Primary Sequential Memory:
an activation-based connectionist model

Colin Allen and Trent Lange

Department of Philosophy
Texas A&M University
College Station
TX 77843-4237, USA
tel: 409-845-5660
fax: 409-845-0458
colin.allen@tamu.edu

Department of Computer Science,
University of California, Los Angeles
Los Angeles
CA 90024, USA
tel: 310-825-5199
lange@cs.ucla.edu

Abstract

Storage and retrieval of ordered sequences from a single, serial presentation of each element in the sequence is typically not explained by existing connectionist models. Some models finesse the issue by presenting all the elements in a sequence simultaneously. Others rely on weight-changing algorithms that require multiple presentations of the sequence. We present a model for short-term storage and recall of ordered information that relies on gated activation mechanisms. Activation from each element presented serially recruits randomly connected responder nodes whose combined activation represents the element and its position in the sequence. The sequence is later recalled by feeding activation back to the elements from the recruited responder nodes. We discuss the relevance of the model to various results from cognitive psychology, including the facts that the length of human sequential memory is very limited, that for novel sequences recall is better for elements at the beginning and ends of sequences than for elements in the middle, and that humans have greater difficulty recalling the second occurrence of an element in a sequence containing a repeated element.

KEYWORDS: Serial order, short-term memory, neural network, Ranschburg effect.

Introduction

Sequential memory--the ability to store and recall items in a particular order--is crucial to many cognitive tasks. For example, the individual digits of a telephone number are of no use unless remembered in the correct order. Information about order may be presented spatially (e.g. telephone numbers listed in a directory) or temporally (e.g. numbers given out by the directory enquiries operator). Spatial encoding can be transformed into temporal encoding when the elements in the sequence are scanned serially. Humans are capable of memorizing some sequences after a single serial presentation of ordered elements.

The question naturally arises whether there is a limit to the number of items in sequences that humans can easily memorize for accurate recall. Miller (1956) introduced the idea of a limit to human information processing capacity which he described as "seven, plus or minus two" items or "chunks" of information. Sequences can be stored for periods ranging from a few seconds (short-term) to many years (long-term). Whereas the labels *short-term* and *long-term* apply to the temporal duration of memory, we use the terms *primary* and *secondary* to denote different mechanisms according to their precedence in memorization tasks. We are all familiar, for example, with the psychological strategy of silently repeating a new telephone number to oneself in order to consolidate it in memory. Clearly, however, this involves a *secondary* mechanism because *one could not repeat the sequence if it had not already been stored in some form*. While sequential information can be stored in long term memory, there is psychological evidence for different mechanisms between primary and secondary memory. Lee and Estes (1977) describe secondary memory as "associative structures representing alphanumeric characters or familiar sequences of characters" while primary memory "is limited to information that such previously established units were activated in a particular context." They describe short-term memory as a mixture of the processes of primary and secondary memory, and devise experiments to separate the two processes observationally (see discussion below). They attribute long term memory almost wholly to secondary memory processes.

In this paper we are interested only in providing a model of *primary* sequential memory. The input specification for primary sequential memory is a single presentation of a temporally ordered sequence of elements. The output specification is the recall of those elements in the order of presentation. Between input and output, any model of primary sequential memory must provide some way of representing all the elements of the sequence simultaneously, without losing the sequential information that is inherent in the temporal ordering of the input.

From an engineering point of view, to provide accurate sequential memory is a trivial task. A very simple model can be implemented using an array and inserting elements in the sequence (or memory pointers to those elements) into consecutive locations in the array. Each array position constitutes a special register for the corresponding item in the sequence to be memorized. Simple text editing programs using such a scheme can store and recall millions of characters with perfect fidelity. From a cognitive perspective, the story cannot be quite so simple. First, it is hard to see why human sequential memory should be limited to so few elements if the brain uses special registers. Second, human sequential memory is prone to systematic recall errors (discussed below) that are not easily explained by the special register model.

From a neurocomputing perspective, the challenge is even greater. In some models the problem of serial input is finessed by mapping the temporal dimension onto special input nodes for each position in the sequence. This is approach taken, for example, by Sejnowksi & Rosenberg (1986) with the NETtalk model for learning the correct pronunciation of syllables in context by backward error propagation. NETtalk learns to pronounce the fourth syllable in a sequence of seven, all of which are given simultaneously as input to the network. Reading a text is simulated by moving the seven syllable "window" through the text. Clearly, this neural net equivalent of the special registers model relies on a prior capacity to represent a temporal sequence spatially and it therefore provides no explanation of primary sequential memory. Hopfield (1982) considers the use of metastable states in a Hopfield net to represent sequences, but comments that "sequences longer than four states proved impossible to generate, and even these were not faithfully followed." Other connectionist models, such as the recurrent backpropagation models of Elman (1988) and Servan-Schreiber et al. (1989), do learn and store temporally coded sequential information. These models all, however, store their sequential information by employing time-intensive methods for modifying connection weights that require multiple presentations of the sequence to be learned. Thus even if such models can account for some aspects of sequential memory, they cannot account for the *primary* mechanism that must work given just a single presentation of the sequence.

A number of connectionist models have addressed reasoning on the time-scale of short-term, primary memory. Such models have been built to approach the problems of short-term reasoning with pre-existing semantic knowledge and rules (e.g. Touretzky &

Hinton 1989; Shastri & Ajjanagadde 1994; Lange & Dyer 1989). However, the problem of remembering short-term information that has some form of pre-existing connections (such as long-term rules) is quite distinct from the problem of remembering arbitrary, non-related sequences. None of these models, have therefore attempted to temporarily store and retrieve order information for novel sequences.

Estes (1972) proposed a psychological model of short-term ordered recall with a connectionist flavor. Estes' model proposed that memory of sequence involves "activated excitation flows to all elements in the sequence". He also proposed that proper sequential output is controlled by inhibitory connections between items. Subsequent research by psychologists has expanded on Estes' basic idea, attempting to explain recall phenomena in terms of inhibition and excitation (see e.g. Lee & Estes 1977; Bjork & Healy 1974; and Bjork 1988). However, none of these psychological models have been detailed enough to be actually implemented as a neural network.

Our research has led to a connectionist model of short-term sequential memory that uses only activation changes to temporarily store and recall the exact order of any novel short sequence of previously defined concepts. While neural activation has been proposed by psychologists as the mechanism for primary sequential memory our model is the first, to our knowledge, that is sufficiently detailed for implementation and testing.

A Model of Primary Sequential Memory

Our model for primary sequential memory consists of three areas of nodes that we have labeled *Semantic Memory, Responder Groups,* and *Sequence Clusters*. Figure 1 shows a small, simplified example of their basic connectivity.

Place Figure 1 here

The semantic memory consists of a relatively small number of nodes used to represent previously known concepts. Activation of semantic memory provides the "chunks" of information to be stored and recalled. For simplicity, we implemented the semantic memory as a localist winner-take-all network, with a separate node for each concept, although there is nothing in principle to prevent the sequential memory working with distributed representations in semantic memory.

Each node in semantic memory is randomly connected to a subset of the nodes in the responder group network. The responder groups serve to temporarily store both the identity of the items presented to semantic memory and their ordering information. They consist of hundreds or thousands of groups of nodes. The precise number does not matter although the general range affects the length of sequences that can be reliably stored. For ease of presentation, Figure 1 shows just ten responder groups and two of their nodes. The bottom nodes of the responder groups are randomly connected to a subset of the nodes in semantic memory. They serve simply as input nodes to directly pass an excitatory signal to the upper member of the responder group unless inhibited. The upper responder nodes have a random threshold (set within a predetermined range). When this threshold is exceeded for a given responder group, that responder group is said to have been "recruited" to partially represent that item in the sequence. The inhibitory connection to the bottom input responder then shuts off any further input from future elements in the sequence.

Storing the Sequence: An ordered sequence is presented to the model for storage by activating, in turn, each of the nodes in semantic memory representing that element in the sequence. The sequence cluster nodes will be ignored in this early discussion. When a given semantic element in the sequence is activated, activation propagates from it to the responder groups that are connected to it. This new activation causes a subset of those responder groups to go over threshold and be recruited to represent that element and its position in the sequence. They stay activated (through a self connection not shown) to continue representing that element, but the inhibitory connections to their input nodes remove them from pool of responder groups that can potentially respond to the next items in the sequence. Responder units that receive activation but do not go over threshold retain their activation, thus making them more likely to go over threshold and be recruited for subsequent elements. The crucial aspect of the model that allows sequence ordering information to be retained is that with relatively low thresholds, elements earliest in the sequence generally recruit more responder groups than later elements. As each element in the sequence is presented, the pool of unrecruited responders that can potentially represent a new element becomes smaller -- allowing the numbers of responders recruited for each subsequent element to implicitly hold their ordering information.

Place Figure 2a here

Figures 2a-c walk through an example of this process for a sequence of three elements (D - A - C). To start the sequence, semantic memory node D is clamped to an activation of 1 (Figure 2a), with all other semantic memory nodes having activation 0. Activation spreads from D to each of its potential responders through the responders' input nodes (bottom layer of the responder groups). Responder 1's activation is now 1, but its threshold is 1.4, so it does not fire. The same is true with Responder 5, whose threshold is 1.5. Responders 4, 8, and 10, however, have thresholds under 1, and so do fire. These responders have been recruited to represent the fact that D was the first element in the sequence, with the inhibitory link to their input nodes shutting them off from further input.

<u>*Place Figure 2b here*</u>

Node D is then shut off, and the second node in the sequence, A, is clamped to an activation of 1 and its activation propagated to its responder groups (Figure 2b). Responder 1, which had an activation of 1 before (from D), now gets enough activation (2 overall) to fire, and is recruited by A. Responder 3 is likewise recruited by A, with Responder 7 gaining activation but not firing. A has no effect on Responders 4 and 10, however, since they have shut themselves off from further input after having been recruited by D. Finally, the third node in the sequence, C, is clamped, and activation spread (Figure 2c). C is only able to recruit one responder group, Responder 9.

<u>*Place Figure 2c here*</u>

A total of six of the responder groups in Figure 2c were recruited during presentation of sequence D - A - C. The order of the elements in the sequence is implicitly represented by the activation of the responder groups, since three of the responders were recruited by D, two were recruited by A, and one was recruited by C. This ordering information of the network will also occur with other sequences that are presented. For example, if the sequence presented had instead been A - C - D, there would have been (a different) three responders recruited for A (3, 4, and 10), two for C (8 and 9), and one for D (1). As long as the thresholds of the responder groups are randomly set within a certain range, the number of recruited responders for each element will generally vary with its position in the sequence -- the first element in the sequence will nearly always recruit the most responder groups, the second element the next most, and so on, as the pool of eligible responder groups becomes smaller for each element in the sequence. This ordering becomes increasingly likely with larger responder group networks.

Retrieving the Sequence: The simplest scheme for recalling a sequence that has been presented to the network would be to simply feed activation back from the responder groups to the nodes in semantic memory. Though some of the responder groups recruited for a given element in the sequence will feed back into other elements (because of the random connectivity of Figure 1), only the semantic node for that actual element will get feedback from all of its recruited groups. Since the responder groups implicitly encode the sequence's ordering information through a decreasing number of recruited responder groups, the first element in the sequence should win the winner-take-all competition, because its connections from all of its recruited responder groups (the largest group of recruited responders) will generally cause it to get the most activation.

Unfortunately, such a simple scheme will all too often fail, especially on sequences with repeated elements. If the sequence F - G - G is presented, for example, and F recruits 30 responders, the first G recruits 20, and the second G recruits 15, then the total number of responders feeding back into G will be 35, causing it to be recalled first.

To handle these kind of retrieval problems, the model has a network of Sequence Clusters, each of which is randomly connected to a subset of the responder groups

(Figure 1). The sequence clusters serve to collect large portions of the otherwise unorganized responder groups into separate clusters of responders representing each individual element of the sequence. One sequence cluster (whichever happens to best group the recruited responders) becomes activated for each element of the sequence, allowing the responders recruited for each element to be differentiated between, therefore stopping problems such as the combined effect of the two G's in the F - G - G example listed above.

Selection of sequence clusters works basically as follows. After a new element of the sequence is presented to the network and has caused a subset of responder groups to be recruited, the newly fired responder groups drives a winner-take-all competition between the sequence clusters. The sequence clusters are connected in a dynamically-adapative winner-take-all network (Lange 1992) that allows a single winning sequence cluster to win no matter how many or few active inputs (responder groups) the winner has. The cluster that best represents the space of newly recruited responder groups (because it happens to have connections to more of the responder groups than any other cluster) will win the competition and cluster those responders into a group separate from the responders recruited by previous and future elements in the sequence.

Figure 3 shows an example of this after presentation of element D in the network of Figure 2a. Sequence cluster 3 happens to have more connections (two, to responders 8 and 10) to the three responder groups recruited for D than do the other sequence clusters. It will therefore win the competition to cluster the responders of this element, grouping two out of its three responders into an identifiable cluster. The winning sequence cluster will stay activated for later sequence recall, but will remove itself from the competitions for subsequent elements of the sequence (through inhibitory connections not shown). This process repeats for each element in the sequence.

Place Figure 3 here

At the end of the presentation of the sequence, there will be one sequence cluster active for each element in the sequence. Because the first element recruited more responder groups than any of the later elements, the cluster representing the first will generally (but not always) have input from the largest number of active responders. Similarly, the second cluster will have more active inputs than the third, and so on. An example run, implemented in the DESCARTES connectionist simulator (Lange 1990), is shown in Figure 4.

Place Figure 4 here

On recall, the active clusters will compete, and the cluster for the first element will win because of its greater number of responder groups (22 in Figure 3). This cluster will then feed its activation back to its responder groups and through them down to the semantic memory causing the first element in the sequence (R) to get the most activation and be recalled.

Place Figure 5 here

A simple example for the sequence D - A - C is shown in Figure 5. In Figure 5, the sequence clusters representing A (cluster 1) and C (cluster 4) are in the middle of competion with the cluster for D (cluster 3). Sequence cluster 3 is winning the competition because more responder groups feed into it (groups 8 and 10) than feed into the other sequence clusters. Activation then feeds *back* from it through its clustered responder groups (through a responder output node not shown) into the semantic memory. In Figure 5, responders 8 and 10 feed back into all of the semantic memory nodes they could potentially respond to. Here, group 8 can respond to C or D and group 10 can respond to A or D. Multiple elements of semantic memory therefore receive feedback from the responders grouped by the winning sequence cluster. However, one of the elements, and almost certainly only one of them (D in Figure 5), will receive feedback from all of the winning cluster's responders, and will therefore win the competition for retrieval as the first element.

The first cluster then removes itself from the competition (through additional nodes and gating not shown), allowing the cluster representing the second element in the sequence to win and recall the second element. The rest of the sequence is recalled in the same way. The complete sequence can be recalled repeatedly, until a new sequence is stored or all of the activation in the responder groups decays away.

Network Dynamics: Figure 3 shows only a simplified version of the full network. The full network includes additional nodes and connections to tell the network when to start memorization and retrieval and to control the dynamics of the storage and retrieval processes.

There are three "control" nodes that are used to start the memorization and retrieval processes: a *Memorize* node, a *Start-Retrieval* node, and a *Next-Element* node. Initially the network starts with no activation on any of its nodes. To start memorizing a new sequence, the initial semantic memory element of the sequence (e.g. "D") is clamped to an activation of 1. The *Memorize* node is then given an activation of 1 for one cycle. The *Memorize* node has a multiplicative connection to all of the responder groups' input nodes, allowing activation to pass from semantic memory to the responder groups and starting the storage process as described previously. After the network settles (i.e. responder groups recruited and a single sequence cluster selected to group them), the next element in the sequence is clamped to 1 and the *Memorize* node activated again to start the storage process again. The *Start-Retrieval* node is activated for a single cycle when retrieval of the sequence is desired. After the first element has been retrieved during subsequent settling of the network, the *Next-Element* node is activated for a single cycle to start competition for retrieval of the next element. *Next-Element* is then activated to start retrieval of each subsequent element, until there are no more.

Each responder group actually has three nodes: the input and upper responder nodes shown in Figures 1-3, and a responder output node as mentioned above. As described earlier, the responder input nodes (lower responder nodes in the figures) receive input from a subset of the semantic memory elements, and propagates it on to the main (upper) responder node when the *Memorize* signal has been received. The main responder node has an outgoing connection to the competion nodes of all of the sequence clusters that can possibly group it (e.g. responder 10 is connected to sequence clusters 2 and 3 in Figure 1). The responder output node, in turn, has a connection back from the competition nodes of all of those sequence clusters. It has outgoing connections to all of the semantic memory elements its responder input responds to. The responder output node therefore serves as the conduit to feed activation back from the retrieval competion of the sequence clusters to the semantic elements that will be retrieved.

The sequence clusters are slightly more complicated. An exact description of their connectivity is not important for the purposes of this paper, but we will describe their nodes and functions briefly. Each sequence cluster also has three nodes. The competition node matching those of Figure 3 and 5 receives activation from the main responder nodes of its random subset of responder groups and sends activation to their responder output nodes. The competition nodes of all of the sequence clusters are connected within a dynamically-adapative winner-take-all network (Lange, 1992) to allow a single winner grouping the most responders to be selected on storage and retrieval, as described above. Each sequence cluster also has a "storage" node that becomes activated (and stays activated) when a cluster has won the competition to represent a single element of the sequence. When it is active, the storage node inhibits its competition node, allowing other sequence clusters to compete to represent the next elements in the sequence. The third node of each sequence cluster is a "still competing" node that is activated when the *Start-Retrieval* signal is received, but which is inhibited (turned off) for the winning cluster when the *Next-Element* signal is received. Because the "still competing" node's activation is required for the competition, this allows the sequence clusters of the next elements to be retrieved.

Finally, it is important to note that the connections between the responder groups and their sequence clusters are gated (through inhibitory links not shown) in a special way. As shown in Figure 5, connections from active sequence clusters to responders that they did not actually cluster and from active responders to inactive sequence clusters become gated closed and do not propagate activation (dashed lines). This allows the sequence clusters to differentiate between the responders that they actually clustered for their element and responders that became activated from later (or earlier) sequence elements (e.g. responder 3 for sequence cluster 3 in Figure 5). If such false connections are not closed, then both the ordering information implicit in the size of the sequence clusters and the element implicitly represented by the clustered responder group can become lost. Because activation only flows between groups of responders and the sequencers that actually clustered them, ordering information is preserved (since the relative number of responders recruited for each element in the sequence is preserved), as is the element each sequence cluster represents (since only responders recruited for that element will actually get feedback from the sequence cluster).

Discussion

We have presented a connectionist model that is able to temporarily store and recall ordered information of any novel short sequence of items presented one at a time. The model performs this solely by the spread of activation, which recruits randomly connected nodes to represent each element and its position in the sequence. A temporally presented sequence is therefore represented by a spatially distributed pattern of activation. Any given responder group may represent different elements in different positions in different sequences, thus there is no sense in which our model contains special registers for elements or positions.

Although a considerable amount is known about the neural mechanisms of many aspects of memory, unfortunately little is known about the neural mechanisms specifically underlying primary sequential memory (possibly because the *apparent* triviality of the task has caused it to be ignored). While the types of nodes and connections in our model are comparable to many other attempts to model neural systems, it would not be reasonable to make any strong claims about correspondences between the model we have proposed and biological nervous systems, especially at the level of overall structure. One intriguing aspect of our model, however, is the way in which randomly connected layers can be used to represent ordered information. Until the relevant neurological work is done, there is no way to know whether layers such as these play any role in sequential memory in biological systems.

Although we are still investigating the properties of the model, it has several features that are very interesting from a cognitive perspective:

The "magical" number seven: Miller (1956) described research indicating that human ability to memorize a sequence of items is dependent on the way in which those items are "chunked". If one naively attempts to memorize a sequence of ones and zeros, the rough guide of seven ones or zeros holds. But if one mentally converts binary numbers into octal or hexadecimal (i.e. chunking them into groups of three or four) it is possible, with practice, to become proficient at scanning and then reproducing sequences of ones and zeros three or four times longer than without chunking. Using different chunking strategies it is possible to markedly extend the length of sequences that can be accurately recalled. It does not seem possible to markedly increase the number of chunks that can be memorized, although where order of recall is not important, it is possible to recall more items (Bjork 1988).

Why is human primary sequential memory limited to so few chunks? Our model provides a plausible explanation. Because of the way responder groups are recruited by items in semantic memory, the number of responder groups needed to reliably store sequences grows exponentially with the length of the sequence. Each input event causes roughly the same fraction of unrecruited responder groups to go over threshold and hence be recruited. Let p $(0 < p < 1)$ be that fraction. Letting N be the number of responder groups, and assuming no decay of activation, the jth element in the sequence will recruit approximately $N.p.(1-p)^{j-1}$ responder groups.

Accurate recall of a sequence requires that the difference between the number of responder groups recruited by the jth and kth elements must meet or exceed some threshold, d (≥ 1). That is, unless

$$N.p.(1 - p)^{j-1} - N.p.(1 - p)^{k-1} \geq d$$

the jth element will not reliably be recalled before the kth element. In the crucial case where the jth and kth elements are consecutive, i.e. $k = j+1$, this condition can be rewritten as

$$N.p.(1 - p)^{j-1} - N.p.(1 - p)^{j} \geq d$$

which can be transformed into

$$N \geq d.(1 - p)^{1-j} / p^2$$

and from which it can easily be seen that N grows exponentially with respect to j if we consider that $1-p = 1/c$ for some $c > 1$, and hence that the inequality is equivalent to

$$N \geq d.c^{j-1} / p^2.$$

The exponential cost of increasing memory capacity associated with our model distinguishes it from conventional network models where capacity is usually linearly related to network size. For instance, Hopfield (1982) estimates the storage capacity of a Hopfield net to be approximately 15% of the number of nodes. This agrees with the range of 10-20% established for other types of network (Anderson & Rosenfeld 1988).

However, as commented previously, such networks are not well suited to the task of primary sequential memory. Alternative schemes for representing sequential information spatially seem inevitably to depend on special registers. Assuming a linear cost for adding more registers, and the obvious advantage of being able to memorize longer sequences, it is natural to wonder why there is not more variation in capacity between individuals and why the capacity for memorizing very long sequences from a single presentation has not evolved.

Exponential growth places a severe biological cost on the evolution of the capacity for storing longer sequences. Furthermore, the random nature of the connections between semantic memory and responder groups and between responder groups and sequence clusters means that this estimate of N represents an ideal that is not generally achieved in practice. Because, however, the responder groups are capable of representing any sequence of events in semantic memory our model is capable of exploiting whatever chunking strategies are available to the semantic memory.

The Ranschburg Effect. Human subjects who are given the task of recalling elements in a serially presented sequence are less accurate at recalling repeated elements. This phenomenon is known as the Ranschburg Effect (Jahnke 1969). Our model displays a similar propensity, although the correspondence between our model and the psychological data is not exact.

A possible reason for this is that some parameters of the model should be adjusted. We are presently investigating the effects of manipulating the percentage of semantic nodes connected to each responder group, the range of the responder group thresholds, and activation decay rates in responder groups, and we hope that by manipulating these parameters we can find a more precise model for the Ranschburg Effect. Another reason for the discrepancy between the model and the psychological data is the enormous difficulty of separating primary mechanisms from secondary mechanisms in human subjects. A technique commonly used is to present each item in the sequence for a fraction of a second and at the end of the sequence to give subjects a "distractor task" to try to prevent any attempts to mentally go over the sequence. Although such distractor tasks have a significant effect on recall performance, there is, of course, no guarantee that they are completely effective in blocking secondary memory mechanisms. Indeed, Hinrichs et al. (1973) concluded that the psychological data for the Ranschburg effect was due the interaction of two factors, the failure to detect repetitions (see also Kanwisher 1987) and "inappropriate guessing strategies."

U-shaped recall: A significant difference between our model and the performance of human subjects is that human subjects are typically least accurate about elements in the middle of sequences (Jahnke 1969), whereas the model described above is least accurate on elements at the end of a sequence (because of the relatively small number of responder groups that remain to be recruited). This feature of our model is, however, under the control of the parameter determining activation decay in responder groups. The results described in this paper assumed no activation decay. (The model was simply wiped clean at the beginning of each new sequence.) However, by setting a decay factor, it is possible to change the recall performance. To see how this works, consider a set of responder groups that have been recruited by the first element in a sequence. As activation levels in those groups decay, some of them will drop below threshold and can be recruited by subsequent elements in the sequence. If the decay rate is very high, then only later elements in the sequence will have recruited responder groups that are still active during the recall phase. In other words, the beginning of the sequence will have been forgotten. If the decay rate is intermediate, then some of the initially recruited responder groups will be recruited by later elements in the sequence. This can have the effect of raising the number of groups recruited by elements at the end of the sequence above those in the middle of the sequence, while keeping it below the number recruited by elements at the beginning of the sequence. Similar effects can be achieved by altering the range of thresholds for responder group nodes--because higher thresholds may delay recruitment--or by slowing the relative presentation rate of the sequence--because responder groups recruited early in the sequence may become active again before the end of the sequence.

We are currently experimenting with the model to try to find a reasonable approximation to the data from the psychological experiments, by varying these parameters. As before, however, it is not clear that our model can be expected to exactly reproduce those results because of questions about the dependence of those results on primary memory alone. This is especially acute when rate of presentation is the variable because slow rates of presentation may allow secondary mechanisms to play a more significant role.

Conclusion

Modelling short-term sequential memory is an important task that has thus far been overlooked by connectionist researchers. Besides being a first pass at a model of human short-term sequential memory, our model is a step towards a connectionist model of the short-term buffer that is necessary to hold sequential training data for long-term backpropagation models and thus eliminating one of their remaining symbolic crutches. Further collaboration between cognitive psychologists, neuropsychologists and computer scientists will be needed to improve the biological and psychological relevance of the mechanisms proposed.

Acknowledgements

We wish to thank audiences at the Texas A&M Cognitive Psychology Colloquium, Texas A&M University Neural Modelling Society (NeMoS), University of Houston-Downtown NeMoS, and the UCLA Artificial Intelligence Laboratory, and especially Sergey Aityan, Michael Dyer and Michael Smith for helpful comments on this research. We are also grateful for helpful comments on the manuscript from the anonymous referees for this journal. Trent Lange was partially supported by a contract with the JTF program of the DOD while this research was conducted.

References

[1] J.A. Anderson and E. Rosenfeld, eds., *Neurocomputing: Foundations of research* (MIT Press, Cambridge, MA, 1988).

[2] R.A. Bjork and A.F. Healy, Short-term order and item retention, *Journal of Verbal Learning and Verbal Behavior* 13 (1974) 80-97.

[3] R.A. Bjork, Retrieval inhibition as an adaptive mechanism in human memory, in: J.L

Roediger and F.I.M. Craik, eds., *Varieties of Memory and Consciousness: Essays in Honour of Endel Tulving* (Erlbaum, Hillsdale, NJ,1988).

[4] J.L. Elman, Finding structure in time, Tech. Rept., Center for Research in Language, UCSD, San Diego, CA,1988.

[5] W.K. Estes, An associative basis for coding and organization in memory, in: A.W. Melton and E. Martin, eds., *Coding Processes in Human Memory* (Winston, Washington DC, 1972) 161-190.

[6] J.V. Hinrichs, S.P. Mewaldt and J. Redding, The Ranschburg effect: repetitions and guessing factors in short-term memory, *Journal of Verbal Learning and Verbal Behavior* 12 (1973) 64-75.

[7] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proceedings of the National Academy of Sciences* 79 (1982) 2254-2258.

[8] J.C. Jahnke, The Ranschburg effect, *Psych. Review* 76 (1969) 592-605.

[9] N.G. Kanwisher, Repetition blindness: Type recognition without token individuation, *Cognition* 27 (1987) 117-143.

[10] T. Lange, Simulation of heterogeneous neural networks on serial and parallel machines, *Parallel Computing* 14 (1990) 287-303.

[11] T. Lange, Dynamically-adaptive winner-take-all networks, in: J. E. Moody, S. J. Hanson and R. P. Lippmann, eds., *Advances in Neural Information Processing Systems* 4 (Morgan Kaufmann, San Mateo, CA, 1992) 341-348. (Collected papers of the IEEE Conference on Neural Information Processing Systems - Natural and Synthetic, Denver, CO, December 1991).

[12] T. Lange and M.G. Dyer, High-level inferencing in a connectionist network. *Connection Science* 1 (2) (1989) 181-217.

[13] C.L. Lee and W.K. Estes, Order and position in primary memory of letter strings, *Journal of Verbal Learning and Verbal Behavior* 16 (1977) 395-418.

[14] G.A. Miller, The magical number seven, plus or minus two: some limits on our capacity for processing information, *Psych. Review* 63 (1956) 81-97.

[15] T.J. Sejnowski and C.R. Rosenberg, NETtalk: a parallel network that learns to read aloud, Tech. Rept. JHU/EECS-86/01, Electrical Engineering and Computer Science, The Johns Hopkins University, Baltimore, MD, 1986.

[16] D. Servan-Schreiber, A. Cleeremans and J.L. McClelland, Learning sequential structure in simple recurrent networks, in: D.S. Touretzky, ed., *Advances in Neural Information Processing Systems 1* (Morgan Kaufmann, San Mateo, CA, 1989) 643-652.

[17] L. Shastri and V. Ajjanagadde, From simple associations to systematic reasoning: A connectionist representation of rules, variables, and dynamic bindings using temporal synchrony, *Behavioral and BrainSciences* (1994).

[18] D.S. Touretzky and G.E. Hinton, A distributed connectionist production system, *Cognitive Science* 12 (1988) 423-466.

Colin Allen is associate professor of philosophy at Texas A&M University. He received graduate training in philosophy and computer science at UCLA where he collaborated with Trent Lange on the model of primary sequential memory. Besides this work, his research is on the conceptual foundations of cognitive ethology.

Trent Lange is a member of the UCLA Artificial Intelligence laboratory. His research is on connectionist implementations of symbolic reasoning, particularly the use of hybrid symbolic and connectionist systems.

Figure Captions

Figure 1: A miniature version of the model with four Semantic Memory nodes (A-D), ten Responder Groups (1-10), and three Sequence Cluster nodes (1-3), showing random connections between units in Semantic Memory and Responder Groups (50% chance of connection), and between Sequence Clusters and Responder Groups (30% chance of connection). Links with dark triangles are inhibitory, while links with white triangles or no triangles are excitatory. The number in the upper node of each Responder Group pair represents a threshold randomly chosen between 0 and 2. For simplicity, all excitatory connections are of unit weight.

Figure 2a: Activation of Responder Groups after D is activated as the first element of the sequence. A unit signal is passed along all the lines from semantic memory node D to Responder Groups. If the threshold on the upper node of the Responder Group was less than 1.0, then the group is recruited and sends an output of 1.0 (shown by shaded rectangle). If the threshold is greater than 1.0 then the unit activation is stored by the upper node (shaded circles).

Figure 2b: Activation of Responder Groups after A has been activated as the second element of the sequence. The Responder Groups in the shaded rectangles have been recruited by the semantic node indicated by the letter in the top left corners. In this example, group 1's upper node has a total activation of 2.0 (from the combination of the earlier stored activation from D and the present activation from A), which is greater than its threshold (1.4), and so is recruited to represent A.

Figure 2c: Activation of Responder Groups after C has been activated as the third element of the sequence.

Figure 3: Simplified version of Sequence Cluster nodes arranged in a winner-take-all network and randomly connected to responder groups. Figure shows the activation of the sequence clusters during competition after presentation of D (see Figure 2a). Here, Sequence Cluster 3 has the most inputs from D's recruited Responder Groups, and so will win the winner-take-all competition and serve to cluster those responders.

Figure 4: Responder Groups and Sequence Clusters activated when the sequence R - B - F - Q - B-R was presented to a network where each semantic element was randomly connected to 100 of the 200 total responder groups. The network had 50 sequence clusters, each of which had random connections to 100 of the responder groups. The first column shows the element presented, the second column shows the number of responder groups recruited by that element, the third column shows which sequence cluster won to represent that element, and the fourth column shows how many of the responders that cluster actually represents.

Figure 5: Simplified version of the network of Figure 2c and 3 during retrieval of the first element. Dashed lines show connections gated closed that do not spread activation for this retrieval (see section on Network Dynamics). Here, Sequence Cluster 3 (representing D) has active connections to more clustered Responder Groups than do the other competing sequence clusters (1 and 4), and so will win the competition. Already feedback from it through its clustered responder groups (8 and 10) is causing its element in the semantic network (D) to start to win for retrieval of the sequence's first element.

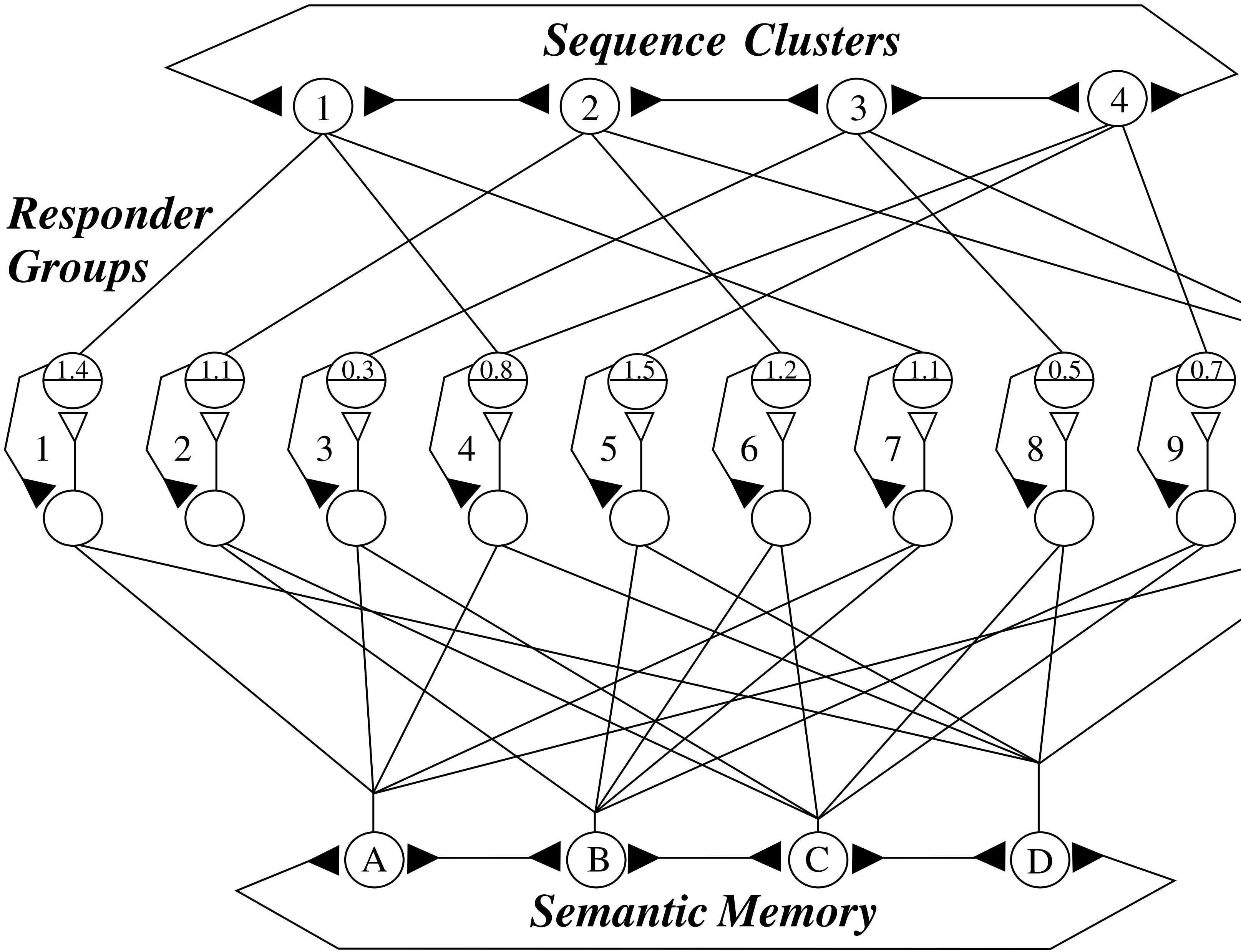
Sequence Clusters
1
2
3
4
Responder Groups
1.4
1.1
0.3
0.8
1.5
1.2
1.1
0.5
0.7
0.9
1
2
3
4
5
6
7
8
9
10
A
B
C
D
Semantic Memory

figure 1

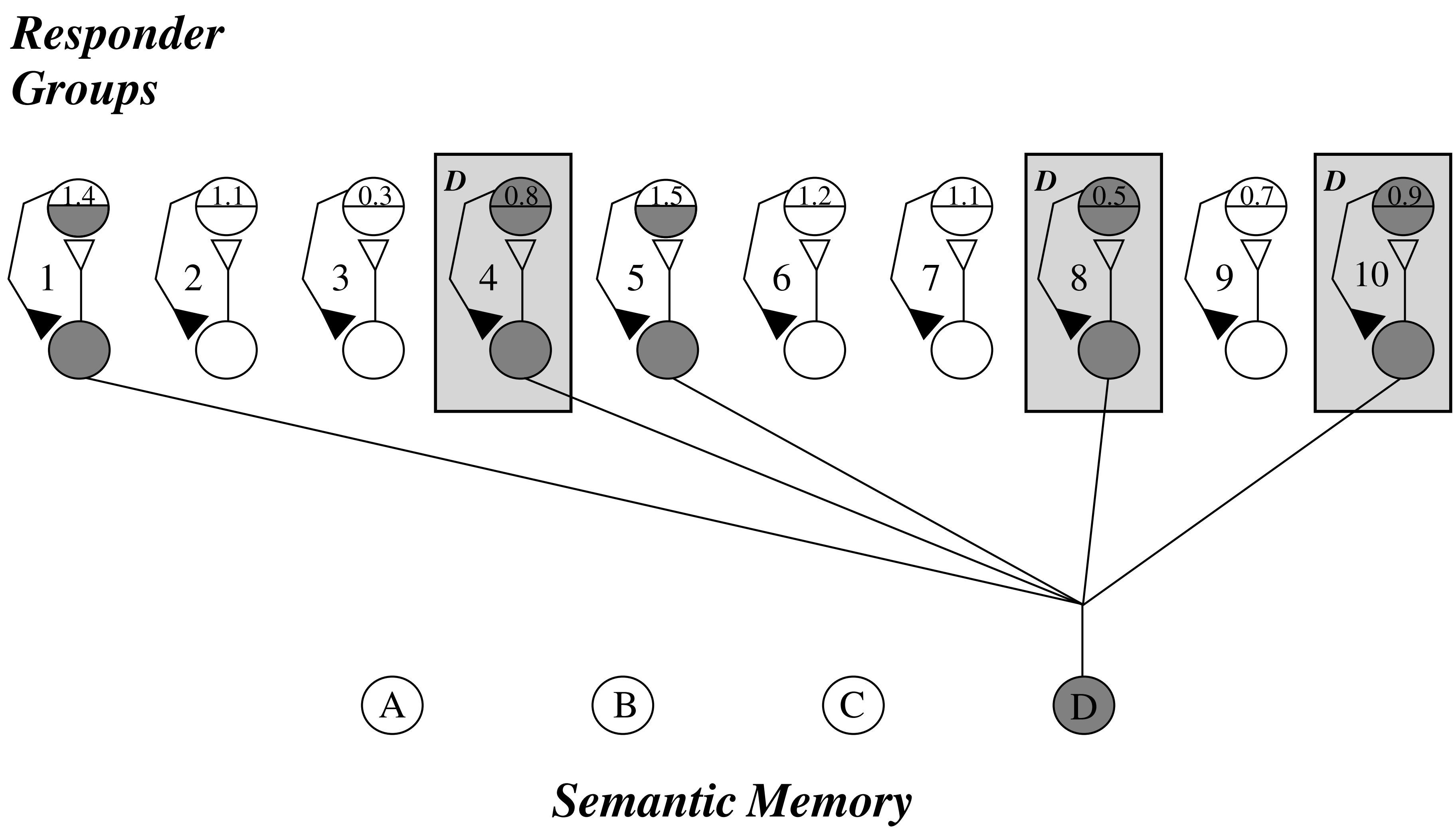
Responder
Groups
1.4
1.1
0.3
D
0.8
1.5
1.2
1.1
D
0.5
0.7
D
0.9
1
2
3
4
5
6
7
8
9
10
A
B
C
D
Semantic Memory

figure 2a

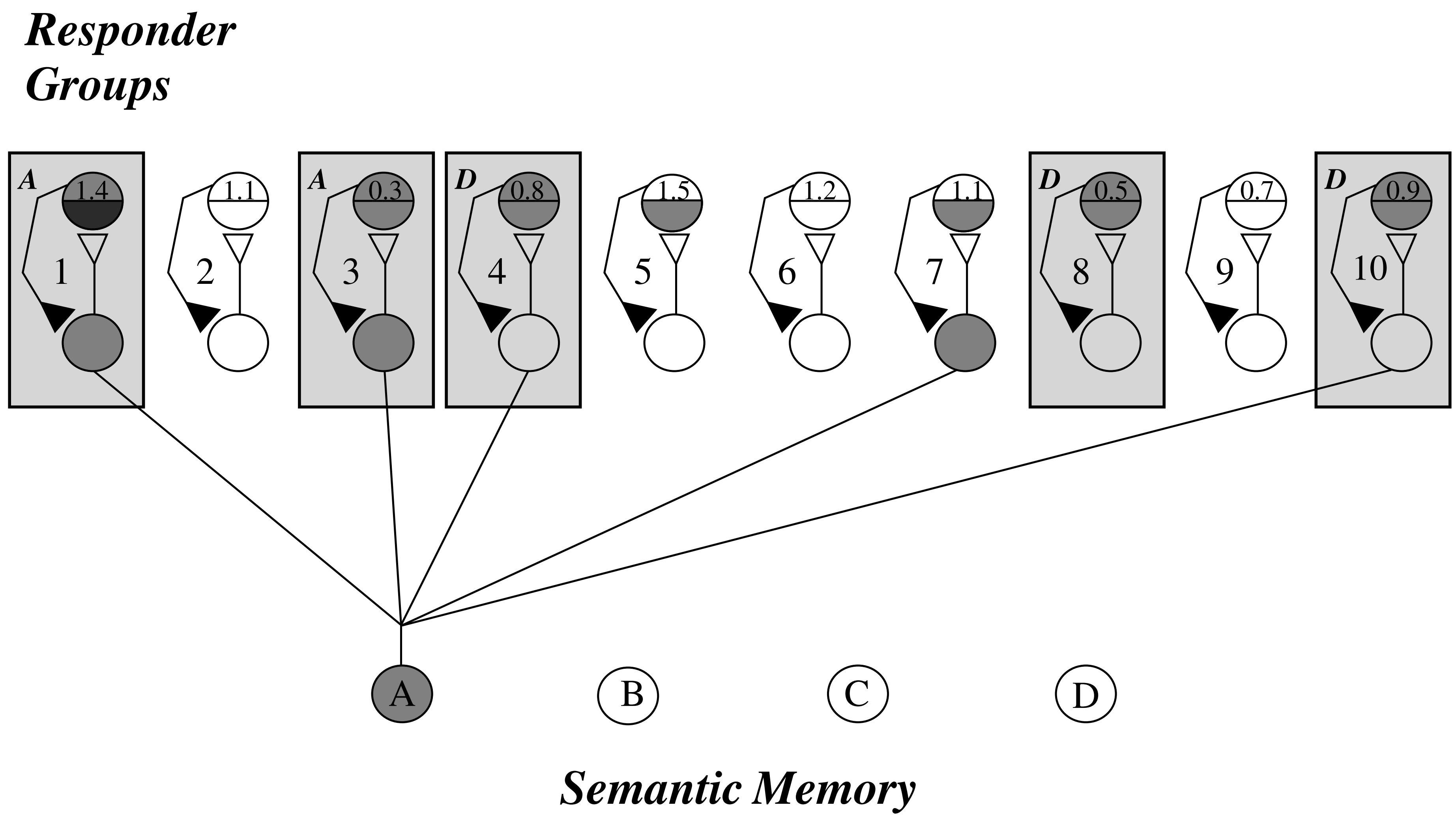
Responder
Groups
A
1.4
1
2
1.1
A
0.3
3
D
0.8
4
1.5
5
1.2
6
1.1
7
D
0.5
8
0.7
9
D
0.9
10
A
B
C
D
Semantic Memory

figure 2b

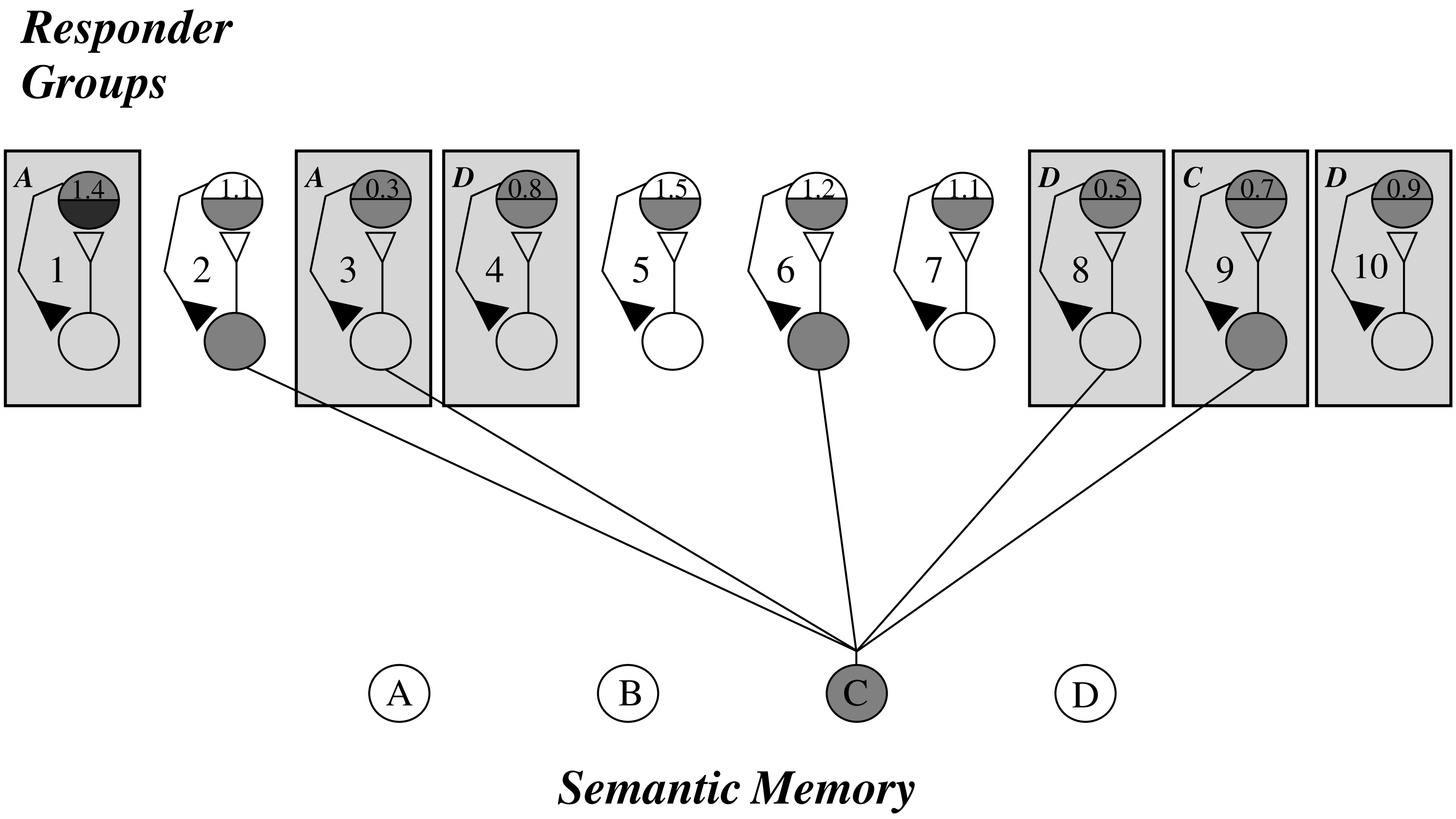
Responder
Groups
A
1.4
1
1.1
2
A
0.3
3
D
0.8
4
1.5
5
1.2
6
1.1
7
D
0.5
8
C
0.7
9
D
0.9
10
A
B
C
D
Semantic Memory

figure 2c

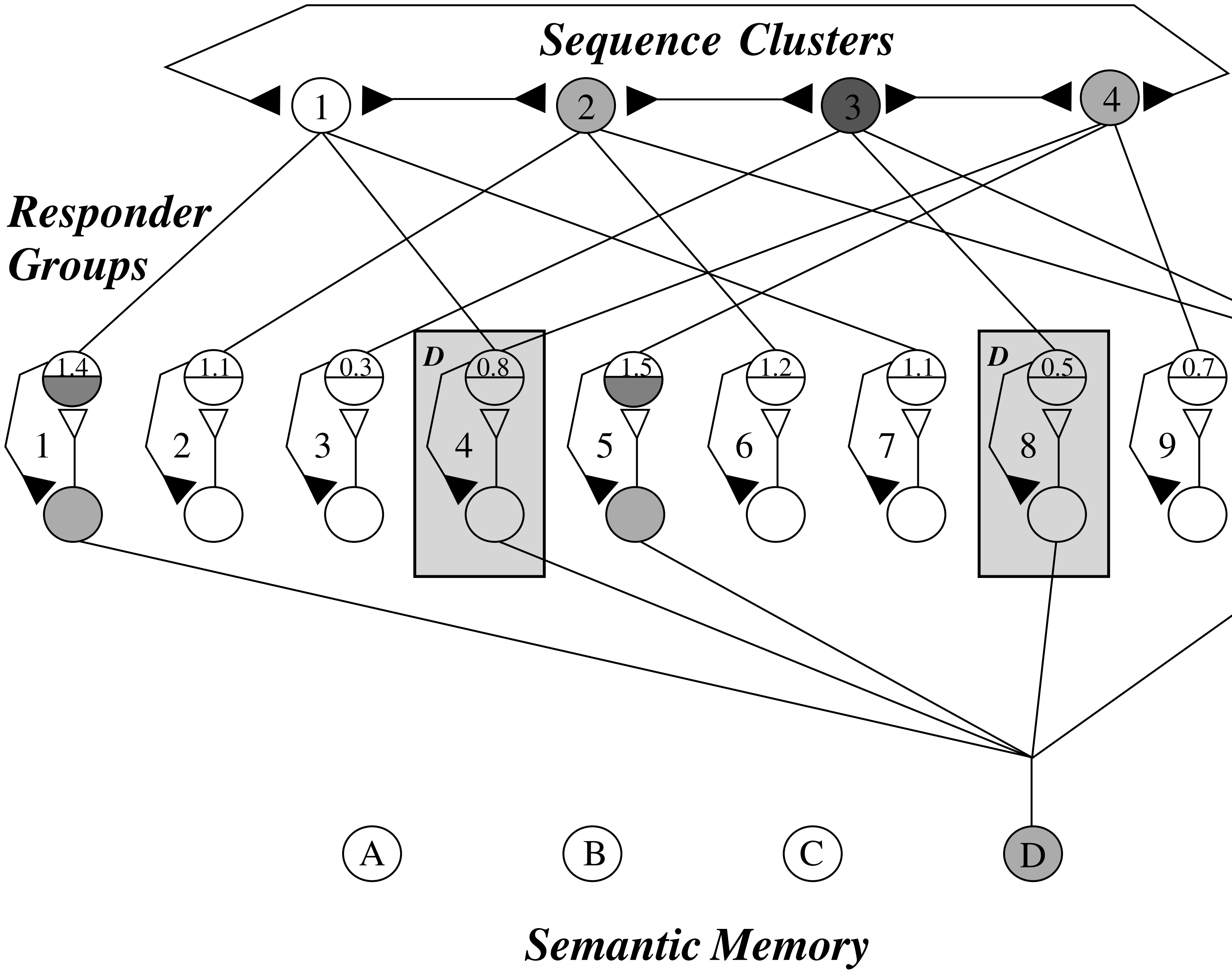
Sequence Clusters
1
2
3
4
Responder Groups
1.4
1.1
0.3
D
0.8
1.5
1.2
1.1
D
0.5
0.7
D
0.9
1
2
3
4
5
6
7
8
9
10
A
B
C
D
Semantic Memory

figure 3

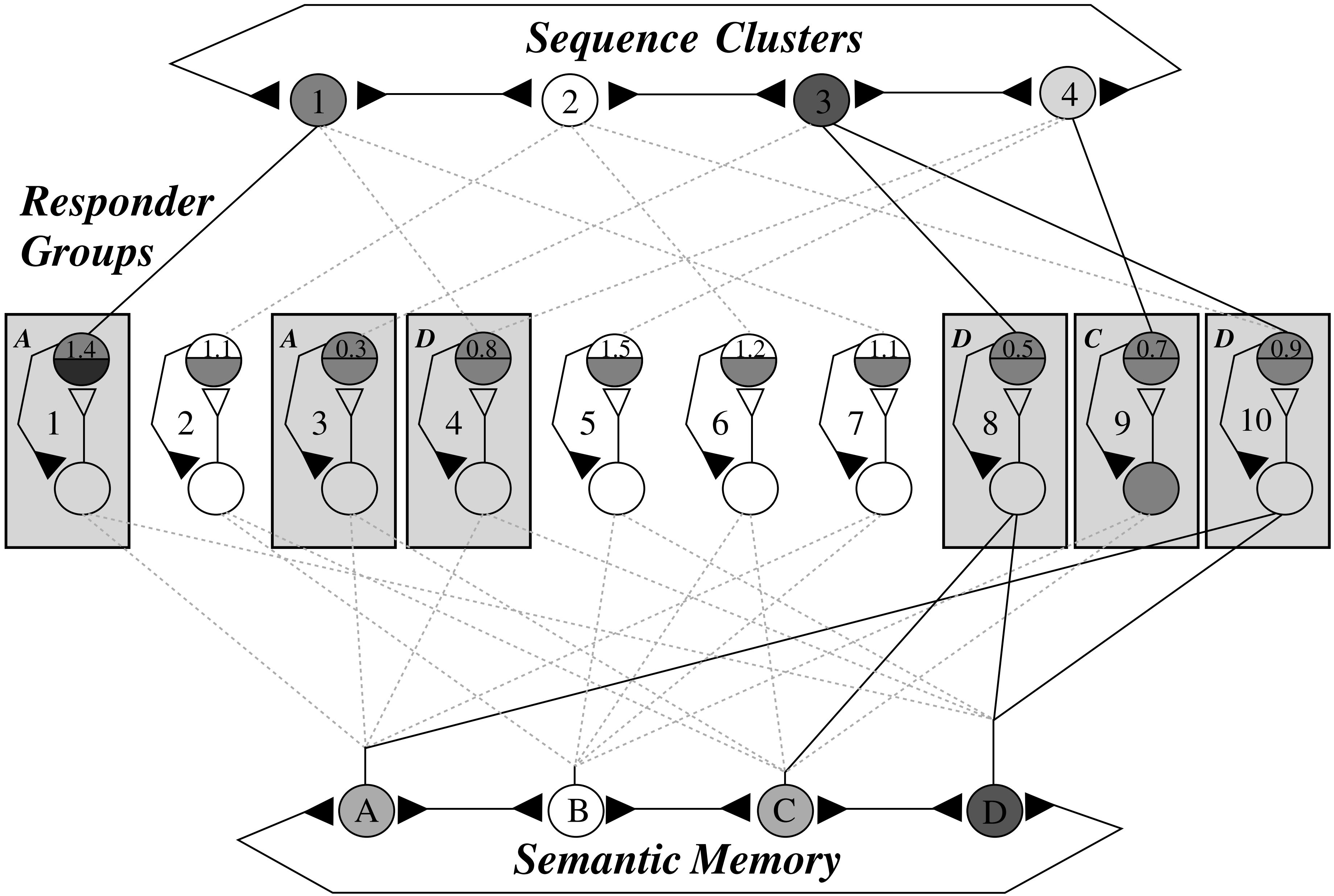
Sequence Clusters
1
2
3
4
Responder Groups
A
1.4
1
1.1
2
A
0.3
3
D
0.8
4
1.5
5
1.2
6
1.1
7
D
0.5
8
C
0.7
9
D
0.9
10
A
B
C
D
Semantic Memory

figure 5

Element Presented	**Responders Recruited**	**Sequence Cluster #**	**Responders Clustered**
R	58	19	36
B	52	36	29
F	37	9	23
Q	27	42	17
B	7	23	5
R	3	7	2

figure 4