



*The Non-Action-Centered
Subsystem*

Sebastien Helie, Nick Wilson, Ron Sun
Cognitive Science, Rensselaer Polytechnic Institute

Outline

1. Representation
2. Reasoning
3. Learning
4. Coordination of the NACS and the ACS
5. Episodic memory
6. Simulation examples
7. Theorems and related phenomena
8. Summary

Outline

1. Representation

2. Reasoning
3. Learning
4. Coordination of the NACS and the ACS
5. Episodic memory
6. Simulation examples
7. Theorems and related phenomena
8. Summary

Representation

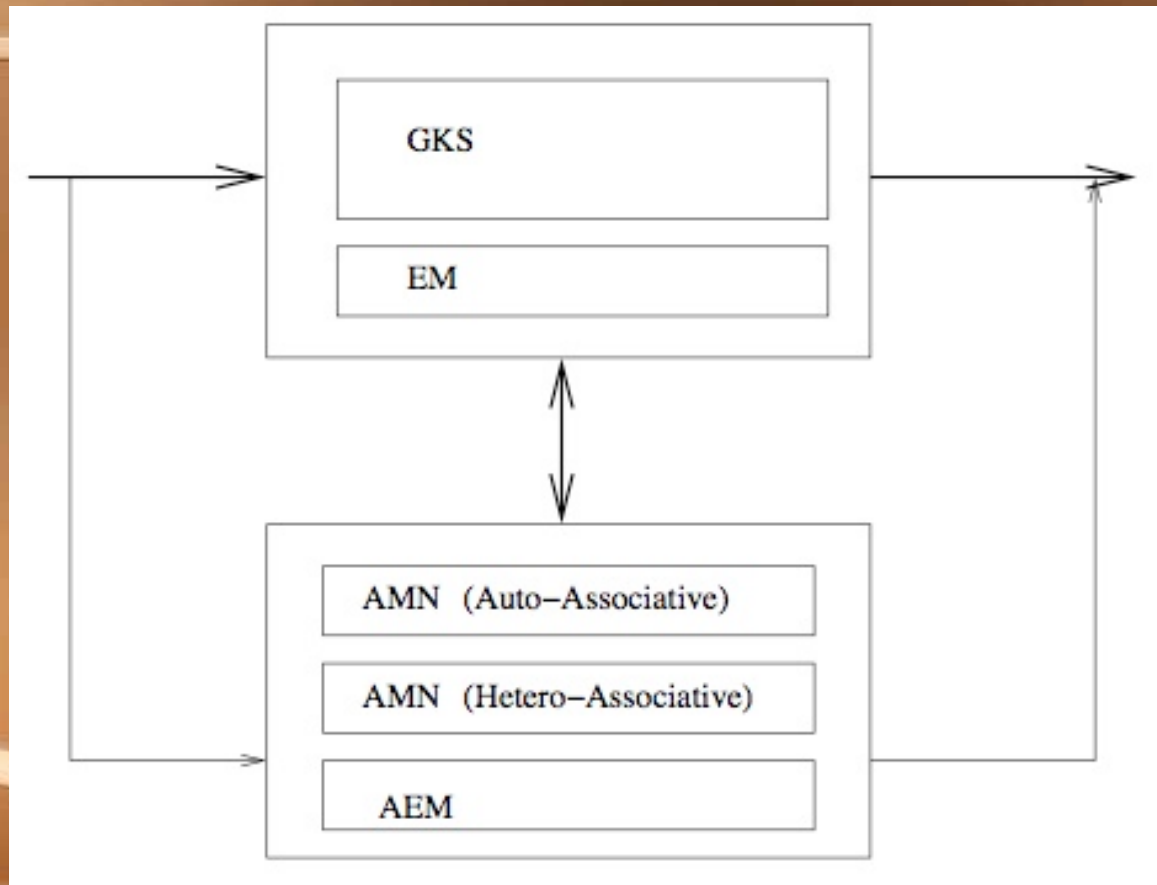
Basic principles (Hélie & Sun, submitted):

1. The co-existence of and difference between explicit and implicit knowledge;
2. The simultaneous involvement of implicit and explicit processes in most tasks;
3. The redundant representation of explicit and implicit knowledge;
4. The integration of the results of explicit and implicit processing;
5. The iterative (and possibly bidirectional) processing.

Representation

- Representing general knowledge about the world -- i.e, the 'semantic' memory
- Performing various kinds of memory retrievals and inferences
- Under the control of the Action-Centered Subsystem (through its actions)
- Formed through acquiring and assimilating general knowledge, from external sources (e.g., the ACS) or from summarizing experiences during action decision making
- NACS includes long-term semantic memory and long-term episodic memory; ACS includes long-term (and short term) procedural memory.

Representation



Representation

- General knowledge store (at the top level) encodes explicit, non-action-centered knowledge.
 - Chunks encode co-occurrences of features.
 - Links across chunks (associative rules) encode explicit associations between chunks. -- Unidirectional or bidirectional
- Similar to the ACS, the bottom level of the NACS can have multiple networks, each for a different kind of information.
- Correspondingly, the top level of the NACS can be divided into multiple rule groups.

Representation

- Chunk-id: $(dim_{i1}, val_{i1})(dim_{i2}, val_{i2}) \dots (dim_{in}, val_{in})$
 - e.g., table-1: (type, table)(size, large)(color, white)(number-of-legs, 4)
- Chunk-id may be externally given (if presented from external source) or generated (randomly) internally.
- Each chunk is represented by a node in the top level
- Each chunk is represented by its feature nodes in the bottom level (more later)

Representation

- Chunks are connected in the top level by explicit associative rules
- An associative rule represents a mappings from a set of chunk nodes to another in an explicit form but without overly complex structures.
- The condition of an associative rule contains one or more chunks.
- The conclusion of an associative rule contains one chunk.

Representation

- Chunks may be activated:
 - As a result of receiving input from the environment or the ACS.
 - By applying an associative rule within the top level.
 - From an associative mapping from the bottom level of the NACS.
 - By similarity-based reasoning.
- The strength of a chunk in the top level is determined by:

$$s_k^{c,GKS} = \max_x (s_k^{c,x})$$

where $s_k^{c,GKS}$ is the activation of chunk k in the top level and x is a particular activation source.

Representation

- Chunks and rules in the top level have base-level activations (as in the ACS):

- Chunks:

$$b_j^c = ib_j^c + c_c \sum_{l=1}^n t_l^{-d_c}$$

where ib_j^c is the initial BLA, c_c is the amplitude, d_c is the decay rate, and t_l is the l th use of the chunk.

- Associative rules:

$$b_j^r = ib_j^r + c_r \sum_{l=1}^n t_l^{-d_r}$$

where the same symbols are used except for the r superscript/superscript.



Representation

Questions?

Representation

- In the bottom level, Associative Memory Networks encode non-action-centered implicit knowledge (e.g., MLP trained with BP).
- Each top-level chunk is represented by a set of features (1 feature = 1 node)
- Bottom-up activation through associative mapping
- Bottom-up activation through similarity-based reasoning

Representation

- Various possibilities of capturing implicit associations in the bottom level:
 - Auto-associative: observed nodes are set as both the input and desired output (e.g., auto-encoder networks, Hopfield networks).
 - Hetero-associative: one set of nodes are presented as the input and another is set as the desired output to create an association between the two (e.g., regular MLPs trained with BP).
- These different ways of using the bottom level may be implemented as separate networks (or a unique network, as needed).

Representation

- The process of top-down activation (Sun, 2002):
 - When a chunk is inferred (activated) in the top level but not in the bottom level, a top-down activation process may be needed to activate corresponding feature nodes in the bottom level.
 - The activation of a feature node (in the bottom level) is set to the strength level of its corresponding chunk.
 - If the feature receives activation from several chunks, the maximum strength is set.

Representation

- The process of bottom-up activation (Sun et al., 2001):
- When the result from the bottom level is sent bottom-up, it activates all chunks compatible with it.
- A *Max* function is used between the strength of activated chunks from both bottom-up activation and from within the top level to determine the overall strength of a chunk.

$$s_i^c = \max\left(s_i^{c,GKS}, s_i^{c,AMN}\right)$$

where s_i^c is the activation of chunk i , $s_i^{c,GKS}$ is the top-level activation of chunk i , and $s_i^{c,AMN}$ is the bottom-up activation of chunk i .

Outline

1. Representation
- 2. Reasoning**
3. Learning
4. Coordination of the NACS and the ACS
5. Episodic memory
6. Simulation examples
7. Theorems and related phenomena
8. Summary

Reasoning

- Starts with input to the NACS by ACS actions (either to the bottom level, top level, or both).
- Activation levels of input are always 1 (full activation.)
- Bottom-up and top-down activation ensures that a full activation of all appropriate chunk nodes occurs regardless of type of input provided.

Reasoning

- Each round of NACS reasoning:
 - Starts with the activated feature nodes on the input side of the bottom level for associative mapping.
 - One round of associative mapping activates a set of feature nodes on the output side.
 - Concurrently, an iteration of inference occurs in the top level starting from all the currently activated chunks.
 - All applicable associative rules fire simultaneously (there is no competition/selection among associative rules).
 - New chunks are inferred in the top level as a result.



Reasoning

- The outcomes of the bottom and top levels are integrated by bottom-up activation;
- Top-down activation is used to activate the features of the newly activated chunks in the input of the bottom-level networks;
- Another round of reasoning may take place.

Reasoning

- Similarity-based reasoning may be employed
 - During reasoning, a known (given or inferred) chunk may be automatically compared with another chunk. If the similarity between them is sufficiently high, then the latter chunk is inferred.
- Mixed rule-based and similarity-based reasoning
 - Accounting for a large variety of commonsense reasoning patterns (including ‘inheritance reasoning’). See Sun (1994, 1995).

Reasoning

- Reasoning methods in the top level of NACS:
 - Forward chaining reasoning: For drawing all possible conclusions in a forward direction -- from known conditions to new conclusions (Smith, Langston, & Nisbett, 1992)
 - Similarity-based forward chaining reasoning -- for drawing all possible conclusions, using rules as well as similarity-based inference (Sun, 1994, 1995)
 - In both cases, there is a threshold that determine whether a conclusion is acceptable or not.
 - (By default, rule utility is not used in the NACS.)

Reasoning

- Rule-based reasoning:

$$s_j^a = \sum_i s_i^c \times w_{ij}^r$$

where s_j^a is the activation of chunk j received from chunk i , s_i^c is the activation of premise chunk i , and w_{ij}^a is the strength of the rule between chunk i and chunk j .

- When several rules activate chunk j , the maximum received activation is used:

$$s_{c_k}^{c,a} = \max_{j \in c_k} (s_j^a)$$

where $s_{c_k}^{c,a}$ is the activation of a chunk from RBR.

Reasoning

- Similarity-based reasoning:

$$s_{c_j}^{c,s} = \max_i \left(s_{c_i \sim c_j} \times s_{c_i}^c \right)$$

where $s_{c_j}^{c,s}$ is the activation of chunk c_j from SBR, $s_{c_i \sim c_j}$ is the similarity between chunks c_i and c_j , and $s_{c_i}^c$ is the total activation of chunk c_i (from both RBR and SBR).

Reasoning

- And the default similarity measure is:

$$s_{c_i \sim c_j} = \frac{n_{c_i \cap c_j}}{f(n_j)} = \frac{\sum_{k \in c_i \cap c_j} V_k^{c_j} \times A_k}{f\left(\sum_{k \in c_j} V_k^{c_j} \times D_k\right)}$$

where $n_{c_i \cap c_j}$ is the number of features overlapping between chunks c_i and c_j , n_j is the number of features in chunk c_j , A_k is the activation of the k th feature included in $c_i \cap c_j$, $V_k^{c_j}$ is the weight of the k th feature, and $f(\cdot)$ is a slightly superlinear function.

- Note that the similarity measure is bounded in the interval $[0, 1)$.

Reasoning

The reverse containment principle (assumption):

- The feature representations of the NACS chunks are organized to emulate an “ideal” categorical hierarchy;
- Hence, if chunk i represents a category that is a superset (e.g., furniture) of the category represented by chunk j (e.g., table), the feature set of chunk j contains the feature set of chunk i ($n_{ci \cap cj} = n_i$).

Reasoning

- Mixing RBR and SBR (Similarity-based forward chaining reasoning; Sun & Zhang, 2006):

$$s_{c_i}^c = \max(\alpha \times s_{c_i}^{c,a}, \beta \times S_{c_i}^{c,s})$$

where $s_{c_i}^c$ is the final activation of chunk c_i , α and β are scaling parameter for RBR and SBR respectively, $s_{c_i}^{c,a}$ is the activation of chunk c_i from RBR, and $s_{c_i}^{c,s}$ is the activation of chunk c_i from SBR.

- This reasoning method can be applied iteratively;
- It can also be sequenced with pure RBR or pure SBR.

Outline

1. Representation
2. Reasoning
- 3. Learning**
4. Coordination of the NACS and the ACS
5. Episodic memory
6. Simulation examples
7. Theorems and related phenomena
8. Summary



Learning

- LTM in the NACS
 - Learning in the top level
 - Learning in the bottom level
 - Top-down learning in the NACS
 - Bottom-up learning in the NACS

Learning

- Learning explicit knowledge
 - Encoding of externally given explicit knowledge (chunks and rules)
 - Under the control of the ACS
 - Serves as a LTM (with an encoding probability parameter)
 - Extraction of explicit knowledge
 - Extraction from the bottom level in the ACS (same as in the top level of the ACS)
 - Extraction from the bottom level of the NACS (using thresholds, as in ACS)

Learning

- Learning implicit knowledge
 - Training of the bottom-level networks is under the control of the ACS
 - Assimilation of explicit knowledge through training a bottom-level network using associations stored in the episodic memory
 - At each step, a subset of items from EM is used to train the bottom level (with a certain probability)



Outline

1. Representation
2. Reasoning
3. Learning
- 4. Coordination of the NACS and the ACS**
5. Episodic memory
6. Simulation examples
7. Theorems and related phenomena
8. Summary

Coordination of the NACS and the ACS

- Usually, the NACS is under complete control of the ACS (action-directed reasoning)
- For instance, an ACS action might be to perform a round of reasoning
- The ACS (or MCS) specifies the type of reasoning to be done in the NACS (e.g., RBR, SBR, RBR + SBR, and so on).

Coordination of the NACS and the ACS

- The outcome of reasoning in the NACS may be sent back to the ACS
- If only one outcome from the NACS needs to be selected and sent back to the ACS, Boltzmann selection is used.
- Alternatively, the ACS may choose to retrieve all outcomes from NACS reasoning or only a certain type of outcomes (e.g., outcomes that were not part of the input)

Outline

1. Representation
2. Reasoning
3. Learning
4. Coordination of the NACS and the ACS
- 5. Episodic memory**
6. Simulation examples
7. Theorems and related phenomena
8. Summary

Episodic memory

- Episodic memory (Tulving, 1983)
 - EM stores past events, with time stamps of when those occurred
 - Action-oriented experience
 - Non-action-oriented experience
 - EM is recency-filtered (BLA + threshold)
 - EM chunks have an encoding probability parameter

Episodic memory

- EM chunks have a bottom-level (feature-based) representation
- The time stamp can be viewed as a special feature
- EM may be used to help learning
 - EM stores all the associative rules applied, all the associations given externally, all the associations representing the mapping from the input to the NACS and each of the resulting (inferred) chunks.
 - Any of those can be selected for training the bottom level of the NACS or the ACS.

Episodic memory

- Abstract episodic memory
 - AEM summarizes information of past episodes experienced by the ACS
 - Used to help learning and extract explicit knowledge from the bottom level of the ACS
 - AEM is constituted by
 - An action frequency network
 - "State --> action" frequency distribution
 - A state frequency network
 - "State, Action --> next state" frequency distribution
 - "State, Action --> immediate reinforcement"

Episodic memory

- Both networks contain localist representations
- Both networks are trained using backpropagation learning
- Training is based on the content of EM
- AEM may be used to help training the ACS (just like EM)

Outline

1. Representation
2. Reasoning
3. Learning
4. Coordination of the NACS and the ACS
5. Episodic memory
- 6. Simulation examples**
7. Theorems and related phenomena
8. Summary

Simulation

- The categorical inference task (Sloman, 1998; Sun & Zhang, 2006)
 - Premise specificity
 - All flowers are susceptible to thrips => All roses are susceptible to thrips
 - All plants are susceptible to thrips => All roses are susceptible to thrips
 - Inclusion similarity
 - All plants contains bryophytes => All flowers contain bryophytes
 - All plants contains bryophytes => All mosses contain bryophytes

Simulation

- Which argument is the stronger?
 - Premise specificity: flower => rose (82%) vs. plant => rose (18%)
 - Inclusion similarity: plant => flower (91%) vs. plant => mosses (9%)
- Average likelihood of arguments
 - Premise specificity: 0.86 (flower => rose)
 - Inclusion similarity: 0.89 (plant => flower)
- These results show the presence of SBR
 - If only RBR was used, argument strength ~ 50%
 - Likelihood of arguments ~ 1.

Simulation

- Scaling parameter: $\alpha = 0.5$, $\beta = 1.0$
- The top level includes inclusion rules
 - "Flowers are plants"
 - "Mosses are plants"
 - Etc...
- The features of the chunks (e.g., "flowers", "mosses") were represented in the bottom level (e.g., "petal", "stem", and other unrecognizable features)

Simulation

1. The chunks represented in the premises of the conclusion statements were activated in the top level;
2. Which in turn activated their features in the bottom level;
3. RBR was performed in the top level;
4. SBR was performed in the bottom level;
5. The results of RBR and SBR were integrated (*Max* function)

Simulation

- Simulation results
 - Which argument is the stronger?
 - Premise specificity: flower => rose (83%) vs. plant => rose (17%)
 - Inclusion similarity: plant => flower (82%) vs. plant => mosses (18%)
 - Average likelihood of arguments
 - Premise specificity: 0.87 (flower => rose)
 - Inclusion similarity: 0.86 (plant => flower)
- These results provide a good match to the human data.

Simulation

- Other experimental conditions
 - Explicitly stating the inclusion argument (0.99)
 - All plants contain bryophytes. All mosses are plants. => All mosses contain bryophytes.
 - Having participants make the categorical inclusion judgment before estimating the likelihoods (0.92):
 - Are all mosses plants?
 - All plants contain bryophytes. => All mosses contain bryophytes.

Simulation

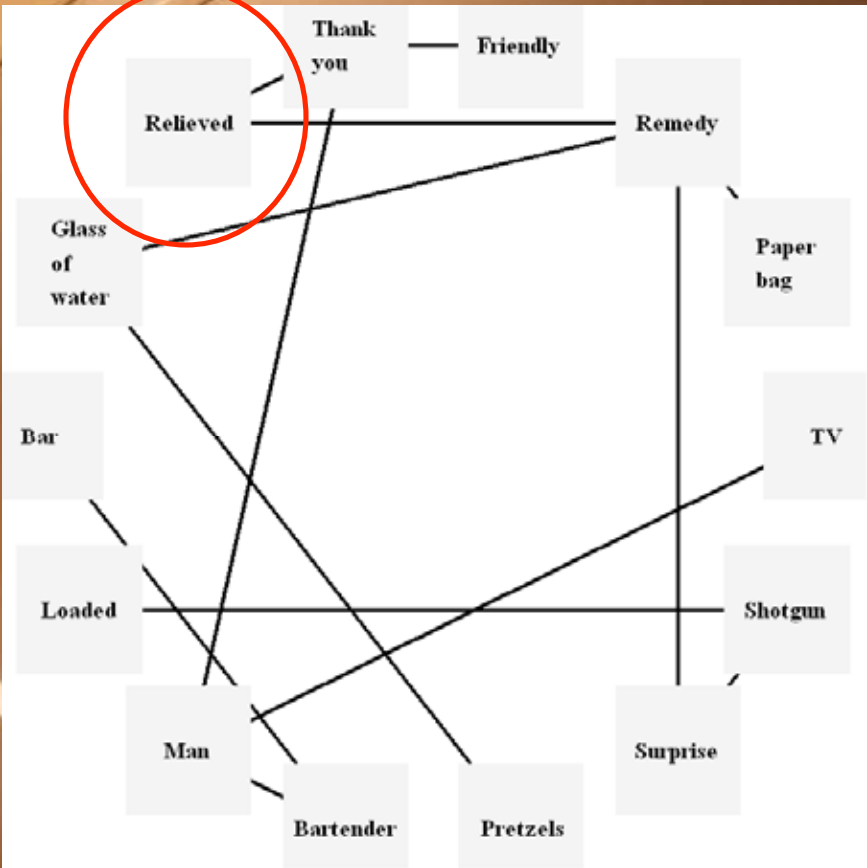
- In CLARION, this amounts to changing the weight of RBR in knowledge integration
 - Explicitly stating the inclusion argument:
 - $\alpha = \beta = 1.0$
 - Mean likelihood: 0.99
 - Having participants make the categorical inclusion judgment:
 - $\alpha = 0.88, \beta = 1.0$
 - Mean likelihood: 0.91

Simulation

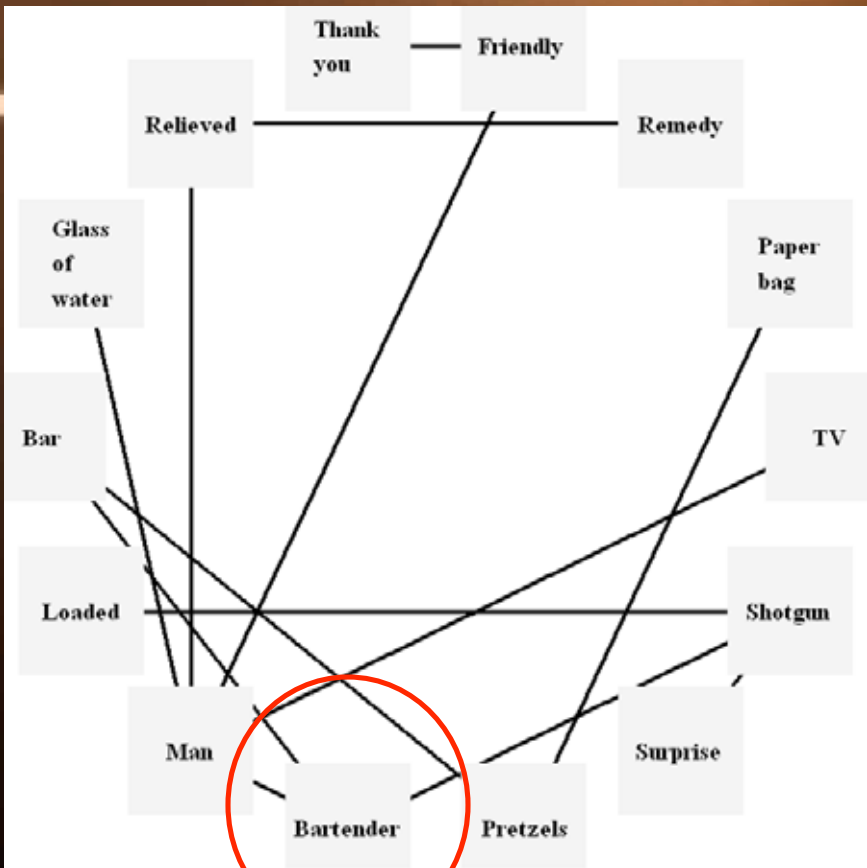
- Insight in problem solving (Durso, Rea, & Dayton, 1994; Hélié & Sun, submitted)

“A man walks into a bar and asks for a glass of water. The bartender points a shotgun at the man. The man says ‘thank you’, and walks out.”

Simulation



Solvers



Non-solvers

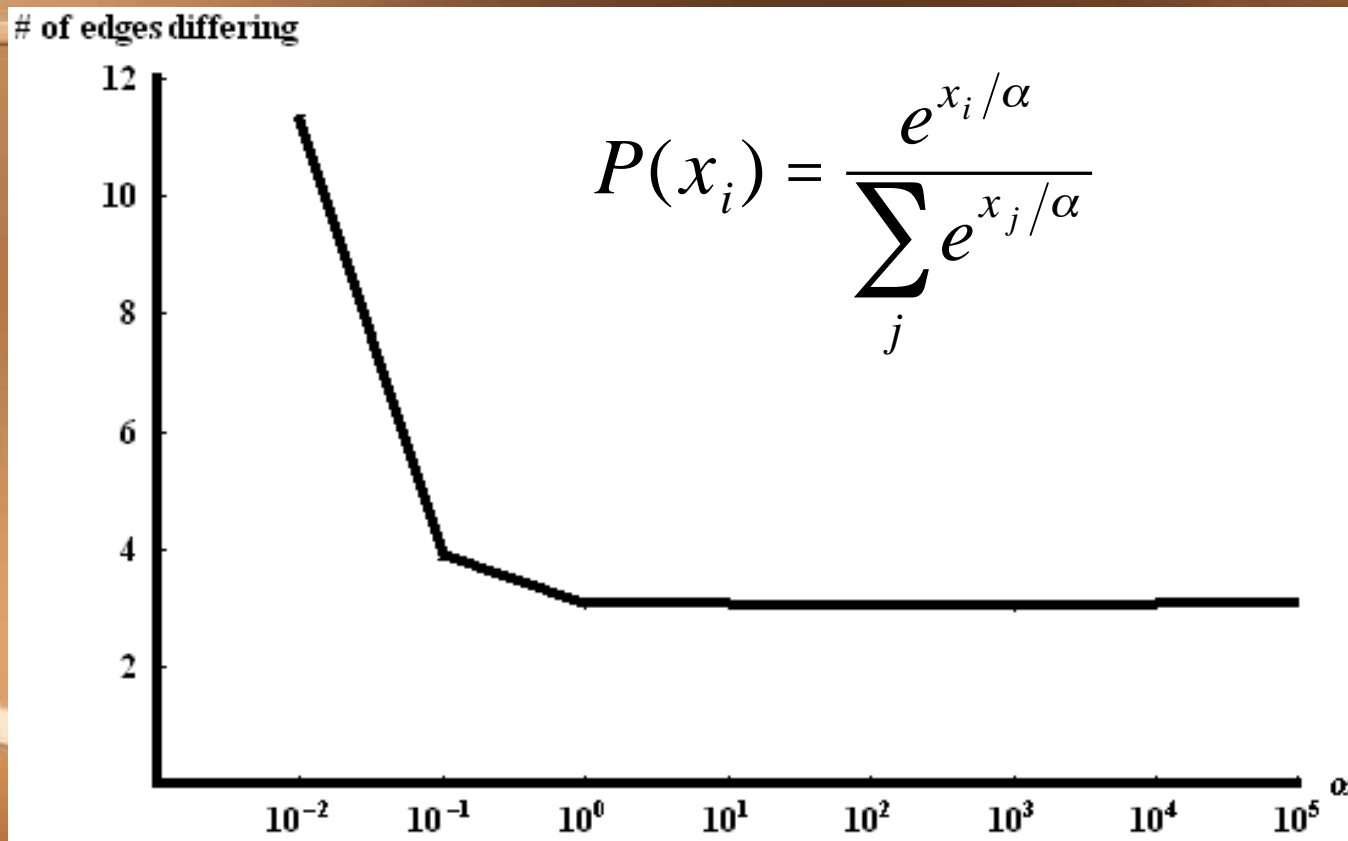
Simulation

- Here, each concept (node in the graphs) is represented in the top level by a chunk;
- Each chunk is represented by a random set of features in the bottom level;
- Top-level rules represent the links in the non-solvers' graph (i.e., culturally shared semantic knowledge);
- Positive exemplars of the top-level rules are used to encode similarity-based associations in the bottom level.

Simulation

- A round of SBR is initiated in the bottom level using random activation;
- The result of SBR is sent bottom-up to activate a chunk;
- RBR is used to activate other chunks in the top level;
- A response is stochastically chosen to be sent to the ACS.

Simulation



Outline

1. Representation
2. Reasoning
3. Learning
4. Coordination of the NACS and the ACS
5. Episodic memory
6. Simulation examples

7. Theorems and related phenomena

8. Summary

Theorems

- Many common reasoning patterns can be handled by CLARION:
 - Inexact information
 - Incomplete information
 - Similarity matching
 - Superclass to subclass inheritance
 - Subclass to superclass inheritance
 - Cancellation of inheritance (both types)
 - Etc.

Theorems

Case 1: Inexact information

Network state: Let c_i and c_j be chunks in the top level of the NACS, w_{ij}^r be a rule in the top level of the NACS linking chunks i and j . Assume that $s_i < 1$.

Result: $s_j = s_i$.

Case 2: Incomplete information

Network state: Let $c_{i1}, c_{i2}, c_{i3}, c_j$ be chunks in the top level of the NACS, $w_{i1j}^r, w_{i2j}^r, w_{i3j}^r$ be rules in the top level of the NACS linking chunks i_k and j . Assume that $s_{i1} = s_{i2} = 1$ and that $s_{i3} = 0$.

Result: $s_j = 2/3$.

Theorems

Case 3: Similarity matching

Network state: Let c_i, c_j, c_k be chunks in the top level of the NACS, $s_{c_i-c_j}$ be the similarity between chunks i and j , and w_{jk}^r be a rule in the top level of the NACS linking chunks j and k . Assume that $s_i = 1$.

Result:

$$s_k = \frac{n_{c_i \cap c_j}}{f(n_j)}$$

Case 4: Superclass to subclass inheritance

Network state: Let c_i, c_j, c_k be chunks in the top level of the NACS, the category represented by c_i is a proper subset of the category represented by c_j , and w_{jk}^r be a rule in the top level of the NACS linking chunks j and k . Assume that $s_i = 1$.

Result: $s_k \approx 1$.

Theorems

Case 5: Subclass to superclass inheritance

Network state: Let c_i, c_j, c_k be chunks in the top level of the NACS, the category represented by c_i is a proper subset of the category represented by c_j , and w_{ik}^r be a rule in the top level of the NACS linking chunks i and k . Assume that $s_j = 1$.

Result: $s_k < 1$.

Case 6: Cancellation of superclass to subclass inheritance

Network state: Let c_i, c_j, c_k, c_m be chunks in the top level of the NACS, the category represented by c_i is a proper subset of the category represented by c_j , and w_{jk}^r and w_{im}^r be rules in the top level of the NACS. Assume that $s_j = 1$.

Result: $s_m = 1 > s_k$.

Theorems

Case 7: Cancellation of subclass to superclass inheritance

Network state: Let c_i, c_j, c_k, c_m be chunks in the top level of the NACS, the category represented by c_j is a proper subset of the category represented by c_i , and w_{jk}^r and w_{im}^r be rules in the top level of the NACS. Assume that $s_i = 1$.

Result: $s_m = 1 > s_k$.

Case 8: Mixed rules and similarities

(1) *State of the network:* Let c_i, c_j, c_k be chunks in the top level of the NACS, $s_{c_j \sim c_k}$ be the similarity between chunks j and k , and w_{ij}^r be a rule in the top level of the NACS between chunks i and j . Assume that $s_i = 1$.

Result:
$$s_k = \frac{n_{c_j \cap c_k}}{f(n_k)}$$

Theorems

- (2) *State of the network:* Let c_i, c_j, c_k be chunks in the top level of the NACS, and w_{ij}^r, w_{jk}^r be rules in the top level of the NACS. Assume that $s_i = 1$.

Result: $s_k = 1$.

- (3) *State of the network:* Let c_i, c_j, c_k, c_m be chunks in the top level of the NACS, s_{ci-cj} is the similarity between chunks i and j , and w_{jk}^r and w_{km}^r be rules in the top level of the NACS. Assume that $s_i = 1$.

Result:

$$s_m = \frac{n_{c_i \cap c_j}}{f(n_j)}$$

Theorems

(4) *State of the network*: Let c_i, c_j, c_k, c_m be chunks in the top level of the NACS, $s_{c_j \sim c_k}$ is the similarity between chunks j and k , and w_{ij}^r and w_{km}^r be rules in the top level of the NACS. Assume that $s_i = 1$.

Result:
$$s_m = \frac{n_{c_j \cap c_k}}{f(n_k)}$$

(5) *State of the network*: Let c_i, c_j, c_k, c_m be chunks in the top level of the NACS, $s_{c_k \sim c_m}$ is the similarity between chunks k and m , and w_{ij}^r and w_{jk}^r be rules in the top level of the NACS. Assume that $s_i = 1$.

Result:
$$s_m = \frac{n_{c_k \cap c_m}}{f(n_m)}$$

Theorems

(6) *State of the network*: Let c_i, c_j, c_k, c_m be chunks in the top level of the NACS, $s_{c_i \sim c_j}$ and $s_{c_k \sim c_m}$ are similarity measures between the chunks, and w_{jk}^r is a rule in the top level of the NACS between chunks j and k . Assume that $s_i = 1$.

Result:

$$s_m = \frac{n_{c_i \cap c_j}}{f(n_j)} \times \frac{n_{c_k \cap c_m}}{f(n_m)}$$

Theorems

Case 4: Superclass to subclass inheritance

Network state:

Let c_i , c_j , c_k be chunks in the top level of the NACS, the category represented by c_i is a proper subset of the category represented by c_j , and w_{jk}^r be a rule in the top level of the NACS linking chunks j and k . Assume that $s_i = 1$.

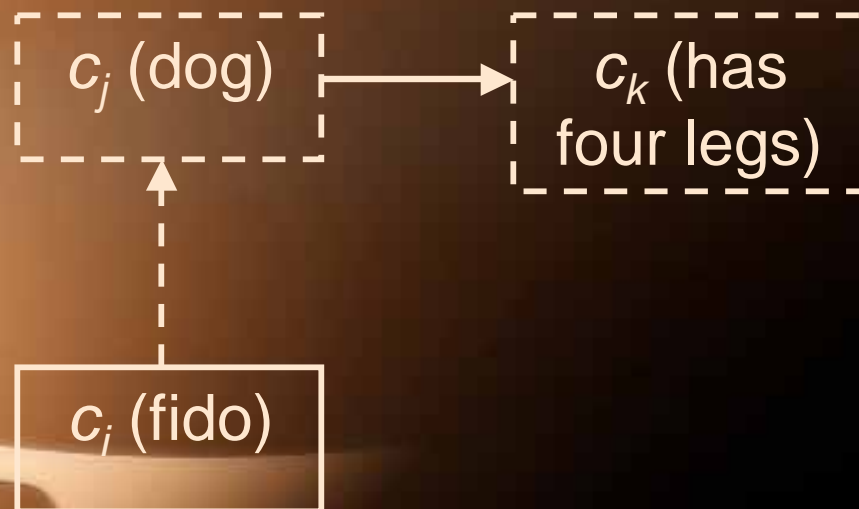
Derivation:

$$s_k = s_i \times s_{c_i \sim c_j} \times w_{jk}^r = \frac{n_{c_i \cap c_j}}{f(n_j)} = \frac{n_j}{f(n_j)} \approx 1$$

In words, chunk k is activated because chunk i fully activates chunk j (up to the slight non-linearity of $f()$, which is negligible). Chunk j has a top-level rule that transmits its activation to chunk k .

Theorems

Case 4: Superclass to subclass inheritance



Theorem

Case 6: Cancellation of superclass to subclass inheritance

Let c_i, c_j, c_k, c_m be chunks in the top level of the NACS, the category represented by c_i is a proper subset of the category represented by c_j , and w_{jk}^r and w_{im}^r be rules in the top level of the NACS. Assume that $s_i = 1$.

Derivation:

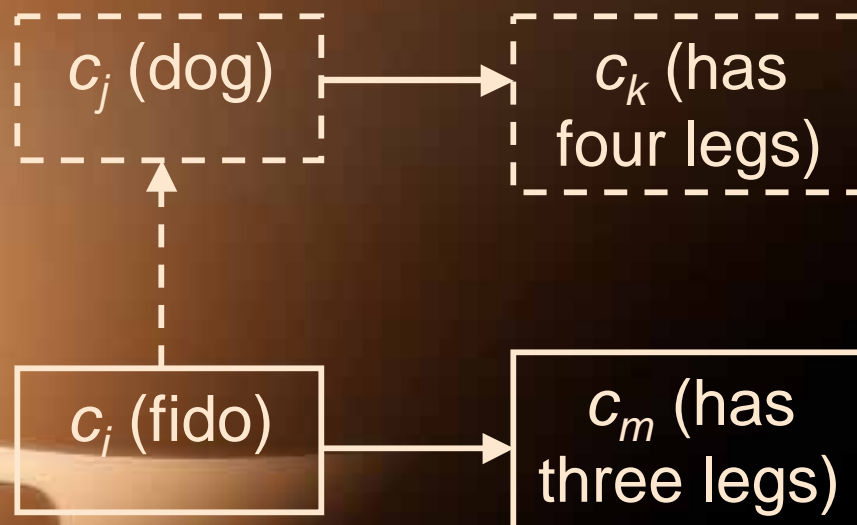
$$\begin{aligned} s_k &= s_i \times s_{c_i \sim c_j} \times w_{jk}^r \\ &= \frac{n_{c_i \cap c_j}}{f(n_j)} \\ &= \frac{n_j}{f(n_j)} \\ &< 1 \end{aligned}$$

$$\begin{aligned} s_m &= s_i \times w_{im}^r \\ &= 1 \end{aligned}$$

Hence, $s_m > s_k$. In words, chunk k is almost fully activated, but the denominator is slightly bigger than the numerator in its derivation [because $f()$ is super-linear]. In contrast, chunk m is fully activated, because top-level rules are exact. This shows the superiority of rule-based reasoning over similarity-based reasoning.

Theorems

Case 6: Cancellation of superclass to subclass inheritance



Theorems

Case 7: Cancellation of subclass to superclass inheritance (i.e., reversing induction)

Network state:

Let c_i, c_j, c_k, c_m be chunks in the top level of the NACS, the category represented by c_i is a proper subset of the category represented by c_j , and w_{im}^r and w_{jk}^r be rules in the top level of the NACS. Assume that $s_j = 1$.

Derivation:

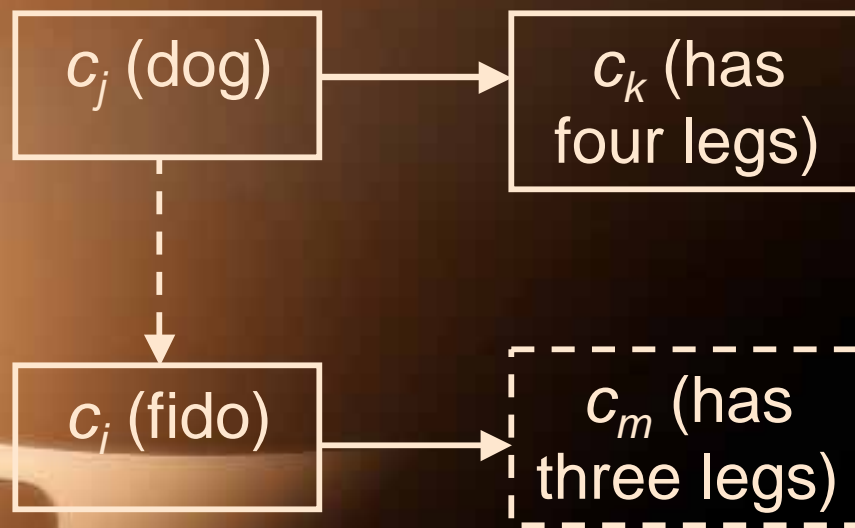
$$\begin{aligned} s_m &= s_j \times s_{c_j \sim c_i} \times w_{im}^r \\ &= \frac{n_{c_j \cap c_i}}{f(n_i)} \\ &= \frac{n_j}{f(n_i)} \\ &< 1 \end{aligned}$$

$$\begin{aligned} s_k &= s_j \times w_{jk}^r \\ &= 1 \end{aligned}$$

Hence, $s_k > s_m$. In words, chunk m is partially activated, because chunk i has more features than chunk j (remember that chunk i represents a proper subset of chunk j). On the other hand, chunk k is fully activated, because top-level rules are exact.

Theorems

Case 7: Cancellation of subclass to superclass inheritance (i.e., reversing induction)



Outline

1. Representation
2. Reasoning
3. Learning
4. Coordination of the NACS and the ACS
5. Episodic memory
6. Simulation examples
7. Theorems and related phenomena

8. Summary

Summary

- The Non-Action-Centered Subsystem in CLARION is divided into a top level and a bottom level
- The top level contains explicit knowledge in the form of rules and chunks
- The bottom level contains implicit knowledge in the form of feature similarity

Summary

- RBR is included in the top level
- SBR is included in the bottom/top level
- The outcomes of the two levels are integrated with the *Max* function
- Explicit knowledge can be learned from the environment or extracted from the bottom level of the ACS and the NACS

Summary

- Implicit learning can be done by assimilation of explicit knowledge (or by training using the EM content)
- CLARION can account for a variety of reasoning phenomena involving varying amount of SBR and RBR.



Summary

Questions?