A cognitively realistic left-corner parser with visual and motor interfaces
An extensible framework for mechanistic processing models

Adrian Brasoveanu & Jakub Dotlačil

CUNY 31, UC Davis · March 17, 2018
Overarching goal

- build formally and computationally explicit processing models for natural language syntax and semantics
- specifically, cognitively realistic models for incremental parsing of discourse representations structures (DRSs, Kamp 1981; Kamp and Reyle 1993) or similar representations
- the semantic and syntactic representations are created in parallel

Main goal for today:

- modeling *syntactic* representations
An extensible framework for processing models

- **Mechanistic** models of processing
  “in most instances, cognitive scientists would ultimately prefer an explanatory process over mere characterization”
  Lewandowsky and Farrell, 2010

- common approach: use an independently motivated, general cognitive architecture

- parsing easy to embed in hybrid cognitive architectures
  - Soar
    Hale, 2014; Young and Lewis, 1999
  - ACT-R
    Dillon et al., 2013; Engelmann et al., 2013; Kush, 2013; Lewis and Vasishth, 2005; Nicenboim and Vasishth, 2018; Rij, 2012; Taatgen and Anderson, 2002; Vasishth et al., 2008
Mainly used to model recall of syntactic structures

Dillon et al., 2013; Engelmann et al., 2013; Lewis and Vasishth, 2005; Nicenboim and Vasishth, 2018; Vasishth et al., 2008

This focus on recall-related modeling does not take advantage of the generality of ACT-R as a cognitive architecture and its “no magic” policy

Implemented fully in LISP (not a very popular programming choice now)

ACT-R comes with many parameters; these are set to their default values or manually changed

Modeling is hard to replicate; systematic quantitative model comparison hard to perform
A new Python3 implementation of ACT-R (pyactr; Brasoveanu and Dotláčil 2018, in prep.)
https://github.com/jakdot/pyactr

ACT-R + Bayes: ACT-R models embedded in Bayesian models, hence systematic exploration of parameter values, model comparison, modeling easy to replicate

the ACT-R component: a working, extensible parsing framework for syntax and semantics, with visual and motor interfaces (today, only syntax)

modular structure: alternative models for peripherals (visual, motor) & other components possible
Grodner and Gibson (2005, Exp. 1): self-paced reading, matrix subject is modified by a subject or object-extracted relative clause (RC)

(1) The reporter who sent the photographer to the editor hoped for a story.

(2) The reporter who the photographer sent to the editor hoped for a story.

9 ROIs: word 2 through word 10 (underlined above)
Red circle is the visual focus. Temporal trace incrementally produced by the model is visible in the background.
1 Introduction: framework & case study

2 ACT-R & left-corner parsing

3 Results

4 Conclusion
1 Introduction: framework & case study

2 ACT-R & left-corner parsing

3 Results

4 Conclusion
Memory in ACT-R

Two types of memory:

- DECLARATIVE MEMORY: knowledge of facts
  facts represented as chunks (attribute-value matrices)

  | ISA:      | word   |
  | FORM:     | car    |
  | MEANING:  | [car]  |
  | CATEGORY: | noun   |
  | NUMBER:   | sg     |

- PROCEDURAL MEMORY: behavior as a series of productions
  productions – conditionalized actions

  Goal> TASK: reading car
  ⇒ Retrieval> ISA: word
  TASK: retrieving category
  FORM: car
Modules and buffers in ACT-R

- ACT-R mind is composed of modules, which include declarative and procedural memory
- Modules are not directly accessible – they can only be accessed through buffers
- Buffers represent agent’s current state; productions fire based on contents of buffers
- Buffers can hold only one chunk
- Only one production can fire at any given time
Parser components:

- lexical knowledge $\rightarrow$ declarative memory
- knowledge of grammar $\rightarrow$ procedural memory
- expectations about upcoming syntactic categories, which guide parsing $\rightarrow$ goal buffer
- information about the current syntactic parse $\rightarrow$ secondary goal buffer
- visual information from environment $\rightarrow$ visual buffer
- key press commands $\rightarrow$ manual buffer
- visual module – EMMA Salvucci, 2001 (other choices possible)
An eager left-corner parser in ACT-R

Rules:
S → NP VP  
NP → Det N  
VP → V

Visual input:
A boy sleeps.
A — — — — —.

Input
- Stack: S (Goal)
- Found: a, Det  
  (Visual + Retrieval)

Output
- Stack: N NP S (Goal)
- Structure:
  S  
  |  
  NP  
  |  
  Det  
  |  
  a
An eager left-corner parser in ACT-R

Rules:
- S → NP VP
- NP → Det N
- VP → V

Visual input:
- A boy sleeps.
  - boy ----.

Input
- Stack: N NP S (Goal)
- Found: boy, N
  (Visual + Retrieval)

Output
- Stack: VP (Goal)
- Structure:
  S
    NP
      Det
        a
    VP
      N
        boy
An eager left-corner parser in ACT-R

Rules:
S → NP VP
NP → Det N
VP → V

Visual input:
- A boy sleeps.
- -- sleeps.

Input
- Stack: VP (Goal)
- Found: *sleeps*, V
  (Visual + Retrieval)

Output
- Stack: {} (Goal)
- Structure:

```
S
   /\    /
  VP  NP
     /\    /
    V  Det N
         /\    /
        V  boy
             /\    /
            V  sleeps
```
Flow chart of parsing process per word

1. **attend word**
2. **retrieve lex. information about word**
3. **move visual attention**
4. **retrieve syntactic parse if applicable (e.g., wh-word)**
5. **parse**
6. **press key**
Parameters – visual encoding (EMMA)

- Visual encoding ($T_{enc}$) dependent on visual distance $d$ and object properties, $D$

  \[ T_{enc} = K \cdot D \cdot e^{kd} \text{(parameter } k \text{ – angle)} \]

- $D = \text{word length, } K = 0.01$
Parameters – rule firing and memory recall

- Rule firing = \( r \) (parameter \( r \))
- Retrieval latency is a function of activation, modulated by parameters \( F \) (latency factor) and \( f \) (latency exponent)
  \[ T = F \cdot e^{-f \cdot A} \]
- Base activation, \( A \), is a function of time elapsed since previous word usages
  \[ A = \log \left( \sum_{k=1}^{n} t_k^{-0.5} \right) \]
The model is fit to data by estimating the 4 free parameters \((k, r, F, f)\).

Standardly, relying on default values or manually changing the values; subjective & time consuming.

**pyactr** enables us to easily interface ACT-R models with standard statistical estimation methods implemented in widely-used Python3 libraries.

We use ACT-R models as the likelihood component of full Bayesian models, and fit the ACT-R parameters to experimental data.
Bayesian model structure

\[
\begin{align*}
&k \sim \text{halfnormal}(0; 1) \\
r \sim \text{halfnormal}(0; 0.05) \\
F \sim \text{halfnormal}(0; 0.3) \\
f \sim \text{halfnormal}(0; 0.5) \\
\Rightarrow & \quad \text{ACT-R}(k; r; F; f) \\
\Rightarrow & \quad \text{Latency} \\
\Rightarrow & \quad \text{RT} \\
\Rightarrow & \quad \text{normal}(\text{Latency}; 10)
\end{align*}
\]
1. Introduction: framework & case study
2. ACT-R & left-corner parsing
3. Results
4. Conclusion
Posterior predictions (Model 1)

Predicted RTs (95% CRIs) and observed RTs (ms)
Posterior predictions (Model 1)

Predicted RTs (95% CRIs) and observed RTs (ms)
Posterior predictions (Model 1)

Predicted RTs (95% CRIs) and observed RTs (ms)

- **subj**
  - reporter: 350 ms
  - who: 400 ms
  - the: 450 ms
  - photographer: 300 ms
  - sent: 350 ms

- **obj**
  - to: 400 ms
  - the: 350 ms
  - editor: 500 ms
  - hoped: 450 ms
Predicted RTs (95% CRIs) and observed RTs (ms)
Posterior predictions (Model 1)
Model 2: no postulated subject gaps
Model 2: no postulated subject gaps
Model 2: no postulated subject gaps

wh-word and following word not modeled well; Model 1 better
Model 2: no postulated subject gaps

Predicted RTs (95% CRIs) and observed RTs (ms)
Model 3: parallel reader

- **attend word**
- **retrieve lex. information about word**
  - **retrieve syntactic parse if applicable (e.g., wh-word)**
  - **parse**
  - **press key**
  - **move visual attention**
Model 3: parallel reader

- attend word
- retrieve lex. information about word
- retrieve syntactic parse if applicable (e.g., wh-word)
- parse
- press key
- move visual attention

- Model 1 completes all available parsing before key press (serial)
- Model 3: first lexical retrieval, then structure building & key press in parallel
- Outcome: spillover on word after object gap captured
Model 3: spillover after object gap captured
Model 3: spillover after object gap captured

Predicted RTs (95% CRIs) and observed RTs (ms)
Model 3: predictions for individual items

- we compare predictions per item
  (linear regression: observed RT $\sim$ predicted RT)
  1ms increase in predicted RT corresponds to 1ms increase in observed RT ($SE=0.009$) ($t = 5.7$)
Model 3: predictions for individual items

- we compare predictions per item
  (linear regression: observed RT $\sim$ predicted RT)
  1ms increase in predicted RT corresponds to 1ms increase in observed RT ($SE=0.009$) ($t = 5.7$)

Eye-tracking and self-paced reading (data from Frank et al. 2013; a variety of syntactic structures, no RCs)

- SPR: 1ms increase in predicted RT corresponds to 0.79ms increase in observed RT ($t = 2.1$)

- ET: 1ms increase in predicted RT corresponds to 0.82ms increase in observed RT ($t = 3.31$)
we introduced a modular and extensible framework for mechanistic processing models

case study: an incremental left-corner parser with visual and motor interfaces for subject/object gap relative clauses

framework used to quantitatively compare hypotheses about processing, e.g., predictively postulating subject gaps
Conclusion: future directions

- we have only done informal quantitative comparisons based on posterior predictions

- but systematic across-the-board model comparison via Bayes factors is possible in this framework

- framework can model other tasks (eye tracking, lexical decision)
Acknowledgments

We are grateful to Amanda Rysling, Donka Farkas, Abel Rodriguez, Matt Wagers, the UCSC S-lab audience (March 2018) and our CUNY 2018 anonymous reviewers for comments and discussion.

We want to thank Ted Gibson and Dan Grodner for providing the items and full datasets for the two experiments reported in their paper (Grodner and Gibson 2005). Jakub Dotlačil was supported by the NWO VENI 275-80-005 grant. The usual disclaimers apply.

Models coming soon here:
https://github.com/jakdot/conferences/2018


Frank, Stefan L et al. (2013). “Reading time data for evaluating broad-coverage models of English sentence processing”. In: *Behavior Research Methods* 45.4, pp. 1182–1190.

References II


