Learning to Use Episodic Memory

Nicholas A. Gorski (ngorski@umich.edu)
John E. Laird (laird@umich.edu)
Computer Science & Engineering, University of Michigan
2260 Hayward St., Ann Arbor, MI 48109 USA

Abstract

This paper brings together work in modeling episodic memory and reinforcement learning. We demonstrate that is possible to learn to use episodic memory retrievals while simultaneously learning to act in an external environment. In a series of three experiments we investigate learning what to retrieve from episodic memory and when to retrieve it, learning how to use temporal episodic memory retrievals, and learning how to build cues that are the conjunctions of multiple features. Our empirical results demonstrate that it is computationally feasible to learn to use episodic memory in all three experiments, and furthermore, that learning to use internal episodic memory accomplishes tasks that reinforcement learning alone does not. These experiments also expose some important interactions that arise between reinforcement learning and episodic memory.

Keywords: Artificial Intelligence; Cognitive Architecture; Episodic Memory; Intelligent Agents; Reinforcement Learning.

Introduction

In this paper, we study possible mechanisms for learning to use the retrieval of knowledge from episodic memory. This unifies two important related areas of research in cognitive modeling. First, it extends prior work on the use of declarative memories in cognitive architecture where knowledge is accessed from declarative memories via deliberate and fixed cued retrievals (Wang & Laird, 2006; Anderson, 2007; Nuxoll & Laird, 2007) by exploring mechanisms for learning to use both simple and conjunctive cues. Second, it extends work on using reinforcement learning (RL) (Sutton & Barto, 1998) to learn not just control knowledge for external actions, but also to learn to control access to internal memories.

Earlier work has investigated increasing the space of problems applicable to RL algorithms by including internal memory mechanisms that can be deliberately controlled: Littman (1994) developed an RL agent that learned to toggle internal memory bits; Pearson et al. (2007) showed that an RL agent could learn to use a simple symbolic long-term memory; and Zilli & Hasselmo (2008) developed a system that learned to use both an internal short-term memory and an internal spatial episodic memory, which could store and retrieve symbols corresponding to locations in the environment. All three cases demonstrated a functional advantage from learning to use memory.

Our work significantly extends these previous studies in four ways: first, our representation is fully relational, which complicates both the structure of episodic memory and RL; second, our episodic memory system automatically captures all aspects of experience; third, our system learns not only when to access episodic memory, but also learns conjunctive cues and when to use them; and fourth, it takes advantage of the temporal structure of episodic memory by learning to advance through episodic memory when it is useful (this property is also shared by the Zilli & Hasselmo system, but for simpler task and episodic memory representations).

Our studies are pursued within a specific cognitive architecture, namely Soar (Laird, 2008), which incorporates all of the required components: perceptual and motor systems for interacting with external environments, an internal short-term memory, a long-term episodic memory, an RL mechanism, and a decision procedure that selects both internal and external actions. In comparison, ACT-R (Anderson, 2007) has many similar components but does not have an episodic memory. Its long-term declarative memory stores only individual chunks, and it does not store episodes that include the complete current state of the system. To do so would require storing the contents of all ACT-R's buffers as a unitary structure, as well as the ability to retrieve and access them, without having the retrieved values being confused with the current values of those buffers. Moreover, ACT-R's declarative memory does not inherently encode the temporal structure of episodic memory, where temporally consecutive memories can be recalled (Tulving, 1983). While the work presented in this paper is specific to learning to use an episodic memory, similar work could be pursued in the context of ACT-R by learning to use its declarative memory mechanism. However, we are unaware of existing work in that area, and even if there were, it would fail to engage the same issues that arise with episodic memory.

Background

Soar includes an episodic memory that maintains a complete history of experience (Nuxoll & Laird, 2007), implemented so as to support efficient memory storage and retrieval (Derbinsky & Laird, 2009). "Snapshots" of Soar's working memory, which is a relational graph structure, are automatically stored in episodic memory so that learning is not required to control how and when information is stored.

To retrieve an episode, a *cue* is created in working memory by Soar's procedural knowledge, which is encoded as rules. A cue is a relational structure that describes a subset of working memory elements that may exist in an episode. The cue is compared to the stored episodes, and the episode that best matches the cue is retrieved to working memory. If there are multiple episodes with the same degree of match, the most recent of those episodes is retrieved. Once an episode is retrieved to working memory, other knowledge (such as procedural knowledge) can access it.

After performing a cue-based retrieval, the agent can use the unique temporal structure of episodic memory and retrieve the next episode, providing a mechanism for the agent to move forward through its memories, recalling sequences of experiences, in addition to specific instances.

Although it is straightforward to create agents that use episodic memory for a variety of purposes (Nuxoll, 2007), this requires endowing the agent with knowledge as to when to access episodic memory and what structures should be used for cueing retrievals. In this research, we study the possibility of learning when to use episodic memory as well as learning which cues to use from experience using Soar's RL mechanism. Soar uses a type of RL called Q-Learning (Nason & Laird, 2005). Q-Learning learns the value for potential actions using temporal-difference updates of reward (Sutton & Barto, 1998) and in Soar this can be used to learn to control external actions as well as internal actions that retrieve information from episodic memory.

Well World

In order to explore how an agent might learn to use an internal episodic memory, we constructed several tasks within a domain we call "Well World." The domain is simple enough to be tractable for an RL agent, but rich enough such that episodic memory can potentially improve performance. The goal in Well World is to be safe when not thirsty, and to quench thirst as soon as possible when thirsty.

In Well World, the agent moves between objects and can consume resources, such as *water* or *shelter* if they are present. The agent perceives the object that is present at its current location, features of the object (including resources that are present), and adjacent objects that it can move to.

Figure 1 shows the base Well World environment. There are two wells which can provide the water resource ("r: water" in the Figure). Well 1 is currently empty, while well 2 has water available. There is also a shelter, which allows the agent to feel safe when the agent is not thirsty.

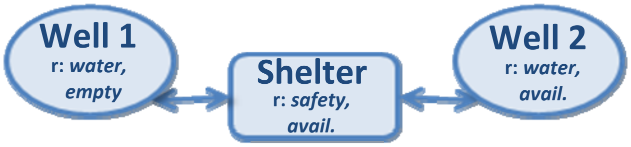

Figure 1: Objects, resources, and adjacency in Well World.

An agent in Well World possesses two internal drives: thirst and safety. When its thirst is quenched, an agent's thirst drive is 0; on every time step after it has been quenched, the thirst drive is incremented by a small amount. After passing a threshold, the agent is thirsty until it quenches its thirst, which requires that the agent move to a well object that contains water and consume water from it.

Only one well contains water at any given time; once water is consumed from a well, it is empty and water becomes available in the other well. In Figure 1, well 2 has water available while well 1 does not. Once the water at well 2 is consumed, well 2 will be empty while well 1 will have water available, and so on.

The agent's other internal drive is to feel safe. The agent satisfies this drive when not thirsty or when it consumes the safety resource from the shelter (which is always available).

Two of Well World's characteristics make it challenging for RL: the agent can only perceive the status of the object in its current location, and wells alternate in containing water and being empty. To perform optimally, an agent must maintain a memory of the environment (the status of the wells) – something a conventional RL agent lacks.

Reinforcement in Well World

The reward signal used by an RL agent in Well World is determined by the state of the agent's internal drives, as well as changes in the states of those drives. Reinforcement in Well World is internally calculated by the agent based on its internal drives, rather than determined by the environment as in a conventional RL setting.

The most important aspects of the agent's reward structure are that: there is a cost for taking external actions and it is greater than the cost of internal actions; there is a reward for not staying at the wells when the agent is not thirsty; there is a significant reward for performing the action (consuming water when thirsty) that is made possible by the episodic retrieval; and there is no explicit reward for using episodic memory, rather such control knowledge must be learned while seeking to satisfy thirst. The reward values are as follows. External actions result in -1 reward, while internal actions result in -0.1 reward. On every time step that the agent is thirsty, it receives -2. On every time step that the agent is not thirsty and consumes the safety resource, it receives +2. Finally, the agent receives +8 for satisfying its thirst. Concurrent rewards (e.g. the agent is thirsty and takes an external action) are summed together.

Experiments in Well World

Within the Well World domain, we developed a suite of three experiments to evaluate various strategies for using episodic memory. In the first experiment, we tested an agent's ability to learn to select a single cue for episodic memory retrieval. The second experiment tested an agent's ability to learn to use the temporal aspects of episodic memory retrievals. The third experiment investigated the agent's ability to create a conjunctive cue (i.e. a cue that contains more than one feature). This set of experiments investigated all of the ways retrievals can be used to access Soar's episodic memory. Before discussing the experiments and results, we present the details of our agent.

Agent Design and Implementation

To explore learning to use episodic memory, we created a Soar agent. In our agent, procedural knowledge determines what actions can be taken in the external environment as well as what actions can be taken to access the internal episodic memory. On each time step of the environment, the procedural knowledge proposes applicable actions based on the agent's current perception of the environment and its internal state. It proposes consuming resources that are present, and it proposes moving to any adjacent objects. There are two internal actions that it can propose for controlling episodic memory (depending on the experiment, as described below): create a cue to initiate a retrieval, or if there has been a retrieval, advance episodic memory forward in time. In experiments where the agent must learn which retrieval cue to use, multiple retrieval actions are proposed, one for each cue.

The decision procedure selects actions probabilistically, based on what has been learned by Q-learning. A central problem in RL is the exploitation vs. exploration trade-off (Sutton & Barto, 1998) - an agent must balance between choosing actions based on what it has already learned (exploitation), with choosing other actions to gain more knowledge about the effects of those actions (exploration). Our agent uses a linearly decaying exploration rate; initially, the agent selects a random action half of the time and the other half selects actions according to their learned values. As time goes on, the agent takes random actions less often.

Although the Well World is presented in terms of "water," "thirst," "empty," and "wells," the agent does not know the semantics of these terms. To the agent, consuming water is simply a possible action that it can take; it must learn that it is good to consume water when thirsty, that water is available at a particular well, and so on.

In contrast to many learning systems that are "reset" after a performance or learning trial, our agent has a continual existence and once it begins acting in the environment, it continues to move about Well World, performing actions, until the end of the experiment.

Instead of using episodic memory, the agent could have maintained task-specific events in working memory (such as which well the agent last consumed). This memory would provide the agent with sufficient knowledge to learn to act in the domain. However, this approach requires task-specific background knowledge while our approach is completely general and applies to any task without additional task-specific knowledge.

Results presented in this paper are the average of 250 trials, were smoothed with 4253 Hanning, and normalized so that an average reward of 0 per action is optimal.

Learning to Retrieve Episodic Memories

The first experiment tests the basic behavior of using RL to learn to use an internal episodic memory, and its purpose is to determine whether an RL agent can learn what to retrieve and when retrieval is appropriate. In this experiment, an agent must learn to retrieve information from memory using a single cue, where the retrieved episode provides sufficient information to perform the task. In one condition, there is only one cue available to the agent to use for retrieval; in another, the agent selects from six possible cues, only one of which is useful.

In Well World (Fig. 1), the optimal *policy* (where a policy is a mapping of every state, or situation, to an action) is for the agent to move to the shelter and consume the safety resource when it is not thirsty, and to move to the well that contains water and consume water there when it is thirsty. As agents are unable to perceive which well contains water, an agent that does not possess an internal memory does not know which well it must move to and wastes time while trying to find available water. An agent that possesses an internal memory, however, can retrieve the episode for the last visited well.

Figure 2 shows the results under the following conditions: only the correct cue is available to be learned (labeled "No distracters"); the correct cue and five distracters are available to be learned ("5 distracters"); and a baseline condition in which episodic memory is lesioned and all retrievals fail ("Lesioned ep. mem.").

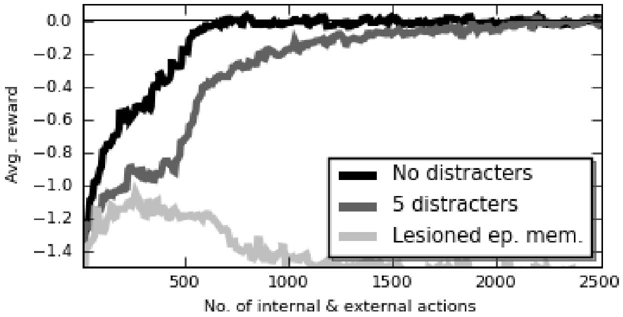

Figure 2: Performances of agents learning to retrieve episodic memories.

When only a single cue is available for retrieval, the agent quickly learns both to act in the environment and to use its internal memory so as to receive the maximum amount of possible reward (it follows the optimal policy). When distracter cues are present, the agent learns more slowly but also converges to the optimal policy. These results indicate that the agent can learn to use its internal memory while simultaneously interacting with its environment.

Learning to Retrieve What Happened Next

A unique aspect of episodic memory is that events are linked and ordered temporally. In Soar's episodic memory, memory retrievals can be controlled temporally by advancing to the next (or previous) memory after performing a cue-based retrieval, providing a primitive envisioning or planning capability where the agent can use its prior history to predict potential future situations. Through RL, the system has the potential of learning when and how to perform such primitive planning.

In the previous experiment, agents retrieved episodic memories of the last time that they had perceived the water resource, which was sufficient information to determine which well to move to in order to find water. An alternative strategy, explored in this experiment, is to retrieve a situation that closely resembles the agent's current situation and then advance to the next memory to remember what the agent did the last time that it was in a similar situation.

In this experiment, the agent has available the normal actions in the environment (moving and consuming resources). It also has two internal actions available to it: a cue-based episodic memory retrieval, which uses structures from its current perceptual state to retrieve the most recent situation that most closely resembled its current situation; and an action (called *advance*) that retrieves the next episode (the episode that was stored after the episode most recently retrieved). Thus, the agent must learn when to do a cue-based retrieval and when to advance its retrieval, and these actions are always competing with the other actions.

For this task, the optimal policy for the agent when it is not thirsty is to move to the shelter and consume the safety resource. When it becomes thirsty, the optimal policy is to perform a retrieval cued by its current state, which results in the agent remembering the last time it was thirsty at the shelter. The next step is to perform an advance retrieval, which results in the agent remembering where it moved to after it was last thirsty at the shelter. This is followed by moving to the other well, where the agent will find water, as the well that it previously visited will be empty.

An important characteristic of this task is that the agent must learn to use its memory while simultaneously learning to act in the world. The best policy for memory usage depends on the agent's prior actions in the environment; if the agent does not visit and consume resources in the appropriate order (i.e. follow the optimal policy for external actions), then the agent is not guaranteed to gain useful information from internal memory retrievals.

The performances of the agent under three conditions are plotted in Figure 3: using a fixed policy to automatically advance episodic memory after a cue-based retrieval, making only the initial cue-based retrieval open to learning; learning when to select both retrieval and advance actions; and a baseline comparison where episodic memory is lesioned.

There are several features of the results in Figure 3 worth further discussion. First, the performances of both agents that use episodic memory are very similar. This was unexpected. The agent that learns to use the temporal action has a larger action space, which implies that it would initially perform worse than the agent that had a fixed policy to advance to the next memory after retrieving. Second, the agents reach asymptotic performance after about 4,500 actions, but do not reach the optimal level of performance. Third, while the agents are exploring while selecting actions (until the 4,000th action), the agent that deliberately selects actions outperforms the agent that has a fixed policy to advance after retrieving. Fourth, there is a dramatic improvement in performance just after exploration ends. The agent retrieves episodes from memory that are similar to its current situation, and uses its past actions to determine how to act in the present situation. If the agent takes an exploratory action when it is thirsty or is not at the shelter when it becomes thirsty because of an exploratory action, then the behavior that results is no longer correct. In effect, although exploration of the problem space is necessary for the agent to learn, it hinders the agent's performance in the task and once there is no exploration the agent can perform significantly better.

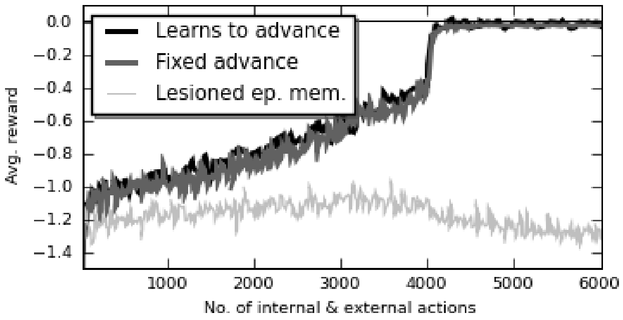

Figure 3: Performances of agents using temporal control of episodic memory after retrieval.

All four of these phenomena are explained by the difficulty of the learning problem that was identified above - for the agent to learn the optimal policy for using its internal memory, it must also learn a near optimal policy for acting in the environment. The learning problem is partially observable, in that the effects of the agent's memory actions depend on the history of the agent's actions in the environment, but the agent cannot perceive that history. The agent is faced with a conundrum: it must learn how to use its memory while settling on a good policy in the environment, but it must also settle on a good policy in the environment without knowing how to use its memory. Often the agent is successful in learning to simultaneously control both memory and external action, but occasionally the agent is unable to converge to the best policy.

The asymptotic behavior of the agent is very near to optimal, which demonstrates that the agent still learns to perform relatively well in the environment. In fact, in all trials the agent converged to one of two policies: the optimal policy, or a policy in which the agent uses episodic memory retrievals to toggle a conceptual bit, as in the agents in Littman (1994) and Pearson et al. (2007). In this second policy, when the agent becomes thirsty, it immediately moves to one of the wells (the same well every time). If the well contains water, it consumes it; if not, it performs a retrieval and moves back to the shelter. At the shelter, the agent now knows that it has performed a retrieval and instead of moving to the same well again (the one that it just visited and knows is empty), it moves to the other well and consumes water there. Essentially, the agent learns which well to move to when it is thirsty based on whether a retrieval has been performed, and not based on the contents of what was retrieved.

From Figure 3 it is also clear that the agent requires many more actions before converging to near-optimal behavior in comparison with the agents from the previous experiment. For the agent to converge to the optimal control policy, it must explore significantly longer than in the previous experiment; however, as noted above, this exploration can hinder the agent's performance in the task as well. We investigated how different exploration policies affected the agent's convergence to the optimal policy and the results are presented in Table 1. In all three cases, the rate of random action selection decays linearly over time. Table 1 presents data gathered when random action selection decayed over 500 steps, 5,000, and 50,000. These results suggest that there are important interactions between the exploration rate decay and learning that need to be pursued in future work.

Table 1: Percentage of trials that converged to optimal memory control policy when using temporal control for different periods of exploration.

Condition	500	5,000	50,000
Fixed	26%	60%	25%
Deliberate	36%	71%	38%

Learning To Construct a Retrieval Cue

In the first experiment, one condition involved the agent learning to select between multiple cues when retrieving from memory. In the second experiment, the agent used cues with more than one feature (multiple features of its current state) in order to retrieve from memory. The purpose of this third experiment is to investigate whether an agent can learn to select multiple features to use as cue, combining aspects of both previous experiments.

In order to test this capability, it was necessary to extend the base Well World configuration so that there were more wells and more features that could be used for retrieval. A third well was added to the environment, and a color feature was added to all objects; the modified environment is shown in Figure 4. As in the base environment, only wells 1 and 2 ever contain water, and they continue to alternate between full and empty as before. Well 3 is always empty and never contains water; it was added to the environment to serve as a distracter to the agent when it performs a cue-based retrieval with features not present on the other two wells.

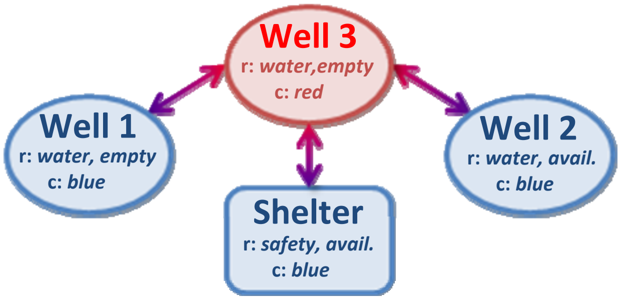

Figure 4: Well World modified with an additional well and an additional feature, color.

In this task, the optimal policy when the agent is not thirsty is still to navigate to the shelter and consume safety. When thirsty, the agent must construct a cue containing features corresponding to the two wells that can contain water in order to determine which well it visited last; these features are "resource: water" and "color: blue". After retrieving the memory of the last blue well that it visited, the agent must then navigate to the *other* blue well and consume water there to satisfy its thirst.

If the agent constructs a cue with some other combination of features, the result of its retrieval depends on its previous behavior – but the retrieved episode will not provide sufficient information for the agent to determine which well to visit next, because the agent must always visit the red well before visiting the shelter. As Soar's episodic memory mechanism is biased towards more recent episodes when multiple memories are perfect matches to the cue, building a cue that contains only "resource: water" or "color: blue" will not result in the agent remembering the last well that it visited (assuming that it has moved back to the shelter). Color: blue will lead to the retrieval of the shelter, while retrieval of resource: water will lead to retrieval of well 3.

The performances of the agent that constructs retrieval cues in the modified Well World are shown in Figure 5 for three conditions: learning to construct a cue from the two correct possibilities ("No distracters"), learning to construct a cue when two distracters are present, and a baseline where episodic memory is lesioned. In the two conditions, there are different sets of features with which an agent may construct the cue: the first has only the two correct features available (resource: water, and color: blue), while the other also has their complements (resource: water/shelter, and color: blue/red). Cues can contain any combination of features so the agent must learn to construct the cue from the correct combination in both cases.

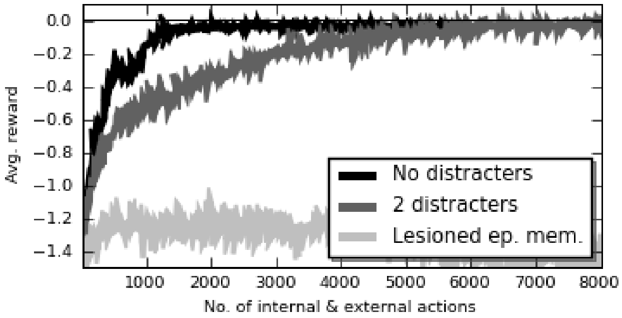

Figure 5: Constructing cues with more than one feature in order to retrieve from episodic memory.

The agent converges to the optimal policy under both conditions, more slowly when two distracter features are present, as expected. These results indicate that an agent can learn to build conjunctive cues from raw features, and use them in a task to retrieve from episodic memory.

Discussion and Conclusions

Although in all three experiments the agent is faced with learning to use its memory while acting in the environment (and thus affecting what information will be retrieved from memory in the future), the interaction of memory and action in the environment is significantly more intertwined in the second experiment. There, the agent's past actions directly impact the usefulness of information retrieved from episodic memory. In all experiments, the agent learns very early on to consume safety when it is not thirsty, and to immediately move to the shelter as soon as it is not thirsty. In the first and third experiments, this means that when the agent retrieves an episode from memory using features of a well as a cue, it will typically be the well that it last consumed water from. However, in the second experiment, the agent is retrieving memories of the first action that it took to quench its thirst, and *not* the memory of when it finally managed to quench it. It not only takes longer to learn how to best act in this setting, but the eventual result is that sometimes instead of converging to the optimal policy it instead converges to a local maximum in the policy space. One issue for future research that we identified in the second experiment is that our approach lacks task-independent strategies for controlling exploration.

In all experiments, the cost of an internal action is less than the cost of external action in the environment. The rationale behind this decision is that it takes significantly more time to act in the world than it does to perform an internal action. Although internal rewards are structured in this way, we have gathered results (not presented here in the interest of space) that demonstrate that this feature of our reward structure does not affect the eventual learned behaviors, but does serve to speed up the learning process by encouraging the selection of internal actions initially.

These three experiments demonstrate that RL can be applied successfully to learn to use internal actions over an episodic memory mechanism while simultaneously learning to act in its environment. Additionally, RL alone cannot be successfully applied to those same tasks, demonstrating that there is a functional advantage to combining RL with an episodic memory in some settings. We also demonstrated that RL can be used to learn when to retrieve, learn which cue to use for retrieval, learn when to use temporal control, and learn to build a cue from a set of possible features.

More broadly, this research opens up the possibility of extending the range of tasks and behaviors modeled by cognitive architectures. To date, scant attention has been paid to many of the more complex properties and richness of episodic memory, such as its temporal structure or the fact that it does not capture just isolated structures and buffers but instead captures working memory has a whole. Similarly, although RL has made significant contributions to cognitive modeling, it has been predominantly used for learning to control only external actions. This research demonstrates that cognitive architectures by incorporate both episodic memory and RL, they can learn behavior that is possible only when they are combined.

Although our research demonstrates that it is possible to learn to use episodic memory, it also raises some important issues. Learning is relatively fast when the possible cues lead to the retrieval of an episode that contains all of the information that an agent requires in order to determine how to act in the world. When retrieving episodes that most closely match the current state and then using temporal control of memory to remember what happened next, however, learning is slower and does not always converge to the best possible behavior. Learning to use episodic memory to project forward is difficult – requiring many trials to converge and without a guarantee that optimal behavior will be achieved. Do these same issues arise in humans or do they have other mechanisms that avoid these issues? One obvious approach to avoid the issues encountered in our experiment is to use one method, such as instruction or imitation, to initially direct behavior so that correct behavior is experienced and captured by episodic memory, and then learning to use those experiences would probably be much faster.

Acknowledgments

The authors acknowledge the funding support of the Office of Naval Research under grant number N00014-08-1-0099.

References

Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe?* New York, NY: Oxford U. Press.

Derbinsky, N. & Laird, J. E. (2009). Efficiently Implementing Episodic Memory. *Proc. of the 8th Intl. Conf. on Case-Based Reasoning.* Seattle, WA.

Laird, J. E. (2008). Extending the Soar Cognitive Architecture. *Proc. of the First Artificial General Intelligence Conference* (224-235). Memphis, TN.

Littman, M. L. (1994). Memoryless Policies: Theoretical Limitations and Practical Results. *Proc. of the 3rd Intl. Conf. on Simulation of Adaptive Behavior*. (238-245).

Nason, S. & Laird, J. E. (2005). Soar-RL, Integrating Reinforcement Learning with Soar. *Cog. Sys. R., 6,* 51-59.

Nuxoll, A. (2007). *Enhancing Intelligent Agents with Episodic Memory.* Doctoral dissertation, Computer Science & Engineering, U. of Michigan, Ann Arbor.

Nuxoll, A. & Laird, J. E. (2007). Extending Cognitive Architecture with Episodic Memory. *Proc. 21st National Conference on Artificial Intelligence.* Vancouver, BC.

Pearson, D., Gorski, N.A., Lewis, R.L. & Laird, J.E. (2007). Storm: A Framework for Biologically-Inspired Cognitive Architecture Research. *ICCM-07.* Ann Arbor, MI.

Sutton, R. S. & Barto, A. G. (1998). *Reinforcement Learning*. Cambridge, MA: MIT Press.

Tulving, E (1983). *Elements of Episodic Memory*. Oxford: Clarendon Press.

Wang, Y. & Laird, J. E. (2006). *Integrating Semantic Memory into a Cognitive Architecture*. (Tech. Rep. CCA-TR-2006-02). Ann Arbor, MI: Center for Cognitive Architectures, University of Michigan.

Zilli, E. A. & Hasselmo, M. E. (2008). Modeling the Role of Working Memory and Episodic Memory in Behavioral Tasks. *Hippocampus. 18,* 193-209.