Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Functional Interactions between Memory and Recognition Judgments

Justin Li and **Nate Derbinsky** and **John Laird**
University of Michigan
2260 Hayward Street
Ann Arbor, MI 48109-2121
{justinnh, nlderbin, laird}@umich.edu

Abstract

One issue facing agents that accumulate large bodies of knowledge is determining whether they have knowledge that is relevant to its current goals. Performing comprehensive searches of long-term memory in every situation can be computationally expensive and disruptive to task reasoning. In this paper, we demonstrate that the recognition judgment — a heuristic for whether memory structures have been previously perceived — can serve as a low-cost indicator of the existence of potentially relevant knowledge. We present an approach for computing both context-dependent and context-independent recognition judgments using processes and data shared with declarative memories. We then describe an initial, efficient implementation in the Soar cognitive architecture and evaluate our system in a word sense disambiguation task, showing that it reduces the number of memory searches without degrading agent performance.

Introduction

An agent that wishes to act intelligently in its environment must accumulate a large body of knowledge over its lifetime. Performing a knowledge search (Newell 1990) is often necessary to progress in problem solving, but the agent must often deliberately search its long-term memory for the small portion of relevant knowledge. Despite much work on creating efficient memory retrieval algorithms, it remains the case that memory retrievals are expensive relative to other reasoning. Moreover, the agent may pay this high computational cost without return if there is no relevant knowledge to be retrieved. In order for agents to make informed choices as to whether resources should be allocated for retrievals, it is beneficial to have a reliable, inexpensive indicator for the existence of relevant knowledge in memory.

One such indicator is the recognition judgment, which signals the agent if an object, a situation, or an event has been encountered before (Mandler 1980). Imagine, for example, being on a bus and realizing that you have seen one of the faces before; after some thinking, you remember that it is the butcher from the market. This butcher-on-the-bus example demonstrates how recognition is common and useful, in this case contributing to social relationships by signaling the presence of an acquaintance.

In this paper, we demonstrate that the recognition of features in the current situation serves as an indication that there is relevant knowledge to bring to bear. Furthermore, recognition judgments can be obtained with little computational cost by leveraging the existing processes and data of the agent's memory systems. This allows agents to quickly decide whether there is utility in searching memory, reducing the likelihood that the resources used in the retrieval will be wasted.

To evaluate this work, we implemented both context-dependent and context-independent recognition judgments in the Soar cognitive architecture (Laird 2012). The system is then tested on a word sense disambiguation task over the SemCor corpus (Miller et al. 1993). We show that for this task, automatic recognition judgments do not present a computational burden and are predictive of the state of memory. The result is a reduction in the number of memory retrievals needed while maintaining the agent's performance on the task.

In the following sections, we first review prior work on recognition and its interactions with memory, from both psychological and computational standpoints. We then define the desirable properties of recognition, introduce an approach for recognition, and detail its implementation in Soar. Finally, we present results from a formulation of the word sense disambiguation task, showing that agents with and without recognition perform comparably while the latter makes fewer searches of memory.

Related Work

The field of metamemory in psychology includes the study of how humans know whether knowledge exists in memory. This signal is called the feeling of knowing (Koriat 1998), of which recognition plays a part. Although there are accounts of human involuntary memories — where knowledge from memory is retrieved without deliberation — feelings of knowing are mostly discussed in the context of human-initiated memory searches. As with other metamemory phenomena, recognition acts not only as a source of information to the agent, but may also guide the agent in its actions (Nelson and Narens 1990). It has been found that deliberate human memory retrievals are partially directed by feelings of knowing about the cue (Reder and Ritter 1992), such that cues leading to unsuccessful retrievals are filtered out (Burgess and Shallice 1996). However, research has mostly focused on other functional benefits of recognition, such as its use in the recognition heuristic (Goldstein and Gigerenzer 1999), which states that "if one of two objects is recognized and the other is not, then infer that the recognized object has the higher value with respect to the criterion." Both the recognition heuristic and the use of recognition to control memory retrievals aim to reduce the cognitive load of the agent, the former by ignoring additional information and the latter by not searching memory to conserve resources. We focus on the interaction between recognition and the memory system in this work, and draw inspiration from the use of recognition judgments in humans.

Although there has been no computational model that empirically demonstrates the functional benefits of recognition in memory retrievals, other computational models of recognition exist. Most relevant to this paper is a model of the recognition heuristic in the ACT-R cognitive architecture (Schooler and Hertwig 2005). In that work, recognition is simply taken to be the binary result of retrieval: a cue is considered recognized if the retrieval is successful and unrecognized otherwise. Although the model accurately predicted the results of the recognition heuristic in humans, it conflates the recognition and retrieval operations. The use of memory retrieval as a direct component of recognition begs the question of how a feeling of knowing may be generated prior to retrieval; such a recognition judgment could never be used to decide whether memory *should* be searched. To the best of our knowledge, there has been no empirical demonstration of the functional benefits of interactions between recognition and the memory system in artificial agents.

Defining Recognition

The term "recognition" and what underlying processes it entails are the subject of some debate within the psychology literature. For this functional evaluation of recognition, however, we take recognition to be a binary signal of whether a "subject" has been previously perceived. Although this definition stands in the intersection of most other definitions, an ambiguity remains as to the subject of recognition. In the literature, recognition has been applied to words, sentences, pictures, and even audio signals (Shepard 1967), with little discussion of how these structures are represented in memory. To simplify the discussion of recognition, this work focuses on the recognition of the smallest unit of representation available in an artificial agent, which we call a "feature." It seems reasonable to assume that the recognition of higher-level objects involves the recognition of features, but how this may be done is beyond of the scope of this work.

Despite the lack of consensus over the processes behind recognition, there are three agreed-upon characteristics of the recognition judgment in humans (Goldstein and Gigerenzer 1999):

Frugality: Recognition is cheap and fast, exacting little-to-no cognitive cost and remains independent of other reasoning.

Predictiveness: Recognition reliably reflects the (non-) existence of knowledge in memory; false negatives are negligible, although false positives are more common.

Automaticity: Recognition of currently perceived features does not require deliberate action by the agent, but occurs regardless of its potential future use.

These characteristics of recognition are crucial to their use in deciding whether to retrieve from memory. Recognition, by virtue of it being frugal, becomes a suitable heuristic for the relatively expensive memory retrieval. For agents operating in the real world, this computational resource is measured in time — that is, recognizing the existence of knowledge in memory should be faster than attempting to retrieve it. An inexpensive recognition judgment minimizes the additional cost to retrieval; this is less important when the knowledge exists in memory (since a retrieval is necessary for the agent to reason over the information), but in the case where knowledge does *not* exist, the difference in cost between retrieval and recognition defines the utility of this approach. Another factor in the utility of recognition is whether it correctly predicts the state of memory; a noisy recognition judgment presents no advantage to the agent, as a retrieval may be necessary to verify the judgment. Finally, although a deliberate recognition judgment is possible, a sufficiently cheap automatic mechanism may allow the agent to eliminate the cost of deliberately iterating over features to judge their familiarity as well.

An Approach for Computing Recognition

In this section we introduce a general approach for obtaining recognition independent of the specific memory system, guided by the three constraints of frugality, predictiveness, and automaticity.

Recognition and declarative memories are both part of a larger memory ecosystem, in which many processes are under the constraints of efficiency. Indeed, the problem of efficient memory storage and retrieval has been of great concern to researchers (Douglass, Ball, and Rodgers 2009; Derbinsky, Laird, and Smith 2010). In order to create algorithms that scale well with the size of memory, auxiliary data structures are often used to maintain indices on the knowledge stored in memory. These indices must be updated upon the storage of new knowledge, thereby connecting knowledge-to-be-stored with knowledge-already-stored. This need for new memory elements to be "assimilated and integrated" into existing elements may be universal (Koriat, Goldsmith, and Pansky 2000).

At a high level, recognition similarly requires connecting new knowledge to what already exists in memory. For memory storage, new knowledge must be mapped to the indices into memory in order to update the latter; for recognition, this mapping must be done to determine whether current perceptions match what was previously stored. This similarity between recognition and memory storage suggests that there are benefits for these operations to share processes and data. More generally, we hypothesize that the indices used to create scalable algorithms for memory can be reused to make cost-effect recognition judgments, and our approach exploits this hypothesis. This approach is also makes progress towards two of the desired properties of recognition. First, the reuse of processes and data may allow recognition to take advantage of previous algorithms used to improve the efficiency of memory storage, keeping recognition frugal. Second, because the data for recognition directly reflects the contents of memory, it is likely to result in accurate judgments, subject only to distortions due to optimizations of the memory system.

This general approach to recognition judgments has two implications. First, since recognition is dictated by the data structures used by a particular memory system, the properties of the resulting recognition judgment will be affected by the properties of the underlying system. In particular, information discarded by the memory system could not be taken into account by the recognition judgment, since that information does not exist in any data structure. For example, applying this approach to a memory system that stores features independent of their original context will result in a context-independent recognition judgment. Although forcing recognition to take on properties of the memory system may seem like a limitation of this approach, the different properties of the resulting recognition judgments may be complementary. Just as a single agent may contain multiple specialized, dissociated memory systems, multiple recognition judgments may complement each other to provide more nuanced information to the agent.

Second, although this approach suggests a method for computing recognition, other variables in an implementation of recognition remain to be explored. One parameter thus far ignored is when the recognition judgment should be made. The similarity between recognition and storage offers an obvious choice of using the same trigger for recognition as for storage. There is, however, no reason why these two operations must coincide; one could imagine scenarios where recognition is useful even when no new knowledge is gained. The impact of different triggers on the utility of recognition depends on the domain, and an evaluation of such triggers remains to be done.

Although we have only implemented this approach to recognition within the memory systems of Soar, we note that the need for auxiliary memory structures is universal among cognitive architectures with the goal of scaling to large stores of knowledge. Thus, while the exact processes and data shared between recognition and memory may vary between architectures (and indeed between different memories within the same architecture), the general approach would still apply. As a functional approach to recognition, however, we do not make any claims as to whether this approach is psychologically plausible. Architectures with a focus on matching human data (e.g. ACT-R) may therefore take our work as a starting point for creating a recognition judgment that more closely matches human data.

Implementation

In the following section, we describe how we implemented this approach of shared processes between recognition and memory in the Soar cognitive architecture. We begin by giving an overview of Soar and its declarative memories, then describe the data that is being shared. The properties of the resulting recognition system are sketched, together with how the recognition judgment is presented to the agent and how different recognition judgments may be integrated.

The Soar Cognitive Architecture

This work is implemented using the semantic and episodic memories of the Soar cognitive architecture (Laird 2012). Soar is an architecture that has been used for developing intelligent agents that respond to their environments in real time. In Soar, agent state is contained in its symbolic short-term *working memory*, represented as a connected, directed graph. Working memory contains the agent's current goals and any knowledge the agent has brought to bear on the task. Reasoning proceeds in decision cycles, during which procedural knowledge, in the form of if-then rules, modifies the agent state. One such action may be the retrieval of knowledge from long-term memories, which contain other background knowledge the agent may need. Soar contains two long-term declarative memories: *semantic memory* (SMem), which stores decontextualized facts about the world, and *episodic memory* (EpMem), which stores the contextualized experiences of the agent. The agent does not have direct access to knowledge stored in either memory; instead, these memories are accessed through buffers in working memory. To retrieve a fact or an episode, the agent constructs a *cue* — an acyclic graph of features that describes the desired knowledge — in a buffer. The specified memory system then retrieves the fact or episode that "best" matches the description and recreates it in the buffer, where the agent can reason over the retrieved knowledge.

Since the semantic and episodic memories in Soar are specialized for different types of knowledge, the underlying processes and data kept by each are very different. For semantic memory, efficient retrievals require the tracking of commonly occurring features; these counters are then updated upon memory storage to reflect the features of the new memory element. The agent is assumed to know what knowledge may be useful in the future, and semantic memory therefore requires agent deliberation for knowledge be stored. In contrast, the storage of knowledge into episodic memory is automatic; at specific intervals, episodic memory captures the entire working memory state of the agent. In order to search through episodes efficiently, the memory system tracks the historically distinct structures that have appeared and the temporal intervals in which they existed in working memory. On storage, the intervals for removed structures are concluded, while the addition of new structures results in new intervals being tracked.

Recognition in Soar

Although recognition has not been a core part of Soar theory, we believe recognition may functionally benefit the architecture. In Soar, the features on which the recognition judgment operates are the edges in its graph structure; this is the data into which memory systems index and which the recognition judgment reuses. In semantic memory, recognition uses the counters for features to determine whether a feature has been perceived before, while in episodic memory the indices to historically distinct graph structures — some of which correspond to features — are used.

Since the semantic and episodic memories were designed to store different types of knowledge, the resulting recognition judgments have different properties. The recognition judgment for semantic memory operates on the features of objects currently perceived, and returns a "recognized" judgment if the counter for that feature is above a threshold, in this work set to zero for simplicity. As with semantic memory, this recognition judgment is decontextualized; as long as the particular feature has been perceived before (and stored in semantic memory), a "recognized" judgment is returned, regardless of the object or structure the feature originally described. For this reason, recognition for semantic memory cannot differentiate between instances of features in one object versus another.

Recognition for episodic memory operates with different properties. Since episodic memory is fully contextualized, this recognition judgment is sensitive to the object to which a feature belonged. By accessing only the indices into graph structures (and not the graph structures themselves), a recognition judgment does not need to perform expensive matching to determine whether a feature has been previously perceived in the same structure. Due to the importance of context, however, episodic recognition does not give a positive response to previously-perceived features if they are in new structures. For example, episodic memory would not recognize the butcher in the introductory example, as the context in which the feature appeared (the bus) does not match the context of its previous appearance (the market).

For both memories, the recognition judgment is automatically applied to all new features currently perceived by the agent. These results are represented to the agent in the respective buffers for the different memories. We make the simplifying assumption that both recognition judgments are triggered at the same time as episodic storage; this gives the agent a complete assessment of its working memory state automatically at specified intervals. In particular, since both recognition judgments refer to the same agent state, the agent can integrate the two signals via deliberate reasoning. An example of knowledge derivable from the two signals — which neither judgments could indicate on their own — is the perception of an old feature in a new context. Semantic recognition would judge the feature as old, but be unable to point out its new context, while episodic recognition would judge the feature as new but be unable to point out that the feature has been perceived before elsewhere. The full meaning of the different possible values of recognition for the two memories is shown in Table 1. The response of the recognition judgments to different properties of features allow the judgments to complement each other when combined.

	Not Recognized by EpMem	**Recognized by EpMem**
Not Recognized by SMem	new feature	old, un-stored feature
Recognized by SMem	old feature in new context	old feature in old context

Table 1: Facts inferable by combining the recognition judgments of both semantic and episodic memories.

Evaluation

To evaluate the frugality, predictiveness, and reduction of memory retrievals of our work, we tested our system in a word sense disambiguation task. Word sense disambiguation is well-suited as an evaluation domain as it involves the repeated perception of features (words) under different contexts. All experiments were run on a 2.8GHz Core 2 Extreme processor with 4GB of RAM.

Domain

Word sense disambiguation is an important problem in the field of natural language processing (Navigli 2009), as many words in English are polysemous (i.e. have two or more meanings). In this task, an agent is given a sentence and must determine which of several meanings a particular word carries. For example, the word "ran" has different meanings in the following sentences:

- He *ran* the race.
- He *ran* the code.

In a corpus of sentences, the correct meaning (or *sense*) of a word is indicated by an index into some dictionary. For this evaluation, we use the SemCor corpus (Miller et al. 1993), which contains words tagged with senses from WordNet (Miller 1995), the most-used dictionary for word sense disambiguation.

We adopt a formulation of the word sense disambiguation task that focuses not on natural language processing but on the functional benefits of recognition. Sentences from SemCor were converted into a parse tree using the RASP parser (Briscoe, Carroll, and Watson 2006). A small portion of sentences were parsed incorrectly and were discarded. These errors were due to a mismatch between the sense-tag data and the parse tree, where tagged phrases (e.g. "put down") were split into multiple leaf nodes in the parse tree despite representing a semantic unit. For each sense-tagged word in correctly-parsed sentences, the environment supplies the agent with the word to be disambiguated and the sentence parse tree, and the agent responds with a WordNet word sense corresponding to the word's meaning. The environment then gives feedback to the agent: the true sense of the word and whether the given sense was correct.

All correctly-parsed sentences in SemCor were used in this evaluation, presenting 91,046 words for the agent to disambiguate, 17,196 of which are distinct. There are therefore $91{,}046 - 17{,}196 = 73{,}850$ repeated exposures of individual words. Taking into account the syntactic location of a word, this task presents the agent with 61,973 distinct location-word pairs, meaning $91{,}046 - 61{,}973 = 29{,}073$ location-word pairs should be recognizable. Together, an oracle should achieve 70.9% performance if it began with no knowledge but could correctly identify the sense of a word once it has seen the pairing.

Agent Design

When the agent is presented with a parse tree and a word, the agent considers searching its episodic memory and semantic memory (in that order) for knowledge of the word's sense. For both memories, the retrieval only occurs if the word is recognized with respect to that memory. In episodic memory, the syntactic location of the word is used as its context; a more realistic definition of context would require a detailed linguistic representation of the sentence, which is outside the scope of this work. For both memories, if a word sense is found from retrieval, it is used as the response; otherwise, the agent responds with "don't know". After the environment provides the agent with feedback, the agent stores the correct word sense in both memories for future retrieval. All agents begin the task with no word sense knowledge and must accumulate this knowledge through the feedback given as it progresses.

We examine the benefits of recognition by looking at agents with their semantic and episodic memories lesioned to various degrees. For both memories, we define three conditions:

full (F) the agent uses its recognition judgment to decide whether to attempt a retrieval; only if the to-be-disambiguated word is recognized does the agent search memory.

retrieve (R) the agent always attempts a retrieval regardless of the results of its recognition judgment; this is equivalent to the agent recognizing all words.

disabled (D) the agent never retrieves from the memory; this is equivalent to the memory not containing relevant knowledge.

Comparing agents across the two memories in these three conditions allows us to separate the effects of recognition from retrieval and of one memory from the other. For ease of description, we refer to a particular agent by the conditions of its memories; an agent with **S**emantic memory on ***F**ull* and **E**pisodic memory on ***R**etrieve* would be the *SFER* agent.

This evaluation is designed to answer three questions about our approach to recognition and its benefits:

Frugality: How expensive is recognition in terms of computation time?

Predictiveness: How reliable is recognition? What are the false-positive and false-negative rates?

Decreased Reduction: How effective is recognition in reducing the number of retrievals from memory? How is task performance affected as a result?

The main hypothesis is that a recognition judgment that reuses processes and data from the memory system is frugal, predictive, and can reduce the number of retrievals from memory while correctly disambiguating the same number of words.

Frugality Results

To determine whether our approach to recognition incurs significant cost to the agent, we tested the *SDED* agent on the word sense disambiguation task, but limited the recognition judgment to a particular stage. In the *baseline* stage, the architecture does not compute recognition information at all, although both semantic and episodic memories are active and in use (for storing the correct word senses). In the *calculated* stage, the recognition judgment is calculated internally, but it is not represented to the agent. Finally, in the *represented* stage, the recognition judgment is calculated and represented to the agent as a structure in working memory. The separation of the costs of calculation and representation is due to previous work suggesting that the addition and removal of elements from working memory could be expensive; the difference between the two shows the cost of changing working memory. It should be noted that we are not committed to representing recognition judgments in working memory; this is simply a first step in exploring how recognition can be integrated with a general cognitive architecture. Also note that these stages are *different* from the conditions of memory defined above; here the difference is in whether the Soar architecture computes recognition, while the same *SDED* agent is used throughout. In each of these variations, we measure both the average and maximum time needed for a decision cycle. While the former measures the expected cost of recognition to the agent, the latter conveys whether recognition significantly delays other processes in the worst case.

	Avg. Time (msec)	Max Time (msec)
baseline	0.436	3.66
calculated	0.458	4.00
represented	1.09	8.67

Table 2: Average and maximum decision times for agents with and without recognition. The middle row shows the cost of the computation of recognition, without the cost of representing it to the agent. The differences between variations are significant ($p < 0.001$).

The results for this test are shown in Table 2. By comparing the first two rows, it can be seen that the calculation of the recognition judgment itself has very low costs, using only 5-9% more time than if recognition was not calculated. However, there is a high cost associated with conveying recognition information to the agent, more than doubling the time required. Further experiments showed that the additional costs were incurred when the agent accesses the recognition judgments; the number of unrecognized features led to the creation of hyperedges, which are expensive in rule matching. These results suggest that although this approach to recognition is efficient, care must be taken in how this information is represented to the agent.

Predictiveness Results

To develop a good understanding of the correctness of recognition, we divide this evaluation into false-positive and false-negative rates. As these statistics are often used to measure the performance of an agent on a task, it should be emphasized that these are *not* the rate at which the agents correctly disambiguate words; rather, it is the rate at which the recognition judgment correctly predicts whether retrievals will succeed or fail. To illustrate this point, consider attempts to disambiguate "ran" in these two sentences:

- He *ran* the race.
- He *ran* the code.

Since the word "ran" has the same lexical form and is in the same syntactic location in both sentences, both semantic and episodic memory would produce incorrect word senses when disambiguating the second sentence after the answer for the first is stored. However, both recognition judgments will correctly recognize the word "ran," and correctly predict that information will be successfully retrieved from memory, *despite the retrieved knowledge being incorrect*. Thus, the second sentence counts towards the true-positive rate of both recognition judgments.

False positives occur when a feature is recognized but does not exist in memory. Since Soar's memories guarantee completeness — that is, a searched-for memory is guaranteed to be returned if it exists — a false positive recognition judgment would cause the agent to attempt a retrieval and fail. To measure the false-positive rate of each recognition judgment, we compared the number of words recognized to the number of words successfully retrieved by the *SFED* and *SDEF* agents. Again, note that this metric is not concerned with the correctness of the retrieved word sense, only that the retrieval was successful. Whether the retrieved knowledge is correct depends on the retrieval biases of the memory system and has been explored in previous work (Derbinsky and Laird 2011).

In contrast with a false positive, a false negative occurs when a feature is not recognized but can in fact be retrieved from memory. This metric can be calculated by comparing the number of words an agent recognizes to the number of words that should be recognized. Note that the latter number is different for semantic and episodic memory, as they have different representations of knowledge.

For the word sense disambiguation task, both recognition judgments perfectly predicted the existence of knowledge in memory, leading to false-positive and false-negative rates of 0%. However, it should be noted that this result is highly dependent on the representation of knowledge; a previous iteration of the agents — which represented the agent state differently — had a false-positive rate as high as 38.6%. The errors stem from having a sentence-independent edge to the ambiguous word, which was interpreted as part of the syntactic location by the architecture. Since episodic memory recognition relies on the syntactic location of the word as context, the non-changing "context" led to undesirable matches in memory.

More generally, errors may occur in recognition when information about previous knowledge is not directly accessible, such that recognition cannot account for the discrepancy. Examples of changes in state that would lead to errors are shown in Figure 1. Although recognition in semantic memory works on single features, cues for retrieval may specify multiple features, all of which must exist in the retrieved memory element. The recognition of individual

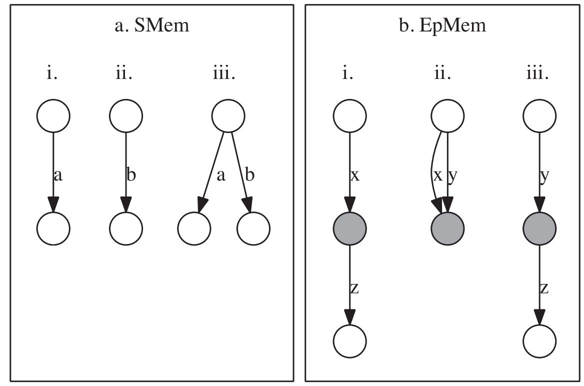

Figure 1: Scenarios in which SMem and EpMem would give false-positive recognition judgments. The individual graphs represent an agent state over time.

features in Figures 1.a.i and 1.a.ii will incorrectly predict the successful retrieval of the pattern in Figure 1.a.iii. For semantic memory, the co-occurrence of individually recognized features cannot be predicted with this approach. A similar error occurs in episodic memory. Optimizations in memory storage leads to the aliasing of the shaded node in Figure 1.b.ii; this causes a false recognition of the edge labeled z in Figure 1.b.iii, as it has been perceived under an alias in Figure 1.b.i.

These issues suggest that the design of both memory architectures as well as agents need to take into account how recognition operates. The space of tradeoffs between agent design, memory systems, and recognition judgments remains to be explored.

Decreased Retrievals Results

The third question to be addressed in this evaluation is whether taking the recognition judgment into account could reduce the number of potentially expensive retrievals from memory. We would like to understand whether reducing the number of retrievals with recognition would degrade the overall performance of the agent in the word sense disambiguation task, and if that is the case, which of the two types of recognition have a larger effect. For this evaluation, we looked at agents in each of the permutations of the conditions of the memory systems. The disabling of a particular memory shows the baseline contribution the other memory makes to the agent's performance.

The results are shown in Table 3; the top and bottom numbers represent the number of retrievals in EpMem and SMem respectively. Recognition for episodic memory (comparing the top number across the first two rows) reduced the number of retrievals from 91,046 to 29,073, for a 68.1% decrease. Recognition for semantic memory had a similar effect: both with and without episodic memory, the number of retrievals from semantic memory (comparing the bottom numbers across the first two columns) were reduced by at least 18.8%. This effect is smaller due to more words being recognized (since context was not taken into account).

Although the number of retrievals is greatly reduced,

	SMem *full (SF)*	SMem *retrieve (SR)*	SMem *disabled (SD)*
EpMem	29,073	29,073	29,073
full (EF)	44,776	61,972	0
EpMem	91,046	91,046	91,046
retrieve (ER)	44,776	61,972	0
EpMem	0	0	0
disabled (ED)	73,850	91,046	0

Table 3: The number of retrievals made by the memories. On top is the number of retrievals made by EpMem, on bottom is the number of retrievals made by SMem. Note that when EpMem isn't *disabled*, SMem only retrieves if EpMem fails.

this is not a desirable result if task performance drops correspondingly. However, this is not the case. Due to the perfect predictions of both recognition memories, there was no negative effect on the task performance of the agent. Both the *SFEF* and the *SRER* agents (top-left and middle-center in Table 3) correctly disambiguated 44.7% of the words, out of a possible 70.9% by the oracle. Overall, these results suggest that for this task, recognition is effective in filtering out retrievals that would fail, at no cost to task performance.

Discussion

This paper has demonstrated that recognition is useful in predicting whether or not memory retrievals will fail. In a word sense disambiguation task, we demonstrated that recognition can reduce the number of retrievals up to 60%, while suffering no decrease in task performance. Furthermore, the recognition judgment is frugal and predictive, although care must be taken to represent this information efficiently to the agent. These results provide evidence that tightly coupling recognition with the memory system is an effective and useful approach.

More generally, this paper has shown that metamemory phenomena may provide functional benefits to the agent. This suggests that there may exist a tradeoff between the resources spent accessing memory and the amount of information returned. Recognition is one extreme in this space, where a single bit of information is returned at little cost, while a comprehensive search of memory may return the full context of some knowledge at the cost of time. Whether there exists other points of interest on this continuum remains a topic for exploration; however, psychology literature suggests that tip-of-the-tongue states, where information is partially retrieved, may be one such option.

Acknowledgments

The authors acknowledge the funding support of the Office of Naval Research under grant number N00014-08-1-0099 and the Air Force Office of Scientific Research, contract FA2386-10-1-4127.

References

Briscoe, T.; Carroll, J.; and Watson, R. 2006. The second release of the RASP system. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, COLING-ACL '06, 77–80. Stroudsburg, PA, USA: Association for Computational Linguistics.

Burgess, P. W., and Shallice, T. 1996. Confabulation and the control of recollection. *Memory* 4(4):359–412.

Derbinsky, N., and Laird, J. E. 2011. A functional analysis of historical memory retrieval bias in the word sense disambiguation task. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI)*, 663–668.

Derbinsky, N.; Laird, J. E.; and Smith, B. 2010. Towards efficiently supporting large symbolic declarative memories. In *Proceedings of the 10th International Conference on Cognitive Modeling (ICCM)*, 49–54.

Douglass, S.; Ball, J.; and Rodgers, S. 2009. Large declarative memories in ACT-R. In *Proceedings of the 9th International Conference on Cognitive Modeling (ICCM)*.

Goldstein, D. G., and Gigerenzer, G. 1999. The recognition heuristic: How ignorance makes us smart. In Gigerenzer, G.; Todd, P. M.; and the ABC Research Group., eds., *Simple Heuristics That Make Us Smart*. Oxford University Press. 37–58.

Koriat, A.; Goldsmith, M.; and Pansky, A. 2000. Toward a psychology of memory accuracy. *Annual Review of Psychology* 51(1):481–537.

Koriat, A. 1998. Metamemory: The feeling of knowing and its vagaries. In Sabourin, M.; Craik, F. I. M.; and Robert, M., eds., *Advances in Psychological Science, Volume 2: Biological and Cognitive Aspects*. Psychology Press.

Laird, J. E. 2012. *The Soar Cognitive Architecture*. MIT Press.

Mandler, G. 1980. Recognizing: The judgment of previous occurrence. *Psychological Review* 87(3):252–271.

Miller, G. A.; Leacock, C.; Tengi, R.; and Bunker, R. T. 1993. A semantic concordance. In *Proceedings of the Workshop on Human Language Technology*, HLT '93, 303–308. Stroudsburg, PA, USA: Association for Computational Linguistics.

Miller, G. A. 1995. WordNet: A lexical database for english. *Commun. ACM* 38(11):39–41.

Navigli, R. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)* 41(2).

Nelson, T. O., and Narens, L. 1990. Metamemory: A theoretical framework and new findings. volume 26 of *Psychology of Learning and Motivation*. Academic Press. 125–173.

Newell, A. 1990. *Unified Theories of Cognition*. Harvard University Press.

Reder, L. M., and Ritter, F. E. 1992. What determines initial feeling of knowing? familiarity with question terms, not with the answer. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 18(3):435–451.

Schooler, L. J., and Hertwig, R. 2005. How forgetting aids heuristic inference. *Psychological Review* 112(3):610–628.

Shepard, R. N. 1967. Recognition memory for words, sentences, and pictures. *Journal of Verbal Learning and Verbal Behavior* 6(1):156–163.