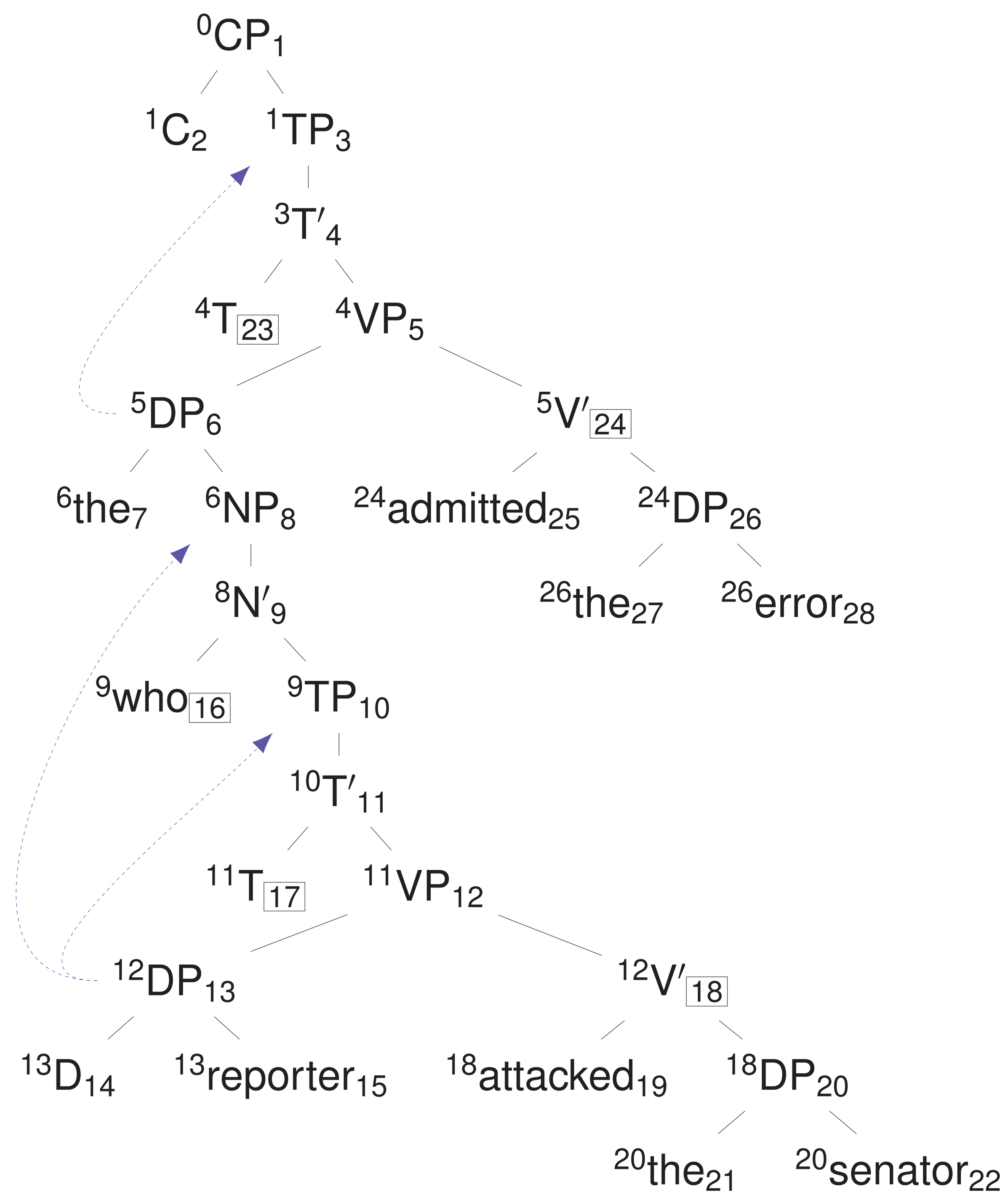
General Strategy: Variant of recursive descent parser; Hypothesize structure top-down and verify that words in structure match input string

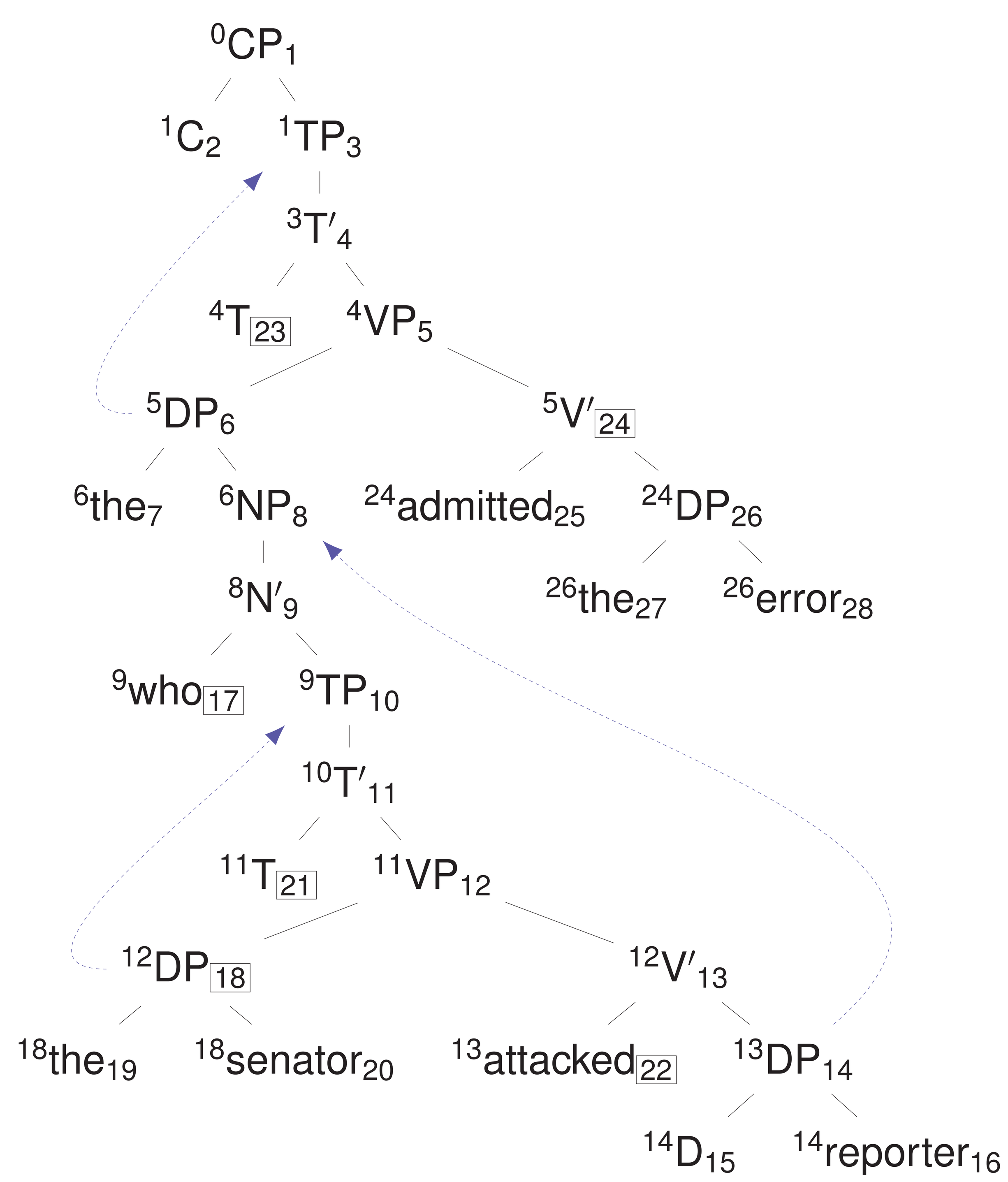
Details of Procedure

- Hypothesize top of structure and add nodes downward (toward words) and left-to-right.
- Move prediction triggers search for mover ⇒ build the shortest path towards predicted mover
- Once the mover has been found, continue from the point where it was predicted.

Role of Memory: if a node is hypothesized at step i but cannot be worked on until step j, it must be stored for $j - i$ steps (e.g. in a priority queue).

Example Trees: Subject and Object Relative Clauses





Measuring Memory Usage

- Three cognitive notions of memory usage:

Tenure how long a node is kept in memory
Payload how many nodes must be kept in memory
Size how much information is stored in a node

- Memory-based parsing metrics measure difficulty:

MaxTenure
$\max(\{\text{tenure-of}(n) \mid n \text{ a node of the tree}\})$
SumTenure $\sum_n \text{tenure-of}(n)$
BoxPayload $|\{n \mid n \text{ has tenure } > 2\}|$
Gap $\sum_{m \text{ a mover}}(\text{index of target for } m - \text{index of } m)$

Evaluation

- The best metric across a variety of languages is **MaxTenure** coupled with **Gap**.
- Phenomena tested:
  - subject VS object relative clause
  - relative clause VS sentential complement
  - center embedding VS right embedding
  - nested dependencies VS crossing dependencies

(Kobele et al. 2012; Graf and Marcinek 2014; Graf et al. 2015)

Summary and Next Steps

- **Memory Usage**
  - Decisive factor for processing difficulty is maximum time that nodes must be kept in memory.
  - Size of node is less relevant.
  - Number of memorized nodes does not matter.
- **Future**
  - Modeling meaning preferences
  - Incorporate into production models

References

Graf, Thomas, Brigitta Fodor, James Monette, Gianpaul Rachiele, Aunika Warren, and Chong Zhang. 2015. A refined notion of memory usage for minimalist parsing. In *Proceedings of MoL 2015*, 1–14 • **Graf**, Thomas, and Bradley Marcinek. 2014. Evaluating evaluation metrics for minimalist parsing. In *Proceedings of CMCL 2014*, 28–36. • **Kobele**, Gregory M., Sabrina Gerth, and John T. Hale. 2012. Memory resource allocation in top-down minimalist parsing. In *Proceedings of FG 2012.* • **Stabler**, Edward P. 2013. Bayesian, minimalist, incremental syntactic analysis. *Topics in Cognitive Science* 5:611–633.