Cognitive Architecture

and Conversational Agents

Eric Saund, Ph.D.
February, 2019

www.saund.org
The organization of structural and functional elements to achieve some purpose.
Cognitive Architecture: Outline

- **Architecture in Information Systems**
  "An organization of elements to achieve some information processing purpose."

- **History**
  Computational theory of mind

- **The Baseline/Generic Cognitive Architecture**

- **Examples**
  Soar, LIDA, CopyCat

- **Architecture in NN / Deep Learning Networks**

- **Conversational Agents**
  - NLP
  - Dialog Management
  - Knowledge Graphs
  - Deep Networks
Information Architecture: A Web Site

HTML1.0

HTML5/CSS/Javascript

- active nav widgets
- guided tours
- search
- state
  - cookies
  - navigation history
- contingent access (passwords)
- personalization
Information Architecture: A Computer

Diagram showing the relationship between I/O Devices, CPU, Memory, Data Memory, Program Memory, Long Term Memory (Disk), and Short Term Memory (RAM). The Operating System is depicted below with layers including Applications, System Libraries, System Call Interface, Device Drivers, Resource Controls, Firmware, and Metal.
Cognitive Architecture: Basic Agent

Cognitive Architecture: Historical Roots

1800s | 1900s

Psychoanalytic Theories
Freud, Jung

Perceptual & Phenomenal Psychology
Helmholtz, William James

Behaviorism
Pavlov, B.F. Skinner

Artificial Intelligence
McCarthy, Minsky

Cybernetics
Norbert Wiener

Computational Theory of Mind
McCulloch and Pitts

Turing, von Neumann

Chomsky

Newell & Simon

Guiding Metaphor
- engines and hydraulics
- signal transmission
- signal processing
- calculating machines
- computers

https://plato.stanford.edu/entries/computational-mind/
Cognitive Architecture: Standard Model

Knowledge Resources
Long Term Memory
(declarative, procedural, semantic, episodic, skill…)

Common Workspace
(blackboard)
(Working Memory)

perception

motor action

sense organs

environment
effector organs
Cognitive Architecture: Big Questions

- **What are the types of content held in the workspace?**
  - percepts
  - beliefs
  - memories
  - goals
  - intentions & plans
  - emotions
  - attitudes

- **What are the representations for state and knowledge?**
  - activation patterns over fixed vectors
  - graphs of objects and relations
  - frequencies and phases of waveforms

- **How is processing controlled?**
  - automatic processes
  - conscious deliberation
  - selection of operators
Soar Cognitive Architecture

Soar

- Definition of intelligence:
  - problem states and transitions
  - solutions found through *search* in state space

- Representation:
  - graphs of objects and relations

- Control: production system
  - Working Memory blackboard
  - procedural knowledge
  - declarative knowledge
  \[ \text{Long-Term Memory} \]
**Definition of intelligence:**
- problem states and transitions
- solutions found through *search* in state space

**Start state:**
both jugs empty.

**Goal state:**
3-gallon jug contains 1 gallon of water.
Representation in Soar

Graph

- data objects
- attributes & relations
- operators
- Working Memory (state)
- Long-Term Memory (knowledge)
**Production System**

- Working Memory blackboard
- declarative knowledge - what
- procedural knowledge – how
  - rules
  - operators
- subgoal states

**Executive Function** (Psychology, Cognitive Neuroscience):
- update Working Memory from sensory and Long-Term Memory resources
- focus attention, inhibit distractors
- shift task context
Cognitive Architecture: Outline

• Architecture in Information Systems
  “An organization of elements to achieve some information processing purpose.”

• History
  Computational theory of mind

• The Baseline/Generic Cognitive Architecture

• Examples
  Soar, LIDA, CopyCat

• Architecture in NN / Deep Learning Networks

• Conversational Agents
  • NLP
  • Dialog Management
  • Knowledge Graphs
  • Deep Networks
Marr’s Three Levels of Abstraction

David Marr: Theoretical Neuroscience \[\rightarrow\] Computational Intelligence

what?

why?

- **Computational Theory**
  What is the computation and by what principles is it accomplished?

- **Algorithm**
  What representations and algorithms are used to carry forth computation?

- **Implementation**
  On what physical hardware and firmware is the algorithm run?

Example

Textbook:
ToC, body text, index

Scrolling text roll, manual + automatic positioning

Projection of markings printed on acetate film, keyboard controls scrolling per ToC and index.
Cognitive Architecture: Reactive Agent

perception

REACTIVE PROCESSES

THE ENVIRONMENT

action
Cognitive Architecture: Deliberative Agent

Diagram showing the relationship between perception, action, deliberative processes, reactive processes, and the environment. The diagram includes labels for long-term memory, motive activation, variable threshold attention filter, and the environment.
Cognitive Architecture: Reflective Agent

META-MANAGEMENT (reflective) processes

DELIBERATIVE PROCESSES (Planning, deciding, scheduling, etc.)

REACTIVE PROCESSES

THE ENVIRONMENT

perception

action

Long term memory

Motive activation

Variable threshold
attention filter
Reactive vs. Deliberative

**Reactive**
- automatic & strictly determined
- modest internal state
- implicit representations
- Kahneman System 1
  - Example: thermostat

**Deliberative**
- makes choices
- rich internal state
- explicit world models
- Kahneman System 2
  - Example: building temperature management system
Reactive vs. Deliberative Building Temperature Controller

**Reactive**
- Automatic & strictly determined
- Modest internal state
- Implicit representations
- Example: thermostat

**Deliberative**
- Makes choices
- Rich internal state
- Explicit world models
- Example: building temperature management system

**Example: thermostat**

```
Temp sense  ->  Heat control  <-  Setpoint
```

**Example: building temperature management system**

[Diagram showing modular model predictive controller]
Reactive vs. Deliberative

**Reactive**
- automatic & strictly determined
- modest internal state
- implicit representations

**Deliberative**
- makes choices
- rich internal state
- explicit world models

Soar
A Reactive Water Jug Solution in Soar

Deliberative search space

Reactive program rules

If 0, 0 Then Fill(3) => 0, 3

If 0, 3 Then Pour(3, 5) => 3, 0

If 3, 0 Then Fill(3) => 3, 3

If 3, 3 Then Pour(3, 5) => 5, 1
Neural Network Architectures

- **Architectural Elements**: Layer dimensions, weights, nonlinearities
- **Organization**: Layer connectivity
- **Purpose**: Function approximation

Input image

Convolutional

Fully connected

Output category

Alexnet
Neural Network Architectures

Reactive or deliberative?

choices?
state?
implicit/explicit
Cognitive Architecture

- Naturally Intelligent Agent
- Artificially Intelligent Agent

Where does knowledge reside?

How is state represented and utilized in decisions?

Forms of Memory

Localist vs. Distributed Representations

What is the control mechanism: what to do & think next?

Distinction between Program Control and Data

How Learning Happens
LIDA Cognitive Architecture

LIDA Cognitive Cycle

WORKSPACE

Structure Building Codelets
Conscious Contents Queue

Current Situational Model

Sensory Memory

Perceptual Associative Memory
Spatial Memory
Transient Episodic Memory

Declarative Memory
Attention Codelets

Global Workspace

Add Conscious Content
Add Coalitions
Form Coalitions

Internal & External Environment

Motor Plan Execution
Sensory Motor Memory
Action Selection
Procedural Memory

Perceptual Learning
Spatial Learning
Episodic Learning
Procedural Learning
Recruit Schemes

Sensory Motor Learning
Selected Behavior
Update Behaviors
Instantiated Schemes (Behaviors)

Perimis
Primings

Short Term
Long Term
Conscious

Stan Franklin
Task: Letter string analogy

\[ \text{abc} \quad \Rightarrow \quad \text{abd} \]

\[ \text{mrrjjj} \quad \Rightarrow \quad ? \]
Task: Letter string analogy

\[ abc \Rightarrow abd \]

\[ xyz \Rightarrow ? \]
Copycat Architecture

Figure 7: A feedback loop between perceptual and conceptual activity.
Copycat/Metacat Slipnet
Running Metacat
Cognitive Architecture: Outline

- Architecture in Information Systems
  “An organization of elements to achieve some information processing purpose.”

- History
  Computational theory of mind

- The Baseline/Generic Cognitive Architecture

- Examples
  Soar, LIDA, CopyCat

- Architecture in NN / Deep Learning Networks

- Conversational Agents
  - NLP
  - Dialog Management
  - Knowledge graphs
  - Deep Networks
Question answering task

“Alexa, who won the 1934 world series?

“The Saint Louis Cardinals beat the Detroit Tigers 4-3 in the 1934 World Series.”

“Alexa, what is Sam Taylor’s phone number?

“Sam Taylor’s telephone number in Boston is area code six-five-eight, one two three, four five six seven.”
Architecture of a Conversational Agent

- **Dialog Manager**
  - **ASR** (Automatic Speech Recognition)
  - **LogicalForm Knowledge**
  - **Linguistic Knowledge**
  - **TTS** (Text To Speech)
  - **Words**
  - **Natural Language Processing**
  - **Response out**
  - **Speech in**
"What is Sam Taylor's phone number?"

"What is the number for Sam in Boston?"

"Tell me Sam Taylor's telephone number."

"What do I dial to reach Sam Taylor?"

**LogicalForm:**

Query ( AddressBookItem ( Person ( name "Sam Taylor" ), ( telephoneNumber ? ) ) )
NLP Pipeline Elements

Theoretical, Comprehensive

Part-of-Speech Tagging

Stemming

Named Entity Recognition

Entity Labeling

Intent Classification

Practical, Shortcuts

Co-Reference Resolution

Sentence Parsing

Semantic Slot Filling
Intent Classification

Practical but Limited

What is Sam Taylor’s phone number?

Data-Value-Question

Input text → features → ML Classifier → Intent class → Intent-specific parameter & slot extraction

bag-of-words

Himang Sharatun
“What is John’s address in Garden Grove?”

1. **common nouns:**
   - garden
   - grove

2. **but uncommon to see the bi-gram,**
   - .....”garden grove”...

3. **City database:**
   - “Garden Grove, California”

4. **Lexical environment**
   - address in \{ location \}

**NER: Named Entity Recognition**

- rules
- standard ML
Can Neural Networks Produce Symbolic Structures from Sentences?

Input type:
Words = sequence of symbols

“What is Sam Taylor’s phone number?”

Output type:
Logical Form = hierarchical tree of symbols

Query ( AddressBookItem ( Person ( name “Sam Taylor” ),
( telephoneNumber ? ) ) )
Figure 3: Sequence-to-tree (SEQ2TREE) model with a hierarchical tree decoder.
Word Vector Embedding

**Purpose:**
- lower dimensionality reduces data sparsity for learning
- similar words :: similar vectors leads to improved generalization

<table>
<thead>
<tr>
<th>word vocabulary size</th>
<th>embedding vector size</th>
</tr>
</thead>
<tbody>
<tr>
<td>e.g. 10k</td>
<td>e.g. 300</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>what</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>.</th>
<th>.</th>
<th>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>why</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>who</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>&lt;person&gt;</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>&lt;location&gt;</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>phone</th>
<th>1</th>
<th>0</th>
<th>.</th>
<th>.</th>
<th>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>telephone</td>
<td>0</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

- .7
- .3
- .1
- .04
- .6
- .7
- .15
Purpose:
- lower dimensionality reduces data sparsity for learning
- similar words :: similar vectors leads to improved generalization
NLP: From Words to Logical Forms (Meaning)

"What is Sam Taylor's phone number?"

"What is the number for Sam in Boston?"

"Tell me Sam Taylor's telephone number."

"What do I dial to reach Sam Taylor?"

**Query (AddressBookItem (Person (name "Sam Taylor"), (telephoneNumber ?)))**
Address Book Knowledge Base: Ontology Schema

**AddressBook**
- owner (SL)
- hasItem
  - isInAddressBook
  - hasItem

**AddressBookItem**
- hasPerson
- hasAddress

**Person**
- firstName (SL)
- preferredName (SL)
- lastName (SL)
- telephoneNumber (SL)
- isRelatedTo
- hasAddress

**Mailing Address**
- street (SL)
- city (SL)
- state (SL)
- zip (SL)

**Key**
- Object Type Name
- propertyName
- SL = String Literal value
**Address Book Knowledge Base: Data Example**

**Knowledge Graph**

**Key**
- **Object Type Name**
- **propertyName**
- **SL = String Literal value**

**AddressBook**
- **owner**: Emily Jones
  - **isInAddressBook**
    - **hasItem**
      - **Tamara**
        - **preferredName**: Tam
        - **lastName**: Rogers
        - **telephoneNumber**: (568) 234-1234
      - **Samuel**
        - **preferredName**: Sam
        - **lastName**: Taylor
        - **telephoneNumber**: (658) 123-4567
    - **isInAddressBook**
  - **isInAddressBook**

**AddressBookItem**

**Person**
- **hasPerson**
- **hasPerson**
  - **Tamara**
    - **preferredName**: Tam
    - **lastName**: Rogers
    - **telephoneNumber**: (568) 234-1234
  - **Samuel**
    - **preferredName**: Sam
    - **lastName**: Taylor
    - **telephoneNumber**: (658) 123-4567

**Mailing Address**
- **hasAddress**
  - **street**: 235 Newberry St.
  - **city**: Boston
  - **state**: Mass.
  - **zip**: 02468
"What is Sam Taylor's phone number?"

Intent terms

Entity terms

LogicalForm:
Query ( AddressBookItem ( Person ( name "Sam Taylor" ), ( telephoneNumber ? ) ) )

SPARQL KB Query:
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?telephoneNumber
WHERE { ?addressBookItem i aBI .
?addressBookItem aBI:person ?person .
?person foif:name "Sam Taylor" .
?addressBookItem aBI:telephoneNumber ?telephoneNumber }
SPARQL KB Query:

```sparql
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?telephoneNumber 
WHERE { 
  ?addressBookItem a ABI:person ?person . 
  ?person foif:name "Sam Taylor" . 
  ?addressBookItem a ABI:telephoneNumber ?telephoneNumber 
}
```
Architecture of a Conversational Agent

External Knowledge Resources

Dialog Manager

NLP

ASR

NLG

TTS

LogicalForm

Words

Speech in

Response out
Cognitive Architecture: Standard Model

Knowledge Resources
Long Term Memory
(declarative, procedural, semantic, episodic, skill…)

Common Workspace
(blackboard)
(Working Memory)

perception

motor action

sense organs
effector organs

environment
Why Are Conversational Agents So Dumb?

“Alexa, who won the 1934 world series?”

“The Saint Louis Cardinals beat the Detroit Tigers 4-3 in the 1934 World Series.”

“Alexa, who was the starting pitcher for the Tigers?”

“The probable pitcher for the Tigers on Saturday February 23 has not been announced yet.”
“Alexa, who won the 1934 world series?"

“The Saint Louis Cardinals beat the Detroit Tigers 4-3 in the 1934 World Series.”

“Alexa, who was the president then?”

“This might answer your question. The president of the United States is Donald Trump.”

Needed: A cognitive architecture with contextual memory.
• Remember past Intents, Entities
  • Query terms:
    • baseball
    • world series
    • 1934
  • Response terms:
    • Saint Louis Cardinals
    • Detroit Tigers
    • 4-3

“Who was the manager of the Cardinals?” (St. Louis Cardinals, in 1934)

• Dialog State Tracking
  - slot filling for restricted tasks, e.g. ordering pizza

  “Pizzabot I want a pepperoni pizza with tomato sauce”

<table>
<thead>
<tr>
<th>size</th>
<th>sauce</th>
<th>topping1</th>
<th>topping2</th>
<th>topping3</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>tomato</td>
<td>pepperoni</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

  “Sure, what size?”
Summary: Taxonomy

Intelligent Beings

Natural
- Animal

Human Intelligence

Alien

Artificial Intelligence

Classical AI

Machine Learning
- Symbolic
- Statistical
- Neural Networks
  - Deep Learning

Cognitive Architecture

NN Architecture
- Research scientist in Cognitive Science and AI.
- I build stuff.
LSTM Module

http://colah.github.io/posts/2015-08-Understanding-LSTMs/
Deep Learning Architecture for Parsing to Frame Graph

Figure 1: Neural network architecture of the SLING parser. The input is encoded by a bi-directional LSTM and fed into a recurrent feed-forward (FF) unit that proposes transition system actions. The hidden layer activations and the transition system state are combined to create the input feature vector for the next step. The FF unit is run repeatedly until the transition system has reached a final state.