AN INTEGRATIVE ACCOUNT OF MEMORY AND REASONING

PHENOMENA

Sébastien Hélie
Purdue University

Ron Sun
Rensselaer Polytechnic Institute

Running head: An integrative account of memory and reasoning

For correspondence,

Sébastien Hélie
Department of Psychological Sciences
Purdue University
703 Third Street
West Lafayette, IN 47907-2081

Phone: (765) 496-2692
E-mail: shelie@purdue.edu

Version R3, last modified April 25th 2014.
Abstract

There is growing consensus that human memory is mediated by multiple qualitatively different systems co-evolved to function in a complementary way. As a result, memory should be studied not only using direct tests of memory but also using other tasks that naturally require memory access. This article presents an attempt at using the declarative memory systems in CLARION (termed the Non-Action-Centered Subsystem or NACS) to account for a wide range of psychological phenomena involving both the direct and indirect use of declarative memory. We advocate an architectural approach, which is broad-based (rather than depth-based). As such, the explanations presented herein, from psychological domains as diverse as human memory, deductive reasoning, inductive reasoning, and heuristic reasoning, were based on architectural properties of CLARION and most of them did not require the adjustment of any numerical parameter. This article concludes with a comparison of CLARION with alternative views of memory systems.

Keywords: psychology, declarative memory, reasoning, cognitive architecture, CLARION.
Introduction

There is growing consensus that human memory is mediated by multiple qualitatively different systems (Ashby, Alfonso-Reese, Turken, & Waldron, 1998; Rolls, 2000; Squire & Schacter, 2002; Tulving, 2002) that have co-evolved to function in a complementary way (e.g., Klein, Cosmides, Tooby, & Chance, 2002; Schneider, 1993; Sun, 2002). As a result, dissociations among memory systems may exist under some circumstances, so that different memory systems may serve different purposes, but dissociations may not be prominent under some other circumstances, so that multiple memory systems may be brought together to bear on one task.\(^1\) Investigating the evolutionary problems faced by each memory system may help to achieve a better understanding and delineate the design features of each memory system. Thus, it can be useful to take a careful look at memory systems from a non-memory-centered, broader perspective (Sun, 2012). In the process, we advocate an architectural approach toward memory systems (and cognitive systems in general), which is broad-based rather than depth-based and almost parameter-free and simulation-free.

Theoretical dichotomies

When focusing on the type of tasks (functions) that can be achieved by different memory systems, the declarative/procedural dichotomy immediately appears as a useful concept. The distinction between procedural and declarative memories has been proposed by, e.g., Anderson (1983), Squire (1987), and others (although some details vary across these proposals). Procedural memory contains knowledge that is specifically concerned with actions in various circumstances, that is, how to do things. Declarative memory
contains knowledge that is not specifically concerned with actions, but more about objects and events in generic terms. For example, declarative memory may contain propositions about the state of the world with a measure of their support (or truth value). The major factor that distinguishes procedural and declarative memories seems to be the action-centeredness or the lack thereof (that is, the procedural versus non-procedural nature of knowledge; Sun, Zhang, & Mathews, 2009). Evidence in support of this distinction includes voluminous studies of skill acquisition in both high- and low-level skill domains (e.g., Anderson & Lebiere, 1998; Helie, Waldschmidt, & Ashby, 2010; Kanfer & Ackerman, 1989).

Another (alternative) dichotomy is concerned with memory accessibility (i.e., explicit vs. implicit). Specifically, a theoretical distinction between accessible and inaccessible processing has been proposed (for a review, see, e.g., Eichenbaum, 1997). According to this framework, explicit knowledge is based on conceptual (consciously accessible) processing and implicit knowledge is based on (consciously inaccessible) subconceptual processing. This distinction is also supported by many empirical results (for reviews, see, e.g., Helie & Sun, 2010b; Sun, Slusarz, & Terry, 2005).

While these dichotomies are often merged (e.g., the claim that procedural memory is implicit while declarative memory is explicit), Sun et al. (2009) argued that the dichotomies may be better treated as orthogonal, with procedural memory being either (or both) explicit and implicit, and declarative memory also being either (or both) explicit and implicit. This is the approach advocated in the CLARION cognitive architecture (Sun, 2002; Sun et al., 2005). In CLARION, procedural and declarative memories are represented as separate modules, and each of these modules is further subdivided into an
implicit and an explicit component (thus decoupling the two dichotomies). So far, most of the published work on the CLARION cognitive architecture (e.g., Sun et al., 2005; Sun, Merrill, & Peterson, 2001; Sun & Peterson, 1998) has focused on the interaction of explicit and implicit processing within the procedural system (called the Action-Centered-Subsystem or ACS). However, the declarative system (called the Non-Action-Centered-Subsystem or NACS) has not received as much attention (with some exceptions; e.g., Helie & Sun, 2010b; Sun & Helie, 2013). This article aims at filling this gap. We present a more detailed presentation of the NACS and show how the interaction between explicit and implicit processing within the NACS can be used to account for a variety of psychological phenomena in memory and reasoning. This is important because, as argued in more detail below, memory serves an important function in supporting intelligent behavior, and testing models/theories using only memory-centered tasks and functions does not inform us on the role of memory in other intelligent behaviors. This is done by first assessing the CLARION NACS as a model of declarative memory, and then showing that it is sufficient to support a wide range of reasoning activities in humans.

An architectural approach to cognition

In the present work, we advocate an architectural approach toward cognitive modeling, which means that we are advocating a broad-based account rather than depth-based (in the same way as architectural sketches, necessary before buildings are constructed). It is our view that this is especially important for the advancement of psychology because it allows for the exploration of how the different component models interact and fit together (instead focusing on individual smaller-scoped computational models that focus on smaller details). The study of architectural issues provides new
An integrative account of memory and reasoning 6

insight, and narrows down possibilities to delineate the processes involved in cognition (Helie & Sun, 2014).

The architectural approach is also what Newell (1990) argued for: more data could be used to constraint a cognitive theory if the theory was designed to explain a wider range of psychological phenomena. In particular, these ‘unified’ (i.e., integrative) psychological theories could be put to the test against well-known (stable) regularities that have been observed in psychology (e.g., the power law of practice, the serial position curve in free recall, etc.; Murdock, 1962; Newell & Rosenbloom, 1981). We would add that one advantage of using such a broad-based architectural approach is that it allows for accounting for data with (almost) parameter-free explanations by deriving or numerically estimating mathematically predicted outcomes from intrinsic properties of the architecture. This is accomplished by not focusing our work on overly fine-grained modeling attempting to capture all the nuances of data involved in the selected phenomena, but rather on general psychological ‘laws’. This is justified because (1) finer-grained data can sometimes represent noise (Pitt & Myung, 2002), (2) minute details can usually be captured by adding some minor activation mechanisms within the same components (e.g., for short-term priming, etc.) and, (3) abstracting away from the details facilitates the exploration of the interactions of architectural components in explaining empirical data. The main objective of this article (and of the CLARION cognitive architecture) is to select and include only a minimum set of mechanisms, structured in a parsimonious but effective way, to account for a maximum set of psychological data and phenomena to explore the interaction between the architectural components of memory.
In this article, we review a range of psychological phenomena, and present explanations of the phenomena using the CLARION NACS. The first set of phenomena considered is related to human memory, to ensure that the CLARION NACS can account for basic declarative memory phenomena. Then we build on this model by showing that the CLARION NACS is a model of declarative memory sufficient for supporting other cognitive functionalities, such as deductive reasoning, inductive reasoning, and heuristic reasoning. Newell (1992) pointed out that it was unclear whether most memory models available in the research literature could support intelligent behavior. The issue of the functionality of memory was taken up again in Anderson (2007), who argued that studying integrative cognitive architectures was a good means of addressing this issue.

With such a wide area of applications, the list of phenomena included in this article is obviously not exhaustive. The phenomena were selected based on (1) their historical importance (e.g., as evidenced by inclusion in major reviews of the fields) and (2) reproducibility. Furthermore, because we believe that the interaction between explicit and implicit declarative memories is essential to the explanation of these psychological phenomena, CLARION is a natural candidate architectural theory. First, the interaction between explicit and implicit processing is one of the main principles underlying the development of this cognitive architecture (Helie & Sun, 2010b; Sun, 2002; Sun et al., 2005). Second, the interaction between explicit and implicit processing in CLARION has already been shown to yield synergistic learning and performance (Helie & Sun, 2010b; Sun, 2002; Sun et al., 2001, 2005). Third, the interaction between explicit and implicit processing in the NACS has been shown to be essential to model similarity effects in human reasoning (Sun & Zhang, 2006) and decision-making (Sun & Helie, 2013).
Fourth, the main focus of the CLARION cognitive architecture is the selection and inclusion of a minimum set of mechanisms, structured in a parsimonious but effective way, to account for a maximum set of psychological data and phenomena. This leads to the present focus on general structure, instead of parameter tweaking. For all these reasons, we chose to explore the interaction of explicit and implicit declarative knowledge in memory and reasoning. Ample justification for architectural choices in the CLARION cognitive architecture can be found in Sun (2002, 2012). A more recent detailed comparison of CLARION with alternative cognitive architecture models of memory can be found in Helie and Sun (2014). The detailed comparison in Helie and Sun suggest that CLARION is currently the only ‘psychologically-oriented’ cognitive architecture that focuses on the interaction of explicit and implicit memory, which is essential to the present objective.

The remainder of this article is organized as follows. First, a general description of CLARION is presented, including a brief summary of past simulation results. Second, the NACS and the working memory of CLARION are introduced, along with a summary presentation of other CLARION subsystems. Third, a number of psychological phenomena are reviewed (i.e., human memory, deductive reasoning, inductive reasoning, and heuristic reasoning), along with explanations of the phenomena using the NACS of CLARION. This presentation is followed by a general discussion that summarizes the major findings, and compares the CLARION account with other integrative accounts of human memory.
The CLARION cognitive architecture

Overview

CLARION is a cognitive architecture that is, in part, based on two basic assumptions: representational differences and learning differences of two different types of knowledge: implicit versus explicit (Helie & Sun, 2010b; Sun, 2002; Sun et al., 2001, 2005). These two types of knowledge differ in terms of conscious accessibility and attentional requirement. The top level of CLARION (as in Figure 1) contains explicit knowledge (easily accessible, but requiring more attentional resources) whereas the bottom level contains implicit knowledge (harder to access, more automatic). The result of top- and bottom-level processing is integrated in order to capture the interaction of implicit and explicit processing in humans (see also Helie & Sun, 2010b).

CLARION includes two major subsystems (see Figure 1): the Action-Centered Subsystem and the Non-Action-Centered Subsystem (Sun, 2002). The Action-Centered Subsystem (with implicit and explicit levels) contains procedural knowledge concerning actions and procedures (i.e., it serves as procedural memory), while the Non-Action-Centered Subsystem (with both levels) contains declarative knowledge (i.e., it serves as declarative memory, both semantic and episodic; Sun et al., 2005). The Non-Action-Centered Subsystem also carries out various types of reasoning (Helie & Sun, 2010b; Sun & Helie, 2013; Sun & Zhang, 2006).

The second assumption in CLARION concerns the existence of different learning processes in the top and bottom levels, respectively (Sun et al., 2001, 2005). In the bottom level, implicit associations are learned through gradual trial-and-error learning. In
contrast, learning of explicit rules in the top level is often “one-shot” and represents the abrupt availability of explicit knowledge following “explicitation” of implicit knowledge or new acquisition of linguistic (or otherwise explicit) information.

The Action-Centered Subsystem of CLARION has been used to model navigation in mazes and mine fields (Sun et al., 2001; Sun & Peterson, 1998) and a number of sequence learning experiments (Sun et al., 2005). The Non-Action-Centered Subsystem of CLARION has been used to simulate simple forms of reasoning (e.g., Sun & Zhang, 2006) and decision-making (Sun & Helie, 2013). Rule-based reasoning is supported by top-level processes (explicit) whereas similarity-based reasoning is supported by the synergistic interaction of bottom-level processes (implicit) with top-level processes (explicit). More recently, the Non-Action-Centered Subsystem played an important role in modeling the role of creativity in problem solving (Hélie & Sun, 2010b). The following subsections present a more complete description of the subsystems. For clarity, the details of the Non-Action-Centered Subsystem and working memory (which are the focus of the present paper) are included, but the Action-Centered Subsystem is only briefly sketched.

A sketch of the Action-Centered Subsystem

In CLARION, the Action-Centered Subsystem (ACS) is the main subsystem (Sun et al., 2001, 2005). In addition to being the procedural memory, the ACS captures some executive functions (i.e., the control of some of the other subsystems). As such, the ACS receives all the inputs from the environment, and provides action recommendations. The description of the ACS here is conceptual only, because technical details are not needed
in this paper. Readers interested in the technical aspects of the ACS are referred to Sun (2002) and Sun et al. (2005).

In the top level of the ACS, explicit knowledge is represented using condition and action chunk nodes. Condition chunks can be activated by the environment (e.g., a stimulus) or other CLARION subsystems (e.g., working memory). Action chunks can represent motor programs (i.e., a response) or queries/commands to other CLARION subsystems. In particular, an action of the ACS can query the Non-Action-Centered Subsystem for a reasoning cycle (as detailed below). In this case, the Non-Action-Centered Subsystem can return one or several chunks resulting from the reasoning cycle, which can be used in the ACS as action recommendations or as conditions for computation at a future time step.

The bottom level of the ACS uses feature-based representations to capture implicit procedural knowledge. Each top-level chunk node is represented by a set of feature (or microfeature) nodes in the bottom level (i.e., a distributed representation). The features (in the bottom level) are connected to the chunk nodes (in the top level) so that they are usually activated together through bottom-up activation (when the features are activated first) or top-down activation (when the chunk nodes are activated first). The features of the condition and action chunk nodes are connected in the bottom level using nonlinear connectionist networks. Each network can be thought of as a highly efficient behavior routine (once properly trained) that can be used to accomplish a particular type of task. Training of the bottom-level networks is iterative and done using backpropagation (Haykin, 2009) implementing Q-learning (Sun et al., 2001; Watkins, 1989).
The Non-Action-Centered Subsystem

This section presents a detailed conceptual description of the core processes of the Non-Action-Centered Subsystem (NACS) of CLARION. In CLARION, the NACS captures declarative (both semantic and episodic) memory (Sun, 2002). The inputs and outputs of this subsystem usually come from the CLARION ACS. Some technical/mathematical details are presented in Appendix B, but readers interested in a more complete technical presentation are referred to Sun and Helie (2013).

Top level

In the top level of the NACS, explicit knowledge is represented by chunk nodes (as in the ACS top-level). However, unlike in the ACS, NACS chunks are not divided into condition and action chunks: all chunks represent concepts that the agent is already familiar with that can be used as a condition or a conclusion in rule application. Each chunk node can be activated by: (1) an ACS query, (2) its association with another chunk node (via an associative rule), or (3) its similarity to another chunk (via a similarity measure). When a NACS chunk node is activated by an ACS query (rule), its activation is generally set to full activation (because rules transmit information exactly in CLARION).

In addition to receiving activation from the ACS, NACS chunk nodes can be linked together to represent ‘associative’ rules (somewhat similar to a semantic network). In the simplest case, by representing the associative rules using connection weights, the top level of the NACS can be represented by a linear connectionist network (Haykin, 2009). Hence, a concept in CLARION is typically accompanied by a numerical value representing its level of support (the activation in the connectionist network), with full
activation representing high confidence that something is true while zero activation representing the lack of support for a fact or an association.\textsuperscript{3}

NACS chunks also share a relationship through similarity, which enables reasoning by similarity. In CLARION, the activation of a chunk node caused by its similarity to other chunk nodes is termed \textit{similarity-based reasoning}. Crucially, the similarity measure is defined through the interaction of the top and bottom levels of the NACS, and CLARION is currently the only cognitive architecture to emphasize this kind of interaction (Helie & Sun, 2014). Overall, the activation of each chunk node in the top level of the NACS is equal to the maximum activation it receives from the three previously mentioned sources, i.e., ACS queries, rule-based reasoning, or similarity-based reasoning.

Regardless of the activation source, chunks that are inferred (activated) in the NACS are sent to the ACS for consideration in action decision-making. Every chunk that is sent back to the ACS is accompanied by an internal confidence level (Helie & Sun, 2010). The internal confidence level is calculated by transforming the chunk node activations into retrieval probabilities using a Boltzmann distribution (Sun & Helie, 2013). The internal confidence level is used to assess certainty in a result and for probability estimation. If only one chunk is to be returned to the ACS, a chunk is stochastically selected using the Boltzmann probability.

\textit{Bottom level}

As in the ACS, the bottom level of the NACS (i.e., the \textit{Associative Memory Networks}) uses feature-based representations to encode the top-level chunk nodes with distributed representations (Helie & Sun, 2010b). The features are connected to the top-
level chunk nodes so that, when a chunk node is activated, its corresponding bottom-level feature-based representation (if exists) is also activated and vice versa. Alternatively, any bottom-level feature in the NACS can be directly activated by an ACS query.

The connections between top-level chunk nodes and their corresponding bottom-level feature-based representations allow for a natural computation of similarity. By default, the similarity between two chunk nodes is calculated by counting the number of bottom-level features shared by the chunks (i.e., the feature overlap in the bottom level), divided by a simple, slightly super linear, monotonic function of the total number of features of the inferred chunk. However, feature weights can be learned and varied to account for prior knowledge or context (e.g., the context emphasizes a particular feature or past experience suggests that a particular feature is more useful). Thus, similarity-based reasoning in CLARION is naturally accomplished using (1) top-down activation by chunk nodes of their corresponding bottom-level feature-based representations, (2) feature overlap between any two chunks in the bottom level, and (3) bottom-up activation of the top-level chunk nodes. This kind of similarity calculation is naturally accomplished in a two-level cognitive architecture and represents a form of synergy between the explicit and implicit modules.

According to the reverse containment principle (Quillian, 1968; Sun, 1994), if chunk $i$ represents a category that is a superset of the category represented by chunk $j$, all the (bottom-level) features of chunk $i$ are included in the (bottom-level) feature-based description of chunk $j$. For instance, chunk $i$ could represent the category ‘bird’ while chunk $j$ could represent the category ‘sparrow’. In this case, the feature-based description of ‘sparrow’ would include the feature-based description of ‘bird’, plus additional
features unique to sparrows. One important form of similarity-based reasoning is inheritance-based inference. In CLARION, this is accomplished based on the reverse containment principle. The reverse containment principle allows for the emulation of a hierarchy of concepts (in the ideal case; Sun, 1994) without representing the actual hierarchy (see the Deductive reasoning subsection below).

In addition to enabling the calculation of the similarity measure through synergistic interaction with the top level, the bottom level of the NACS allows for implicit non-action-centered processing. Implicit processing is accomplished by using a number of networks connecting the distributed feature-based representations (as in the bottom level of the ACS). Some bottom-level networks are auto-associative, thus allowing for the retrieval of chunks using partial feature match (among other functionalities). These auto-associative networks can be implemented using Hopfield-type networks (Anderson, Silverstein, Ritz, & Jones, 1977; Hopfield, 1982). In particular, the Nonlinear Dynamic Recurrent Associative Memory (NDRAM; Chartier & Proulx, 2005) has been used in the NACS of CLARION (see, e.g., Hélie & Sun, 2010b). The NDRAM is a synchronous Hopfield-type model that allows for learning continuous-valued patterns and minimizes the number of spurious memories (Chartier & Proulx, 2005). Learning is online in NDRAM, i.e., it occurs each time a stimulus is presented to the model.

The weight space of Hopfield-type neural networks can be interpreted as a phase space in a dynamical system (e.g., Anderson et al., 1977; Haykin, 2009). The learned patterns are represented by fixed-point attractors, and all the trajectories (e.g., initiated by stimuli or activation patterns) in NDRAM have been shown to converge to a fixed-point
An integrative account of memory and reasoning

attractor (Chartier, Renaud, & Boukadoum, 2008; Hélie, 2008). This interpretation of Hopfield-type neural networks can be used to draw a useful analogy between the phase space in the system and a conceptual space (Shepard, 1987; Tenenbaum & Griffiths, 2001): Items in conceptual spaces are represented by fixed-point attractors in the phase space, and new concepts (stimuli) can be learned (i.e., added to the conceptual space, which corresponds to creating a new fixed-point attractor in the phase space) or existing concepts can be retrieved from the conceptual space (i.e., through convergence to an existing fixed-point attractor in the phase space). This interpretation of NACS bottom-level network is useful in accounting for many memory phenomena.

Working memory

Working Memory (WM) is a secondary subsystem in CLARION controlled by the ACS (not represented in Figure 1). One main role of WM is to temporarily hold items transferred between the ACS and the NACS. WM is “deliberate”, in that cognitive effort must be made to maintain the information in WM. Like the ACS and the NACS, WM is composed of a top and a bottom level. In the top level, memory items are represented by chunk nodes. However, top-level chunk nodes are not organized as a network: They fill slots. At every time step, each item in working memory has a certain probability of being encoded in the NACS, and one item is chosen randomly to be refreshed (which increases its base-level activation, keeping it accessible in WM, as explained below). More details on this process are provided in Appendix A.

Each WM chunk node has a base-level that decays exponentially and corresponds to the odds of needing the chunk based on past experiences (Anderson, 1990). Each new item entering WM may displace an existing chunk. The probability of being bumped is
an inverse function of the base-level activations of an existing item (see Appendix A). This is because chunks that are being used (attended to) are more likely to be needed in the future and should not be forgotten (Anderson, 1990).

Similar to the ACS and the NACS, top-level chunk nodes are also represented by their features in the bottom level. However, similar to the top-level chunk nodes in WM, the feature nodes in the bottom level of WM are not organized in a network. The main role of the bottom-level distributed representations in WM is to activate the bottom level of the ACS. Because this is not used here, we will not get into details (but see Sun, 2002).

**Accounting for a range of psychological phenomena**

In this section, the CLARION NACS is used to explain some general psychological phenomena relying on declarative memory. This is done in four different areas: human memory, deductive reasoning, inductive reasoning, and heuristic reasoning. Several psychological phenomena are explained and captured in each subsection concerned with a particular psychological area. It should be noted that the goal here is not to simulate particular data sets; instead, it is to show that the CLARION NACS provides a natural (mostly architectural) explanation for these psychological phenomena, and their manifestations, using either generic simulations of processes or mathematical proofs, whichever is appropriate. Hence, most of the following explanations are parameter-free (exceptions will be noted).

As a reminder, throughout this section, chunk nodes in the top level represent concepts that are familiar (already learned, e.g., a fire truck), feature nodes in the bottom level represent features (or microfeatures) of the chunk nodes (e.g., the color red) and, together, a chunk node and its feature nodes form a chunk. The connections in the top
level represent (explicit) rules while the connections in the bottom level represent (implicit) associations. The connections between a chunk node (in the top level) and its feature nodes (in the bottom level) are used for similarity-based reasoning, and each bottom-level feature can have a different weight suggested by past experience or context. The activation of each chunk node is processed through a Boltzmann distribution to represent its retrieval probability (Sun & Helie, 2013). The retrieval probability is called the internal confidence level and is used to represent uncertainty.

**Human memory**

Before showing that the declarative memory in CLARION is sufficiently powerful for reasoning, it will be shown that the NACS can account for some key memory phenomena. In this subsection, we show how CLARION can be used to explain frequency effects, priming, cued recall, the list length/list strength effect, and the serial position curve. These phenomena are widely accepted in psychology and have been reproduced a number of times.

**Frequency effect**

Kintsch (1970) has shown that higher frequency words are better recalled in free recall tests. In CLARION, new words are learned in both the top and bottom levels. Learning in the top level is “one-shot” (Sun, 2002; Sun et al., 2001). However, the bottom level of the NACS involves an attractor neural network, which has strong pattern completion capabilities (as shown by Chartier & Boukadoum, 2006; Chartier & Proulx, 2005). Relearning a word that is already known increases the number of times that the attractor neural network is trained with this particular word.
The density estimation capability of the attractor neural network in the bottom level of the CLARION NACS allows for a natural explanation of this effect. Higher frequency words, coded as attractors in the bottom-level network, have larger attractor fields (Anderson et al., 1977). When retrieval is not initiated by a clue, memory search is initiated by random activation in the bottom level of the NACS (Helie & Sun, 2010b). Helie et al. (2006) trained the NDRAM on three different categories with different base rates corresponding to a uniform distribution, a multi-normal distribution, a step-distribution, and an exponential distribution. After training was completed, a series of random vectors were presented to the network and their converging attractor (category) was recorded. Figure 2 shows the proportion of random activation converging in each category/attractor for each training condition as well as the true base rate. As can be seen, the model did learn all four environmental distributions, and this observation was confirmed by statistical testing. Thus, words that were encoded more frequently (e.g., by being more likely in the environment) were better recalled. This simulation shows that the CLARION NACS can account well for the frequency effect.

**Proming**

**Positive priming.** In lexical decision tasks, human participants are usually faster at identifying sequences of related concepts, e.g., the word “butter” when it follows the word “bread” (Meyer & Schvaneveldt, 1971). In CLARION, the explanation of priming is similar to Masson (1995). The analogy between the phase space of the attractor neural network (in the bottom level of the NACS) and the conceptual space (Shepard, 1987; Tenenbaum & Griffiths, 2001) is important: the phase space represents a semantic space,
and attractors in the phase space represent words in the semantic space. In a priming experiment, the presentation of the first word (e.g., “bread”) produces the convergence of the retrieval trajectory into the “bread” attractor. When the second word is presented (e.g., “butter”), the new stimulus perturbs the stable state and initiates a new retrieval trajectory. However, the starting point of the new trajectory is near the final position from the preceding trial (the “bread” attractor). If the new starting point is closer to the attractor representing the next item (i.e., “butter”) than a randomly selected starting point (the starting point when there is no priming), convergence takes fewer iterations, which captures positive priming.

The above-mentioned process may be illustrated by a simulation example. Figure 3 presents the number of iterations required for the convergence of 101 stimuli (representing first words) according to their correlation with (distance/semantic relatedness to) the target attractor (representing the second word) (the simulation details are presented in Appendix A). As shown, higher absolute correlations yielded faster convergence. The filled circle represents the average number of iterations required for the convergence of random activation. As can be seen, this average is 28, which equals the number of iterations needed for convergence when a vector has a correlation of ±0.16 with the memorized patterns. When the correlation between the starting state (representing the first word) and the target state (representing the second word) is larger than 0.16, positive priming is present.

Negative priming. Negative priming refers to longer reaction times when the first word is unrelated to the second (e.g., Tipper, 1985). In CLARION, the explanation of
negative priming is very similar to the explanation of positive priming (see Figure 3 above). When the second word is unrelated to the first word, the attractors representing these two words are usually far apart (because the phase space represents a semantic space and distance represents semantic relatedness). Therefore, the starting point of the trajectory representing the second word can be far away (i.e., further than a randomly generated starting point on average) from the target attractor (i.e., the attractor representing the second word). This leads to a natural explanation for negative priming.

In Figure 3, the filled circle represents the average number of iterations required for the convergence of random activation. As can be seen, this average is 28, which equals the number of iterations needed for convergence when a vector has a correlation of \( \pm0.16 \) with the memorized patterns. When the absolute correlation between the starting state (representing the first word) and the target state (representing the second word) is smaller than 0.16, negative priming is present.\(^4\)

**Cued recall**

The efficiency of recall generally increases with the number of cues (e.g., Anderson, 1983). In CLARION, the activation in the top level of the NACS is a monotonic non-decreasing function of the number of cues (because the activation and the connection weights are always positive). Hence, adding cues generally result in more activation in the top level. In the bottom level of the NACS, the attractor neural networks (which are instances of the Cohen-Grossberg additive model; Cohen & Grossberg, 1983) always converge toward an attractor (Grossberg, 1988; Hélie, 2008). As a result, the activation of the corresponding top-level chunk nodes always increases following bottom-level processing and bottom-up activation (if it converges to the correct attractor,
which is more likely if there are more cues). In addition, more cues add constraints to both top- and bottom-level processing, which increases the likelihood that explicit and implicit processing would synergistically suggest the same response (as in, e.g., Helie & Sun, 2010b). The resulting higher activation leads to a higher internal confidence level accompanying the chunks retrieved with more cues. This higher internal confidence level is directly translated into a higher retrieval probability.

Helie and Sun (2010a) simulated a cued recall experiment were a series of cues were increasingly associated with a target word to be retrieved (Bower, Regehr, Balthazard, & Parker, 1990). Figure 4 shows the CLARION NACS activation as a function of number of cues. As can be seen, integrated activation monotonically increased with the number of cues. This increased the internal confidence level and retrieval probability. This simulation shows that the CLARION NACS can account for the effect of cues on recall probability.

Insert Figure 4 about here

List length / list strength effects

The list length effect refers to a decline in performance in both recognition and free recall when the list grows longer (Roberts, 1972). The list strength effect refers to better performance for items that were presented more frequently or for a longer duration in the memorization phase (Roberts, 1972). The bottom level of the NACS in CLARION consists of an attractor neural network, an earlier version of which was shown to reproduce both effects (Proulx & Chartier, 1998). In short, the list length effect can be explained by an increase in the number of spurious attractors in an attractor neural network when the number of stimuli becomes large compared with the number of nodes
in the network, which decreases recall performance. This phenomenon is illustrated using random activation vectors in Figure 5 for a network composed of 50 nodes (the simulation details are presented in Appendix A).

In CLARION, the list strength effect is explained by frequency effects, as detailed in the explanation of the frequency effects earlier (i.e., more frequent stimuli are represented by attractors with larger attractor fields; see Figure 2). This is because items that are presented for a longer time period are used multiple times during this period by the NACS for encoding, which emulates frequency effects.

*Serial position curve*

The serial position curve in free recall includes mainly two effects (Murdock, 1962). First, items presented at the beginning of the list are more likely to be recalled (i.e., the *primacy effect*). Second, items that were presented at the end of the list are also more likely to be recalled (i.e., the *recency effect*). Typically, the recency effect is stronger than the primacy effect.

In CLARION, items are stored in working memory, which requires (covert) rehearsal. In every trial, each item in working memory has a probability of being transferred into long-term declarative memory. Also, an item is selected in each trial to be refreshed. The working memory size is limited and, when a new item enters an already full working memory, one of the old items is replaced. In this framework, the primacy effect can be explained by the fact that the first few items (up until the working memory capacity) are more likely to stay in the working memory longer. Hence, they have more opportunities of being encoded into long-term declarative memory (the NACS). The
recency effect simply results from the last few items being more likely to be available in working memory with higher base-level activations when the recall phase of the experiment starts (for details, see Eq. A1 in Appendix A).

This process has been simulated based on CLARION for various list lengths and the results are shown in Figure 6a (the simulation details are presented in Appendix A). As can be seen, the simulated data reproduced this classical human result. Also, it should be noted that the above explanation of the phenomenon is consistent with the classical explanation of the serial position curve effects (e.g., Ashcraft, 1989).

**Inhibition of the recency effect.** Glanzer and Cunitz (1966) conducted a free recall experiment using 15 words. However, a delay was inserted between the study phase and the recall phase to inhibit the effect of working memory. The results showed that the recency effect disappeared while leaving the primacy effect untouched. This result is often cited as evidence for separate long-term memory and working memory systems (Ashcraft, 1989).

The explanation by CLARION is that the base-level activations of the items in working memory decrease during the delay, and the last few items in working memory are replaced by new material (unrelated to the experiment) and are thus unavailable for recall. The simulation of this experiment with CLARION was similar to the preceding simulation used to generate the serial position curves (see Appendix A). However, the base-level activations of the working memory items decay exponentially during the interval between the two phases (see Eq. A1). Therefore, the content of working memory
was not recalled automatically. The results are shown in Figure 6b. As can be seen, the primacy effect is reproduced, but the recency effect is missing (as in the human data).

**Inhibition of the primacy effect.** Baddeley and Hitch (1974) conducted a free recall experiment with 16 words. However, there was a “pre-loading” of working memory: three digits were presented to the participants prior to the study phase. These three digits had to be recalled at the end of the study phase. The results showed that the primacy effect in free recall was inhibited, while leaving the recency effect untouched. Baddeley and Hitch interpreted this result as evidence that longer rehearsal of the first few items was responsible for the primacy effect in free recall.

The explanation by CLARION is similar to Baddeley and Hitch: Distracting items pre-loaded in working memory at the beginning of the experiment have higher base-level activations than random (e.g., unrelated to the experiment) thoughts. Thus, the first few items of the list had similar base-level activations as the pre-loaded items and were bumped at the same rate as all the other items in the list. This experiment was simulated in CLARION by pre-loading the working memory with three distracter items. All the remaining simulation details were the same as in the above simulations (Appendix A). The results are shown in Figure 6c. As can be seen, the primacy effect disappeared without affecting the recency effect. This captures the human data and supports the adequacy of CLARION as an architectural model of free recall.

**Deductive reasoning**

Deductive reasoning is an important cognitive capacity that relies on the application of general rules to particular cases. However, empirical research has shown that human reasoning is often a mixture of rule-based and similarity-based processing
An integrative account of memory and reasoning (for reviews and arguments, see Sun, 1994). Due to the involvement of similarity-based reasoning, under many circumstances, deductive reasoning is uncertain, not guaranteed to be correct, and relies on the interaction of explicit and implicit declarative knowledge (e.g., retrieval, activation propagation, etc.). The CLARION NACS can account for many types of uncertain deductive reasoning (as reviewed in, e.g., Sun, 1994). This subsection presents a conceptual description of the cases and their explanations in CLARION, while mathematical derivations are included in Appendix B.

Uncertain information

When information regarding the premise of a rule is not known with absolute certainty, a conclusion can still be reached albeit with uncertainty. For example, one might have the following rule related to American football: “If Tom Brady plays, the New England Patriots is going to win the game”. Even if the playing status of Tom Brady is somewhat uncertain, one can infer that because he is likely to play, the New England Patriots is likely to win.

In CLARION, this case can be captured by rule-based reasoning in the top level of the NACS. Uncertainty of information is captured by partial activation (as opposed to full activation) of the premise chunk. In CLARION, if the premise chunk is partially activated, the conclusion chunk is also partially activated, proportional to the activation of the premise chunk.

Incomplete information

When a rule has more than one premise, a conclusion can be reached, with some uncertainty, even if only some of the premises are present (Sun, 1994). Keeping with the
preceding example, one could have a rule: “If Tom Brady and Rob Gronkowski play, the New England Patriots is going to win the game”. If it is known that Tom Brady plays but the playing status of Rob Gronkowski is unknown, the conclusion that the New England Patriots is going to win can still be made with some uncertainty.

In CLARION, this case is accounted for by rule-based reasoning in the top level of the NACS. Each premise in a rule has a weight, and the weights of the premises add to no more than one. When not all the premise chunks are activated, the conclusion chunk is partially activated, proportional to the number of activated premise chunks.

*Similarity matching*

When no known rule allows for answering a question directly, one can make an inference based on the similarity to known facts (Sun, 1994). For example, when asked: “Is the Chaco a cattle country?”, one answered: “It is like western Texas, so in some sense I guess it’s a cattle country” (Sun, 1994). In this example, the answer was based on similarity matching.

In CLARION, this case is explained by similarity-based reasoning (via the interaction of the top and bottom levels of the NACS). When two chunks share a subset of features, the activation of one chunk is automatically (and partially) transmitted to the other. Here, the activation of a chunk node representing “Chaco” (partially) activates the chunk node representing “western Texas” (by bottom-up activation flow, because of feature overlap), which in turn (partially) activates all the rules associated with western Texas (e.g., “western Texas → cattle country”). Hence, activating the chunk node representing “Chaco” automatically activates (partially) the chunk node representing “cattle country”, proportional to the similarity between Chaco and western Texas.
Superclass to subclass inheritance

In superclass to subclass inheritance, one uses properties from the superclass to answer a question about a subclass (Collins & Quillian 1969; Sun, 1994). For example, when asked if Snoopy the dog has four legs, one may respond “yes” without knowing anything about Snoopy, because the generic (prototypical) dog has four legs (e.g., this may be accomplished through a rule: “dog → four legs”).

In CLARION, superclass to subclass inheritance is a special case of similarity-based reasoning, because the bottom-level feature-based representation in CLARION can capture a categorical hierarchy without explicitly representing an actual hierarchy. Chunks representing subcategories (e.g., Snoopy) have all the features of the chunk representing their superclass (e.g., dogs), plus additional features making them unique (i.e., the reverse containment principle). Hence, superclass to subclass inheritance is explained in CLARION by similarity-based reasoning as applied to superclass-subclass relations.6

Subclass to superclass “inheritance”

Here, properties of a superclass are inferred from knowledge about the properties of particular subclasses. Subclass to superclass “inheritance” may be viewed as a case of induction. More complex cases of induction (and exceptions) will be presented later in the Inductive reasoning subsection.

CLARION treats subclass to superclass “inheritance” as a case of similarity matching. Note that the superclass does not necessarily have the property being inferred using knowledge of the subclass chunk because subcategories have more features and properties than supercategories. Hence, inference using subclass to superclass inheritance
is weaker than inference using superclass to subclass inheritance: It produces less activation, because there is no convergence toward full activation.

Cancellation of superclass to subclass inheritance

As explained above, one can infer that Snoopy the dog has four legs because the prototypical (generic) dog has four legs (superclass to subclass inheritance). However, an exception rule might also be present, stating that Snoopy the dog does not have four legs, because he was in an accident (cancellation of superclass to subclass inheritance).

According to CLARION, superclass to subclass inheritance is the most reliable form of similarity-based reasoning. However, it is still not as reliable as rule-based reasoning, because full activation can never be reached (for details, see Eq. B3 in Appendix B). In contrast, rule-based reasoning can fully activate a chunk. Hence, rules can be used to reject conclusions reached by similarity-based reasoning.

Cancellation of subclass to superclass “inheritance”

As explained earlier, subclass to superclass “inheritance” is a weaker form of similarity-based reasoning than superclass to subclass inheritance. Hence, in CLARION, rules can be used to reject a conclusion reached by subclass to superclass “inheritance” (the same way as the cancellation of superclass to subclass inheritance).

Mixed rules and similarities

In addition to the cases described above, rule-based and similarity-based reasoning can be chained in many other ways. For instance, a chunk can be activated by similarity matching, and the newly inferred chunk can fire a rule. The opposite can also happen (i.e., inferring a chunk using rule-based reasoning and activating another chunk
by similarity to the conclusion chunk). There are many such cases that can be explained by chaining the explanations of the preceding seven cases. In Appendix B, predictions for six examples of mixtures of rule-based and similarity-based reasoning are derived.

**Inductive reasoning**

Induction is an essential process that generates general conclusions from the observation of instances (Heit, 2000). While this form of reasoning is known to be error-prone (i.e., not guaranteed to be correct), it is a crucial part of cognition in that it allows humans to function in their environment by making predictions and adjusting their actions (e.g., as in science; Jain, Osherson, Royer, & Sharma, 1999). This form of reasoning relies on retrieval from declarative memory that is mostly similarity-based. Here, we examine a few essential phenomena observed with either single or multiple premises, along with their explanations using the NACS of CLARION. The different effects of the properties of a concept/category are also explored. In this subsection, only parameter was varied to account for the phenomena, i.e., the relative feature weight in similarity-based reasoning. As a reminder, the relative weights of features in CLARION can be emphasized by the context (e.g., compare concepts $x$ and $y$ according to their color) or learned through past experience (e.g., learning what feature is useful in a specific task using reinforcement learning).

**Similarity between the premise and the conclusion**

Human inductive reasoning is affected by the similarity between the premise and the conclusion (Osherson et al., 1990; Rips, 1975). For example, participants make stronger inference from rabbits to dogs than from rabbits to bears (Heit, 2000). In
CLARION, the similarity between two chunks is a function of the number of overlapping features in the bottom level of the NACS. Assuming that the strength (activation) of the premise is unvarying, the strength of the conclusion chunk is a positive, increasing function of the number of overlapping features between the premise and the conclusion (i.e., a positive, increasing function of the similarity between the two chunks). Therefore, CLARION captures the basic similarity effect in inductive reasoning.

Unequal properties

In human induction, some properties are more projectable than others (Heit, 2000). For instance, participants in Nisbett et al.’s (1983) experiment were unwilling to generalize that all members of a tribe are obese based on one observation. However, they were more inclined to generalize skin color to all members of the tribe based on one observation. This difference can be explained by prior knowledge, which provides different explanations for obesity and skin color. Hence, not all properties are treated equally for generalization due to such prior knowledge.

Although in CLARION all the bottom-level features of a chunk are weighed equally in similarity calculation by default, this need not be the case. As discussed earlier, different features may be given different weights in similarity calculation to represent prior knowledge or context (e.g., knowledge about the variance in attribute values; Rips, 1989). Hence, if prior knowledge or the context emphasizes a particular set of features, these features may be given more weights and the strength of the conclusion chunk will be increased accordingly. In the preceding example, obesity may have a weight of, for example, 0.01 while skin color may have a weight of, for example, 1. Both of these two features are present in the feature-based representation of the chunks. However, the latter
feature would have much stronger effects on similarity matching. In this case, unequal weighing of the features may result from previous experience showing that body weight cannot be generalized to entire populations (hence the smaller feature weight) whereas skin color generally can (hence the higher feature weight). This accounts for the inequality of features on inductive reasoning.

*Functional attributes*

Although we have already discussed that similarity increases induction strength, this is not always the case (Heit & Rubinstein, 1994). Compare the following two cases:

1. Chickens have a liver with two chambers
2. Hawks have a liver with two chambers

is stronger than

1. Tigers have a liver with two chambers
2. Hawks have a liver with two chambers

This is because chickens and hawks are more similar than tigers and hawks. This was described earlier. However, consider the following arguments:

1. Chickens prefer to feed at night
2. Hawks prefer to feed at night

and

1. Tigers prefer to feed at night
2. Hawks prefer to feed at night

In this case, the second argument is considered stronger by participants (Heit & Rubinstein, 1994). This is explained by feeding habits being more similar between hawks
and tigers (i.e., they are both predators) than between hawks and chickens. This phenomenon is known as “exception to similarity due to functional role” (Heit, 2000).

Although a review of the literature on categorization suggests that it is unclear what constitutes a feature exactly (e.g., Schyns, Goldstone, & Thibault, 1998), functional attributes, such as feeding habits, are incorporated as features within CLARION. These functional features (Schyns & Rodet, 1997) are given large weights when they are emphasized by the context (e.g., through meta-cognitive regulation within CLARION; see Sun, 2007 for details). Hence, functional attributes are part of the similarity computation in CLARION and affect the strength of the conclusion reached (without any additional assumptions or mechanisms). An exception to the similarity effect can be observed when the weight of a feature (here feeding habit) is larger than the difference between the weighted feature overlap of the two induction cases. This case is analyzed mathematically in Appendix C.

Multiple premises

The number of premises affects the strength of the conclusion in human induction experiments (Nisbett et al., 1983; Osherson et al., 1990). For example, the argument:

Hawks have sesamoid bones.

Sparrows have sesamoid bones.

Eagles have sesamoid bones.

All birds have sesamoid bones.

is stronger than the argument:

Sparrows have sesamoid bones.
Eagles have sesamoid bones.

All birds have sesamoid bones.

In the NACS of CLARION, the activation of chunks is monotonic and non-decreasing (for details, see Eqs. B1-B3 in Appendix B). Adding more premises can only increase the strength of the conclusion (computationally, this is similar to adding cues in memory recall; see Figure 4). Hence, adding premises increases or at least maintains the strength of the conclusion.

**Heuristic reasoning**

In real-life situations, humans rarely have the time or desire for optimal reasoning strategies. Still, decisions must be made and conclusions must be reached. As such, heuristic reasoning (i.e., using simple, sometimes suboptimal, but generally useful strategies) is ubiquitous (Gigerenzer & Goldstein, 1996). Below, four classical phenomena and heuristics studied in uncertain reasoning along with the explanations provided by CLARION are presented. Only one parameter was varied in this subsection, i.e., the relative feature weight in similarity-based reasoning (same as for Inductive reasoning). As a reminder, the relative weights of features in CLARION can be emphasized by the context (e.g., compare concepts $x$ and $y$ according to their color) or learned through past experience (e.g., learning what feature is useful in a specific task using reinforcement learning).

**Representativeness heuristic**

The representativeness heuristic is the evaluation of how well a new situation represents (i.e., is similar to) a stored prototypical situation, and using this evaluation to
estimate the probability of the new situation (e.g., Tversky & Kahneman, 1974). In CLARION, each prototypical situation is conceptually represented by a separate top-level chunk node in the NACS. In addition, each top-level chunk node is represented by a set of corresponding features in the bottom level. When a new situation is encountered, a chunk node representing this new situation may not be present in the top level (because the situation/concept hasn’t been learned yet), but the corresponding features in the bottom level (representing the situation/concept) can still be activated by the stimulus. These features are used to activate existing top-level chunk nodes related in some way to the new situation (by similarity-based reasoning). All the chunk nodes representing prototypical situations that are activated are sent back to the ACS along with their internal confidence levels (used for probability estimations). This similarity-based bottom-up activation within the NACS is responsible for the representativeness heuristic, as it generates similar (i.e., representative) instances as a basis for further reasoning. The representativeness heuristic has been used to account for several biases in human uncertain reasoning (for a review, see Tversky & Kahneman, 1974). Some of the most well known biases are described below.

**Base-rate neglect.** In estimating the probability that a fictional character ‘Steve’ is a librarian or a farmer, the total number of librarians and farmers should be considered (in a normative sense, as prescribed by Bayes’ theorem; Tversky & Kahneman, 1974). However, in many cases, human participants do not consider this base-rate information and only rely on the representativeness heuristic – i.e., is the description of Steve more representative of librarians or farmers? If the description is more representative of
An integrative account of memory and reasoning

librarians, the estimated probability that Steve is a librarian is higher than the estimated probability that Steve is a farmer (notwithstanding the actual probabilities).

In CLARION, the mechanism used to account for the representativeness heuristic accounts for the base-rate neglect. In relation to the example above, ‘Steve’ may not be represented by a top-level chunk node in the NACS (because Steve is an unknown character), but its feature description can be used to activate a set of feature nodes in the bottom level. The feature similarity between Steve’s description and the chunks representing ‘farmer’ and ‘librarian’ activate these two chunk nodes bottom-up. These two chunks are sent back to the ACS along with their internal confidence levels. The probability judgments made by the ACS are based on the internal confidence levels, which in this case represent chunk node activation through similarity-based reasoning. Base rates are not considered in the estimations because the feature activation in the bottom level was not processed by the settling procedure of the attractor neural network. Hence, CLARION displays base-rate neglect when no elaborate implicit processing is performed.

Conjunction fallacy. Human participants were asked to estimate the probability that Linda is a bank teller, and the probability that she is a feminist bank teller (Garnham & Oakhill, 1994; Tversky & Kahneman, 1983). The results showed that participants estimated the former to be less probable than the later, even though the first category (i.e., bank tellers) includes the second (i.e., feminist bank tellers). According to Tversky and Kahneman (1983), this anomaly results from the application of the representativeness heuristic, because Linda’s description was more similar to (more representative of) a feminist bank teller than a (regular) bank teller.
The explanation in CLARION for this phenomenon is similar to that for the base rate neglect. Although Linda may not be represented by a chunk node in the top level of the NACS (again, because she is an unknown character), separate top-level chunk nodes exist representing the concepts ‘bank teller’, ‘feminist’, and ‘feminist bank teller’ (because these are all concepts familiar to the participants). The description of Linda given in the story activates a set of features in the bottom level, which compute the similarity between Linda and ‘bank teller’ and between Linda and ‘feminist bank teller’, and activate the corresponding two chunk nodes bottom-up. Both chunk nodes are sent back to the ACS along with their respective internal confidence level. If Linda is more similar to a ‘feminist bank teller’ than a ‘bank teller’, the ‘feminist bank teller’ chunk should have a higher internal confidence level and yield a higher probability judgment than the ‘bank teller’ chunk. This captures the conjunction fallacy.

Availability heuristic

It has been shown that in many human experiments, participants estimate the probability of an event based on the ease with which a similar event can be retrieved from memory (Garnham & Oakhill, 1994; Tversky & Kahneman, 1974). As explained in the Representativeness heuristic subsection above, stimuli that are more similar to existing chunks in the CLARION NACS yield higher activation for these existing chunk nodes. The chunk node activations are turned into a Boltzmann distribution and one of the chunks is stochastically selected to be sent back to the ACS (i.e., retrieved from memory). In general, higher chunk node activation makes a chunk easier to retrieve because it increases the probability that it is chosen in stochastic selection. Likewise, stimuli that are more frequent have larger attractor field are thus easier to retrieve. The
retrieval process is repeated multiple times, and the probability is estimated by the ACS based on the frequency of retrieval of chunks similar to the stimulus/event (Barsalou, 2003). This provides an intuitive explanation for the availability heuristic. The availability heuristic has been used in the literature to account for several biases (Tversky & Kahneman, 1974). Two of the most well known cases are presented below.

**Effectiveness of search set.** It is well known that cues can improve memory search (Ashcraft, 1989; Bower et al., 1990). Some cues are better than others. For instance, the last three cues in Bower and his colleagues experiment were more strongly associated with the target word than the other cues (hence the fast increase in Figure 4). According to Tversky and Kahneman (1974), participants tend to make probability estimates by trying to recall as many examples as possible from memory and choose the cue that led to more recalls. As an example, the first letter of a word is a much better cue to recall the word than its third letter. Hence, if trying to decide whether more words start with the letter ‘r’ or have an ‘r’ in the third position, participants try to recall words with ‘r’ in first or third positions, and tend to respond (incorrectly) that there are more words with an ‘r’ in the first position because they can retrieve more words with an ‘r’ in the first position (Tversky & Kahneman, 1974).

In CLARION, cued recall can be seen as a special case of similarity-based reasoning. Familiar memory items (including words) are represented by chunk nodes in the top level of the CLARION NACS. Hence, the activation of a chunk node is proportional to the number of its features that are activated in the bottom level. However, some features are more closely associated with the chunk node and constitute better cues for recall (i.e., they have higher cross-level weight values). Thus, when features with
higher weights are activated by the cue, the chunk node activation is higher. As in the preceding explanations, this activation is normalized using a Boltzmann distribution and a chunk is stochastically chosen to be sent back to the ACS (which is the recalled item). Chunk nodes that are more activated are more likely to be selected. Therefore, some cues are better than others. In this example, the feature corresponding to the first letter of a word has more weight than the feature corresponding to its third letter. This is likely caused by past experiences in which words are often organized in alphabetical order (based on the 1st letter) but are rarely (if ever) ordered according to their 3rd letter (suggesting that the 1st letter of a word is more important). Therefore, chunk nodes representing words starting with the letter ‘r’ are more activated than chunk nodes representing words with a ‘r’ in the third position and are more likely to be retrieved. The ACS counts the number of retrieved words and produces a response.

**Retrievability of instances.** In a now classic experiment, participants were read a list of man and woman names and asked to judge if there were more man names or more woman names in the list (Tversky & Kahneman, 1974). The results showed that when famous man names were included in the list, participants estimated that there were more man names in the list (notwithstanding the actual number). According to the availability heuristic, participants tried to recall names from the list and made an estimate based on the recalled names; if they could remember more man names, they assumed that there were more man names in the list. Famous names are easier to retrieve from memory (Ashcraft, 1989).

In CLARION, each time a name is seen/mentioned/used, it is (re)learned by the bottom-level network of the NACS. Famous names appear more often and thus are
relearned more often. As discussed earlier, the bottom level of the NACS is affected by training frequency (because each time a name is encountered, it is relearned and strengthened; see Figure 2). Hence, generally, attractors representing famous names have larger attractor fields. In contrast, regular names are likely to be used less often, and have smaller attractor fields. As discussed in the Human memory subsection, memory search that is not initiated by a cue (e.g., free recall) is initiated by random activation in the bottom level (Helie & Sun, 2010b), and attractors with larger attractor fields are more likely to be settled into from random starting positions (Helie et al., 2006). Thus, concepts represented by larger attractor fields (e.g., representing famous names) are more likely to be retrieved and sent back to the ACS (after stochastic selection as detailed before). Therefore, famous names from the list are more likely to be retrieved by the ACS and yield a higher estimate.

**General Discussions**

At the beginning of the present article, we argued that memory systems should be studied using both direct memory tasks (e.g., free recall) and indirect tasks that, while not focusing on memory, involve the use of memory (e.g., reasoning). This work is an attempt at using the declarative memory modules of CLARION (the Non-Action-Centered Subsystem or NACS) to account for psychological phenomena involving both the direct and indirect use of memory. The explanations, from psychological domains as diverse as memory, deductive reasoning, inductive reasoning, and heuristic reasoning, were based on architectural properties of the CLARION NACS and most did not require the adjustment of any numerical parameter (one numerical parameter was varied in a minority of cases).
It should be noted that the main goal of the present article was to highlight the importance of the synergistic interaction between implicit and explicit processes within declarative memory in supporting human reasoning. Thus, capturing the finer details of specific data sets was not our focus, and a detailed statistical analysis of matching between CLARION and empirical data was not essential. Instead, the focus of this research was on multiple tasks and/or multiple data sets in order to extract general principles or general phenomena that are applicable to a broad range of tasks. As a comparison, a number of other projects took a similar approach; for example, see McClelland, McNaughton and O’Reilly (1995) or Osherson et al. (1990). The reader may find the arguments for such generic cognitive architectures work in, for example, Newell (1990). Generic cognitive architectures have their inherent advantages and shortcomings.

Synergistic memory representations

One of the most basic assumptions in CLARION is the difference between explicit and implicit processes, and a dual representation framework has been used since its inception (e.g., Sun, 1994, 2002; Sun & Peterson, 1998, Sun et al., 2001, 2005). The inclusion of “redundant” representations in the top and bottom levels has often lead to synergistic learning and performance (e.g., Sun & Peterson, 1998; Sun et al., 2005), and is a core principle of CLARION. Knowledge is represented using features in the bottom level (in an implicit form) and chunk nodes in the top level (in an explicit form). Furthermore, the interaction and co-activation between the two levels allows for a natural computation of similarity between chunks/concepts (i.e., similarity-based processing), which has proven essential to most of the explanations provided herein. However, these are not the only instances of synergistic explanations provided by CLARION. For
example, the Explicit-Implicit Interaction theory of creative problem solving (Helie & Sun, 2010b) relies on the synergistic interaction of redundant explicit/implicit representations and used CLARION to specify a computational model and to simulate many datasets. It would be difficult for other cognitive architectures to provide a similar simple/intuitive computational model for this phenomenon. Other examples of synergy within the ACS can be found in Sun et al. (2001, 2005).

Comparison with other integrative memory models

The closest integrative memory model to CLARION is ACT-R. ACT-R is a production system aimed at explaining psychological processes (Anderson et al., 2004; Anderson & Lebiere, 1998). It is based on three key ideas (Taatgen & Anderson, 2008): (a) rational analysis, (b) the distinction between procedural and declarative memories and, (c) a modular structure linked with communication buffers. According to the rational analysis of cognition (Anderson, 1990), the cognitive architecture is optimally tuned to its environment (within its computational limits). Hence, the functioning of the architecture can be understood by investigating how optimal behavior in a particular environment would be implemented.

Similar to CLARION, ACT-R includes more than one long-term memory stores, distinguishing between procedural and declarative memories. In addition, ACT-R has a rudimentary representation of explicit and implicit memories: explicit memory is represented by symbolic structures (i.e., chunks and production rules) while implicit memory is represented by the numerical activation of the structures. In contrast, the distinction between explicit and implicit memories in CLARION is one of the main focuses of the architecture, and a more detailed representation of implicit knowledge has
allowed for a natural representation of similarity-based reasoning as well as natural simulations of many psychological data sets (e.g., Helie & Sun, 2010b; Sun et al., 2001, 2005). Yet, ACT-R memory structures have been adequate for simulating many datasets. Thus, it seems likely that ACT-R could provide alternative (albeit different) explanations for many of the phenomena included herein.

**Concluding remarks**

This paper has explored the possibility of providing conceptual/mathematical explanations for a wide range of psychological phenomena using the CLARION declarative memory modules (i.e., the non-action-centered subsystem or NACS). Here, we showed that the declarative memory modules in CLARION were sufficient to explain many psychological phenomena related to human memory, deductive reasoning, inductive reasoning, and heuristic reasoning using only architectural properties of the memory system. This intellectual exercise is important, because integrative psychological models generally use numerical approximation to fit psychological data, and it is unclear whether the explanation depends on the architecture of the model only or also on the numerical values assigned to the parameters. As such, this work is a step in showing that one can provide simple and direct explanations for diverse phenomena based on the structure of declarative memory modules.
References


Appendix A: Human memory

This appendix presents some technical details on the functioning of working memory in CLARION and the modeling details of the simulations in the Human memory subsection. The free parameters in all the simulations were set to their default values. Before describing the simulation details, some technical details of the top-level of working memory need to be introduced.

Working memory

Top-level chunk nodes in the CLARION working memory (WM) are not organized as a network: They fill slots (there are $wm_{size}$ slots in WM; by default, $wm_{size} = 4$). At every time step, each item in working memory has probability $p$ of being encoded in the NACS (by default, $p = 0.1$), and one item is chosen randomly to be refreshed (which increases its base-level activation, keeping it accessible in WM).

Each WM chunk node has a base-level activation defined as (similar to ACT-R; Anderson et al., 2004):

$$b_j^c = ib_j^c + c \sum_{t=1}^{n} t_i^{-d}$$

where $b_j^c$ is the base-level activation of chunk node $j$, $ib_j^c$ is the initial base-level activation (by default, $ib_j^c = 0$), $c$ is the amplitude (by default, $c = 2$), $d$ is the decay rate (by default, $d = 0.5$), and $t_l$ is the $l$th use of the chunk node. This measure decays exponentially and corresponds to the odds of needing chunk $j$ based on past experiences (Anderson, 1990).

Each new item entering WM may displace an existing chunk. The probability of being bumped is an inverse function of the (softmax) normalized base-level activations of
an existing item (Sun & Helie, 2013). This is because chunks that are being used
(attended to) are more likely to be needed in the future and should not be forgotten
(Anderson, 1990). Specifically,

\[
P(bumped = i) = \frac{1}{k \left[P(chunk \ i)\right]} \quad \text{(A2)}
\]

where \( P(bumped = i) \) is the probability that chunk \( i \) is chosen to be replaced by the new
chunk, \( P(chunk \ i) \) is the probability of chunk \( i \) being needed in the future where

\[
P(chunk \ i) = \frac{e^{b_i}}{\sum_j e^{b_j}} \quad \text{(A1, after softmax normalization)}, \quad \text{and } k \text{ is a constant where}
\]

\[
k = \sum_j \frac{1}{P(chunk \ j)}.
\]

**Priming**

The model was composed of 100 nodes, and a bipolar stimulus was randomly
generated and used to train the model for 100 Epochs. One hundred and one test stimuli
were generated by randomly flipping \{-1, 1\} the activation of between 0 and 100 nodes
in the training stimulus. Each test stimulus was presented to the network and processed
until convergence. The number of iterations needed for each stimulus to converge was
recorded. The average number of iterations required for convergence of a random vector
was computed by randomly generating 1,000 bipolar vectors. Each random vector was
processed in the network until convergence, and the number of iterations was recorded.
The simulation results are presented in Figure 3.
List length

The model was composed of 50 nodes. Fifty lists of stimuli were created by randomly generating between 1 and 50 bipolar stimuli (one list was generated for each list length). The model was trained with each list of stimuli for 50 Epochs. After training, each training stimulus was presented to the network, and the capacity of the network to restore the training stimulus was recorded (correct vs. incorrect). The proportion of correct recall for each list length is presented in Figure 5.

Serial position curve

The simulation of the serial position curve did not involve the attractor network in the bottom level of CLARION; only the interaction between the working memory and the NACS was required. The list lengths were: {10, 15, 20, 30, 40}. Each list length was simulated 100 times, and the proportion of recalls for each item was recorded. Because there was no time pressure, it was assumed that all the words encoded into the NACS and/or present in working memory at the end of the simulation were automatically recalled. The results are presented in Figure 6a.

The inhibition of the recency effect (Figure 6b) was simulated exactly as described above but with a list length of 15 and without recalling items from working memory (to simulate the delayed recall condition). The inhibition of the primacy effect (Figure 6c) was simulated exactly as described above with a list length of 16 words and a preloading of 3 items (as in the human experiment).
Appendix B: Deductive reasoning

This appendix presents a detailed derivation of the proofs related to the cases described in the *Deductive reasoning* subsection. In the simplest case, associative rules in CLARION can be represented using connection weights, and the top level of the NACS can be represented by a linear connectionist network (Haykin, 2009):

\[
s_j^r = \sum_i s_i \times w_{ij}^r
\]  

(B1)

where \(s_j^r\) is the activation of chunk node \(j\) following the application of an associative rule, \(s_i\) is the activation of chunk node \(i\), and \(w_{ij}^r\) is the strength of the associative rule between chunk nodes \(i\) and \(j\) (by default, \(w_{ij}^r = 1/n\), where \(n\) is the number of chunk nodes in the condition of the associative rule). The application of Eq. B1 is referred to as rule-based reasoning (Sun, 1994).

NACS chunks also share a relationship through similarity, which enables reasoning by similarity. In CLARION, the activation of a chunk node caused by its similarity to other chunk nodes is termed similarity-based reasoning. Specifically,

\[
s_j^s = s_{ci\sim cj} \times s_i
\]  

(B2)

where \(s_j^s\) is the activation of chunk node \(j\) caused by its similarity to other chunks, \(s_{ci\sim cj}\) is the similarity from chunk \(i\) to chunk \(j\), and \(s_i\) is the activation of chunk node \(i\). The similarity measure \(s_{ci\sim cj}\) is defined through the interaction of the top and bottom levels of the NACS:
An integrative account of memory and reasoning

\[ s_{c_i \rightarrow c_j} = \frac{n_{c_i \cap c_j}}{f(n_{c_j})} \]

\[ = \frac{\sum_k w_k^{c_j} h_k(c_i, c_j)}{f\left(\sum_k w_k^{c_j}\right)} \quad \text{(B3)} \]

where \( w_k^{c_j} \) is the weight of feature \( k \) in chunk \( j \) (by default, \( w_k^{c_j} = 1 \) for all \( k \)s), \( h_k(c_b, c_j) = 1 \) if chunks \( i \) and \( j \) share feature \( k \) and 0 otherwise, and \( f(x) \) is a slightly super linear, monotonically increasing, positive function [by default, \( f(x) = x^{1.1} \)]. By default, \( n_{c_i \cap c_j} \) counts the number of features shared by chunks \( i \) and \( j \) (i.e., the feature overlap in the bottom level) and \( n_{c_j} \) counts the total number of (bottom-level) features in chunk \( j \). However, the feature weights are learned and can be varied to account for prior knowledge or context (e.g., the context emphasizes a particular feature or past experience suggests that a particular feature is more useful). Thus, similarity-based reasoning in CLARION is naturally accomplished using (1) top-down activation by chunk nodes of their corresponding bottom-level feature-based representations, (2) calculation of feature overlap between any two chunks in the bottom level (as in Eq. B3), and (3) bottom-up activation of the top-level chunk nodes (Eq. B2). This kind of similarity calculation is naturally accomplished in a multi-level cognitive architecture and represents a form of synergy between the explicit and implicit modules.

**Inexact information**

Let \( c_i \) and \( c_j \) be chunks in the top level of the NACS, and \( w_{ij} \) be a rule in the top level of the NACS linking chunks \( i \) and \( j \). Assume that \( s_i < 1 \).

**Derivation.**
\[ s_j = s_i \times w_{ij}^r \]

\[ = s_i \]

In words, chunk \( j \) is partially activated, proportional to the activation of chunk \( i \).

The strength of chunk \( j \) can be used to represent the confidence in the inference.

**Incomplete information**

Let \( c_i, c_j, c_k, c_l \) be chunks in the top level of the NACS and \( w_{il}^r, w_{jl}^r, w_{kl}^r \) be rules in the top level of the NACS linking chunks \( i, j, \) and \( k \) with chunk \( l \). Assume that \( s_i = s_j = 1 \) and that \( s_k = 0 \).

**Derivation.**

\[ s_l = s_i \times w_{il}^r + s_j \times w_{jl}^r + s_k \times w_{kl}^r \]

\[ = w_{il}^r + w_{jl}^r \]

\[ = 2/3 \]

In words, chunk \( l \) is partially activated, in correspondence with the proportion of its premises that are activated.

**Similarity matching**

Let \( c_i, c_j, c_k \) be chunks in the top level of the NACS, \( s_{ci-cj} \) 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 \( s_i = 1 \).

**Derivation.**

\[ s_k = s_i \times s_{ci-cj} \times w_{jk}^r \]

\[ = \frac{n_{c_i \cap c_j}}{f(n_{c_i})} \]
In words, chunk \( k \) is partially activated, proportional to the similarity between chunks \( i \) and \( j \).

**Superclass to subclass inheritance**

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

**Derivation.**

\[
\begin{align*}
s_k &= s_i \times s_{c_i-\neg c_j} \times w_{jk}^r \\
&= \frac{n_{c_i \cap c_j}}{f(n_{c_i})} \\
&= \frac{n_{c_j}}{f(n_{c_i})} \\
&\approx 1 \text{ (but } < 1) 
\end{align*}
\]

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

**Subclass to superclass “inheritance”**

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

**Derivation.**
In words, chunk $k$ is partially activated, proportional to the ratio of the number of features (i.e., bottom-level nodes) of chunks $j$ and $i$. Because chunk $i$ represents a category that is a proper subset of the category represented by chunk $j$, chunk $i$ is represented by more bottom-level features than chunk $j$. This represents the uncertainty of inductive reasoning.

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 chunk $i$ is a proper subset of the category represented by chunk $j$, and $w_{jk}^r$ and $w_{im}^r$ are rules in the top level of the NACS linking chunks $j$ and $k$ and chunks $i$ and $m$ (respectively). Assume that $s_i = 1$.

**Derivation.**

\[
s_k = s_j \times s_{c_j \cap c_i} \times w_{jk}^r
\]
\[
= \frac{n_{c_j \cap c_i}}{f(n_{c_j})}
\]
\[
= \frac{n_{c_j}}{f(n_{c_j})}
\]
\[
< 1
\]

while,

\[
s_m = s_i \times w_{im}^r
\]
\[
= 1
\]
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 $f(\bullet)$ 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.

Cancellation of subclass to superclass “inheritance”

Let $c_i$, $c_j$, $c_k$, $c_m$ be chunks in the top level of the NACS, the category represented by chunk $i$ is a proper subset of the category represented by chunk $j$, and $w_{ik}'$ and $w_{jm}'$ are rules in the top level of the NACS linking chunks $i$ and $k$ and chunks $j$ and $m$ (respectively). Assume that $s_j = 1$.

Derivation.

\[
\begin{align*}
  s_k &= s_j \times s_{c_j \sim c_i} \times w_{ik}' \\
  &= \frac{n_{c_j \cap c_i}}{f(n_{c_j})} \\
  &= \frac{n_{c_j}}{f(n_{c_j})} \\
  &< 1 \\

  s_m &= s_j \times w_{jm}' \\
  &= 1 \\

  \text{Hence, } s_m > s_k.
\end{align*}
\]

In words, chunk $k$ 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 $m$ is fully activated, because top-level rules are exact. This restates the prominence of rule-based reasoning over similarity-based reasoning.
Mixed rules and similarities

Here, we present six subcases involving different amounts of rule-based and similarity-based reasoning.

(1) Let $c_i$, $c_j$, $c_k$, be chunks in the top level of the NACS, $s_{c_j-c_k}$ 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 $s_i = 1$.

\[ s_k = s_i \times w_{ij}^r \times s_{c_j-c_k} \]

\[ = \frac{n_{c_j\cap c_k}}{f(n_{c_k})} \]

In words, chunk $k$ is partially activated, proportional to its similarity with chunk $j$ (because chunk $j$ is fully activated by rule-based reasoning).

(2) 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 linking chunks $i$ and $j$ and chunks $j$ and $k$ (respectively). Assume that $s_i = 1$.

\[ s_k = s_i \times w_{ij}^r \times w_{jk}^r \]

\[ = 1 \]

In words, chunk $k$ is fully activated because top-level rules are exact (i.e., rule-based reasoning is transitive).

(3) Let $c_i$, $c_j$, $c_k$, $c_m$ be chunks in the top level of the NACS, $s_{c_i-c_j}$ is the similarity between chunks $i$ and $j$, and $w_{jk}^r$ and $w_{km}^r$ are rules in the top level of the NACS linking chunks $j$ and $k$ and chunks $k$ and $m$ (respectively). Assume that $s_i = 1$.

\[ \text{Derivation.} \]
In words, chunk $m$ is partially activated, proportional to the similarity between chunks $i$ and $j$. This is because the activation of chunk $j$ is a function of its similarity to chunk $i$. However, the activation from chunk $j$ to chunk $k$ to chunk $m$ is transmitted exactly.

(4) Let $c_i$, $c_j$, $c_k$, $c_m$ be chunks in the top level of the NACS, $s_{cj-ck}$ is the similarity between chunks $j$ and $k$, and $w_{ij}^r$ and $w_{km}^r$ are rules in the top level of the NACS linking chunks $i$ and $j$ and chunks $k$ and $m$ (respectively). Assume that $s_i = 1$.

**Derivation.**

$$s_m = s_i \times s_{c_j} \times w_{jk}^r \times w_{km}^r$$

$$= \frac{n_{c_j}}{f(n_{c_j})}$$

In words, chunk $m$ is partially activated, proportional to the similarity between chunks $j$ and $k$.

(5) Let $c_i$, $c_j$, $c_k$, $c_m$ be chunks in the top level of the NACS, $s_{ck-cm}$ is the similarity between chunks $k$ and $m$, and $w_{ij}^r$ and $w_{jk}^r$ are rules in the top level of the NACS linking chunks $i$ and $j$ and chunks $j$ and $k$ (respectively). Assume that $s_i = 1$.

**Derivation.**

$$s_m = s_i \times w_{ij}^r \times s_{c_k} \times s_{c_m}$$

$$= \frac{n_{c_j}}{f(n_{c_m})}$$
In words, chunk $m$ is partially activated, proportional to its similarity with chunk $k$.

(6) Let $c_i, c_j, c_k, c_m$ be chunks in the top level of the NACS, $s_{ci-cj}$ and $s_{ck-cm}$ are similarity measures between chunks $i$ and $j$ and chunks $k$ and $m$ (respectively), and $w_{jk}^r$ is a rule in the top level of the NACS between chunks $j$ and $k$. Assume that $s_i = 1$.

**Derivation.**

$$s_m = s_i \times s_{c_i-c_j} \times w_{jk}^r \times s_{c_k-c_m}$$

$$= \frac{n_{c_i-c_j}}{f(n_{c_i})} \times \frac{n_{c_k-c_m}}{f(n_{c_m})}$$

This case is a little more complex to interpret. Chunk $m$ is partially activated, proportional to its similarity to chunk $k$. However, unlike in the previous cases, chunk $k$ is not fully activated: the activation of chunk $k$ is a function of the similarity between chunks $i$ and $j$. Hence, all else being equal, the activation of chunk $m$ is smaller here than in the preceding subcase (5).
Appendix C: The role of functional attributes in inductive reasoning

In CLARION, induction is accounted for by similarity-based reasoning. The reader is referred to Eqs. B2-B3 in Appendix B above for a formal treatment of similarity-based reasoning in CLARION. Only one numerical parameter was varied to derive the following role of functional attributes (i.e., $w^c_{kj}$, the weight of feature $k$ in chunk $j$). The relative weights of features in CLARION can be emphasized by the context (e.g., compare concepts $x$ and $y$ according to their color) or learned through past experience (e.g., learning what feature is useful in a specific task using reinforcement learning).

Let chunk $i$ represent ‘chicken’, chunk $t$ represent ‘tiger’, and chunk $j$ represent ‘hawk’. If all the features are weighed equally, formalization using Eqs. B2 and B3 yields:

$$s_i \times \frac{n_{c_i \cap c_j}}{f(n_{c_j})} > s_t \times \frac{n_{c_i \cap c_j}}{f(n_{c_j})} \Rightarrow s_i \times \frac{\sum w^c_{kj} h_k(c_i, c_j)}{f\left(\sum w^c_{kj}\right)} > s_t \times \frac{\sum w^c_{kj} h_k(c_t, c_j)}{f\left(\sum w^c_{kj}\right)}$$

where $h_k(c_i, c_j) = 1$ if chunks $i$ and $j$ share feature $k$ and 0 otherwise, and $w^c_{kj}$ is the weight of feature $k$ in chunk $j$. The denominators are the same so they can be dropped. Also, let feature $k = 0$ represent the functional attribute of feeding habit:

$$s_i \times \left[w^c_0 \times h_0(c_i, c_j) + \sum_{k>0} w^c_{kj} h_k(c_i, c_j)\right] > s_t \times \left[w^c_0 \times h_0(c_t, c_j) + \sum_{k>0} w^c_{kj} h_k(c_t, c_j)\right]$$

Because $k = 0$ represents feeding habits, $h_0(c_i, c_j) = 0$ and $h_0(c_t, c_j) = 1$.

$$s_i \times \sum_{k>0} w^c_{kj} h_k(c_i, c_j) > s_t \times \left[w^c_0 + \sum_{k>0} w^c_{kj} h_k(c_t, c_j)\right]$$
The above expression represents a regular case of similarity effect in induction. What is the condition that would reverse this inequality and create an exception to similarity?

\[ s_i \times \sum_{k > 0} w^c_k h_k (c_i, c_j) < s_i \times \left[ w^c_0 + \sum_{k > 0} w^c_k h_k (c_i, c_j) \right] \]

\[ \Rightarrow w^c_0 > \frac{s_i}{s_i} \times \sum_{k > 0} w^c_k h_k (c_i, c_j) - \sum_{k > 0} w^c_k h_k (c_i, c_j) \]

Hence, an exception to the similarity effect can be observed when the weight of a feature (here feeding habit) is larger than the difference between the weighted feature overlap (excluding feeding habit) of the two induction cases (where the feature overlap of the normal case is modulated by chunk activation). This situation is mathematically described by the derived inequality above. In this way, CLARION accounts for exceptions to similarity in induction when the context sufficiently emphasizes a particular exception feature.
Acknowledgments

This research was supported by a postdoctoral research fellowship from Le Fonds Québecois de la Recherche sur la Nature et les Technologies to the first author, and by the Army Research Institute contract W74V8H-05-K-0002 and by the Office of Naval Research grant N00014-08-1-0068, both to the second author.
Footnotes

1 Note that memory systems may, alternately, be viewed either as functional modules or as physiological modules (for more discussions of this and other distinctions, see Fodor, 1983). However, our main focus here is on functional modules.

2 Note that the CLARION cognitive architecture is not about computational techniques (although there have been many innovations in this regard, as published by the authors in the late 1990's and early 2000's); the focus here is on providing psychological explanations through integration of mechanisms.

3 If the model needs to encode knowledge that a statement is false, a separate chunk and/or rule is created to represent this information. Lateral inhibition can be used so that a chunk and its negation cannot be both highly activated simultaneously.

4 Note that these phenomena (i.e., positive and negative priming) are not dependent on network size, but the exact measures are.

5 This framework is similar to Anderson (1990). However, the choice of the item to be discarded when the working memory is full was uniform in Anderson’s rational analysis of memory.

6 This case is a “weak” form of deduction, because the superclass to subclass relationship is similarity-based; not rule-based. Similarity-based reasoning can converge toward full activation, but never reaches it (unlike rule-based reasoning, which exactly transmits activation).
Figure captions

**Figure 1.** A high-level representation of CLARION.

**Figure 2.** Frequency effect as modeled in Helie et al. (2006). In each panel, the black bars represent the (discretized) environmental distribution (target frequencies) and the grey bars represent the model distribution (estimated frequencies). The x-axis represents the category where the random vector converged (1...3) and the y-axis represents the proportion of convergence. (a) Uniform distribution, (b) Multi-normal distribution, (c) Step distribution, (d) Exponential distribution.

**Figure 3.** Simulation results for the convergence of 101 test stimuli. The filled circle represents the average number of iterations required for the convergence of a random vector. Simulation details are provided in Appendix A.

**Figure 4.** Synergistic activation for cued recall in the CLARION NACS. Simulation details can be found in Helie & Sun (2010a).

**Figure 5.** List length effect in CLARION. Simulation details are presented in Appendix A.

**Figure 6.** (a) Serial position curves simulated by CLARION. (b) Inhibition of the recency effect in CLARION when the content of working memory is not automatically retrieved at the beginning of free recall. (c) Serial position curve simulated by CLARION with a pre-loading of three items. The simulation details for all three panels are provided in Appendix A.
Figure 4

Activation

0  2  4  6  8  10  12  14
# Clue-words