A cognitively realistic left-corner parser with visual and motor interfaces

Adrian Brasoveanu (UC Santa Cruz) & Jakub Dotlačil (ILLC, University of Amsterdam)
abrsvn@gmail.com

I. Introduction. Current psycholinguistic models of syntactic comprehension rarely incorporate an explicit parser, and if they do, they often abstract away from the visual and motor components involved in standard self-paced reading tasks. The main contribution of this paper is a `Python3` reimplementation of ACT-R (Anderson and Lebiere 1998 a.o.) in which we can build *an end-to-end simulation of syntactic parsing in a typical self-paced reading experiment* (Just et al. 1982).

II. The four main features of the model are that: (**i**) it uses a psycholinguistically plausible *left-corner parsing* strategy (Demers 1977, Resnik 1992, Hale 2014 a.o.), implemented as a skill in procedural memory (following Lewis and Vasishth 2005, LV05 for short); (**ii**) it makes use of independently motivated components of the ACT-R framework: a *content-addressable declarative memory* (Wagers and Phillips 2009 a.o.), *goal and imaginal buffers* to drive / monitor the parsing process etc.; (**iii**) it explicitly models the *motor and visual processes* involved in self-paced reading, extending previous work (LV05): the model searches the screen for a string of characters, shuttles it to the declarative memory retrieval buffer, plans the next visual search or key press etc.; (**iv**) the end-to-end syntactic comprehension model can be embedded into a (Bayesian) statistical model that enables us to *estimate its sub-symbolic (quantitative) parameters and do model comparison* based on experimental data.

III. The motor and visual modules and their interaction with the syntactic parsing process. The motor module implements part of EPIC's Manual Motor Processor (Meyer and Kieras, 1997) in `Python3` (following Lisp ACT-R). The module has one buffer that accepts motor commands like key presses. The 'hands' of the ACT-R model are assumed to be positioned in the home row position on a standard (US) English keyboard, with index fingers at F and J. The visual module allows the ACT-R model to 'see' the environment. This interaction happens via two buffers: `visual_location` searches the environment for elements matching its search criteria, and `visual` stores the element found using `visual_location`; these buffers are sometimes called the visual Where and What buffers. The vision module is an implementation of an EMMA model (Salvucci, 2001; Staub, 2011), which in turn is a generalization and simplification of the E-Z Reader model (Reichle et al., 1998).

On the next page, we provide parts of the output obtained when the model reads the sentence *Mary likes Bill*. At 0 ms, the first word (*Mary*) is already on the screen. The visual module encodes its location and retrieves the visual value (word) at that location. At the same time, the `"press spacebar"` rule is selected, which will ultimately trigger a key press that will reveal the second word of the sentence. This rule fires 50 ms later, initiating a manual process in the motor module. Since the visual module already retrieved the first word, we can fire the `"encode word"` rule at 100 ms, taking the retrieved visual value and encoding it in the goal buffer as the word to be parsed. We then attempt the retrieval of the encoded word from declarative memory: rule `"retrieve category"` is fired at 150 ms, starting a retrieval process that takes 50 ms to complete. At 200 ms, the word *Mary* is successfully retrieved with its syntactic category ProperN and we start a cascade of parsing rules triggered by this 'left corner': the `"shift and project word"` rule at 250 ms creates a chunk in the imaginal buffer storing a unary branching tree with ProperN as the mother node and the word *Mary* as the daughter. Meanwhile, the motor module executes the movement preparation and initiation triggered by the `"press spacebar"` rule fired at 50 ms, which happens in parallel to the parsing steps triggered by the word *Mary*. Parsing continues with the next word on the screen (*likes*), interweaved with further syntactic rules triggered by the left-corner *Mary*.

IV. Conclusion. This framework builds on LV05's classical work and incorporates a variety of modern ACT-R features. Reimplementation in Python makes this type of cognitive modeling more accessible to (psycho)linguists, and makes the large ecosystem of Python libraries for natural language processing, statistical modeling etc. easily available. The model can be extended to cover semantics, as well as other experimental methodologies (eye-tracking, priming etc.).

```
****Environment: {1: {'text': 'Mary', 'position': (320, 180)}}
(0, 'visual_location', 'ENCODED LOCATION: _visuallocation(screen_x= 320, screen_y= 180)')
(0.0118, 'visual', 'AUTOMATIC BUFFERING: _visual(screen_pos=
         _visuallocation(screen_x= 320, screen_y= 180), value= Mary)')
(0.05, 'PROCEDURAL', 'RULE FIRED: press spacebar')
(0.05, 'manual', 'COMMAND: press_key')
(0.1, 'PROCEDURAL', 'RULE FIRED: encode word')
(0.15, 'PROCEDURAL', 'RULE FIRED: retrieve category')
(0.15, 'retrieval', 'START RETRIEVAL')
(0.15, 'PROCEDURAL', 'CONFLICT RESOLUTION')
(0.15, 'PROCEDURAL', 'NO RULE FOUND')
(0.2, 'manual', 'PREPARATION COMPLETE')
(0.2, 'retrieval', 'RETRIEVED: word(cat= ProperN, form= Mary)')
(0.2, 'PROCEDURAL', 'RULE SELECTED: shift and project word')
(0.25, 'manual', 'INITIATION COMPLETE')
(0.25, 'PROCEDURAL', 'RULE FIRED: shift and project word')
(0.25, 'imaginal', 'CREATED A CHUNK: parse_state(daughter1= Mary, node_cat= ProperN)')
(0.26, 'manual', 'KEY PRESSED: SPACE')
****Environment: {1: {'text': 'likes', 'position': (320, 180)}}
(0.26, 'visual_location', 'ENCODED LOCATION: _visuallocation(screen_x= 320, screen_y= 180)')
(0.2654, 'visual', 'AUTOMATIC BUFFERING: _visual(screen_pos=
         _visuallocation(screen_x= 320, screen_y= 180), value= likes)')
(0.3, 'PROCEDURAL', 'RULE FIRED: project: NP ==> ProperN')
(0.3, 'imaginal', 'CREATED A CHUNK: parse_state(daughter1= ProperN,
      lex_head= Mary, mother= S, node_cat= NP)')
[...]
```

References

Anderson, John R., and Christian Lebiere. 1998. The atomic components of thought. Hillsdale, NJ: Lawrence Erlbaum Associates.

Demers, Alan J. 1977. Generalized left corner parsing. In Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages, 170–182. ACM.

Hale, John T. 2014. Automaton theories of human sentence comprehension. Stanford: CSLI Publications.

Just, Marcel A., Patricia A. Carpenter, and Jacqueline D. Woolley. 1982. Paradigms and processes in reading comprehension. Journal of Experimental Psychology: General 111:228–238.

Lewis, Richard, and Shravan Vasishth. 2005. An activation-based model of sentence processing as skilled memory retrieval. Cognitive Science 29:1–45.

Meyer, David E, and David E Kieras. 1997. A computational theory of executive cognitive processes and multiple-task performance: Part i. basic mechanisms. Psychological review 104:3.

Reichle, Erik D, Alexander Pollatsek, Donald L Fisher, and Keith Rayner. 1998. Toward a model of eye movement control in reading. Psychological review 105:125.

Resnik, Philip. 1992. Left-corner parsing and psychological plausibility. In Proceedings of the Fourteenth International Conference on Computational Linguistics. Nantes, France.

Salvucci, Dario D. 2001. An integrated model of eye movements and visual encoding. Cognitive Systems Research 1:201–220.

Staub, Adrian. 2011. Word recognition and syntactic attachment in reading: Evidence for a staged architecture. Journal of Experimental Psychology: General 140:407–433.

Wagers, Matthew W., and Colin Phillips. 2009. Multiple dependencies and the role of the grammar in real-time comprehension. Journal of Linguistics 45:395–433.