Memory & Cognition
1995, 23 (5), 631-645

Developing TODAM: Three models for serial-order information

BENNET B. MURDOCK
University of Toronto, Toronto, Ontario, Canada

TODAM2, a theory of distributed associative memory, shows how item and associative information can be considered special cases of serial-order information. Consequently, it is important to get the right model for serial-order information. Here, we analyze and compare three distributed-memory models for serial-order information that use TODAM's convolution–correlation formalism. These models are the chaining model, the chunking model, and a new model, the power-set model. The chaining model associates each item with its predecessor; the chunking model uses multiple convolutions and n-grams to form chunks; and the power-set model interassociates all items in a set in a particular way to form a chunk. The models are compared in terms of their performance on seven basic tests of serial-order information—namely, serial recall, backward recall, recall of missing items, sequential probe tests, positional probe tests, serial-to-paired-associate transfer, and item recognition. The strengths and weaknesses of each model are discussed.

If we are to understand human memory in any detail, we must distinguish three types of information: (1) Item information, which underlies the recognition of single objects ("items"); (2) associative information, which underlies the relation or association between two objects or items; and (3) serial-order information, which preserves the temporal order in a string of three or more items. Evidence for these distinctions was presented in Murdock (1974), and TODAM (an acronym for *theory of distributed associative memory*) has always preserved these distinctions.

TODAM2 (Murdock, 1993b) is an attempt to provide a unified theoretical account of the storage and retrieval of item, associative, and serial-order information. To give a brief overview, it is a distributed-memory model in which items are represented by random vectors, associations are represented by the convolution of item vectors, information is stored in a common memory vector, the dot product is the comparison operation for recognition, and correlation is the retrieval operation for recall; these operations are assumed to go on in working memory, part of the general system architecture. Chunks are represented as sums of n-grams, and n-grams are n-way autoassociations (autoconvolutions) of the sums of n item vectors. If $n = 1$, we have item information; if $n = 2$, we have associative information; and if $n \geq 3$, we have serial-order information. Each of these concepts will be explained shortly.

This research was supported by Grant APA 146 from the Natural Sciences and Engineering Research Council of Canada. I would like to thank Mike Kahana for many helpful comments on the manuscript. Correspondence should be addressed to B. Murdock, Department of Psychology, University of Toronto, Toronto, Ontario M5S 1A1, Canada (e-mail: murdock@psych.toronto.edu).

The TODAM2 analysis makes it clear that item and associative information are special cases of serial-order information, so it is crucial to get the right model for serial-order information. The serial-order model we used in TODAM2 was the chunking model (Murdock, 1992), but that is only one possibility. In this paper, we analyze and compare three possible distributed-memory models for serial-order information: The chaining model of Lewandowsky and Murdock (1989); the chunking model; and the power-set model, a new model for serial-order information that seems to remedy some of the problems of the chaining and chunking models. The intention of this paper is to analyze the strengths and weaknesses of these three models to guide further development.

There are many other memory models besides TODAM, including SAM (Shiffrin & Raaijmakers, 1992), MINERVA2 (Hintzman, 1988), CHARM (Metcalfe-Eich, 1982), the matrix model (Humphreys, Bain, & Pike, 1989), the resonance-retrieval theory (Ratcliff, 1978), and the perturbation model (Estes, 1972). However, with the exception of the perturbation model, all of these models deal with item or associative information, and none have been extended to deal with serial-order information. The perturbation model deals with serial-order recall, but does not really deal with item or associative information (see Murdock, 1992, for a comparison of the chunking model and the perturbation model). Consequently, while we would like to consider other models in this review, we are not able to do so. However, in principle, most, if not all, of these models could be extended to the serial-order case, and it is our hope that this paper will stimulate such development.

We first review the general TODAM framework, then describe and analyze the three models in turn, and end with a summary comparison.

BASIC TODAM FRAMEWORK

Representation

Following Anderson (1973), items (e.g., **f**, **g**) are represented as independent (or possibly correlated) random vectors. A random vector is an N-dimensional vector whose N elements are random samples from some specified feature distribution. Less technically, a random vector is simply an ordered list of features in which each feature is a random variable. With minor variations, this representation is used by most distributed-memory and connectionist models today.

Again following Anderson (1973), this feature distribution is assumed to be a normal (Gaussian) distribution with a mean of zero and a variance of P/N, where, generally, $P = 1$. In this way, item vectors are always unit vectors, regardless of N; that is, they are always statistically normalized to 1.0.

Convolution and Correlation

After Borsellino and Poggio (1973), we use the convolution of item vectors to represent the association of two item vectors and the correlation of one item vector (the probe) with the convolution for the retrieval operation. Convolution (*) is a way of combining or blending two items, and correlation (#) is the approximate inverse of convolution. If two variables, a and b, are multiplied and the product is divided by one of them (say a), then the result is the other (i.e., b). Loosely speaking, convolution is analogous to multiplication and correlation is analogous to division (i.e., multiplication by an inverse). Thus,

$$\mathbf{f}\#(\mathbf{f}*\mathbf{g}) = \mathbf{g}',$$

where $\mathbf{g}'$ is an approximation to $\mathbf{g}$.

If two item vectors were added, they would have to be aligned properly; that is, for item vectors $\mathbf{f}$ and $\mathbf{g}$, one cannot add the ith element of $\mathbf{f}$ to the jth element of $\mathbf{g}$, $\mathbf{i} \neq \mathbf{j}$. The same applies to convolution and correlation; that is, the two vectors must be aligned properly. If $\mathbf{x}$ is the point of alignment, then, at the component level,

$$(\mathbf{f}*\mathbf{g})(x) = \Sigma_i f(i)\, g(x-i).$$

This expression defines convolution for item vectors, and is the discrete analogue of the convolution of continuous functions. The defining expression for correlation is

$$(\mathbf{f}\#\mathbf{g})(x) = \Sigma_i f(i)\, g(x+i),$$

so the only computational difference between correlation and convolution is that the correlation operation involves the sum, not the difference, of the second subscript.[1] For a more detailed explanation, see Metcalfe-Eich (1982) or Murdock (1979).

Deblurring

Since $\mathbf{g}'$ is an approximation to $\mathbf{g}$, we must deblur $\mathbf{g}'$ to $\mathbf{g}$ in order to recall $\mathbf{g}$, so retrieval is a two-step process (correlation followed by deblurring). Just as deblurring a fuzzy picture means bringing it into sharper focus, so deblurring $\mathbf{g}'$ to $\mathbf{g}$ means bringing $\mathbf{g}'$ into sharper focus (i.e., making it more like $\mathbf{g}$). Each item vector is a list of features, and if, relative to $\mathbf{g}$, each feature of $\mathbf{g}'$ is slightly off target (some are too high, and others too low), then deblurring $\mathbf{g}'$ to $\mathbf{g}$ means making each feature of $\mathbf{g}'$ more like the corresponding feature of $\mathbf{g}$. A detailed explanation is beyond the scope of the present paper, but for an illustrative application, see Lewandowsky and Li (1994).

Storage Equation

Information in TODAM, whether of the item, associative, or serial-order type, is always stored in a common memory vector. We use composite storage; that is, if there are two or more types of information, they are all stored in this same common memory vector. For item and associative information, if $\mathbf{M}$ is the common memory vector and α is the forgetting parameter, $0 \leq \alpha \leq 1$, then the memory vector after the jth pair has been presented is

$$\mathbf{M}_j = \alpha\mathbf{M}_{j-1} + \mathbf{f}_j + \mathbf{g}_j + \mathbf{f}_j*\mathbf{g}_j, \qquad (1)$$

where $\mathbf{f}_j$ and $\mathbf{g}_j$ are the two items in the jth pair and $\mathbf{f}_j * \mathbf{g}_j$ is their association (e.g., Murdock, 1982). This is the basic storage equation for item and associative information; it is a first-order–difference equation or recurrence relation, in which the state of the system after the jth pair is presented is a function of the state of the system after the $(j-1)$th pair had been presented plus the addition of the new information.

Suppose $\mathbf{M}_0$ (the memory vector at the start of the list) is zero, and the list consists of three pairs, *A-B*, *C-D*, and *E-F*. Then $\mathbf{M}_1$ (the memory vector after the first pair has been presented) is $\mathbf{a} + \mathbf{b} + \mathbf{a}*\mathbf{b}$, where $\mathbf{a}$ and $\mathbf{b}$ are the encoded representations of Items *A* and *B*, $\mathbf{M}_2$ is $\alpha\mathbf{M}_1 + \mathbf{c} + \mathbf{d} + \mathbf{c}*\mathbf{d} = \alpha(\mathbf{a} + \mathbf{b} + \mathbf{a}*\mathbf{b}) + \mathbf{c} + \mathbf{d} + \mathbf{c}*\mathbf{d}$, and $\mathbf{M}_3$ is $\alpha\mathbf{M}_2 + \mathbf{e} + \mathbf{f} + \mathbf{e}*\mathbf{f} = \alpha^2(\mathbf{a} + \mathbf{b} + \mathbf{a}*\mathbf{b}) + \alpha(\mathbf{c} + \mathbf{d} + \mathbf{c}*\mathbf{d}) + \mathbf{e} + \mathbf{f} + \mathbf{e}*\mathbf{f}$. Thus, Equation 1 says that when each new pair is presented, the memory vector to date is decremented ($\mathbf{M}_0$ is multiplied by α, where $0 \leq \alpha \leq 1$), and then both of the two current items and their convolution are added to the resulting memory vector.

The comparable storage equation for item and serial-order information is

$$\mathbf{M}_j = \alpha\mathbf{M}_{j-1} + \mathbf{f}_j + \mathbf{f}_j*\mathbf{f}_{j-1} \qquad (2)$$

(Lewandowsky & Murdock, 1989; Murdock, 1983). This equation describes a simple chaining model; after the memory vector has been decremented, the current item ($\mathbf{f}_j$) and its association with the prior item ($\mathbf{f}_j*\mathbf{f}_{j-1}$) are both added to the common memory vector. Again using capital letters for list items and small letters for their vectorial counterparts, if we use X for the starting signal, then for a five-item list, *XABCDE*, if we omit the forgetting factor α the list would be represented in memory as $\mathbf{x} + \mathbf{a} + \mathbf{a}*\mathbf{x} + \mathbf{b} + \mathbf{b}*\mathbf{a} + \mathbf{c} + \mathbf{c}*\mathbf{b} + \mathbf{d} + \mathbf{d}*\mathbf{c} + \mathbf{e} + \mathbf{e}*\mathbf{d}$ (and possibly $\mathbf{y}$ and $\mathbf{y}*\mathbf{e}$ if we had a stop signal Y). The chain is *XA*, *AB*, *BC*, *CD*, *DE* (and possibly *EY*).

Retrieval

For the recall of associative information, if $\mathbf{f}_k$ (the left-hand member of the kth pair) is the probe and δ' is an approximation to δ, the delta or "unit" vectors, then we would have

$$\mathbf{f}_k\#\mathbf{M} = \mathbf{f}_k\#\{(\mathbf{f}_k*\mathbf{g}_k)+\ldots\} \cong \mathbf{f}_k\#(\mathbf{f}_k*\mathbf{g}_k)$$
$$= (\mathbf{f}_k\#\mathbf{f}_k)*\mathbf{g}_k = \delta'*\mathbf{g}_k = \mathbf{g}'_k \rightarrow \mathbf{g}_k.^2$$

The arrow denotes deblurring, where the approximation $\mathbf{g}'_k$ must be deblurred to $\mathbf{g}_k$ for recall to be successful. Serial recall will be discussed in the section dealing with the chaining model.

The comparison process for recognition is assumed to be the dot (or inner) product of the probe vector with the memory vector. If $f(i)$ is the ith element of item vector $\mathbf{f}$ and $g(i)$ is the ith element of item vector $\mathbf{g}$, then the dot product of the item vectors $\mathbf{f}$ and $\mathbf{g}$ would be

$$\mathbf{f} \cdot \mathbf{g} = \sum_{i=1}^{N} f(i)g(i).$$

Since item vectors are random vectors, each $f(i)$ is a random variable, and since there are N random variables in each item vector, if we use Z for a random variable, then the expectation or expected value E of the dot product of an item vector with itself is

$$E[\mathbf{f} \cdot \mathbf{f}] = E\left[\sum_{i=1}^{N}\{f(i)f(i)\}\right] = NE[Z^2]$$

$$= N\sigma^2,$$

and since, by our representation assumption, $\sigma^2 = 1/N$, we have

$$E[\mathbf{f} \cdot \mathbf{f}] = 1.$$

For any constant c, $E[c\mathbf{f}] = cE[\mathbf{f}]$, so if we use α_k to mean α as a function of k, and if L is list length and $\alpha_k = \alpha^{L-k}$, then we have

$$E[\mathbf{f}_k \cdot \mathbf{M}] = \begin{Bmatrix} \alpha_k, & k \in L \\ 0, & k \notin L \end{Bmatrix}. \qquad (3)$$

Thus, we have the old- and new-item strength distributions for a signal-detection analysis of recognition memory, and the predicted d' values can be computed from the parameters of the model (N and α). This analysis of recognition memory is the matched-filter model of Anderson (1973) translated into TODAM terminology.

System Architecture

The system architecture is shown in Figure 1, which shows the interconnections between the perceptual or P-system, the query or Q-system, the response or R-system, and working memory. The P-system is for encoding; conceptually, it is where real-world stimuli are mapped into TODAM vectors. The Q-system is where the memory vector lives, and the R-system is for output processes—for instance, it is where deblurring goes on. Working memory is assumed to consist of five "slots," or arrays, for vector processing; it is where all the TODAM operations (convolution, correlation, summation, and the dot product) are carried out.

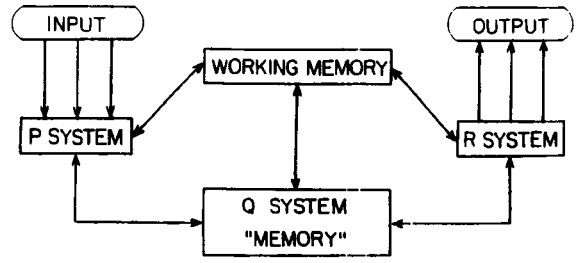

Figure 1. System architecture for the chunking model, showing the relations between the perceptual (P), query (Q), and response (R) systems and working memory. From "A distributed memory model for serial-order information," by B. B. Murdock, 1983, *Psychological Review*, 90, p. 321, Figure 2. Copyright 1983 by the American Psychological Association. Reprinted by permission.

The main point of Figure 1 is to highlight the division of labor that must occur. Encoding goes on in the P-system, and we do not model that; we take item vectors as given. Likewise, we do not model deblurring; we assign it to the R-system.[3] This division of labor allows us to focus on the Q-system and working memory.

THE CHAINING MODEL

The chaining model (Lewandowsky & Murdock, 1989) includes item and serial-order information in the storage equation (Equation 2). The item information is the single item (i.e., $\mathbf{f}_j$), and the serial-order information is the association of the current item with the previous item (i.e., $\mathbf{f}_j*\mathbf{f}_{j-1}$). The former underlies item recognition, as in a Sternberg task (Sternberg, 1969), whereas the latter makes possible serial (and probe) recall. For item recognition, the analysis is straightforward: the dot product is the comparison operation; that is, the dot product of the probe item with the memory vector. Over trials, this gives rise to the old- and new-item strength distributions which form the basis for a signal-detection analysis, as shown in Equation 3.

Serial recall proceeds sequentially. The start signal retrieves the first item, the first item retrieves the second item, the second item retrieves the third item, and so on. More generally,

$$\mathbf{f}_k\#\mathbf{M} = \mathbf{f}'_{k-1} + \mathbf{f}'_{k+1} \rightarrow \mathbf{f}_{k+1}.$$

That is, with starting signal X and list $ABCDE$, $\mathbf{x}\#\mathbf{M} = \mathbf{a}' \rightarrow \mathbf{a}$, $\mathbf{a}\#\mathbf{M} = \mathbf{x}' + \mathbf{b}' \rightarrow \mathbf{b}$, $\mathbf{b}\#\mathbf{M} = \mathbf{a}' + \mathbf{c}' \rightarrow \mathbf{c}$, and so on. Why does $\mathbf{b}\#\mathbf{M} = \mathbf{a}' + \mathbf{c}'$? Because convolution is commutative, and both $\mathbf{a}*\mathbf{b}$ and $\mathbf{b}*\mathbf{c}$ are in the memory vector. Thus, not only does $\mathbf{b}\#(\mathbf{b}*\mathbf{c}) = \mathbf{c}'$, but also $\mathbf{b}\#(\mathbf{a}*\mathbf{b}) = \mathbf{b}\#(\mathbf{b}*\mathbf{a}) = \mathbf{a}'$.

How the backward association (i.e., **a'**) is rejected in the deblurring process was not specified in the original model. However, in response to criticisms of the chaining model by Mewhort, Popham, and James (1994) and Nairne and Neath (1994), Lewandowsky and Li (1994) suggested a possible solution. They proposed a deblurring mechanism for the R-system and used Hebbian antilearning to eliminate items that had already been recalled.

A noteworthy characteristic of the chaining model is that the chain is stronger than its weakest link. If an item is not recalled (i.e., if $\mathbf{f}'_{k+1}$ is not deblurred to $\mathbf{f}_{k+1}$), then one can still use the retrieved item (i.e., $\mathbf{f}'_{k+1}$) as the cue for the next recall. Lewandowsky and Li (1994) showed that when this is done, recall is generally above chance.

The chaining model includes attentional parameters that give rise to primacy effects in recall but to recency effects in recognition. In particular, γ is the attentional parameter for item information, and Ω is the attentional parameter for serial-order information. These parameters are functions of serial position, so the more complete storage equation would be

$$\mathbf{M}_j = \alpha\mathbf{M}_{j-1} + \gamma_j\mathbf{f}_j + \Omega_j(\mathbf{f}_j * \mathbf{f}_{j-1}),$$

where

$$\Omega = \begin{cases} 1, & j=1 \\ \Omega_0 e^{\lambda(j-2)}, & j>1 \end{cases},$$

where parameter λ determines the rate of change of Ω over serial position, and Ω_0 is its starting value.[4] With a limited-capacity assumption, γ and Ω sum to 1.0, so $\gamma_j = 1-\Omega_j$. The parameters λ and Ω_0 would control the relative allocation of attention between item and serial-order information.

Perhaps the main strength of the chaining model is that it is able to account for a large number of empirical effects at a quantitative level with a fairly small number of free parameters (see Lewandowsky & Murdock, 1989, for details). As noted, it has been criticized by Mewhort et al. (1994) and Nairne and Neath (1994), but these criticisms have been answered by Lewandowsky and Li (1994). It gave rise to some novel predictions about the experimental separation of primacy and recency under forward and backward recall—predictions that were confirmed qualitatively, but not quantitatively, by Li and Lewandowsky (1993). The chaining model does have some problems, however; they will be discussed later.

THE CHUNKING MODEL

Multiple Convolutions, *n*-grams, and Chunks

The chunking model uses multiple convolutions, *n*-grams, and chunks for serial-order information. If a*b is a two-way convolution, then **a*b*c** would be a three-way convolution, **a*b*c*d** would be a four-way convolution, and so on. More generally, if we use the symbol X for multiple convolution, then we can represent an *n*-way multiple convolutions as

$$\mathop{\mathbf{X}}_{i=1}^{n}\mathbf{f}_i = \mathbf{f}_1 * \mathbf{f}_2 * \ldots * \mathbf{f}_n.$$

We can also have multiple autoconvolutions, where the same item is convolved with itself n times. Thus, we could have $\mathbf{f}(n=1)$, $\mathbf{f}*\mathbf{f}(n=2)$, $\mathbf{f}*\mathbf{f}*\mathbf{f}(n=3)$ or, more generally, if we use a "star" notation, we can define an n-way multiple autoassociation as

$$\mathbf{f}^{*n} = \mathbf{f}*\mathbf{f}*\ldots*\mathbf{f} \quad (n \text{ times})$$

The star notation is (not accidentally) like exponentiation: just as $a^3 = a*a*a$, so $f^{*3} = f*f*f$.

Both multiple convolutions and multiple autoconvolutions are still vectors, although their dimensionality (number of features, or N) increases with n. However, the increase in dimensionality is linear in n, making multiple convolutions and multiple autoconvolutions ideally suited for the chunking model. By comparison, in a matrix model, one forms an n-way outer-product matrix, and the increase in dimensionality is N^n, so the computational demands could be excessive for large N or n.

Chunks can be of various sizes, but regardless of size, a multiple convolution or autoconvolution is still a single unit. Further, the linear increase in N with n means that the storage demand will not be too great. Of course, the noise level does increase with n as well, but that is probably quite desirable for a psychological model of chunking.

An n-gram is defined as the n-way autoconvolution of the sum of n item vectors. Thus, the single item **a** would be an engram ($n=1$); $(\mathbf{a}+\mathbf{b})^{*2} = \mathbf{a}*\mathbf{a} + \mathbf{b}*\mathbf{b} + 2(\mathbf{a}*\mathbf{b})$ would be a digram ($n=2$); $(\mathbf{a}+\mathbf{b}+\mathbf{c})^{*3} = \mathbf{a}*\mathbf{a}*\mathbf{a} + \mathbf{b}*\mathbf{b}*\mathbf{b} + \mathbf{c}*\mathbf{c}*\mathbf{c} + 3(\mathbf{a}*\mathbf{a}*\mathbf{b} + \mathbf{a}*\mathbf{a}*\mathbf{c} + \mathbf{a}*\mathbf{b}*\mathbf{b} + \mathbf{b}*\mathbf{b}*\mathbf{c} + \mathbf{a}*\mathbf{c}*\mathbf{c} + \mathbf{b}*\mathbf{c}*\mathbf{c}) + 6(\mathbf{a}*\mathbf{b}*\mathbf{c})$ would be a trigram ($n=3$); or, more generally, if we symbolize an n-gram as $\mathbf{G}(n)$, then

$$\mathbf{G}(n) = \left(\sum_{i=1}^{n}\mathbf{f}_i\right)^{*n}.$$

An n-gram consists of the sum of autoconvolutions and multiple convolutions. Autoconvolutions give the chunking model redintegrative capabilities. Thus, if $p\mathbf{f}$ is a "reduced" version of $\mathbf{f}$ (i.e., if some proportion p of the features of $\mathbf{f}$ are present, and the remainder $(1-p)$ are absent, or zero), then

$$p\mathbf{f}\#(\mathbf{f}*\mathbf{f}) = p\mathbf{f}'.$$

That is, $p\mathbf{f}$ correlated with the autoconvolution will redintegrate or reconstruct an approximation to $\mathbf{f}$, and it seems reasonable to assume that the probability of deblurring will be proportional to p. The multiple convolutions can be used in serial recall, and this will be explained shortly.

A chunk is defined as the sum of n-grams 1 to n. Thus, if we symbolize a chunk as $\mathbf{C}(n)$, then

$$\mathbf{C}(n) = \sum_{i=1}^{n} \mathbf{G}(i) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} \mathbf{f}_j \right)^{*i}.$$

Since n-grams are composed of sums of multiple convolutions and multiple autoconvolutions, they are vectors (their dimensionality is approximately nN), and since chunks are sums of n-grams, they, too, are still vectors whose dimensionality is that of the highest order n-gram. Thus, a chunk is a single unit, just as an item vector is a single unit.

Retrieval

Not only is a chunk a single unit, it can also be unpacked into its constituents. We use a scheme developed by Liepa (1977), wherein the multiple convolution of the items recalled to date functions as the retrieval cue for the recall of the next item. Thus, if δ is the delta vector (a vector whose middle element is 1 and whose flanking elements are all zero), then for chunk $\mathbf{C}(n)$,

$$\begin{aligned} \delta \# \mathbf{C}(n) &= \mathbf{a}' \rightarrow \mathbf{a}, \\ \mathbf{a} \# \mathbf{C}(n) &= \mathbf{b}' \rightarrow \mathbf{b}, \\ (\mathbf{a} * \mathbf{b}) \# \mathbf{C}(n) &= \mathbf{c}' \rightarrow \mathbf{c}, \\ (\mathbf{a} * \mathbf{b} * \mathbf{c}) \# \mathbf{C}(n) &= \mathbf{d}' \rightarrow \mathbf{d}, \text{ and so on.} \end{aligned}$$

Actually, this is an oversimplification; an n-way multiple convolution correlated with a larger-than-n chunk retrieves a linear combination of the first $n + 1$ items, but the strength of the $(n+1)$th item is always twice the strength of all the earlier items, regardless of n. Thus,

$$\begin{aligned} \mathbf{a} \# \mathbf{C}(n) &= \mathbf{a}' + 2\mathbf{b}' \rightarrow \mathbf{b}, \\ (\mathbf{a} * \mathbf{b}) \# \mathbf{C}(n) &= 3\mathbf{a}' + 3\mathbf{b}' + 6\mathbf{c}' \rightarrow \mathbf{c}, \end{aligned}$$

or, in general,

$$\left(\sum_{i=1}^{m} \mathbf{f}_i \right) \# \mathbf{C}(n) = \sum_{i=1}^{m+1} c_i \mathbf{f}_i, \; n > m,$$

where $c_{m+1} = 2c_j$, $1 \leq j \leq m$.

As in the chaining model, what is retrieved on one cycle is plugged into the retrieval cue for the next cycle; in the chaining model, however, the retrieval cue is just the retrieved item itself, whereas in the chunking model, it is the convolution of the retrieved item with all the items that preceded it. This has a complex cumulative effect. The more items that are correctly recalled, the less impact any single incorrect recall will have, but on the other hand, the more items that are incorrectly recalled to date, the less likely one will be to recall the next item correctly.

Parameters

A simulation of the model will be reported shortly, but first the parameters of the model need to be discussed. The chunking operations are assumed to go on in working memory (see Murdock, 1992, for details), and we need the forgetting parameter α. This gives us α and N, the number of features in the item vectors, but we also need p for probabilistic encoding. In order for learning to occur, we assume that, at the time of encoding, each feature is either included in the item vector with probability p or not included (set to zero) with probability $1-p$. Thus, we have three parameters: N, α, and p.[5]

Normalization

The other point that needs to be mentioned concerns normalization. Like item vectors, n-grams and chunks are statistically normalized to 1.0. This way, everything is in unit vectors, regardless of whether we are dealing with the item level, the n-gram level, or the chunk level. In the simulations to be reported, scale factors were used; for details, see Murdock (1995).

Simulations

For the simulations, we did both recognition and recall to see whether the model captures the interaction between primacy/recency and recall/recognition found in the literature (Murdock, 1995). We varied chunk size between one and seven, and in each case, the chunks were always the sum of n-grams, where the n-grams used the probabilistically encoded item vectors. For recognition, we took the dot product of the probe item with the chunk where we used both old- and new-item probes with old-item probes from all serial positions, and we computed means and variances over replications for a d' analysis. For recall, we used a scoring rule, whereby the best match of the retrieved information to the n items in the chunk was deemed to be the item recalled. If all dot-product

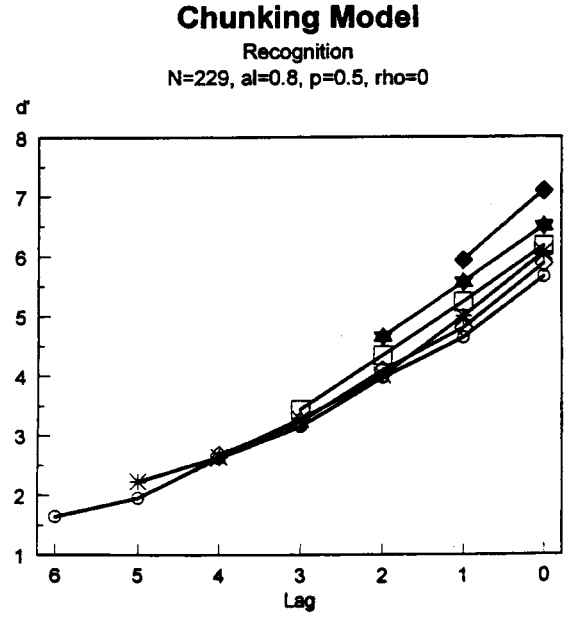

Figure 2. d' as a function of lag from a simulation of the chunking model.

values were negative, we scored an omission. If the correct item was the best match, it was plugged into the retrieval cue for the next item; otherwise, the retrieved information was plugged into the retrieval cue for the next item. We ran 500 replications, with parameter values $N = 229$, $\alpha = 0.8$, and $p = .5$.

The results for recognition are shown in Figure 2, while Figure 3 shows the results for recall. As they should do, recognition shows recency and recall shows primacy. Figure 4 shows recall data from Drewnowski and Murdock (1980) from a memory-span experiment using a staircase method (for each subject, list length was increased by one after each correct recall, but was decreased by one after each incorrect recall). There would seem to be quite good agreement between the simulation and the experimental data.

The simulation demonstrates that lawful behavior can result, even in the face of a very low signal-to-noise ratio. The number of terms (types, not tokens) in a chunk is 1, 4, 14, 49, and 175 for ns of 1, 2, 3, 4, and 5; the larger the value of n, the more each individual term is attenuated by the normalization process. One could well wonder whether anything would show up for chunks of three or four or more. In fact, the data are quite regular, which suggests that the memory system can still function with a high noise level.

The data in Figure 3 show a recency effect for longer lists, but the effect is unreliable. Across many simulations with a variety of parameter values, it was often absent. Serial-recall data often show a small recency effect; it does not show up in Figure 4 because of the scoring method used. Recall probability at early serial positions is too low for the longer lists, but subjects probably

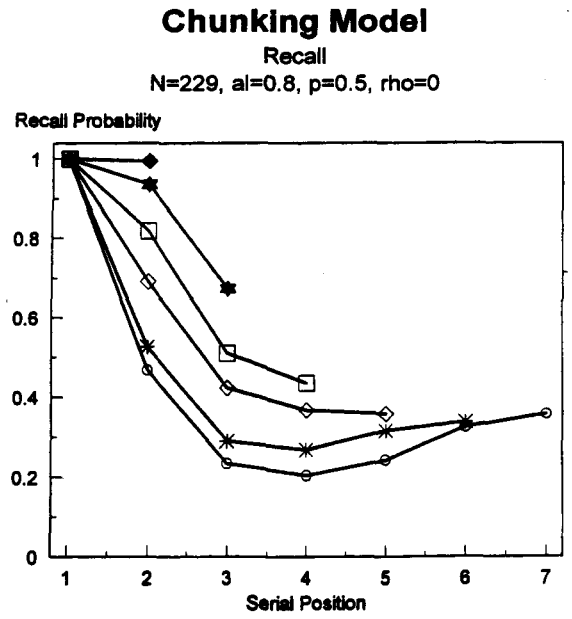

Figure 3. Probability of recall as a function of serial position for lists of 1–7 items from a simulation of the chunking model.

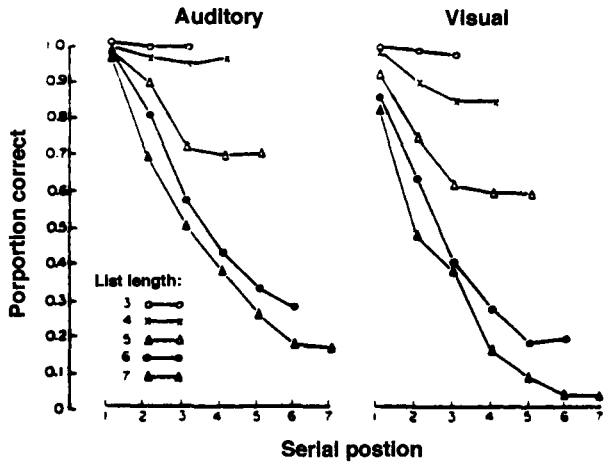

Figure 4. Performance on a memory-span task with auditory or visual presentation. Data from "The Role of Auditory Features in Memory Span for Words," by A. Drewnowski & B. B. Murdock, 1980, *Journal of Experimental Psychology: Human Learning & Memory, 6*, p. 324. Copyright 1980 by the American Psychological Association. Adapted by permission.

break six- and seven-item lists into chunks of three or four (Donaldson, 1980; Ryan, 1969; Wickelgren, 1964).

Like the chaining model, the chunking model can account (at least qualitatively) for a range of data (Murdock, 1993b), and some analytic expressions are available, so we do not always need simulations to know what the model predicts. It only has three or four parameters, so it is quite parsimonious in this regard. It very clearly embodies the notion of a chunk, in that it can function as a unit but it can also be unpacked into its constituents if necessary (Miller, 1956). However, it, too, has some problems, and these will be discussed before we go on to discuss the power-set model.

Problems

The chunking model has at least three major problems, concerning backward recall, probe tests, and error gradients. The only way it can do backward recall is by multiple forward scans, but since subjects can do backward recall with short lists with considerable facility, the question arises as to whether they use multiple forward scans also. Multiple forward scans may well occur sometimes—as in, for example, repeating the alphabet backward—but that is probably not the only mechanism we use. As for probe recall, there is a way this could be done, but it is somewhat complex; it will be discussed later. Finally, the chunking model cannot give a satisfactory account of error gradients (e.g., the distance functions reported by Lee & Estes, 1977, and Nairne, 1990). In the simulation, they were consistently wrong. There was almost always a big recency effect at each serial position, and the intrusions did not decrease in frequency with distance from the target position itself, as they should have done.

In the same way that problems with the chaining model led us to develop the chunking model, so problems with the chunking model led us to develop the power-set model, to which we next turn.

THE POWER-SET MODEL

Storage Equation

In the chaining model, the current item is associated with the prior item, and that (plus the current item) is added to the memory vector. Suppose instead that the current item is associated with *all* the prior items, that the association is normalized, and that it is that (plus the current item) that is added to the memory vector; this would be the power-set model. Specifically,

$$\mathbf{M}_j = \alpha\mathbf{M}_{j-1} + \mathbf{f}_j + |\mathbf{f}_j * \mathbf{M}_{j-1}|, \qquad (4)$$

where the vertical bars denote normalization.[6]

Assume that we start with the first item **a**, so that $\mathbf{M}_1 = \mathbf{a}$. Then

$$\mathbf{M}_2 = \alpha\mathbf{a} + \mathbf{b} + |\mathbf{a}*\mathbf{b}|.$$

We can rewrite $|\mathbf{a}*\mathbf{b}|$ as $h_2(\mathbf{a}*\mathbf{b})$, where h_2 is the normalization constant for a two-way multiple convolution, so that we have

$$\mathbf{M}_2 = \alpha\mathbf{a} + \mathbf{b} + h_2(\mathbf{a}*\mathbf{b}).$$

Then

$$\begin{aligned}\mathbf{M}_3 &= \alpha^2\mathbf{a} + \alpha\mathbf{b} + \mathbf{c} + \alpha h_2(\mathbf{a}*\mathbf{b}) + h_3(\mathbf{c}*\mathbf{M}_2)\\ &= \alpha^2\mathbf{a} + \alpha\mathbf{b} + \mathbf{c} + \alpha h_2(\mathbf{a}*\mathbf{b}) + \alpha h_3(\mathbf{a}*\mathbf{c})\\ &\quad + h_3(\mathbf{b}*\mathbf{c}) + h_2h_3(\mathbf{a}*\mathbf{b}*\mathbf{c}).\end{aligned}$$

Thus, $\mathbf{M}_3$ contains a linear combination of *A*, *B*, *C*, *AB*, *AC*, *BC*, and *ABC*, where the weighting coefficients are the products of the appropriate values of the forgetting factor (α) and normalization factor (h). More generally, for any value of n, the memory vector will always contain a linear combination of the power set of n items (less the empty set ϕ), where convolution replaces the set product as the binding operation; hence the name (see also Hayes-Roth & Hayes-Roth, 1977).

What is the benefit of this? First, it gives us item information. Each individual item will be included in the memory vector, and the weighting coefficient for the ith item will be α_i, where $\alpha_i = \alpha^{n-i}$. Thus, like the chaining model, there is pure recency (i.e., the last item is the strongest, and the strength of earlier items falls off geometrically with distance from the end). The recency effect is automatically produced by the operation of the forgetting parameter α.

Second, it also gives us the multiple convolutions that could underlie serial-order information. Assume serial-recall proceeds as in the chunking model; the retrieval cue is the multiple convolution of all the items retrieved to date. By the filter principle (Murdock, 1993a), a multiple convolution of n items correlated with the memory vector retrieves only single items from order $(n+1)$ multiple convolutions, so we can partition the set to analyze recall. Since

$$\mathbf{M}_n = \sum_{i=1}^{n}\alpha_i\mathbf{f}_i + \left\{\sum_{i=1}^{n}\alpha_{i+1}\prod_{j=1}^{i}h_j\mathbf{f}_j\right\} * \left\{\sum_{k=i+1}^{n}h_k\mathbf{f}_k\right\} + \text{Noise}, \qquad (5)$$

this means that at any point in the sequence, the multiple convolution $(\prod_{j=i}^{i}h_j\mathbf{f}_j)$ that we use as retrieval cue retrieves a linear combination of all the remaining items $(\sum_{k=i+1}^{n}h_k\mathbf{f}_k)$, where the weights are the normalization factor h_k. Since the normalization factor decreases monotonically with serial position, not only is the target item (i.e., the next in line) the strongest, we also get the right gradient. The strength of the retrieved items decreases with the degree of remoteness.

Remote Associations

Another way to put this is in terms of remote associations. Remote associations have been part of the lore of serial learning ever since Ebbinghaus (1913/1964), and even the critics seem to have recanted (e.g., Slamecka, 1985). Figure 5 illustrates how forward remote associations would look in the power-set model. Each "item" is associated with all of the subsequent items, where here an item is a multiple convolution, and the strength of the association decreases with the degree of remoteness.

The gradients of remote association depend upon h, the normalization constant. This constant h depends on α as well as on serial position, and illustrative normalization functions for four values of α are shown in Figure 6. The explicit function[7] is

$$h_j = \left\{\begin{array}{ll} 1, & j = 1 \\ \sqrt{\dfrac{1-\alpha^2}{2-\left(\alpha^{2n-2)} + \alpha^{2n-4}\right)}}, & j > 1 \end{array}\right\}. \qquad (6)$$

As can be seen, the normalization function is quite flat after the third serial position, but its slope is proportional to α.

The fact that the strength of the retrieval cue (the multiple convolution) is also a function of α and h (Equation 5) is probably not important, because the retrieval cue can always be normalized before the correlation is done. However, the target item cannot be normalized be-

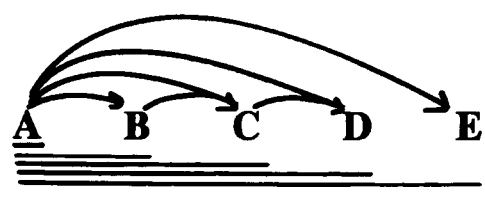

Figure 5. Gradients of remote forward associations for the power-set model.

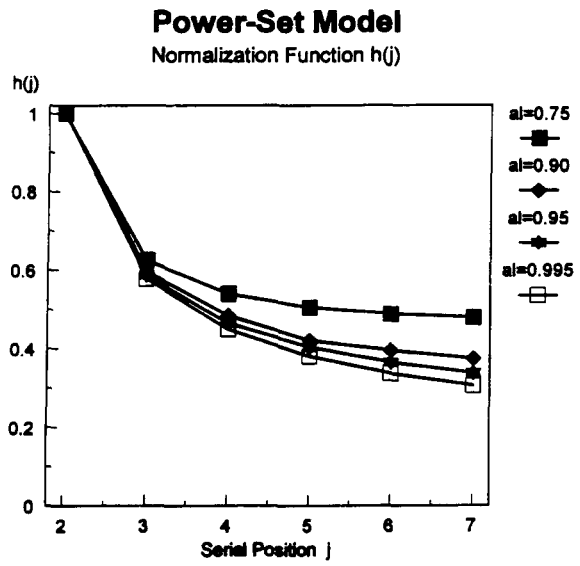

Figure 6. The normalization function for the power-set model for four values of α.

cause it is embedded in a linear combination of all the candidate items, and any normalization would shift the gradient up or down, but not affect its slope.

Backward Recall

What about backward recall? In backward recall, the subject is instructed to start with the last item and work forward. It turns out that we have a counterpart to Equation 5; that is, for backward serial position b,

$$\mathbf{M}_n = \sum_{b=1}^{n} \alpha_b \mathbf{f}_b + \left\{ \sum_{b=1}^{n-1} \sum_{j=1}^{b} h_b \mathbf{f}_b \right\} * \left\{ \sum_{k=b+1}^{n} \alpha_k \mathbf{f}_k \right\} + \text{Noise}, \quad (7)$$

where, again, $\alpha_k = \alpha^{n-k}$. Consequently, at each serial position, there is also a backward gradient of remote associations, but this time the weighting coefficients of the linear combination are a function of α, not of h. Figure 7 shows the analogous diagram for remote backward associations. Again, there is the right gradient, so in backward recall, as in forward recall, the adjacent item is the strongest, and strength is inversely proportional to remoteness.

Since the remote-association gradients for forward and backward recall have slopes that are functions of different parameters (h and α, respectively), it is not unreasonable to expect that an experimental separation might occur. Li and Lewandowsky (1993) have shown that there are variables which affect one and not the other. At least at the qualitative level, this result is consistent with the power-set model.

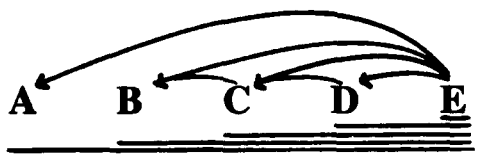

Figure 7. Gradients of remote backward associations for the power-set model.

Initiation of Recall

So far, we have avoided mention of one potential problem—namely, the initiation of recall. If the delta vector is correlated with the memory vector, one will again retrieve a linear combination of items, but this time the items are the item information, so the first item is the weakest; that is,

$$\delta \# \mathbf{M}_n = \sum_{i=1}^{n} \alpha_i \mathbf{f}_i ,$$

where $\alpha_i = \alpha^{n-i}$. This is not what we want; it is one thing to postulate a deblurring mechanism that is designed to deblur a linear combination of items to the strongest item, but it is quite unreasonable to expect it also to be able to deblur a linear combination to its weakest item.

One possible solution is to assume that the first item is special; not only is it added to the memory vector, it is also associated with a context vector and stored in working memory.[8] Then we would initiate recall by correlating the context vector (or, more realistically, a degraded version of the context vector) with the memory vector, and then continue as before. Since subsequent retrievals would always generate a linear combination with a primacy gradient (Figure 6), the problem would not arise once the first item had been recalled.

Simulation

Even though in principle analytic solutions for the power-set model exist, predicted recall values at any serial position are different for every possible cue combination (which is what the plug-ins were), so it seemed easier to simulate the model. In the simulation, we followed the procedure described above. The memory vector was set up according to Equation 4, recall was started by correlating the context vector (simply another item vector that was independent of the list items) with the memory vector, and from there on, we used the convolution of the items recalled to date as the cue for the next item. We used the same scoring rule that we had used in the simulation of the chunking model, but we always used the best match as the plug-in. Recall terminated following an omission.

The storage flowchart for the simulation is shown in Figure 8. As in the chunking model, we assume that there are five registers in working memory, which we label v, w, x, y, and z. We write $z \leftarrow x*y$, to mean that the contents of x are convolved with the contents of y and that the result is stored in z. The flowchart for retrieval is similar, except that we need a deblurring algorithm or

$w \leftarrow Context$

$z \leftarrow Word[i]$

$v \leftarrow w * z$

$for(i = 2,n)$

$w \leftarrow Word[i]$

$x \leftarrow z$

$y \leftarrow w * x$

$y \leftarrow |y|$

$z \leftarrow \alpha z + w + y$

$Done$

Figure 8. Flowchart for the power-set model for encoding a list of *n* items.

a scoring rule; we assume that this does not disturb working memory. We used set sizes 1–5 with forward and backward recall, with parameter values $N = 99$, $\alpha = .75$, $p = .5$, and $\rho = 0$.[9] The only difference between the simulations of forward and backward recall was that for backward recall, we started with the last item rather than with the context vector. The most recent item is always in working memory (in Register w), so we simply used that to initiate the recall process (i.e., to correlate Register w with the chunk); thereafter, everything was the same as in forward recall.

The results for 100 replications, which are shown in Figures 9 and 10, are clear. There is a monotonic primacy effect for forward recall, but a monotonic recency effect for backward recall, and the backward-recall data agree well with data reported by Madigan (1971). In both cases, too, a family of curves exists, one curve stacked above another, and these are ordered by n. The data for forward recall are quite similar to the data from the simulation of the chunking model, but there is no simulation of backward recall for the chunking model because, as noted, the chunking model can only do backward recall by multiple forward passes. In the power-set model, except for the starting value, backward recall is implemented in exactly the same way as forward recall. The resulting data are almost mirror images of one another.

Advantages of the Power-Set Model

1. The power-set model provides a principled account of forward and backward remote associations. Quite literally, everything is associated to everything where the association is by convolution. The multiple convolution of all items recalled to date is associated with each of the remaining items, but with decreasing strength as the degree of remoteness increases. This occurs in both a forward and a backward direction (see Figures 5 and 7). We have known about remote associations for over 100 years, but the power-set model shows how they could come about.

2. The power-set model also provides a principled account of the intrusion gradients that are found in serial recall or probe recall. The probability of an intrusion from any given serial position decreases as its distance from the target serial position increases (see Lee & Estes, 1977, or Nairne, 1990, for serial-recall data, and see Murdock, 1968, for probe-recall data). Qualitatively, these gradients are exactly what one would expect from the hypothesized remote associations. Of course, for quantitative fits, more will be involved, but at least this provides a first step.

3. The power-set model implements the classic Conrad interpretation of order errors in serial recall. According to Conrad (1965), there is no loss of order information in storage; instead, an intrusion occurs (for whatever reason), and then, immediately or eventually, the original target item shows up in recall as a second intrusion, because subjects edit out items they have already recalled. Thus, two items have been transposed, but it is not a rearrangement in storage;[10] rather, in recall, one error leads to another. In the power-set model, whatever item is recalled is used as the plug-in for the next retrieval cue, so again, in the model, one error could lead to another.

4. The power-set model provides a possible explanation for the experimental dissociation of forward and backward recall reported by Li and Lewandowsky (1993). The remote-association gradients for forward recall depend on the normalization factor h, whereas the remote-association gradients for backward recall depend on the forgetting parameter α; these, in turn, determine primacy and recency, respectively. Since at any serial position the normalization factor depends on α, p, and ρ,[11] any

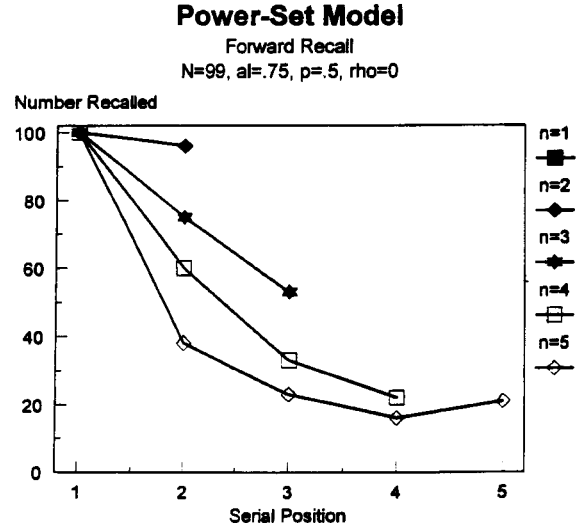

Figure 9. Results from a simulation of the power-set model which shows number recalled as a function of serial position for forward recall.

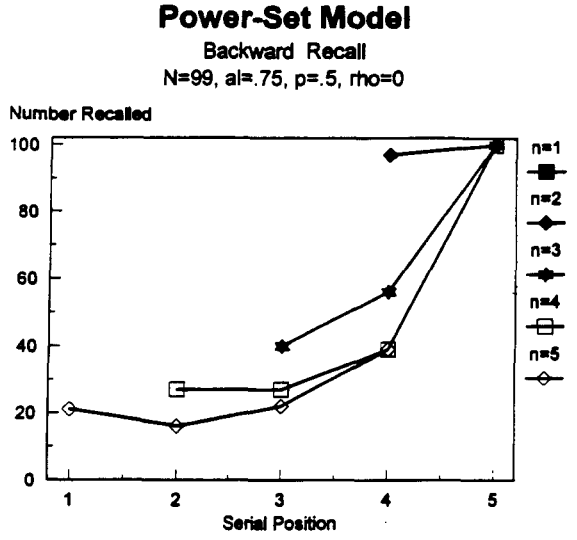

Figure 10. Results from a simulation of the power-set model which shows number recalled as a function of serial position for backward recall.

variable that affects p or ρ should affect forward recall but not backward recall. Li and Lewandowsky found that an intralist distractor task affected forward recall but not backward recall.

5. Since every item is associated with every other item, probe tests are not a problem for the power-set model. There should be intrusion gradients in each direction around the probe, and a proof of this assertion is given in the Appendix. Bidirectional gradients are the typical result (e.g., Murdock, 1968). However, the confusion matrices from experimental data are sharper than the remote-association gradients of the power-set model, so some additional tuning mechanism will probably be necessary. Since, unlike Shiffrin and Cook's (1978) model or the perturbation model of Estes (1972), there are no item-to-position associations, it is not clear how subjects could respond to a position probe. One possibility is that it is mediated by implicit recall, but this seems unlikely, because the results are very similar for sequential probes and position probes (Murdock, 1968).

6. All of the operations can be done with five registers in working memory and, with one exception (the back transfer from z to x; see Figure 8), can be done in a left-to-right (or bottom-to-top) order. Thus, if we wanted to implement the power-set model in a connectionist network, we could consider Registers v, w, x, y, and z as successive layers in a connectionist network. The function of the network would be to implement the operations in the correct order and, with one exception (the back transfer), it would be strictly a feed-forward network.

7. The power-set model predicts associative symmetry for pairs but not for triples (or higher), and this is exactly what was found by Kahana (1995). After a pair has been presented,

$$\mathbf{M}_2 = \alpha\mathbf{a} + \mathbf{b} + (\mathbf{a} * \mathbf{b}),$$

because $h_2 = 1$ (see Figure 6). Consequently, $\mathbf{a}\#\mathbf{M}_2 = \mathbf{b}'$ and $\mathbf{b}\#\mathbf{M}_2 = \mathbf{a}'$, so unless deblurring is easier for $\mathbf{b}'$ than for $\mathbf{a}'$, associative symmetry for pairs would be expected.[12] However, for $n > 2$, the forgetting parameter α and the normalization factor h would lead to violations of symmetry under a variety of conditions, and this is exactly what Kahana (1995) found.

8. The model has only four parameters (N, α, p, and ρ). The parameter N affects the noise level; the parameter α affects recognition directly and recall indirectly; the parameter p affects the learning rate; and the parameter ρ reflects interitem similarity, which affects, among other things, the "sharpness" of the stored-order information. Unlike the chaining model, none of the parameters varies with serial position, so it is the model that is doing the work, not the parameters.

9. Given the representation assumption and the convolution-correlation formalism, everything follows from the storage equation (Equation 5). No additional assumptions or mechanisms are required. The storage equation is a simple recurrence relation, and once that has been specified, everything else follows. Of course, one has to flesh out some of the details to simulate the model or work out analytic predictions, but the basic structure of the model is completely determined by the storage equation.

10. The model can easily fit into the TODAM2 framework, so item and associative information fall out as special cases, and any serial string has the potential to function as a chunk. If item information is all that is wanted (i.e., if people do not have to store associative or serial information), then Equation 5 reduces to

$$\mathbf{M}_j = \alpha\mathbf{M}_{j-1} + \mathbf{f}_j,$$

which, of course, is the matched-filter model of Anderson (1973). If people store item and associative information but not serial-order information, then Equation 5 reduces to

$$\mathbf{M}_j = \alpha^2\mathbf{M}_{j-2} + \alpha\mathbf{f}_{j-1} + \mathbf{f}_j + (\mathbf{f}_{j-1} * \mathbf{f}_j),$$

which is only slightly different from the standard storage equation we have always used for the storage of item and associative information.[13] Thus, the storage equation of the power-set model could be a general formulation which subsumes the storage of item and associative information as special cases.

COMPARISON

We have discussed three possible models for serial-order information: the chaining model, the chunking model, and the power-set model. What are the relative capabilities of each model? Table 1 evaluates the relative capabilities of the three models on seven basic serial-order tasks: forward recall, backward recall, recall of

missing items, sequential probes, positional probes, serial-to-paired-associate transfer, and item recognition.

Forward Recall

For serial recall, all three models retrieve a linear combination of items—but of *which* items? For the chaining model, the answer is the items immediately before and after the probe ($c_{k-1}\mathbf{f}_{k-1}$ and $c_{k+1}\mathbf{f}_{k+1}$); for the chunking model, it is all the items retrieved to date plus the target item (i.e., $\sum_{i=1}^{k+1} c_i\mathbf{f}_i$); and for the power-set model, it is all the items yet to be retrieved (i.e., $\sum_{i=k+1}^{n} c_i\mathbf{f}_i$). For the chunking model, the target item is always twice as strong as its competitors, whereas for the power-set model, the gradients (c_i) are relatively flat (see Figure 6). For the chaining model, c_{k-1} and c_{k+1} depend on the parameter values Ω_0 and λ, but c_{k-1} is always larger than c_{k+1}.

The chunking model is clearly the best here because not only are the gradients much sharper, the competitors in the linear combination are all items that have already been recalled. As a consequence, they should be relatively easy to attenuate before the deblurring process.[14] For the power-set model, the competitors are all items yet to be recalled, so it is not clear how they could be attenuated. The chaining model can do forward recall, even though the retrieval cue is only a single item. However, what is retrieved is only the correct item and one competitor (the prior item). Since the latter is always stronger than the former, the model must depend on the attenuation process to effect recall, so it is this unspecified attenuation process, and not the serial-order part of the model, that is doing the work.

Backward Recall

As we have mentioned, the only way that the chunking model can do backward recall is by multiple implicit forward scans, and this seems to be a weakness. Both the chaining and the power-set models can do backward recall, and they do it in the same way that they do forward recall. For the chaining model, this time the target item is stronger than the competition, so the deblurring process is not so dependent on an attenuation process.

The chunking model can only do backward recall by multiple forward scans. Is this a weakness of the model? The power-set model can do backward recall as easily, and in exactly the same manner, as it does forward recall; is this a strength of the model? In both cases, the answer is not necessarily. Subjects seem to be about equally accurate on forward and backward recall, even when they are not precued (Li & Lewandowsky, 1993), but in the absence of latency data, we do not know whether both types of recall were done in the same way (or if an equal amount of effort was required by each task). Furthermore, these are episodic tasks; anecdotal evidence suggests that in backward recall of long lists (e.g., the months of the year, or the letters of the alphabet), people do resort to forward scans when backward recall runs into trouble.

It may be that overlearning alters the relation between forward and backward recall. While the structure or coherence of the stored information is similar, if not identical, at first, it changes, and one loses some of the initial facility for backward recall. It is not inconceivable that in some as-yet-unexplained manner, the storage changes from something like that envisioned by the power-set model to something like that envisioned by the chunking model.

Table 1
Relative Capabilities of the Chaining, Chunking, and Power-Set Models on Seven Basic Serial-Order Tasks

Task	Models		
	Chaining	Chunking	Power Set
Forward recall	+	++	+
Backward recall	+	−	+
Recall of missing items	−	+	++
Sequential probe	−	+	+(?)
Position probe	−	−	−
Transfer test	+|−	−|+	+|−
Item recognition	+(?)	+	+

Recall of Missing Items

In the recall of missing items following list presentation (Yntema & Trask, 1963), a subset of the list (generally consisting of all but one of the items) is repeated in a scrambled order, and the task of the subject is to recall the missing item(s). It turns out that the chunking model can do this very easily, while the chaining model can only do it with difficulty, if at all (Murdock, 1992). Since the probe is a subset of the list,

$$\mathop{\mathrm{X}}_{i=1}^{n-1}\mathbf{f}_i \# \mathbf{C}(n) \cong \mathop{\mathrm{X}}_{i=1}^{n-1}\mathbf{f}_i \# \mathop{\mathrm{X}}_{i=1}^{n}\mathbf{f}_i = \mathbf{f}_n' \rightarrow \mathbf{f}_n,$$

and because convolution is commutative, the order of items in the probe is immaterial. Consequently, the chunking model can recall a missing item in a very simple and direct manner.

What about the power-set model? Since the chunk will always include the highest order multiple convolution (i.e., $\mathrm{X}_{i=1}^{n}\mathbf{f}_i$), the power-set model can recall a missing item in exactly the same way as the chunking model. However, suppose there is more than one missing item? Hadley, Healy, and Murdock (1992) found that subjects could also recall two missing items, though performance was poorer with two missing items than with one. Unless one of the missing items is the last item, multiple convolutions will not work for the chunking model, because of the dependence on $\mathbf{C}(n-1)$: since (by assumption) the probe includes $\mathbf{f}_n$, the correlation of $\mathrm{X}_{i=1}^{n-2}\mathbf{f}_i$ with $\mathbf{C}(n-1)$ provides no useful information. However, in the power-set model, the chunk includes all possible $(n-1)$-way multiple convolutions, so if we denote the two missing items as $\mathbf{f}_u$ and $\mathbf{f}_v$, then

$$\mathop{\mathrm{X}}_{i=1}^{n-2}\mathbf{f}_i \# \mathbf{C}(n) \cong \mathop{\mathrm{X}}_{i=1}^{n-2}\mathbf{f}_i \# \sum_k \mathop{\mathrm{X}}_{i=1}^{n-1}\mathbf{f}_{i(k)} = \mathbf{f}_u' + \mathbf{f}_v',$$

so a linear combination of the two missing items is retrieved, regardless of the composition of the probe. This

linear combination can certainly be deblurred to the stronger of the two items and, perhaps, to the weaker of the two items as well on a second pass, but it seems reasonable to assume that there should be less accuracy with two missing items than with one.

Sequential Probes

In the chaining model, for a sequential probe, the probe item is correlated with the memory vector, and this again retrieves

$$c_{k-1}\mathbf{f}'_{k-1} + c_{k+1}\mathbf{f}'_{k+1}.$$

If one had to recall the earlier item (a backward sequential probe), at least some of the time, one could deblur the linear combination to the correct alternative because it is stronger. However, for a forward sequential probe, the target item is always the weaker of the two items, and since there is no context of prior recalls in a sequential probe test, it is not at all clear how the chaining model could do a forward sequential probe. Even if it could, performance on a forward sequential probe should be worse than performance on a backward sequential probe, but this seems unlikely. (I do not know of any studies that have made this comparison.)

The chunking model can handle sequential probes, but it requires a judgment of recency (JOR) first. To illustrate this, suppose we have a three-item chunk which, by definition, consists of the sum of an engram **a**, a digram $(\mathbf{a}*\mathbf{b})^{*2}$, and a trigram $(\mathbf{a}+\mathbf{b}+\mathbf{c})^{*3}$. Suppose **b** is used as the probe for **c**. Now, $\mathbf{b}^{*2}$ correlated with the chunk gives $3\mathbf{a}'+\mathbf{b}'+3\mathbf{c}'$, and $\mathbf{b}^{*1}$ (i.e., **b**) correlated with the chunk gives $2\mathbf{a}'+\mathbf{b}'$, and if the latter is subtracted from the former, the result is a linear combination $(\mathbf{a}'+3\mathbf{c}')$, in which the target item **c** is the strongest. More generally,

$$\left\{\mathbf{f}_k^{*j}\#\mathbf{C}(n)\right\} - \left\{\mathbf{f}_k^{*(j-1)}\#\mathbf{C}(n)\right\} = \sum_{i=1}^{j+1} c_i\mathbf{f}'_i,\ j,k < n,$$

where

$$c_i = \begin{cases} 0, & i = k \\ 1, & i \neq k, \quad i < j+1 \\ j+1, & i = j+1 \end{cases}.$$

The JOR comes in in determining j, the number of times to autoassociate the probe. Note that the strongest item (and, consequently, the most likely item to be recalled) depends on j, not on k (the serial position of the probe), so the JOR is crucial. There should be recency in a sequential probe for two reasons: (1) the later the probe serial position, the more accurate JORs are (Hockley, 1991); and (2) even though the number of competitors increases, the differential strength between the target item and its competitors increases with serial position, so this should benefit deblurring.

For a backward sequential probe,

$$\left\{\mathbf{f}_k^{*(j-2)}\#\mathbf{C}(n)\right\} - \left\{\mathbf{f}_k^{*(j-1)}\#\mathbf{C}(n)\right\} = (j-1)\mathbf{f}_{j-1},$$

so a backward sequential probe can be handled in the same way as a forward sequential probe. Again, and for the same reasons, recency would be expected. This analysis does not have to be restricted to adjacent probes. One could probe for the item one back, two back, three back, or further back, or one ahead, two ahead, three ahead, or further ahead. However, unless the chunk size was quite large, there would be a very limited range that one could use without spilling over a chunk boundary.

Sequential probes for the power-set model are much simpler. Since every item is associated with every other item, if the probe item $\mathbf{f}_k$ were correlated with the memory vector, the result would be a linear combination of all the other items in the list. There is a bidirectional gradient around the probe, wherein the forward gradient is a function of the normalization factor h and the backward gradient is a function of the forgetting parameter α (see the Appendix). Thus, the remote-association gradients predict the right pattern of results for a sequential probe.

On a sequential-probe task, the chunking model is more complex, but more sharply tuned, than the power-set model. That is, the discriminability of the target item would be much higher in the chunking model, so deblurring would be more likely to succeed. While the power-set model is simplicity itself and generates the right sort of gradients, the retrieved information is much flatter. This will have to remain an open question until we know more about deblurring.

Positional Probes

By a positional probe is meant a probe test in which a single item from the list is presented and subjects must give its serial position, or vice versa. The former is tantamount to an absolute JOR, while the latter uses a positional cue to probe for a specific item. Since none of the three models uses item-to-position cues, all must rely on strength or familiarity for a JOR, and on some as-yet-unspecified process to respond to a position probe.

Relative JORs can be explained on the basis of absolute JORs (Lockhart, 1969; though see also Hacker, 1980; Muter, 1979), so the problem is to explain absolute JORs. The classical finding here is the overshoot–undershoot effect first reported by Hinrichs (1970), and it is not at all clear how, or whether, any of these three models could do it. Further, Hockley has some unpublished data which strongly suggest that JORs are different from judgments of frequency (JOFs), which, presumably, are based on strength. All in all, positional probes would seem to be a weak spot for all three models.

Transfer

By *transfer*, we are referring to the well-known finding that there is little or no positive transfer in going

from a serial list to a paired-associate list made up of linked pairs (i.e., *A-B*, *C-D*, *E-F*, etc.). Actually, as pointed out by Lewandowsky and Murdock (1989; see their note 1), there is considerable positive transfer on the first few trials; it is on a trials-to-criterion measure that the experimental group (linked pairs) and the control group (learning unlinked pairs, such as *A-D*, *G-B*, etc.) do not differ. There can even be considerable positive transfer when the criterion measure is used (Slamecka, Moore, & Carey, 1972), so the general conclusion would seem to be that there is in fact at least some serial-to-paired-associate transfer.

The chaining model would predict benefits for the linked pairs but no benefit for the unlinked pairs, because there are no remote associations in the model. The power-set model would predict more transfer for the linked pairs than for the unlinked pairs, because adjacent associations are stronger than remote associations (see Appendix), but there should still be some benefit for the unlinked pairs because of the remote associations. The chunking model would predict little or no transfer, because only the first two items in the chunk(s) are directly associated. It therefore probably comes down to a win/lose situation. Both the chaining and the power-set model can predict the early transfer but not its disappearance, whereas the chunking model cannot predict enough early transfer, but is consistent with the lack of difference on the trials-to-criterion measure.

Item Recognition

All of the models can perform item recognition, and all of them predict pure recency. For the chaining and the power-set model, one simply takes the dot product of the probe item with the memory vector (or a register in working memory), but the recency is predicted for different reasons. For the chaining model, it is predicted because of the parameter λ, whereas for the power-set model, its prediction is a direct consequence of the storage equation (Equation 5). For the chunking model, one would do the same thing on an immediate-recognition test, but for a delayed test in which the intervening activity overwrote the recognition register, the item information would have to be derived, since it is no longer directly available (see Murdock, 1993b). However, one would still expect recency on a delayed-recognition test (cf. Wright, Santiago, Sands, Kendrick, & Cook, 1985).

SUMMARY

The main difference among these models lies in the way they perform the various recall tasks. The strength of the chunking model is that it uses a multiple-item cue (the multiple convolution of the items recalled to date) to retrieve the next item. The penalty it pays is that it can only do backward recall by multiple forward scans. The chaining model only uses a single-item cue at each step of the recall process, but because of the way the parameters are set up, $\mathbf{f}'_{k-1}$ is stronger than $\mathbf{f}'_{k+1}$, and this causes problems for sequential probes. The power-set model can do either; that is, it can use a multiple-item cue in serial recall or a single-item cue in a probe test. Further, since it has remote associations, it automatically generates gradients, and these gradients show the right patterns. However, these gradients are rather flat, and we will need to know more about the deblurring process before we can tell whether they are consistent with the experimental data.

REFERENCES

ANDERSON, J. A. (1973). A theory for the recognition of items from short memorized lists. *Psychological Review*, **80**, 417-438.

BORSELLINO, A., & POGGIO, T. (1973). Convolution and correlation algebras. *Kybernetik*, **122**, 113-122.

CONRAD, R. (1965). Order error in immediate recall of sequences. *Journal of Verbal Learning & Verbal Behavior*, **4**, 161-169.

DONALDSON, W. (1980). The category size effect in serial memorization. *Canadian Journal of Psychology*, **34**, 175-178.

DREWNOWSKI, A., & MURDOCK, B. B. (1980). The role of auditory features in memory span for words. *Journal of Experimental Psychology: Human Learning & Memory*, **6**, 319-332.

EBBINGHAUS, H. (1964). *Memory: A contribution to experimental psychology*. New York: Dover. (Original work published 1913)

ESTES, W. K. (1972). An associative basis for coding and organization in memory. In A. W. Melton & E. Martin (Eds.), *Coding processes in human memory* (pp. 161-190). Washington, DC: Winston.

HACKER, M. J. (1980). Speed and accuracy of recency judgments for events in short-term memory. *Journal of Experimental Psychology: Human Learning & Memory*, **6**, 651-675.

HADLEY, J. A., HEALY, A. F., & MURDOCK, B. B. (1992). Output and retrieval interference in the missing-number task. *Memory & Cognition*, **20**, 69-82.

HAYES-ROTH, B., & HAYES-ROTH, F. (1977). Concept learning and the recognition and classification of exemplars. *Journal of Verbal Learning & Verbal Behavior*, **16**, 321-338.

HINRICHS, J. V. (1970). A two-process memory-strength theory for judgment of recency. *Psychological Review*, **77**, 223-233.

HINTON, G. E., PLAUT, D. C., & SHALLICE, T. (1993). Simulating brain damage. *Scientific American*, **269**, 76-82.

HINTZMAN, D. L. (1988). Judgments of frequency and recognition memory in a multiple-trace memory model. *Psychological Review*, **95**, 528-551.

HOCKLEY, W. E. (1991). Interrogating memory: A decision model for recognition and judgment of frequency. In W. Abraham, M. C. Corballis, & K. G. White (Eds.), *Memory mechanisms: A tribute to G. V. Goddard* (pp. 219-245). Hillsdale, NJ: Erlbaum.

HUMPHREYS, M. S., BAIN, J. D., & PIKE, R. (1989). Different ways to cue a coherent memory system: A theory for episodic, semantic, and procedural tasks. *Psychological Review*, **96**, 208-233.

KAHANA, M. J. (1995). *Associative symmetry and memory theory*. Manuscript submitted for publication.

LEE, C. L., & ESTES, W. K. (1977). Order and position in primary memory for letter strings. *Journal of Verbal Learning & Verbal Behavior*, **16**, 395-418.

LEWANDOWSKY, S., & LI, S.-C. (1994). Memory for serial order revisited. *Psychological Review*, **101**, 539-543.

LEWANDOWSKY, S., & MURDOCK, B. B. (1989). Memory for serial order. *Psychological Review*, **96**, 25-57.

LI, S.-C., & LEWANDOWSKY, S. (1993). Intralist distractors and recall direction: Constraints on models of memory. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **19**, 895-908.

LIEPA, P. (1977). *Models of content addressable distributed associative memory (CADAM)*. Unpublished manuscript, University of Toronto, Department of Psychology.

LOCKHART, R. S. (1969). Recency discrimination predicted from absolute lag judgments. *Perception & Psychophysics*, **6**, 42-44.
MADIGAN, S. A. (1971). Modality and recall order interactions in short-term memory for serial order. *Journal of Experimental Psychology*, **87**, 294-296.
METCALFE-EICH, J. M. (1982). A composite holographic associative recall model. *Psychological Review*, **89**, 627-661.
MEWHORT, D. J. K., POPHAM, D., & JAMES, G. (1994). On serial recall: A critique of chaining in TODAM. *Psychological Review*, **101**, 534-538.
MILLER, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, **63**, 81-96.
MURDOCK, B. B. (1968). Serial order effects in short-term memory. *Journal of Experimental Psychology Monographs*, **76**(Pt. 2), 1-15.
MURDOCK, B. B. (1974). *Human memory: Theory and data*. Potomac, MD: Erlbaum.
MURDOCK, B. B. (1979). Convolution and correlation in perception and memory. In L.-G. Nilsson (Ed.), *Perspectives in memory research: Essays in honor of Uppsala University's 500th anniversary* (pp. 105-119). Hillsdale, NJ: Erlbaum.
MURDOCK, B. B. (1982). A theory for the storage and retrieval of item and associative information. *Psychological Review*, **89**, 609-626.
MURDOCK, B. B. (1983). A distributed memory model for serial-order information. *Psychological Review*, **90**, 316-338.
MURDOCK, B. B. (1992). Serial organization in a distributed memory model. In A. F. Healy, S. M. Kosslyn, & R. M. Shiffrin (Eds.), *From learning theory to connectionist theory: Essays in honor of William K. Estes* (Vol. 1, pp. 201-225). Hillsdale, NJ: Erlbaum.
MURDOCK, B. B. (1993a). Derivations for the chunking model. *Journal of Mathematical Psychology*, **37**, 421-445.
MURDOCK, B. B. (1993b). TODAM2: A model for the storage and retrieval of item, associative, and serial-order information. *Psychological Review*, **100**, 183-203.
MURDOCK, B. B. (1995). Primacy and recency in the chunking model. In C. Weaver, S. Mannes, & R. Fletcher (Eds.), *Discourse comprehension: Models of processing revisited* (pp. 49-63). Hillsdale, NJ: Erlbaum.
MUTER, P. (1979). Response latencies in discrimination of recency. *Journal of Experimental Psychology: Human Learning & Memory*, **5**, 160-169.
NAIRNE, J. S. (1990). Similarity and long-term memory for order. *Journal of Memory & Language*, **29**, 733-746.
NAIRNE, J. S., & NEATH, I. (1994). A critique of the retrieval/deblurring assumptions of TODAM. *Psychological Review*, **101**, 528-533.
RATCLIFF, R. (1978). A theory of memory retrieval. *Psychological Review*, **85**, 59-108.
RYAN, J. (1969). Grouping and short-term memory: Different means and patterns of grouping. *Quarterly Journal of Experimental Psychology*, **21**, 137-147.
SHIFFRIN, R. M., & COOK, J. R. (1978). Short-term forgetting of item and order information. *Journal of Verbal Learning & Verbal Behavior*, **17**, 189-218.
SHIFFRIN, R. M., & RAAIJMAKERS, J. (1992). The SAM retrieval model: A retrospective and prospective. In A. F. Healy, S. M. Kosslyn, & R. M. Shiffrin (Eds.), *From learning theory to connectionist theory: Essays in honor of William K. Estes* (Vol. 2, pp. 69-86). Hillsdale, NJ: Erlbaum.
SLAMECKA, N. J. (1985). Ebbinghaus: Some associations. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **11**, 414-435.
SLAMECKA, N. J., MOORE, T., & CAREY, S. (1972). Part-to-whole transfer and its relation to organization theory. *Journal of Verbal Learning & Verbal Behavior*, **11**, 73-82.
STERNBERG, S. (1969). The discovery of processing stages: Extensions of Donders' method. *Acta Psychologica*, **30**, 276-315.
WICKELGREN, W. A. (1964). Size of rehearsal group and short-term memory. *Journal of Experimental Psychology*, **68**, 413-419.
WRIGHT, A. A., SANTIAGO, H. C., SANDS, S. F., KENDRICK, D. F., & COOK, R. G. (1985). Memory processing of serial lists by pigeons, monkeys, and people. *Science*, **229**, 287-289.
YNTEMA, D. B., & TRASK, F. P. (1963). Recall as a search process. *Journal of Verbal Learning & Verbal Behavior*, **2**, 65-74.

NOTES

1. Another way of saying this is that convolution is exactly the same as correlation, except that in convolution, one of the item vectors is reflected about its midpoint before doing the correlation.

2. Since TODAM vectors are statistically normalized to 1, any item vector (e.g., $\mathbf{f}_k$) correlated with itself is δ'. As with δ', primes denote approximations, so $\mathbf{g}'_k$ is an approximation to $\mathbf{g}_k$. The arrows denote deblurring; $\mathbf{g}'_k$ may or may not be deblurred to $\mathbf{g}_k$ where $\mathbf{g}_k$ is the target item (see Note 3).

3. We do need a principle to determine whether or not the retrieved information (e.g., $\mathbf{g}'_k$) is close enough to the target item (e.g., $\mathbf{g}_k$) to conclude that the correct item would be recalled, but this principle amounts to a scoring rule, and is not a model of the deblurring process.

4. In the equation (Equation 5 from Lewandowsky & Murdock, 1989), the index j is one greater than serial position, so $j = 1$ denotes the start signal; see Lewandowsky and Li (1994).

5. We also have the interitem similarity parameter ρ, but $\rho = 0$ in the simulations reported here.

6. The normalization is by the dot product; e.g., $|\mathbf{f}| = h\mathbf{f}$, where $h = 1/\sqrt{\mathbf{f} \cdot \mathbf{f}}$.

7. We can work out an explicit expression for the normalization function because, for any n, $E[X_{i=1}^{n}\, \mathbf{f}_i \cdot X_{i=1}^{n}] = 1.0$.

8. During the presentation of a list, what we are calling $\mathbf{M}$ is a register in working memory, so this scheme would simply use another register in working memory.

9. ρ is the interitem similarity, and here the items are independent ($\rho = 0$). If $\rho > 0$, performance deteriorates slightly, but the pattern is still the same.

10. The perturbation model of Estes (1972) says exactly the opposite. Loss of order information occurs in storage; in every short period of time, there is a small but constant probability of interchange of adjacent items, and this is what leads to the observed transpositions in recall. Some modification of this model would presumably be needed to explain why we do not, in the limit, suffer a complete loss of order information.

11. The effect of ρ on h increases with serial position, and this is so because more terms enter into $\mathbf{f}_j * \mathbf{M}_{j-1}$ as j increases. Further, because we use a Gaussian feature distribution, $E[Z^u Z^v] = 0$, $u+v$ odd, so only selected terms affect the value of h at any given serial position.

12. Even if it is easier for $\mathbf{b}'$ than for $\mathbf{a}'$, associative symmetry for pair recognition would still be expected.

13. The main difference is that formerly we assumed that a pair functioned as a unit, so the memory vector was decremented on a pair-by-pair basis. If items were presented sequentially, it might be slightly different, or this might be optional, just as in TODAM2 one has a branch after two items have been presented (see Figure 2 in Murdock, 1993b).

14. One such scheme was suggested in Murdock (1993b), and similar schemes are also used in connectionist models (e.g., Hinton, Plaut, & Shallice, 1993).

APPENDIX

If we consider a power set of n items to be a chunk of size n, we can write an explicit expression for a chunk—namely:

$$\mathbf{C}(n) = \sum_{i=1}^{n} \alpha_i \mathbf{f}_i + \left\{\sum_{i=1}^{n-1} \alpha_{i+1} \mathbf{f}_i\right\} * \left\{\sum_{j=i+1}^{n} h_j \mathbf{f}_j\right\}$$

$$+\left\{\sum_{i=1}^{n-2} \sum_{i'=2}^{n-1} \alpha_{i+2} h_{i'} (\mathbf{f}_i * \mathbf{f}_{i'})\right\} * \left\{\sum_{j=i'+1}^{n} h_j \mathbf{f}_j\right\}$$

$$+\ldots+\left\{\mathbf{f}_1 * \sum_{i=2}^{n-1} h_i \mathbf{f}_i\right\} * h_n \mathbf{f}_n .$$

For a sequential probe $\mathbf{f}_k$ from serial position k,

$$\mathbf{f}_k \# \mathbf{C}(n) \cong \mathbf{f}_k \# \left[\left\{ \sum_{i=1}^{n-1} \alpha_{i+1} \mathbf{f}_i \right\} * \left\{ \sum_{j=i+1}^{n} h_j \mathbf{f}_j \right\} \right],$$

because the lower order items (engrams) and higher order terms (trigrams and up) are just noise. First, consider those cases in which the probe $\mathbf{f}_k$ matches $\mathbf{f}_i$; then we have

$$\begin{aligned} \mathbf{f}_k \# \mathbf{C}(n) &\cong \mathbf{f}_k \# \sum_{i=1}^{n-1} \alpha_{i+1} \sum_{j=i+1} h_j (\mathbf{f}_i * \mathbf{f}_j) \\ &= \mathbf{f}_k \# \alpha_{k+1} \sum_{j=k+1}^{n} h_j (\mathbf{f}_k * \mathbf{f}_j) \\ &= \mathbf{f}_{k+1} \sum_{j=k+1}^{n} h_j (\mathbf{f}_k \# \mathbf{f}_k) * \mathbf{f}_j \\ &= \alpha_{k+1} \sum_{j=k+1}^{n} h_j \mathbf{f}'_j, \end{aligned}$$

so the result is a linear combination whose coefficients are the normalization constants h_j.

When the probe matches the later item, we can turn the expression around, and write:

$$\begin{aligned} \mathbf{f}_k \# \mathbf{C}(n) &\cong \mathbf{f}_k \# \left[\left\{ \sum_{u=2}^{n} h_u \mathbf{f}_u * \sum_{v=1}^{u-1} \alpha^{u-v-1} \mathbf{f}_v \right\} \right] \\ &= \mathbf{f}_k \# \left\{ h_k \mathbf{f}_k * \sum_{v=1}^{k-1} \alpha^{k-v-1} \mathbf{f}_v \right\} \\ &= h_k (\mathbf{f}_k * \mathbf{f}_k) * \sum_{v=1}^{k-1} \alpha^{k-v-1} \mathbf{f}_v = h_k \sum_{v=1}^{k-1} \alpha^{k-v-1} \mathbf{f}'_v, \end{aligned}$$

so here the coefficients of the linear combination are a function of α. Since the exponent is $k-v-1$, the more remote (i.e., the earlier) the competition, the weaker it is. Thus, for any sequential probe $\mathbf{f}_k$, correlating the probe with the chunk gives us a linear combination of all the other items in the list with a gradient in the forward direction, which is a function of the normalization factor h and with a gradient in the backward direction, which is a function of the forgetting parameter, α.

(Manuscript received May 10, 1994;
revision accepted for publication July 25, 1994.)