Order Information and Distributed Memory Models

Roger Ratcliff

Psychology Department
Northwestern University
Evanston, IL 60201

Abstract

Current versions of distributed models have difficulty in accounting for the representation of order information in matching tasks. In this article, experiments are presented that allow discrimination between physical and ordinal representations of ordinal information, discrimination between position-dependent codes and context-sensitive codes, and generalization of the results of matching tasks from strings of letters to long-term memory for triples of words. Data from these experiments constrain the kinds of models that can be developed to account for matching and order, and present problems for several current memory models, including connectionist models. Suggestions are made for modifications of these models to account for the results from matching tasks.

The representation of order information is important to normal functioning in many cognitive domains. In speech, both perception and production involve the processing of a continuous temporal stream of information that requires the maintenance of order information. In the perception of visual patterns, the processing system is usually required to maintain either the absolute or relative positions of objects in the visual scene. The study of the representation of order information was of importance over 10 years ago as a topic in its own right (Lee & Estes, 1977; Murdock, 1974, pp. 157-174), but more recently it has become a subtopic within different processing domains. The domain studied in this paper concerns the maintenance of order within a simultaneously presented string of letters or words.

The task used in the experiments of this article is a matching task in which subjects study a string of items (letters in Experiments 1 and 2, words in Experiments 3 and 4) and then must decide whether a test string matches the study string (see Murdock, 1984, for a related recall task). When strings of letters are the studied items, a test string is presented immediately after each study string. When words are used, the test is delayed by presenting study strings in blocks, and then presenting test strings for all the study strings in the test block. The primary experimental manipulation considered in this paper is one in which the order of the studied items is rearranged at test (Angiolillio-Bent & Rips, 1982; Proctor & Healy, 1985; Ratcliff, 1981; Ratcliff & Hacker, 1981). When items adjacent in a study string are interchanged in the test string, subjects find it difficult to respond that the test string is "different;" accuracy is poor and reaction time slow relative to "different" conditions in which new items in the test string replace old items in the study string. Results also show that the greater the displacement in the switch (e.g., adjacent letters switched versus remote letters switched), the easier it is for subjects to respond "different." Ratcliff (1981) developed a model to account for these effects that assumes that the representations of items are distributed across position so that when items are interchanged in the test string, there is a contribution from close positions to the match between study and test strings.

This article continues this work with three major aims. First, new experimental results on matching tasks are presented that test some specific hypotheses about the nature of the representation of order and also generalize the results from letter strings to word strings. Second, the results allow discrimination between two main hypotheses about the representation of order, position dependent codes versus context sensitive or associative codes. Third, it is argued that memory models designed to account for memory for single words, pairs of words, and so on, should be capable of representing order, and in the last part of the paper, some of these models are evaluated. Specifically, there are important implications of the experimental data for distributed memory models and connectionist models. The argument is simple: models in which items are represented as vectors of features assume that elements within the vectors are independent. Thus,in their present form, they are incapable of dealing with the transposition data presented here and in Ratcliff (1981).

The empirical part of the paper will present four new experiments. The first demonstrates that manipulations of the order of the items in the study string are not sensitive to exact physical position. The second experiment examines performance on different permutations of the study string to contrast the hypotheses of position dependent versus context sensitive memory codes. The third experiment replicates the letter matching task in the memory domain using triples of words, and Experiment 4 uses a response signal procedure to examine the time course of processing in this memory task.

Experiment 1

The letter matching task has been traditionally called the "perceptual" matching task. This label comes from another class of perceptual letter recognition tasks, in which letters are displayed briefly for identification, and physical variables such as the spacing of letters in the string to be identified affect performance (e.g., Bjork & Murray, 1979; Estes, 1982). A similar effect in the matching task would point to a mental representation based closely on physical features of the stimuli.

Method

To examine whether performance in matching depends on physical location, the spacing of the letters in a string was altered between study and test. Subjects studied three letters presented in the center of the display for 500 ms (e.g., _ABC_). There were two manipulations at test. Spacing was tested by altering the positions of the letters in 5 slots e.g., ABC__, AB_C_, AB__C, A_BC_, (all these would require a positive response) and so on. Spacing was crossed with positive and negative conditions: One third of the trials were positive trials in which the three study letters were presented at test in the same order as at study. For negative trials, there were 5 permutation conditions, and three conditions in which a single letter was replaced by a new letter. The test string was displayed immediately after the study string for 200 ms and then removed to eliminate possible eye movement effects. Subjects were presented with 10 blocks of 120 trials. Eighteen Northwestern undergraduate subjects participated in a one-hour session for course credit. (See Ratcliff, 1981, for further details of the experimental procedure.)

Results

Results are shown in Tables 1 and 2. Table 1 shows accuracy and reaction time for the positive conditions as a function of spacing. The effect of spacing is significant (reaction time: $F(9,153)=3.82$, $p<.05$, mse=1720; error rate: $F(9,153)=2.45$, $p<.05$, mse=.00226), but inspection of Table 1 shows that the effects are quite small. Tukey's HSD = 45 ms, so that differences between pairs of reaction times in Table 1 larger than 45 ms are significant. For error rates, Tukey's HSD is 0.051. Inspection of Table 1 shows that only condition 6 (test A__BC) differs from some of the other conditions in accuracy, and only conditions 3 and 6 in reaction time. The power of these contrasts is high because there are around 600 observations per condition.

For negative conditions, there was no effect of spacing but large effects of permutation, replicating Ratcliff and Hacker (1981) and Ratcliff (1981) (see Table 2). For reaction time: spacing effect, $F(9,153)=1.3$, not significant, negative condition, $F(7,119)=31.5$, $p<.05$, and the interaction between spacing and negative condition, $F(63,1071)=1.02$, not significant. For accuracy: spacing, $F(9,153)=1.7$, not significant, negative condition, $F(7,119)=26.6$, $p<.05$, and the interaction, $F(63,1071)=1.02$, not significant.

Table 1
Reaction Time and Accuracy for Same Conditions in Experiment 4

Condition	Number	Accuracy	Reaction Time (ms)
1 ABC__	607	.916	624.2
2 AB_C_	609	.917	645.3
3 AB__C	603	.910	674.3
4 A_BC_	599	.903	627.1
5 A_B_C	611	.919	629.2
6 A__BC	576	.869	670.8
7 _ABC_	626	.939	612.7
8 _AB_C	603	.904	657.2
9 _A_BC	605	.917	635.8
10 __ABC	612	.911	629.5

Note. The study string was presented as _ABC_ where the symbol _ refers to a blank.

Table 2
Reaction Time and Accuracy for Different Conditions in Experiment 4 averaged over Spacing

Negative Condition	Number of Responses	Accuracy	Reaction Time (ms)
ACB	1289	.777	766.9
BAC	1574	.952	666.6
BCA	1602	.965	630.8
CAB	1608	.972	619.1
CBA	1627	.978	610.4
XBC	1627	.973	597.0
AXC	1589	.958	645.3
ABX	1567	.943	666.2

Note. The study string is denoted ABC and X is a letter other than A, B, or C.

We can conclude that spacing differences of the letters between study and test have small effects that are detectable only with experiments with high power. Thus, it is wise to view the word "perceptual" in the term perceptual matching as a name for the task and not as a description of what kinds of variables are likely to affect performance.

Experiment 2

The second experiment was designed to provide data to distinguish between models in which an item is encoded in terms of its absolute position and models in which relative position is encoded. The idea is that certain test conditions allow these models to be contrasted. If the string ABCDE is studied, then a test string BCDEA has four letters in their correct adjacent order (BCDE) and none in their correct absolute position. In contrast, the test string AECDB has three letters in their correct absolute positions but only one pair in the correct adjacent order (CD). The relative difficulties of such test strings can be used to discriminate the two kinds of models.

To perform this experiment, all permutations of the final four letters were the main conditions studied (Ratcliff, 1981, found that performance when the first letter was changed was near ceiling). There were also conditions in which one letter was replaced by a new letter and some fillers in which the first letter was permuted.

Method

The method was similar to that of Experiment 1, except 5 Dartmouth undergraduates were volunteer subjects (paid at $3/hr) for 7 one-hour sessions. The study string was presented for 1.2 s and the test string was presented for 250 ms.

Results

Results are shown in Table 3. To compare the various negative conditions, Tukey's HSD test was used, and differences in accuracy greater than .05 and in reaction time greater than 41 ms are significant.

Table 3
Reaction Time and Accuracy for the Letter Matching Experiment 2

Condition	Number of Observations	Accuracy	Reaction Time (ms)
Same	6520	.926	650
ABCED	197	.430	734
ABDCE	282	.609	667
ABDEC	353	.762	645
ABECD	349	.746	659
ABEDC	360	.763	643
ACBDE	311	.670	656
ACBED	360	.776	632
ACDBE	347	.732	609
ACDEB	388	.822	606
ACEBD	393	.836	590
ACEDB	396	.843	596
ADBCE	347	.737	626
ADBEC	397	.838	603
ADCBE	351	.756	618
ADCEB	391	.827	582
ADEBC	404	.861	588
ADECB	408	.863	591
AEBCD	392	.838	604
AEBDC	390	.826	608
AECBD	397	.841	625
AECDB	369	.790	597
AEDBC	402	.843	589
AEDCB	401	.851	584
AXCDE	351	.750	614
ABXDE	369	.782	607
ABCXE	307	.667	691
ABCDX	286	.620	722
BACDE	409	.887	607
CBADE	427	.910	578
DBCAE	423	.910	581
EBCDA	426	.908	595
XBCDE	397	.871	601
CABDE	425	.900	574
BCADE	419	.895	579

Note. It is assumed that the study string is ABCDE and X is a letter other than A, B, C, D, or E.

Several comparisons can be made that address the issue of relative order versus absolute location. Assuming that ABCDE is the studied string, the string AEBCD with items BCD in correct order but four items in incorrect location can be compared with ACEBD and ADBEC which also have 4 items in incorrect location but no pairs in correct relative positions. Results in Table 3 show that these conditions produce the same values of reaction time and accuracy. Thus, the position of an item with respect to other studied items is not a dimension that is of importance in matching. In contrast to this test of pairwise order, correct location of an item in the string does have a large effect. For example, ABCED, ABDCE, and ACBDE are hard to reject (accuracy less than 0.7 and reaction time greater than 680 ms).

These results argue for a model with a position dependent code (Ratcliff, 1981) or a position dependent retrieval process (Proctor & Healy, 1985). This does not mean that a position dependent code is always used or that a context dependent code is never used, only that in matching procedures, a position dependent code is used. But it does mean that models must have the capability of using a position dependent code (see McNicol & Heathcote, 1986, for similar arguments and experimental support).

Experiment 3

Ratcliff (1981) presented two perceptual matching experiments that used letters strings as stimuli. The model developed to account for the results assumed that the difficulty in responding <u>different</u> to a reordered letter string was located in the distributed representation of letters in memory. If this is correct, then the results should generalize across paradigms and materials and provide contact with both theoretical and empirical work in memory research (e.g., Gillund & Shiffrin, 1984; Murdock, 1982). Experiments 3 and 4 are analogs of Experiments 1 and 2 in Ratcliff (1981) using word triples instead of 5-letter strings, and a study-test procedure in which 8 triples are studied and then tested. Also, a large pool of different words was used instead of repetitions of the same set of letters.

Method

Subjects were 17 Northwestern University Undergraduates participating for course credit. Thirty-two lists were presented, each made up of 8 study triples (6 seconds per triple) followed by 8 test triples. Subjects were encouraged to be fast and accurate; feedback ("TOO SLOW!!") was given for responses slower than 2500 ms.

Results

Results are shown in Table 4. The results are similar to those found in letter matching (Ratcliff, 1981): adjacent switches (test strings ACB and BAC for the study string ABC, where each letter represents a word) are difficult with low accuracy, other permutations are less difficult, and single replace conditions (ABX, AXC, and XBC) are least difficult. Angiolillio-Bent and Rips (1982) defined displacement count as a measure of the size of the permutation, so ACB has a displacement count of 2 and BCA has a displacement count of 4. Displacement count seems to be the main determiner of accuracy for the permutations in this study. Reaction time results differ from the normal pattern (which is low accuracy, slow responses) because accuracy and reaction time are correlated only when accuracy scores are all above (or below) accuracy .5 will be slowest. In general, the results are qualitatively similar to those found in Ratcliff (1981), namely, the smaller the displacement, the more difficult is a negative response.

Table 4
Reaction Time and Accuracy for Word Triple Matching Experiment 3

Condition	Number of Observations	Accuracy	Reaction Time (ms)
Same	1464	0.717	1346
ACB	119	0.465	1499
BAC	109	0.427	1489
BCA	147	0.576	1511
CAB	154	0.609	1505
CBA	146	0.575	1479
ABX	180	0.711	1424
AXC	178	0.698	1486
XBC	197	0.770	1401

Note. The studied word triple was ABC and X refers to a word other than A, B, or C.

Experiment 4

This experiment uses a response signal procedure to examine the growth of accuracy in the same experimental conditions in Experiment 3. The experiment is designed to provide the kind of evidence obtained in Experiment 2 in Ratcliff (1981) which examined the growth of accuracy in the letter matching task. In that procedure, subjects were required to respond when a signal to respond was presented. Results showed that accuracy grew rapidly, at the same rate for each of the different negative conditions. This provided strong evidence for a parallel holistic matching process.

Method

The method was the same as that of Experiment 3 except for one main difference: Subjects were required to respond upon presentation of a signal (within 200 to 300 ms) and the signal was presented at lags of 150, 300, 600, 900, and 2000 ms after the test string. The signal was a row of asterisks presented directly under the test triple. Eight Northwestern undergraduates served as subjects for course credit and participated in 5 experimental sessions preceded by one practice session.

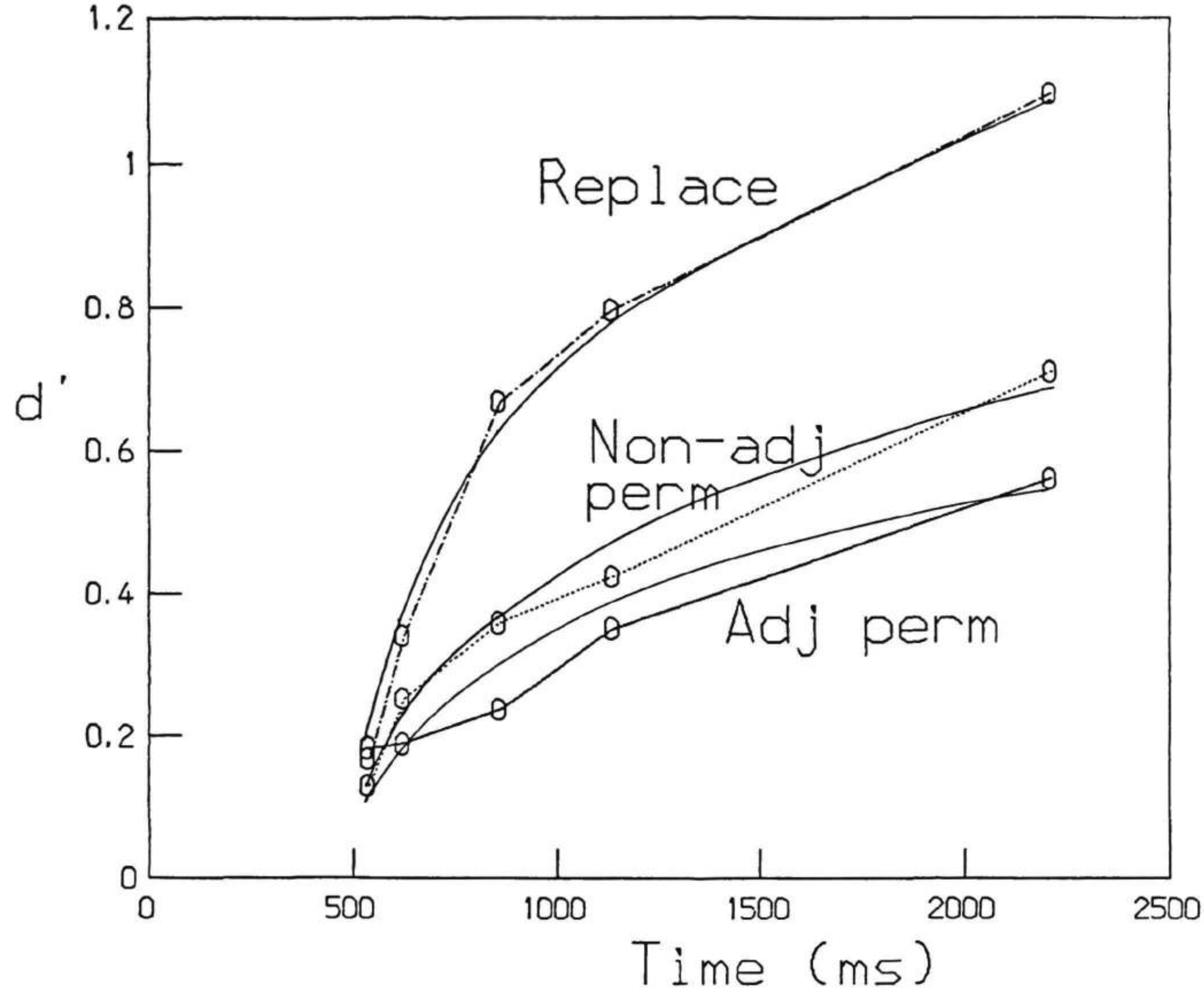

Figure 1

Growth of accuracy as a function of time for Experiment 4. The continuous curves are fits of the diffusion model (Ratcliff, 1978; 1981) to the data. Parameters of the model are T_{er}=501, v=2910, and d'_a for the three conditions (top to bottom)=1.84, 1.12, 0.88. The diffusion equation is: $d'(t)=d'_a/\mathrm{sqrt}(1+v/(t-T_{er}))$.

Results

The main results are shown in Figure 1. The two test conditions ACB and BAC are collapsed to give the "adjacent permutation" condition, the other permutations are collapsed to give the "non-adjacent permutation" condition, and the three single replace conditions are collapsed. Accuracy grows for all the conditions essentially in parallel to different asymptotes. This means that information about goodness-of-match is available for the three different classes of negatives at the same rate. Thus any model that explains order effects by a systematic, serial, element-by-element comparison process is not supported by this data.

The results of this experiment are qualitatively similar to those found in Experiment 2, Ratcliff (1981). All the functions begin at the same point in time and grow in parallel. Fits of the diffusion retrieval model (Ratcliff, 1978; 1981) to the response signal results are shown in Figure 1. The fits are excellent and support the claim for parallel growth of these functions, so that information about all the different kinds of negative conditions is available to the same extent at any time during the course of matching. These results show that the memory procedure provides equivalent results to the letter matching procedure, and so suggests that we should give special attention to models that account for both sets of phenomena with similar representations and/or processes.

Theory

In this section, several memory models are considered with respect to their abilities to account for position dependent effects (Experiment 2), and the immediate availability of position codes for use in letter-matching and word triple matching in long-term memory (Experiment 4).

Vector models. This class of models includes both distributed memory models (e.g., Anderson, 1972; 1973; Murdock, 1982) and connectionist models (Ackley et al, 1985; McClelland & Rumelhart, 1985). The first point to note about current theories is that many have no built-in capability for dealing with position codes and distance effects. In vector models, the first assumption that might be made is that groups of elements within a vector represent letters within a letter string. Thus the memory vector for a five letter string consists of 5 groups of n elements. The main problem with this idea is that each element is assumed to be independent of each other element so switching elements within the vector, the analog of studying ABCDE and testing ABDCE, would produce results identical to testing ABXYE. The data in Ratcliff (1981) and Experiments 1-4 here show that the former is much more difficult, so this approach will not work.

One way of implementing position dependence in the framework of a vector model is to assume that each letter string is a vector (with letters being sub-vectors within that vector) and that the letter string vector is added to a longer memory vector many times at randomly varying positions. I have implemented this model and found that 100 presentations of the letter vector are needed with a memory vector of length 200, a letter vector of length 10, and storage position normally distributed with standard deviation 15. For this model to produce the correct distance effects, more copies of the letter string vector must be stored in the central position (position at which the test vector matched) and the number of copies stored at more distant positions must decrease with distance from the center (this is why the normal distribution of storage positions is used). To illustrate this, suppose that ABCDE is stored and ABDCE is tested. During encoding, some copies of ABCDE are stored in positions BCDE_, i.e., the string is shifted one letter position to the right, so that the C in the test string matches the stored C in position D.

Another possibility for implementing position dependence is to assume that at test time, a cross correlation is computed between the study and test strings. For a vector model, this means that not only the match between the two strings is assessed (e.g., if f and g are the study and test vectors, then the correlation is SIf_i*g_i), but also the match between the study and test strings with the vector shifted in position (SIf_i*g_{i+k} where k varies from 1 to n-k; note some weighting may be needed). Thus, if two items were interchanged, there would be a component of match from the cross-correlation.

I have implemented both these schemes and they both mimic the model presented in Ratcliff (1981) that assumed that letters are distributed across position. The first vector model is a specific implementation of the distributed memory scheme and the cross-correlation model is a retrieval implementation. The critical issue for this class of models is not whether a model can be developed, but whether a unified model with this kind of mechanism could also account for the same range of data as a more traditional model.

Context sensitive models. Instead of the vector scheme outlined above, a different representation could be used in which an item is encoded with respect to its neighboring items (a context sensitive code). Several such models have been developed (e.g., Cohen & Grossberg, 1986; Rumelhart & McClelland, 1986; Wickelgren, 1969). For the domain of speech processing, the context sensitive representation is appropriate, but the results from Experiments 2 and 3 demonstrate that a context sensitive code will not work for the matching task.

Another representation that might be used is one in which both item and position information are represented. However, extra assumptions would be necessary to account for distance effects when letters or words were switched in position. The model of Ratcliff (1981) and the cross-correlation scheme noted above would both provide that metric.

Gillund and Shiffrin (1984) model. This model assumes an associative code for the representation of information. At test time in recognition, the familiarity of the test probe is assessed from pairwise associations between the test probe and memory. There is no position dependence in this code. In recall, the associative code is used in a sampling scheme for recall. Again, the associative code will not explain the results of Experiments 2 and 3. Thus some kind of position dependent code is required for this model to account for the experimental results presented above.

Distributed memory models. For Anderson's (1973) and Murdock's (1982) vector model, we could assume that each letter (or word) is represented as a subvector within the vector representing the whole item string and then strings are entered into the memory vector at different positions. The issue is whether multiple representations will affect the signal to noise ratio and thus make recognition performance too low. The Anderson and Murdock models also have an associative component that stores pairwise associations (Anderson, 1972, a matrix; Murdock, a convolution). This associative component would not account for position dependent effects because an associative code is context sensitive and so would not deal with the results from Experiment 2.

There are two connectionist models that are relevant here. The auto-associative distributed memory model of McClelland and Rumelhart (1985) assumes a vector representation and assumes that memory is represented in a matrix of connections between each element and each other element. The multilayer connectionist model (e.g., Ackley, et al, 1985) assumes that there are three layers of units, e.g., an input layer, a hidden layer, and an output layer. For the encoder problem (essentially learning to produce an output pattern that is the same as the input pattern), memory for the pattern resides in the two matrices of connections between the input and hidden layer and hidden and output layers. Both these models as they stand do not allow for distance effects as noted earlier because they assume that elements of the vectors are independent so that switches of items within the string would be the same as replacements of new letters. One might think that one of the schemes noted above could be used to encode position dependent information, i.e., multiple copies or cross-correlation at retrieval. But multiple copies of the studied materials shifted in position would not help because patterns are stored as units (partly as a result of nonlinear processing) and permutations of letters across position would not produce the required crosstalk. Cross-correlation at test time may be a better candidate, but if cross-correlation were used routinely in all retrieval processes, the signal to noise ratio at test would be severely reduced (because of all the contributions from nonmatching cross-correlations). Perhaps some of the notions of Zipser (1986) could be incorporated into these schemes. The main issue for these distributed models is to come up with a consistent and coherent account that applies across a range of paradigms including tests of order information as well as other phenomena (e.g., those in Gillund & Shiffrin, 1984).

The final issue to be considered is whether positional information is routinely encoded into memory or whether it is only encoded when it is needed. McNicol and Heathcote (1986) argue that within the domain of short-term memory, their results are most compatible with a theory that allows different subsystems in short-term memory each with its own format for preserving order (see Ratcliff & McKoon, 1987 for data showing no position dependent code and availability of order information late in processing). I endorse this and argue that the results of Experiments 3 and 4 extend the use of a position dependent code into the domain of long-term memory. However, the critical point is that a model must have the capability of representing position dependent codes without the invocation of a completely new model just for that task.

References

Ackley, D.H., Hinton, G.E., & Sejnowski, T.J. (1985). A learning algorithm for Boltzmann machines. Cognitive Science, 9, 147-169.

Anderson, J.A. (1973). A simple neural network generating an interactive memory. Mathematical Biosciences, 14, 197-220.

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

Angiolillio-Bent, J.S., & Rips, L.J. (1982). Order information in multiple element comparison. Journal of Experimental Psychology: Human perception and Performance, 8, 392-406.

Bjork, E.L., & Murray, J.T. (1977). On the nature of input channels in visual processing. Psychological Review, 84, 472-484.

Cohen, M., & Grossberg, S. (1986). Neural dynamics of speech and language coding: Developmental programs, perceptual grouping, and competition for short-term memory. Human Neurobiology, 5, 1- 22.

Estes, W.K. (1982). Similarity-related channel interactions in visual processing. Journal of Experimental Psychology: Human Perception and Performance, 8, 353-382.

Gillund, G. & Shiffrin, R.M. (1984). A retrieval model for both recognition and recall. Psychological Review, 19, 1-65.

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

McClelland, J.L., & Rumelhart, D.E. (1985). Distributed memory and the representation of general and specific information. Journal of Experimental Psychology: General, 114, 159-188.

McNicol, D., & Heathcote, A. (1986). Representation of order information: An analysis of grouping effects in short-term memory. Journal of Experimental Psychology: General, 115, 76-95.

Murdock, B.B. (1974). Human memory: Theory and data. Potomac, Md.: Erlbaum.

Murdock, B.B. (1982). A theory for the storage and retrieval of item and associative information. Psychological Review, 89, 609- 626.

Proctor, R.W., & Healy, A.F. (1985). Order-relevant and order-irrelevant decision rules in multiletter matching. Journal of Experimental Psychology: Learning, Memory, and Cognition, 11, 519-537.

Ratcliff, R. (1978). A theory of memory retrieval. Psychological Review, 85, 59-108.

Ratcliff, R. (1981). A theory of order relations in perceptual matching. Psychological Review, 88, 552-572.

Ratcliff, R., & Hacker, M.J. (1981). Speed and accuracy of same and different responses in perceptual matching. Perception and Psychophysics, 30, 303-307.

Ratcliff, R., & McKoon, G. (1987). Item familiarity versus relational information: The time course of retrieval of sentences. In preparation.

Rumelhart, D.E., & McClelland, J.L. (1986). On learning the past tense of English verbs. In J.L. McClelland & D.E. Rumelhart, Parallel distributed processing: Explorations in the microstructures of cognition. Vol. 2: Psychological and biological models. Cambridge, Mass.: MIT Press.

Wickelgren, W.A. (1969). Context-sensitive coding, associative memory, and serial order in (speech) behavior. Psychological Review, 76, 1-15.

Zipser, D. (1986). Biologically plausible models of place recognition and goal location. In J.L. McClelland and D.E. Rumelhart, Parallel distributed processing: Explorations in the microstructures of cognition. Vol. 2: Psychological and biological models. Cambridge, Mass.: MIT Press.

Footnote

This research was supported by NSF grant BNS 82 03061 to Roger Ratcliff and NSF grant BNS 85 16350 to Gail McKoon. I would like to thank Gail McKoon and Scott Gronlund for their comments on this article.