Automaton Theories of Human Sentence Comprehension - Ch. 5: Cognitive Architecture for Linguists

2020. 02. 14

Won Ik Cho

SEOUL NATIONAL UNIVERSITY

Human Interface Laboratory

Cognitive Architecture

- **Cognitive architecture**
  - Term that refers to a collection of **basic elements, or primitives**, out of which **theories of intelligent behavior** might be fashioned
  - What is the cognitive architecture for the thinking human?
  - What is a good architecture? – **Generalizability?**
    - Architecture that allows theorists to re-use the same primitives across domains
    - Models the way humans do multiple tasks that call upon different aspects of their intelligence
  - Re-use of primitive elements across tasks and domains

    >> They **correspond to something fundamental about the mind**
    - e.g., **Newell (1972)**
      - Goal: defining a process theory
      - "Posits a set of processes or mechanisms that produce the behavior of the thinking human"

        ↔ **Chomsky?**

Cognitive Architecture

- **Newell (1972) - Human problem solving**
  - (by Cohen, 1981): Newell used methods alien to cognitive psychology to explore questions alien to AI
    - Not well-known among linguists
    - But offers an elegant way to reconcile generative grammar with the rest of cognitive science
  - Production system:
    - **Scheme for specifying an information processing system**
      - Consists of a set of **productions**
      - Each production consists of **a condition and an action**
      - Also has a collection of data structures - Expressions that encode the info upon which the production system works
        - » On which the **actions operate**
        - » On which the **conditions can be determined to be true or false**
    - A program that computes outputs from some given inputs
      - In a way that affords a particularly direct psychological interpretation of the intermediate states

Cognitive Architecture

- **Newell (1972) - Human problem solving**
  - Example of production rules: IF-THEN
    - *IF you encounter the word "the"*
      *THEN as an action **postulate** a noun phrase*
    - *IF the goal is to add the first block*
      and the heap of blocks is not empty
      *THEN **pick up** the nearest block from the heap*

    - IF-THEN highlight the condition and action part
      - Condition: either T/F in a given situation
      - Action: specifies a list of things to do when condition == True
    - Result: Modifying he current data structures
      - Which leads in the next instance to another (possibly the same) production
    - When process halts
      - When no condition is true (nothing is evoked)
      - When an action containing a stop operation occurs

Cognitive Architecture

- **Newell (1972) - Human problem solving**
  - Commonality: Absence of externally-imposed control
    - In case of multiple matches? – Conflict resolution policy: The rule
      - with most specific consequences
      - that has not fired recently
      - estimated to be most 'useful' according to some metric
  - Which 'Soar' deals with:
    - One conceives of **parsing** as a skill,
      as **procedural knowledge encoded in production rules**,
    - Then a natural way to model **syntactic processing** is to
      assimilate ambiguity resolution to the underlying production system's policy for conflict resolution

Cognitive Architecture

- **Newell (1972) - Human problem solving**
  - Different from past AI/cognitive modeling such as Canonical Systems (1943)
    - A Post canonical system is a triplet (A,I,R), where
      - A is **a finite alphabet**, and finite (possibly empty) strings on A are called ***words***
      - I is **a finite set of *initial words***
      - R is **a finite set of string-transforming rules** (called production rules), each rule being of the following form:

$$\begin{array}{cccccccc} g_{10} & \$_{11} & g_{11} & \$_{12} & g_{12} & \cdots & \$_{1m_1} & g_{1m_1} \\ g_{20} & \$_{21} & g_{21} & \$_{22} & g_{22} & \cdots & \$_{2m_2} & g_{2m_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ g_{k0} & \$_{k1} & g_{k1} & \$_{k2} & g_{k2} & \cdots & \$_{km_k} & g_{km_k} \end{array}$$

$$\downarrow$$

$$h_0 \quad \$'_1 \quad h_1 \quad \$'_2 \quad h_2 \quad \ldots \quad \$'_n \quad h_n$$

https://en.wikipedia.org/wiki/Post_canonical_system

      - » where each *g* and *h* is a specified fixed word, and each *$* and *$'* is a variable standing for an arbitrary word.
    - Which is conceptual origin of formalized transformational grammars
      - as in Chomsky (1957/1975)

Human performance, NL Soar

- Post: Saw himself as exploring the foundations for mathematics
- Chomsky: Saw himself as doing the same for the foundations of linguistics
- **For Newell and Simon,**
  - Production rules took on a psychological interpretation in **models of human performance**
    - **What is human performance?**
      - For psychologists, refers to 'the speed and accuracy with which people do things'
      - Boosted up with the military applications
      - Trials such as Memory and ACT-R (by Anderson)
    - **What is Soar?**
      - **Cognitive architecture** that Newell advanced (1990)
      - Can reason about its own conflict resolution
        - » The conflict-resolution policy for Soar is to apply Soar itself to the problem of deciding what to do
      - Reifies the each step of problem-solving in an abstraction called an operator, rather than via production rules

Human performance, NL Soar

- Individual production rules
  - Related to operator proposal, selection, and application
  - What if **no clear winner between competing operators**?
    - **Tie impasse**
      - Motivates creation of a substate
      - Substate will be destroyed if enough information given to solve the impasse
    - Solving this is the key feature of Soar:
      - Solves its own conflicts using whatever knowledge that is available?
  - The most important for linguists – NL Soar
    - Ambiguity resolution in language processing is simply the resolution of impasse by Soar
    - It does not mean an alternative proposal about grammar
    - Rather, NL Soar shows how **assumptions about grammar and processing could fit together**
    - Built on a premise that a **single decision takes 100ms**

Human performance, NL Soar

- Applications
  - Lewis (1993)
    - A theory of garden pathing using NL-Soar in combination with several suxiliary hypothesis
      - Single-path parsing
      - Limit on the number of attachment sites
      - Multiple levels of representation all the way from lexical referential
  - Parsing automata - Generalized Left-Corner Parsing
    - Basic idea: **To maintain the pushdown stack in working memory**
      - Will be encoded by several cooperating working memory elements
        - » Use the sequence of symbols on top of the stack to condition the operators
        - » IF the top of the stack reads [V, VP, Adv]
          THEN projecting VP → V VP Adv is an acceptable operator
        - » IF projecting VP → V VP Adv was selected
          THEN apply it bottom-up by replacing all those symbols with VP
  - Also used in Attachment, Shift etc.

Seoul National University

EndOfPresentation

Thank you!