From List Learning to Semantic Knowledge: Search and Learning of Associative Memory

Greg E. Cox (gcox@umd.edu)
University of Maryland, Department of Psychology,
Biology/Psychology Building, College Park, MD 20742 USA

J. Isaiah Harbison (isaiah.harbison@gmail.com)
University of Maryland, Department of Psychology,
Biology/Psychology Building, College Park, MD 20742 USA

Eddy J. Davelaar (e.davelaar@bbk.ac.uk)
School of Psychology, Birkbeck, University of London. Malet Street
WC1E 7HX, London, United Kingdom

Abstract
Meanings—semantics—matter in the study of cognition. Recent advances in computational modeling have demonstrated how semantic information can be incorporated into episodic or experiential free recall tasks to better account for participant performance. However, one limitation of current uses of semantic representations with the SAM model is a disconnect between how the semantic representations are created and the method of learning used by SAM. By extending the contextual representation used in SAM to incorporate both temporal drift and item-based drift, inspired by the Temporal Context Model (TCM), we demonstrated how the SAM learning mechanism can be used to create semantic representations.

Keywords: semantic memory; recall; false memories; SAM.

Introduction
How semantic information—meaning—is extracted from experience is one of the most fascinating questions in cognitive science. Recent research defining the meaning of words in terms of their context of appearance has had significant success. HAL (Hyperspace Analogue to Language; Lund & Burgess, 1996), for example, defines the context of a word by the words that appear immediately around it. In contrast, LSA (Latent Semantic Analysis; Landauer & Dumais, 1997) defines the context of a given word in terms of the documents in which the word appears. In both cases, the similarity of contexts of two words is used to predict semantic similarity. Despite the conceptual simplicity of these forms of learning, these models have proven quite robust in mirroring human performance on semantic tasks.

This success has been of special relevance to cognitive models that have previously been unable to account for the contribution of semantics. For example, in memory retrieval, semantic relationships matter. Semantically related items are likely to be retrieved together no matter the order in which the items were presented (Bousfield, 1953; Jenkins & Russell, 1952) and items that were not presented are often falsely recalled if they were strongly semantically associated with the other items presented (Deese, 1959; Roediger & McDermott, 1995).

By adding semantic representations generated by LSA or by the Word Association Space (WAS; Steyvers, Shiffrin, & Nelson 2004), Sirotin, Kimball, and Kahana (2005) extended the SAM (Search of Associative Memory; Raaijmakers & Shiffrin, 1981) model to create the eSAM model. eSAM was able to account for semantic effects in list retrieval. Kimball, Smith, and Kahana (2007) further extended SAM, creating fSAM, in order to account for false recall.

However, adding semantic representations from LSA or WAS is not completely satisfying in terms of parsimony. SAM has a mechanism of its own to learn associations and it is not clear how this learning mechanism relates to the types of representations created by LSA and WAS. The purpose of this paper is to explore whether the same learning mechanism used to learn associations between list items could also be used to create semantic representations. We combined the SAM learning mechanism with a contextual representation similar to that of the Temporal Context Model (TCM; Howard & Kahana, 2002) to create a model that is able to learn semantic representations with the same mechanism that it uses to learn lists. Our model also makes use of several developments in the treatment semantic information within SAM introduced in the eSAM and fSAM models. We show that the model, called SLAM (Search and Learning of Associative Memory), is capable of extracting semantic information from text and is able to predict certain semantic effects in free recall tasks.

Adding Semantic Learning to SAM
As mentioned above, SAM already contains a learning mechanism that allows for the learning of lists of items. In fact, SAM’s item-to-item association learning is quite similar to the method of learning used by HAL. Both methods increment associations between items that occur in
proximity to each other. In SAM, these are items that are still in short-term memory when a new item is presented while in HAL, these are items that are still within the moving item window. However, the resulting associations are used differently in HAL and SAM. Semantic information is extracted from HAL by comparing each item’s vector of associations to the vector of associations of other items. SAM does not make use of the similarity between association vectors and therefore, does not use this as a measure of semantic similarity.

Aside from the HAL-like, item-to-item association learning, SAM also learns associations between items and contexts, where context is normally defined by a single node specific to the list on which the item appears. This use of context is not unlike LSA’s definition of context, where an item is associated with the documents in which it occurs (the assortment of documents in which the item occurs becomes its context). However, this form of context has not been used within SAM to make judgments of the semantic similarity between items. SAM normally has only been applied to learning a few lists at a time. Furthermore, the lists on which the items occurred or co-occurred implied no deeper meaning.

Inspired by the relationship between SAM’s learning mechanism and what is used in successful models of semantic learning, we decided to further develop SAM’s learning mechanism by extending its use of context. Specifically, we made three modifications: First, instead of having only one context node active at a time, we allowed for several context nodes to be active simultaneously, providing more gradations of similarity between contexts. Second, the activation of the context nodes is determined in part by temporal drift, so that items that often occur near one another will have a similar context based on their temporal proximity alone. Finally, the activation of the context nodes is also a function of item-based drift. That is, the current context is adjusted as items are added to STM such that each item partially reinstates its previously learned context. This modified contextual representation, drawn in large part from the TCM, is shown in Figure 1.

With these modifications, the contextual representations of items that occur in similar situations—i.e., that occur simultaneously or that appear with similar items—become similar. For example, one almost never sees (or eats) both pasta and rice at the same time. Yet, both pasta and rice are found in meals as bases or sides for other food items. By virtue of the similar situations in which both pasta and rice occur, their contextual representations would also be similar, despite the fact that they never co-occur. This similarity between contexts allows SLAM to be sensitive to semantic similarity and is a novel addition to the SAM model.

The SLAM Model

SLAM specifies two processes, learning and recall. Aside from the use of context during learning, both learning and recall are based on the e- and f- extensions of the SAM model, and retain the same assumptions regarding episodic learning, e.g., that forward and backward association strengths are different. The main distinction between SLAM and the previous semantic extensions of SAM is in how the semantic representations are generated. eSAM and fSAM make use of LSA or WAS representations; SLAM learns semantic representations on its own.

In SLAM, the strength of association between two items in memory is directly proportional to the amount of time they are both simultaneously active in short-term memory (STM). Further, the strength of association between an item and a given context is directly proportional to the amount of time the item was active in STM within that context. Unlike previous modification of SAM, context is a function of the items currently active in STM, a small amount of random temporal drift, and the previous context.

At recall, input from both the current context and the items currently active in STM is used to retrieve a specific item from memory, thus making it active in STM. The computational details of these processes are specified below.

Computational Objects

SLAM posits two memory modules: a STM rehearsal buffer of size \( r \) and a long term memory (LTM) which stores the associations between items and the associations between items and contexts. Context is represented as a binary vector \( t \) of dimension \( N_C \). Thus, LTM is represented as two matrices, one which stores the episodic associations between items (call this matrix \( E \)) and one which stores the associations between items and each element of the context vector (call this matrix \( C \)).

Learning

When an item is presented to SLAM, it is added to the STM buffer. If the buffer is at capacity, an item is removed from the buffer before the new item is added. The probability that an item will be removed from STM is given by:
where $i$ is the item’s index in the STM buffer (older items have lower indices, which begin at 0), $r$ is the buffer size, and $q$ is a constant. This equation was introduced by Phillips, Shiffrin, and Atkinson (1967) and ensures that older items have a greater chance of being removed from STM. Novel items are initially assigned random sparse contextual associations. \(^1\) While the initial state need not be sparse, sparse initial states help to drive contextual drift and, subsequently, to better differentiate item contexts.

The current context is re-computed, taking input from the items currently in STM (including the most recently presented item), as well as the previous context. The first step in computing the new context is to allow for activation decay. The probability of any currently active context node deactivating is $\beta$, a free parameter.

In the second step, context nodes are probabilistically activated by the items active in STM. Input from the items currently in STM is given by the vector $x$, where

$$x_i = \prod_j^{N} \frac{C_{ji}}{\sum_k^N C_{jk}}$$

where $N$ is the number of items in STM and $C_{ji}$ is the association between item $j$ and context node $i$. Note that contextual input is normalized by dividing by the sum of each item’s contextual associations, thus putting all items “on an equal footing” with regard to influencing context (i.e., items will only help activate a context node to the extent that they are associated with that node and no others).

Third and finally, $x$ is combined with some amount of temporal drift, given by the vector $\phi$, where each $\phi$ is a random value drawn from a Gaussian distribution and restricted to $[0,1]$. The final input vector to context, $t^{|\text{IN}|}$, is thus given by

$$t^{|\text{IN}|}_i = \gamma(x_i / \max(x)) + (1 - \gamma)\phi_i,$$

where $\gamma$ is a free parameter determining the relative contribution of temporal drift and item-based drift, and the components of $x$ are normalized by dividing by the greatest component of $x$. Then, the probability that a given context node $i$ will become active is given by:

$$P(t_i = 1) = t^{|\text{IN}|}_i / \max(t^{|\text{IN}|}).$$

Note that dividing by the value of the greatest component of $t^{|\text{IN}|}$ ensures that at least one context node—the one receiving the greatest input—will be active. (Such nodes are those that are most consistent with all items in STM.)

After these three steps, the new context has been created and learning can commence. All items in STM (including the newly added item) have their association strengths to each other and to the currently active context nodes incremented. The association between two items, item $i$ and item $j$, is incremented by $b_i(d/N)$, where $b_1$ is a free parameter, $d$ is length of presentation, and $N$ is the number of items in STM, if $i$ was presented before $j$ (i.e., if $i$ has a lower index in the STM buffer than $j$) and by $b_2(d/N)$, where $b_2$ is another free parameter, if $i$ was presented after $j$. The associations between each item in STM and the currently active context nodes are incremented by $a(d/N)$, where $a$ is a free parameter.

**Recall**

The current context $t$ and the items currently in STM (if any) are treated as cues to probabilistically retrieve an item from memory. Recall consists of two processes: sampling and recovery. The probability that an item will be sampled from memory is proportional to the strength of its associations to the items in STM as well as its association to the currently active context nodes.

The strength of an item $i$’s association to a context $t$ is given by

$$\alpha_i^C = \left( \sum_j^N t_j \frac{C_{i}}{\sum_k^N C_{ik}} \right)^3.$$

Note that the item’s context is again normalized by dividing by the sum of the values of its contextual association vector, and that the final sum is cubed. Cubing has the effect of better differentiating weak from strong associates.

The strength of an item $i$’s association to the set of items in STM is given by

$$\alpha_i^E = \prod_j^N (E_{ji} + \nu),$$

where $N$ is the number of items in STM and $\nu$ is a small amount of Gaussian noise, restricted to $[0,\infty)$, reflecting the non-zero associations between items that may not have been presented together.

Contextual and episodic cue strengths are combined to form a composite activation:

$$\alpha_i = (\alpha_i^C)^{W_C} (\alpha_i^E)^{W_E},$$

where $W_C$ and $W_E$ are free parameters giving the relative weight of contextual and episodic associations, respectively. The probability that an item $i$ will be sampled is proportional to its activation relative to all $M$ items in memory, specifically

$$P(\text{sample item } i) = \alpha_i / \sum_j^M \alpha_j.$$

Once an item has been sampled from memory, it must be recovered before it can enter STM. The probability that a sampled item will be recovered is

$$P(\text{recover item } i) = 1 - \exp\left(-\alpha_i^E W_C - \alpha_i^E W_E\right).$$

An item cannot be recovered from memory if the same set of cues failed to recover the item before, nor if it has already been recovered. If an item is recovered, it’s forward episodic associations to the items in STM is incremented by

---

\(^1\) Initial context states were assigned by drawing the values of an item’s contextual association vector from a uniform random distribution over $[0,1]$ and taking these values to the fifth power.
the simplifying assumptions, also made in fSAM, that subject results reported in Simulation 2. We further made preliminary simulations designed to fit SLAM to the human effect. The remaining parameters were selected based on reported in this paper, temporal drift did not have a large versions of SAM; STM. If the STM buffer is at capacity, the item that has been in the buffer the longest is removed before the new item is added (this assumes that no maintenance rehearsal is taking place at recall). The current context is recomputed as above, given the new item in STM. Recall continues until a maximum number of retrieval failures is reached, \( K_{\text{MAX}} \). Additionally, if there are \( L_{\text{MAX}} \) consecutive retrieval failures, the items in STM are no longer used as cues; instead, only the context is used until the next successful recovery.

SLAM’s parameters, as well as their values in each of the simulations presented in this paper, are summarized in Table 1. The values of \( r \) and \( q \) were derived from previous versions of SAM; \( N_{\text{C}} \) and \( \beta \) were chosen arbitrarily to reduce the numbers of parameters allowed to vary; and \( \gamma \) was set to 1.0, since on the time scales of the simulations reported in this paper, temporal drift did not have a large effect. The remaining parameters were selected based on preliminary simulations designed to fit SLAM to the human subject results reported in Simulation 2. We further made the simplifying assumptions, also made in fSAM, that \( b_2 = .5b_1, f_2 = .5f_1 \), and \( L_{\text{MAX}} \) is a tenth of \( K_{\text{MAX}} \).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W_e )</td>
<td>Weight of episodic cue strength</td>
<td>0.95</td>
</tr>
<tr>
<td>( W_c )</td>
<td>Weight of contextual cue strength</td>
<td>3.6</td>
</tr>
<tr>
<td>( N_{\text{C}} )</td>
<td>Number of context nodes</td>
<td>50</td>
</tr>
<tr>
<td>( r )</td>
<td>Size of STM rehearsal buffer</td>
<td>4</td>
</tr>
<tr>
<td>( K_{\text{MAX}} )</td>
<td>Maximum number of retrieval failures before free recall stops</td>
<td>25</td>
</tr>
<tr>
<td>( L_{\text{MAX}} )</td>
<td>Maximum number of consecutive retrieval failures before context alone is used as a cue</td>
<td>3</td>
</tr>
<tr>
<td>( \beta )</td>
<td>The probability that any active context node will deactivate in a single update cycle</td>
<td>0.2</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>The amount of contextual drift due to the items in STM, as opposed to random noise</td>
<td>1.0</td>
</tr>
<tr>
<td>( q )</td>
<td>Fitting parameter for calculating the probability of an item’s removal from STM</td>
<td>0.266</td>
</tr>
<tr>
<td>( a )</td>
<td>Amount by which contextual associations are incremented in rehearsal</td>
<td>0.754</td>
</tr>
<tr>
<td>( b_1 )</td>
<td>Amount by which forward episodic associations are incremented in rehearsal</td>
<td>0.276</td>
</tr>
<tr>
<td>( b_2 )</td>
<td>Amount by which backward episodic associations are incremented in rehearsal</td>
<td>0.138</td>
</tr>
<tr>
<td>( f_1 )</td>
<td>Amount by which forward episodic associations are incremented at retrieval</td>
<td>0.431</td>
</tr>
<tr>
<td>( f_2 )</td>
<td>Amount by which backward episodic associations are incremented at retrieval</td>
<td>0.2155</td>
</tr>
</tbody>
</table>

\( f_1 \), and the backward associations by \( f_2 \), both free parameters. Then, the newly recovered item is added to STM. If the STM buffer is at capacity, the item that has been in the buffer the longest is removed before the new item is added (this assumes that no maintenance rehearsal is taking place at recall). The current context is recomputed as above, given the new item in STM. Recall continues until a maximum number of retrieval failures is reached, \( K_{\text{MAX}} \). Additionally, if there are \( L_{\text{MAX}} \) consecutive retrieval failures, the items in STM are no longer used as cues; instead, only the context is used until the next successful recovery.

**Simulation 1: Context and Semantic Similarity**

To demonstrate the use of context within SLAM and its capacity to learn the similarity between items that have not been presented together, we created a set of 24 categories, where each category contained a “label” and 15 members. Each category member was directly associated with its corresponding label, but was not directly associated with the other members of its category. The members and label of one category were not associated with any member or label of another category. We then created a training set of 8000 label-member word pairs randomly selected from this category structure, where each pair represented a new learning episode. That is, episodic associations formed in one episode did not carry over into any others, and only those words occurring in that episode were used to compute context.

We trained 500 instances of SLAM on these 8000 pairs. Then, each was given the label of a category as a retrieval cue, and the first recalled word was tallied. Because item-to-item episodic associations are not used in this recall task, only context can serve as a retrieval cue. Thus, successful recall of category members depends on the extent to which the context evoked by the category label is shared by the category members, and by little or no other words. On average, 99.4% of the recalled words were from the category with the given label, and all 15 category members appeared in the set of recalls for each cue.

Using the same methodology, we then used a randomly selected category member as the cue word. In this case, 85.8% of recalls were other words from within the cue’s category, 13.0% of recalls were the label of the cue word’s category, and the remaining recalled words were from other categories. Again, all of the remaining category members appeared in the set of recalls for each cue.

These free recall tasks show that SLAM has successfully learned the category structure on which it was trained. It is capable of forming robust context representations which reliably produce semantically valid recall. Further, those context representations enable words that were not

\(^2\) Note that while episodic associations are permitted to increase without bound, contextual associations can never exceed one. So, to balance them properly requires that \( W_c \) be high relative to \( W_e \).

\(^3\) This is identical to the free recall method used to generate the word association norms (Nelson, McEvoy, & Schreiber, 1999) from which WAS (Steyvers, et al., 2004) was derived.
Table 2: Data from Simulation 2

<table>
<thead>
<tr>
<th></th>
<th>Veridical recall</th>
<th>Critical item intrusions</th>
<th>Prior-list intrusions</th>
<th>Extra-list intrusions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLAM</td>
<td>.60 (SD=.064)</td>
<td>.52 (SD=.139)</td>
<td>.07 (SD=.21)</td>
<td>.29 (SD=.41)</td>
</tr>
<tr>
<td></td>
<td>(.59)</td>
<td>(.51)</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Human</td>
<td>.50 (Stadler, Roediger, &amp; McDermott, 1999)</td>
<td>.54</td>
<td>.01</td>
<td>.30</td>
</tr>
<tr>
<td></td>
<td>(.52)</td>
<td>(.07)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fSAM</td>
<td>.48 (Kimball, Bjork, 2002; fit of Kimball &amp; Bjork, 2002, above)</td>
<td>.54</td>
<td>.05</td>
<td>.43</td>
</tr>
</tbody>
</table>

presented together (i.e., words within the same category) to cue one another, thus capturing semantic relationships on which SLAM was not explicitly trained. Being able to form such indirect associations is necessary for a successful model of semantic learning.

**Simulation 2: False Recall**

In the Deese-Roediger-McDermott (DRM) paradigm (Deese, 1959; Roediger & McDermott, 1995), subjects are presented with lists of words. The words on each list are all associated with a critical word that is not presented to the subject. When subjects are asked to recall the lists on which they were trained, they reliably produce the critical word in addition to list words (for further examples, see Stadler, Roediger & McDermott, 1999 and Kimball & Bjork, 2002). The SAM model was originally designed to model list recall, and its most recent incarnation, fSAM (Kimball, Smith & Kahana, 2007), is explicitly designed to handle semantic effects in false recall in the DRM paradigm. Though it does an excellent job of this, fSAM invokes, in addition to SAM’s standard episodic and contextual associations, a fixed set of pre-experimental semantic associations to produce these semantic effects. We hoped that with SLAM’s more enriched context representation, SLAM would be able to produce such effects without an additional set of associations, given its prior experience with the words on the lists.

To test this, we used the same category structure as in Simulation 1. Twelve of these categories were chosen to create 12 DRM lists of 15 words each, with the category members as the list words and the category labels as the critical words. The remaining 12 categories served as potential distractors. We once again trained 500 SLAM instances on the word pairs used in Simulation 1. Each SLAM was then presented with the 12 DRM lists in the same pre-determined random order; it is during this process that episodic associations between list words are formed (however, no contextual learning occurs at this time). After each list presentation, the SLAM instance was given an immediate free recall task for the list items.

SLAM first recalls all of the items currently in STM before moving on to use both episodic item-to-item associations and contextual associations to cue recall. In SAM, the list context is treated as a single node to which all the words presented as part of that list become associated. SLAM, in contrast, is able to use its contextual drift mechanisms to build up the list context from the items presented in that list. It is this resulting list context—in addition to episodic, item-to-item associations formed during list learning—that is used to cue recall. Note that recalled items are also capable of influencing the current context via SLAM’s contextual drift mechanism.

The results of this simulation are given in Table 2, as well as data from DRM experiments with human subjects and from DRM simulations with fSAM (Kimball, et al., 2007). SLAM’s performance is similar to that of human subjects on all counts, though it more closely resembles the data from Stadler, et al., (1999) with regards to the proportion of veridical recall from each list. In this task, SLAM differs from human performance chiefly in the fact that critical item intrusions occur much earlier with mean output percentile 35.3 (SD=10.9), as compared to human performance with M=66 (Kimball & Bjork, 2002). This is likely due to the simplicity of the semantics on which SLAM was trained—the only semantically related word that might intrude was the critical word. Additional semantically related words (i.e., additional category members that are not presented on the list) would delay critical item intrusions by both intruding themselves, and by interfering with the critical item, preventing any of them from being recalled (and perhaps promoting episodically associated items for recall, as opposed to purely contextually associated ones).

The results of this simulation suggest that, beyond being able to form associations between semantically related items through their contextual representations, SLAM is capable of combining such associations with item-to-item episodic associations to produce performance that approximates that of humans. Further, it shows that SLAM is capable of using its learned contextual representations to dynamically construct its own momentary context from the items with which it has been presented. This ability is key to both learning and recall, by being able to evoke previous contexts and thereby preferentially activate those words associated with that context—i.e., semantically related words.

**General Discussion**

Models such as LSA and HAL demonstrate how far the context of an item’s occurrence can go in defining the meaning of that item. The incorporation of these semantic representations into SAM by eSAM and fSAM has demonstrated how semantic representations provided by LSA and WAS can be used to account for the effects of semantic information in memory recall. With SLAM, we
hope to demonstrate how people might learn contextual representations. Furthermore we want to explore the relationship between learning semantic representations and learning on a smaller scale, such as that which occurs in experiments found within the free recall paradigm.

The results from Simulations 1 and 2 suggest that the same mechanism used to learn lists within the free recall paradigm could also be used to learn semantic representations. Additional work is required to put this claim on a more solid foundation. For example, the present simulations were limited by the small number of words in the model’s lexicon and the simplicity of their semantic relationships. We are currently working on larger scale simulations. With a larger, richer lexicon and more varied experience with it, we expect a better fit with the DRM data and to be able to compare the semantic representations formed by SLAM with those formed by LSA, WAS, and other models.

Another application of the SLAM model, due to its specification of a semantic learning mechanism with psychologically relevant parameters (e.g., short term memory size, degree of temporal or random and item-based drift), is to semantic learning over development (e.g., Brainerd, Reyna, & Forrest, 2002) and within specific clinical populations. For example, individuals with schizophrenia are often described as having difficulty maintaining context. As discussed by Hemsely (2005), a lack of contextual coherence could contribute to a number of phenomena associated with schizophrenia. Simulations with the SLAM model adjusting the relative contribution of temporal (random), item-based drift, and activation decay may be able to account for the loosening of semantic associations found in individuals with schizophrenia (Mohr et al., 2001; Popescu & Miclutia, 2006). Indeed, the finding that individuals with schizophrenia express a higher degree of association between uncommon or unusual items may be due to a lack of contextual differentiation, as found in the small-scale simulations reported in the present paper.

Acknowledgments

The authors would like to thank Michael R. Dougherty for his assistance and support on this project and Troy A. Smith for his assistance in understanding the fSAM model.

References


