Proceedings of the Remembering Who We Are – Human Memory for Artificial Agents Symposium, Mei Yii Lim and Wan Ching Ho (Eds.), at the AISB 2010 convention, 29 March – 1 April 2010, De Montfort University, Leicester, UK

Comparing Forgetting Algorithms for Artificial Episodic Memory Systems

Andrew Nuxoll[1], Dan Tecuci[2], Wan Ching Ho[3] and Ningxuan Wang[1]

Abstract. Episodic memory is essential for human cognition and has also proven necessary for some intelligent agents. The size of an episodic store grows over time unless some forgetting mechanism is in place to keep it in check. In this research, we investigate the effect that different forgetting mechanisms have on the episodic memory performance. We compare three different forgetting algorithms using three distinct episodic memory architectures in a domain that is designed to highlight the differences between then. Our results show that the choice of forgetting algorithm has a significant impact on overall agent performance.

1 INTRODUCTION

Any intelligent agent must have a memory that it uses to store knowledge that it applies to a domain. At the most fundamental level this contains basic procedural knowledge of how to perform its function. Most agents also possess semantic knowledge, or knowledge of facts about the agent's domain.

Recently, there has been a growing interest in episodic knowledge or memory of specific past events [4, 5, 8, 13, 16, and 17]. Episodic memory is a history, or chronological record of specific events, called episodes. The ability to remember past events enables a system to improve its performance as well as its competence.

In this research, we examine and compare three simple forgetting algorithms for episodic memory:

- Forgetting a randomly selected memory in the episodic store.
- Forgetting the oldest memory in the episodic store.
- Forgetting the memory with a lowest activation value that is calculated based upon the frequency and recency of use.

We conducted our research using three distinct, general-purpose artificial episodic memory systems that have already been applied to a variety of different problems [8, 13, and 17]. Each of the nine unique combinations of system and algorithm has been applied to a domain that we have specifically designed to highlight the effectiveness of the agent.

Our results show that while the random and oldest forgetting algorithms yield similar behaviour, an activation-based forgetting algorithm can be more effective at selecting memories with lower utility and, thus, improving the agent's performance. Our results also show that agent performance does not decline linearly as the size of the episodic store decreases. Instead, there is a sudden decrease in performance when the size drops below a particular threshold.

Because we have performed our experiment on three different systems, we expect the results of the research to be indicative of the overall effectiveness of the forgetting algorithms. Furthermore, the test bed we have designed is well suited for future research into forgetting for episodic memory.

2 EPISODIC MEMORY

Our goal in building artificial episodic memory modules is not to mimic the human episodic memory, but rather to endow artificial agents with such episodic storage and recall strategies and to explore the additional capabilities they allow. Research in human episodic memory plays a guiding role in our endeavors.

An artificial episodic memory functions by storing events (or episodes) from the agent's past. Retrieval is performed based on a cue; the system presents the agent with episodes that are most similar to a given cue. These recalled episodes can help the agent make better decisions faster. For example, an agent might be able solve problems faster by adapting previous solutions. Additional tasks, such as avoiding unwanted behavior by detecting potential problems, monitoring long-term goals by remembering what subgoals have been achieved, and reflection on past actions, become feasible.

The size of the episodic store grows over time as the agent acquires more and more experiences. A large episodic memory raises a number of problems: it increases the search space of the retrieval algorithm thus negatively impacting retrieval time. An agent that indiscriminately stores all its past experiences might also run into external constraints related to the physical memory available to it.

While opportunities for using data compression to reduce the size of the episodic store exist, ultimately it will continue to grow. The only technique for keeping episodic store below a fixed size is for the agent to regularly forget portions of this knowledge. As a result, we anticipate that forgetting must be an essential part of maintaining a long-term episodic store for an intelligent agent.

A successful forgetting algorithm must fulfil two goals. First, the algorithm must remove knowledge that is least likely to be useful in the future. Logically, an agent that always forgets knowledge that will never be needed again performs equally well to an agent that does not forget anything. Second, the algorithm must be efficient. An algorithm that takes too long to select content to remove will affect the performance of the entire agent.

[1] Dept. of Electrical Engineering and Computer Science, Univ. of Portland, Portland, Oregon USA. Email: {nuxoll, wangn10}@up.edu.
[2] Dept. of Computer Science, Univ. of Texas at Austin, Austin, TX, USA. Email: tecuci@cs.utexas.edu.
[3] School of Computer Science, Univ. of Hertfordshire, AL10 9AB, UK. Email: w.c.ho@herts.ac.uk.

Forgetting is particularly important for general-purpose episodic memory systems. These systems do not have domain-specific knowledge that they can use to identify what is important about a given episode. Without this knowledge, the entire content of the episode is potentially important and must be stored by default. As a result, the episodic store grows very quickly.

3 RELATED WORK

The distinction between procedural, semantic and episodic memory is well established in human psychology. Endel Tulving first described the distinction between episodic and semantic memory [19]. Further research in cognitive neuroscience has attempted to define the distinction between the two memory types [3]. One thing is clear; the behaviour of human amnesiacs demonstrates that a lack of episodic memory is a severe handicap to cognition.

It is readily apparent that most humans routinely forget details of their experiences [6]. Whether this is a result of actual loss of information or inability to retrieve it is uncertain. Empirical studies show that human memory performance declines with time or intervening events. This decay is well described by a power function - called the 'power-law of forgetting' [2, 20, 21, and 22]. The decay of memory performance can be attributed to two factors: decay of unused information or interference of new and old information. Tulving [19] argues that newer memories prevent the recall of older ones. Evidence suggests that human memory actively constructs and stores only abstract and general representations (also called schemata) of these experiences [1], which may the source of human forgetting.

Previous research demonstrates that in the past, forgetting mechanisms were required in order to create effective intelligent agents. To date, we are aware of no prior work comparing forgetting algorithms.

The instance-base learning technique used in machine learning is a simple form of episodic memory system. In one implementation, Ram and Santamaría [15] forced the system to maintain a finite memory size, by forgetting the most redundant memory in their store.

McCallum [14] used a short-term episodic memory to distinguish between states in a partially observable environment. This agent would learn a proper maximum size for memory store and, thus, might decide to increase the memory store's size as needed to retain critical memories. When McCallum's implementation did forget, it would discard its oldest memory.

Kennedy and DeJong [10] used a utility-based metric for deciding which memory to discard. Their research also showed that forgetting can actually improve the performance of an agent since retrieval will be faster when searching a smaller episodic store.

Brom et al. [4] have conducted research to create human-like agents for virtual worlds. Because of their desire for realism, their system is based human cognition. Its episodic memory contains the most complex forgetting mechanism that we are aware of. This mechanism selects memories to remove based upon their agent and a measure of their "emotional saliency," an approach that is similar to the activation-based forgetting we use in this research.

While forgetting has already been established as a necessary part of some memory systems, our research aims to compare multiple forgetting mechanisms for their relative merit outside of the context of a specific architecture.

4 EPISODIC MEMORY SYSTEMS

In order to mitigate the architecture-dependent effects in our data, this research was conducted simultaneously on three different episodic memory systems. These systems are described below in brief.

4.1 Generic Memory for Events

Overview: Tecuci [17] investigates building a "generic memory module for events" – an episodic memory that functions separately from the system that is using it. In other words, it encapsulates the episodic functionality as a way to reduce the complexity of the overall system. It focuses on the generic aspects of memory organization and retrieval in isolation of a specific environment and task. This system defines a representation of generic episodes. It uses a multi-layer indexing scheme and has a generic API to external systems.

Application: This memory module has been successfully applied to tasks such as planning, plan recognition and question answering.

Forgetting: The system leaves the decision of when to forget an episode up to the external system.

4.2 Computational Autobiographic Memory

Overview: Ho et al [8] have aims at modelling computationally the psychological concept of autobiographic memory (which is a specific kind of episodic memory). The design of the memory system focuses on identifying the significance of events through measuring the impact they have on agent's synthetic states (e.g. emotions). Therefore, agents with autobiographic memory can remember and recall significant events that originate in their own experiences in order to adapt to the environment and smooth the interaction with a human user in software applications.

Application: Computational autobiographic memory has been integrated into various agent architectures for different research domains, including Artificial Life and synthetic narrative agents [10].

Forgetting: To date, these agents remember every single episode in their memory and no forgetting algorithm has been introduced to the architecture to improve the efficiency of memory storage and retrieval.

4.2 Soar Episodic Memory

Overview: The episodic system created by Nuxoll and Laird [13] is integrated into the Soar cognitive architecture [11]. This system creates episodes by recording a series of snapshots of the current state of the architectures working memory. A new episode is recorded whenever an agent takes an action in its environment. Episodes are retrieved when the agent defines a memory cue in its working memory.

Application: Previous research with Soar's episodic memory system has focused on demonstrating a variety of cognitive capabilities that are granted to an agent by virtue of possessing an episodic memory. These demonstrations consist of performing a specific task that cannot be performed without a given capability.

Forgetting: Up until now, this system has lacked any sort of forgetting mechanism.

These three systems are different implementations of the concept of episodic memory as they were created with strikingly different research goals in mind. However, they all have an important feature in common: they were designed as general-purpose, domain-independent systems. As a result, they are well suited for comparing the effects of various forgetting mechanisms in a mutual domain.

5 THE WAREHOUSE DOMAIN

The domain used for this research was carefully selected to meet the following criteria:

- An environment with a virtually unlimited number of unique possible states. If the number of possible states is limited, then it would be possible for an agent with a large enough episodic memory store to have an exact match for every state. As a result, the agent's performance would be capped in the long term.
- A domain that allows for virtually unlimited degrees of success as opposed to just success/fail. This allows us to compare degrees of success between agents that used different forgetting algorithms.
- A task for which an optimal solution is very difficult to find but that a better-than-random solution is easy to find. This is intended to verify that all our agents can learn to improve their behaviour in the domain but still be better or worse than other agents.
- An environment that is highly configurable so that it can be fine-tuned it to make differences between agents maximally apparent.
- Because we are connecting this domain to multiple different episodic memory systems, we need an environment with a simple interface that will not need to be changed if we adjust its configuration. Specifically, we want a domain with small number of commands and a simple, extensible input format.
- A domain that can run for an unlimited amount of time. In effect, it must have no goal state and yet always present a challenge to the agent.
- Finally, we want a domain with an obvious connection to a real-world problem and not a “toy” task.

The domain we designed in order to meet these criteria is called the Warehouse domain and is depicted graphically in Figure 1. In this domain, the agent is the manager of a warehouse full of various goods that can be shipped to different cities. Every day one truck is sent to each of these cities one at a time. At any point there is only one truck at the warehouse to be loaded.

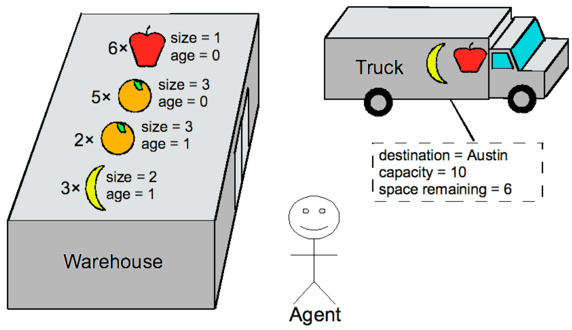

Figure 1. The Warehouse Domain

Each truck has a fixed capacity and therefore is able to hold more of some types of goods than of others. The agent's task is to decide what goods to put on each truck. A given truck could contain multiple different types of goods.

Whenever a truck is shipped to a city, the agent receives some amount of profit for the delivery. The amount of profit that the agent receives will vary based upon a subset of these factors:

- The base value of the goods. For example, gold is more valuable than manure.
- The amount of time that the goods sat in the warehouse before being shipped. In effect, each good has a perish rate that causes its value to change over time. For example, fresh fruit will rapidly lose value but non-perishable items will remain stable in value.
- The combination of goods that are shipped in each truck. Each unique, ordered pair of goods can have a good-to-good relationship that affects the value of one of the goods. For example, a shipment of fine china is worth a lot less if it is shipped in the same truck as livestock
- The relative worth of the goods in the destination city. Each unique combination of good and city can have a city-to-good relationship that describes how that city affects the value of the good. For example, snowshoes are of little value in Rio De Janeiro but would be invaluable in Reykjavik.

The simulation reaches the end of a day when one truck has been shipped to each city. At this point, new randomly selected goods are added to the warehouse's inventory for the next day. In this way, the warehouse never runs out of goods but the combination of goods is relatively unique.

This domain meets all of the requirements specified at the beginning of this section. It provides the agent with a constant stream of unique states for which an optimal action is difficult to select but a better-than-random action is easy to select. Furthermore, the interface, which described in detail in the next section, is simple yet highly configurable.

While we have attached a meaningful veneer to this domain, it is essentially a combination of the knapsack problem with a scheduling problem [7]. These are both NP-Hard problems that regularly appear in real world software applications.

6 WAREHOUSE CONFIGURATIONS

This section describes in detail how we interfaced the warehouse domain with our episodic memory systems and how we configured the warehouse domain for our experiments.

6.1 Input

The following information is provided the agent at each time step:

- A complete list of all the goods that are in the warehouse. The following properties of each good are visible to the agent: name, quantity, age, size and a unique identifier.
- The truck's overall capacity, how much space is currently available on the truck and the truck's destination city.
- A complete list of all the goods that are in the truck. For each good in the truck, the same info is provided that is provided for the goods that are still in the warehouse.
- If the last action the agent took was to ship the truck then the agent is provided with a reward signal indicating the overall profit from that shipment.

It is important to note that no information is provided to the agent about the value of these goods and how they change based on various circumstances. Furthermore, the agent does not know the significance of the properties of the truck and the goods aside from the fact that it cannot load a good onto a truck for which there is insufficient space. In other words, no heuristic knowledge is being provided to the agent about the domain.

6.2 Output

The agent can perform one of the following actions at each cycle:

- *Load one good onto the truck.* Illegal load commands (i.e., unknown good or insufficient space on the truck) are ignored.
- *Ship the truck.* This sends the truck to its destination city. At this point the agent receives a reward signal and a new, empty truck arrives that has the next destination city.

6.3 Relationships

For our experiment we began with a particular configuration of the warehouse domain that allowed an episodic memory system to consistently provide a significant improvement over random behaviour but could not come close to optimal behaviour. As a result, there was plenty of room for the agent's performance to improve or degrade as we varied the configuration of our systems.

Specifically, we began with three goods (apple, orange, and banana) and three cities (Austin, Portland, and Shanghai). The names of these goods and cities are irrelevant to the task but helpful for differentiating the results we present later in this paper.

The city-to-good relationships were defined as shown in Table 1. In this table, a given entry indicates the effect that the city in the column has upon the good in the row. Specifically, the value of the good was multiplied by the given ratio.

	Apple	Orange	Banana
Portland	1.0	4.0	3.0
Austin	6.0	6.0	1.0
Shanghai	2.0	1.0	4.0

Table 1. City-to-Good Relationships

The good-to-good relationships were defined as shown in Table 2. Each entry indicates the effect that the good in the row has upon the good in the column. As with city-to-good relationships, the value of the good was multiplied by the given ratio.

	Apple	Orange	Banana
Apple	1.0	3.0	3.0
Orange	3.0	1.0	1.0
Banana	1.0	1.0	1.0

Table 2. Good-to-Good Relationships

We measured the performance of each agent by computing the difference between each reward received and the optimal reward for an agent that maximizes its immediate profit.

7 AN EPISODIC MEMORY AGENT FOR THE WAREHOUSE DOMAIN

To compare our forgetting mechanisms we built an agent using each episodic memory system. All agents used the following algorithm:

1. The agent creates a set of all possible actions it can take in the current state. For example, the agent in Figure 1 could load one apple, load one orange or either age=0 or age=1, load one banana or ship the truck as it is currently loaded.
2. Since the agent does not have sufficient semantic knowledge to immediately decide which action to take, it evaluates each possible action using its episodic memory. It does this as follows:
   a. The agent creates a memory cue based upon its current state and the to-be-evaluated action that it is taking in that state. The specific content of the cue varies from system to system but the intent is the same: The agent is cueing for an episode wherein it took the candidate action in an identical state.
   b. An exact match to the cue usually does not exist, so a closest partial match is retrieved by the episodic memory system. It is notable that the three systems we used for this research each have different algorithms for determining which episode is the closest match.
   c. The agent examines the outcome of retrieved memory. In particular, it looks for the reward it received following that action. This reward is used as the evaluation score of the action it is considering in the present.
3. Once all the possible actions have been examined, the agent takes the action that has been assigned the highest evaluation score.
4. In the episodic memory systems created by Tecuci [17] and Ho, et al. [8] episodes span the entire loading of the truck. As a result, the best matching memory selected in step 3 is used to make all subsequent decisions by the agent until the truck has been shipped. In the episodic memory system created by Nuxoll and Laird [13] the episodes are recorded

each time the agent take any action (e.g., load or ship), as a result, steps 1-3 are repeated for each agent action. The impact of this difference is apparent in our experimental results.

8 EXPERIMENTS

We collected our data by performing a series of runs that persisted for 1000 virtual days in the warehouse environment. The agent began each run with an empty episodic memory and gathered episodic knowledge as the run progressed.
We varied three variables for each run:

- Episodic memory system – This was one of the three systems described in section 4.
- Memory limit – This was the maximum size of the episodic store. The episodic memory limits were selected such that they were spread evenly across a range from less than 1% of amount of data that could be recorded up to two-thirds of the maximum amount of data that could be recorded. The Tecuci [17] and Ho, et al. [8] systems recorded exactly 3000 episodes over the course of 1000 days, or one per shipment. For these systems, episodic memory limits were set at 10, 100, 200, 600, 1000, and 2000. The Soar episodic memory system records episodes more frequently, so its limits were prorated accordingly. In both cases, the limits were sufficient for the agent to completely remember the events over approximately 3, 33, 67, 200, 333 and 667 days respectively.
- forgetting mechanism – This was one of three forgetting algorithms that was used to decide what episode to remove from the episodic store when necessary:
  - *Random* - a random memory was selected to remove.
  - *Oldest* - the oldest memory in the store was removed.
  - *Activation* - each memory had an activation value that reflected the frequency and recency with which it had been retrieved by the agent. When it is time to remove an episode, the one with the smallest activation value was removed.

    Note: The activation is calculated using the same algorithm as working memory activation in the Soar and ACT-R cognitive architectures [12]. When an episode is created and each time it is retrieved, its activation level is increased by a fixed amount. Over time, however, the activation level decreases exponentially. Specifically, the overall activation level is calculated using this formula:

$$A = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right)$$

    t is a vector of the times when the episode was created or retrieved measured as the number of cycles that have elapsed since that time. *d* is a constant that determines the decay rate of the activation level. For our research, this value was fixed at 0.8 which is the default value when this formula is used for memory decay [12].

For each particular configuration of the three variables, we performed a total of ten runs and averaged them. A typical result is shown in Figure 3 below. The x-axis of the figure measures the elapsed time as measured in virtual days. The y-axis of the figure is the difference from the optimal reward that was achieved by the agent. Since the performance is measured relative to optimal behaviour, smaller values are better. A trend line has been added to the graph to show that the agent's performance is improving slightly.

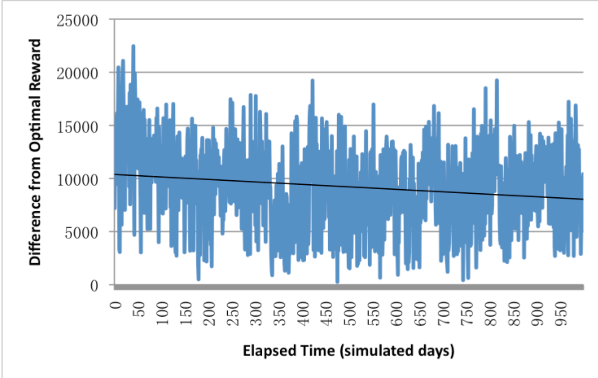

Figure 3. A typical result from one configuration

In this example, the improvement is very modest due to a small episodic memory size. This episodic learning was evident in all the data we collected and became more pronounced as the maximum size of the episodic store was increased.

Throughout this experiment, we gathered a total of 189 results like the one depicted in Figure 3 (or 1890 data runs). To keep the results presented here concise, we have chosen to focus on the results for just one city in our simulation (Austin). The results from the other cities are very comparable.

Figure 4 depicts a compilation of all the results for the computational autobiographical memory created by Ho, et.al. [8]. The horizontal (category) axis depicts the memory limits that were set. The height of each column indicates the average performance of the agent over the course of 10 distinct runs. As before, lower performance numbers are better. The three colors correspond to the three different forgetting mechanisms that were used.

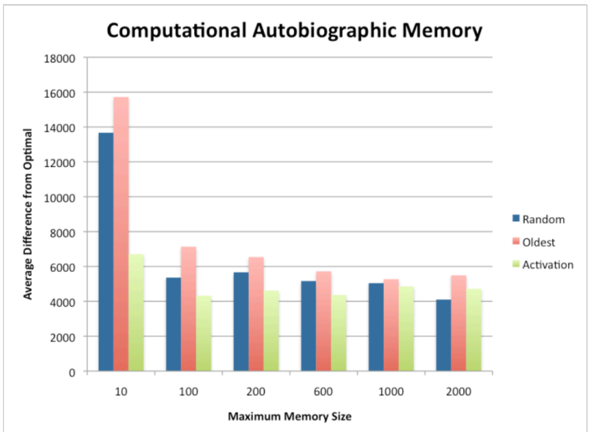

Figure 4. Average agent performance in the city of Austin for each selected memory limit using the computational autobiographical memory system created by Ho, et.al [8].

As the graph depicts, overall performance improves as the memory limit increases. For the random and oldest episode, performance declines sharply at the smallest memory limit.

However, activation-based forgetting maintains much better performance.

Figure 5 depicts the results of the same experiment performed on the episodic memory system created by Tecuci [17]. As shown in the figure, these results are generally comparable to those in Figure 4. The only difference is for a memory size of 10, when activation forgetting performs worse than the other two. Further investigation revealed that this is not due to the forgetting algorithm but the way the Generic Memory for Events performs retrieval. It uses a technique called surface similarity in which the most similar N episodes are retrieved and the best one is used to generate a new load decision. We used N=10, which means all episodes in memory are retrieved at every step. For activation forgetting that means they are all activated. This prevents any new episode (other than the original 10) from being stored in memory. When this N is much smaller than the maximum memory size, learning takes place and activation forgetting performs at least as well as random and oldest forgetting.

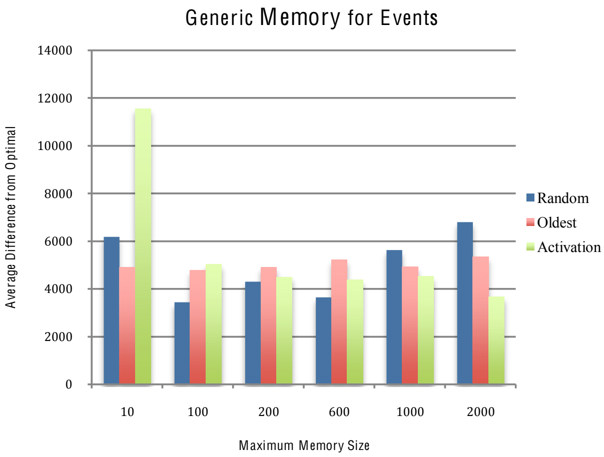

Figure 5. Average agent performance in the city of Austin for each selected memory limit using the generic memory for events created by Tecuci [17].

The results Soar episodic memory system created by Nuxoll and Laird [13] are depicted in Figure 6. These results are somewhat different due to the fact that episodes are recorded more frequently and only one decision is made per episode. However, the general result is the same. As the maximum size of the episodic store decreases, the activation-based forgetting algorithm yields an agent that maintains its performance better than the other two algorithms. Noticeably with the Soar episodic system this effect happens early (when memory size equals 3000) in comparing with the computational autobiographic memory (Figure 4) where it happens when memory size equals 10.

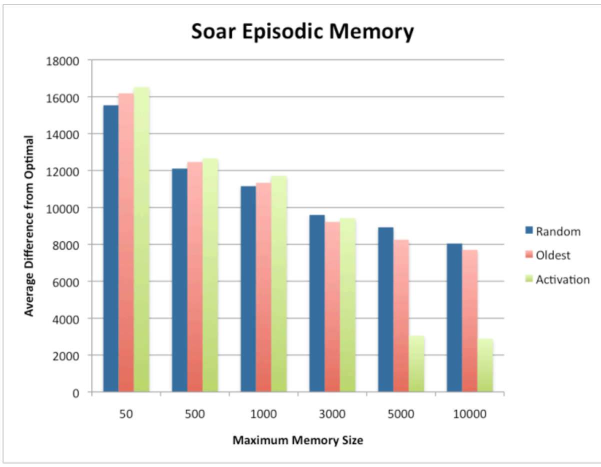

Figure 6. Average agent performance in the city of Austin for each selected memory limit using the Soar episodic memory system created by Nuxoll and Laird [13].

9 CONCLUSIONS AND FUTURE WORK

Current research suggests that forgetting will remain an important component of any episodic memory system that expects to operate over an extended period of time. Without it, the memory store will continue to grow until it exceeds the available storage space.

This paper presents the results of an experiment to study the effects of three forgetting algorithms on agent performance. To make our results as general as possible, the experiment was performed simultaneously on three different, general episodic memory systems. The experiment was also performed using a domain designed to highlight differences between agents.

Some important observations can be gleaned from our results. First, the effects of the memory limits are apparent in all three systems despite significant differences between them. This reinforces the notion that forgetting is an important component of any episodic memory system.

Second, when a forgetting mechanism is put in place the performance of the agent does not decline linearly with the size of the memory store. Instead, performance drops exponentially once the memory limit's size exceeds some threshold. This suggests that the memory limit setting for an episodic memory system is of particular importance. Since the performance of our agents was directly dependent upon the quality of the retrieved episodes, this also suggests that a given domain has a certain minimum number of episodes required to inform effective performance.

Third, the threshold at which performance makes its precipitous drop is partially dependent upon the forgetting algorithm. This suggests that seeking more effective forgetting algorithms is fertile ground for future research.

One factor to consider when studying forgetting mechanisms is the trade-off between the computational cost associated with forgetting and the system's task performance. Selecting the least activated episode to forget yields improved performance, but this comes at the cost of computing that activation. On the other hand, random or oldest forgetting are computationally cheap, but do not offer the same task performance benefits. We plan on investigating this trade-off in the future.

For the future, some approaches to forgetting that show promise are:

- Removing redundant episodes – The forgetting subsystem would locate the two episodes that are most similar and remove one of them. Ram & Santamaría [15] used an approach similar to this in their continuous case-based reasoning work. The time required to locate the redundant memories may be prohibitive, although this might be combined with retrieval.
- Combining redundant episodes – As above, the forgetting subsystem would identify redundant memories. However, instead of removing them it would separate out the redundant portion and save only one copy of that portion. In effect, each episode would consist of a base episode and a small set of changes. This approach bears some similarity to the creation of schemata described by Alba and Hasher [1].
- Memory Decay – Rather than remove entire episodes, the forgetting subsystem would remove individual features of the episodes. To do this effectively, an algorithm would still be required to select features to remove. An extension of the activation mechanism used for this research appears to be a good candidate for this but may be intractable. In addition, any episodic memory system with a memory decay mechanism would require the agents that used it to be able to process incomplete memories.

REFERENCES

[1] J.W. Alba and L. Hasher, Is memory schematic? *Psychological Bulletin 93*, 203–231 (1983).

[2] J. R. Anderson. *Cognitive Psychology and Its Implications*. Worth Publishing, New York, Fifth edition, (2000).

[3] A. Baddeley. The Concept of Episodic Memory. *Episodic Memory: New Directions in Research,* Oxford University Press, Oxford, UK., (2002)

[4] C. Brom and Jiří Lukavský., Towards More Human-Like Episodic Memory for More Human-Like Agents. In: *Proceedings of the 9th International Conference on Intelligent Virtual Agents*, Amsterdam, The Netherlands (2009).

[5] W. Dodd. The Design of Procedural, Semantic, and Episodic Memory Systems for a Cognitive Robot. Master's thesis, Vanderbilt University, (2005).

[6] H. Ebbinghaus. *Memory: A Contribution to Experimental Psychology*, New York: Teachers College, Columbia University, (1885/1913)

[7] M.R. Garey and D.S. Johnson, Computer and Intractability: A guide to the Theory of NP-Completeness, W.H. Freeman and Co., New York, USA pp. 5, 241 (1979).

[8] W.C. Ho, K. Dautenhahn, and C.L. Nehaniv, Computational memory architectures for autobiographic agents interacting in a complex virtual environment: a working model, *Connection Science, 20:1*, 21 – 65 (2008).

[9] W.C. Ho and K. Dautenhahn, Towards a Narrative Mind: Creating Coherent Life Stories for Believable Agents, *Proceeding of Intelligent Virtual Agent (IVA) 2008*, 59-72 (2008)

[10] W.G. Kennedy and K.A. De Jong. Characteristics of Long-Term Learning in Soar and its Application to the Utility Problem, In: *Proceedings of the Twentieth International Conference on Machine Learning*, pp 337-344., AAAI Press, Washington, DC, USA (2003).

[11] A. Newell. *Unified Theories of Cognition*, Harvard University Press, Cambridge, Mass, USA (1990).

[12] A. M. Nuxoll, J.E. Laird and M. R. James, Comprehensive Working Memory Activation in Soar. In: *Proceedings of the International Conference on Cognitive Modeling (ICCM-04).* Pittsburgh, PA, USA (2004).

[13] A. M. Nuxoll and J. E. Laird, Extending Cognitive Architecture with Episodic Memory, In: *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-07)*, Vancouver, B.C., Canada (2009).

[14] A.R. McCallum, Instance-Based State Identification for Reinforcement Learning, *Advances in Neural Information Processing Systems (NIPS 7).* (1995).

[15] A. Ram & J.C. Santamaría. Continuous Case-Based Reasoning. *Artificial Intelligence*, 90(1-2):25-77. (1997).

[16] M. Sànchez-Marrè, U. Cortés, M. Martínez, J. Comas, and I. Rodríguez-Roda. An Approach for Temporal Case-Based Reasoning:Episode-Based Reasoning. In H. Munoz-Avila and F. Ricci, editors, *ICCBR*, volume 3620 of Lecture Notes in Computer Science, pages 465–476. Springer, (2005).

[17] D. Tecuci. A Generic Memory Module for Events. *PhD Dissertation*, The University of Texas at Austin, (2007)

[18] D. Tecuci, B. Porter. "A Generic Memory Module for Events". In Proceedings to *The Twentieth International FLAIRS Conference,* (2007).

[19] E. Tulving. *Elements of Episodic Memory*. Clarendon Press, Oxford, UK., (1983).

[20] J. T. Wixted. Analyzing the empirical course of forgetting. *Journal of Experimental Psychology: Learning, Memory & Cognition*, 16:927–935, (1990).

[21] J. T. Wixted and E. B. Ebbesen. On the Form of Forgetting. *Psychological Science*:2:409–415, (1991).

[22] J. T.Wixted and E. B. Ebbesen. Genuine Power Curves in Forgetting: A Quantitative Analysis of Individual Subject Forgetting Functions. *Memory & Cognition*, 25:731–739, (1997).