THE SIMULATION OF VERBAL LEARNING BEHAVIOR

by Edward A. Feigenbaum

The purpose of this report is to describe in detail an information processing model of elementary human symbolic learning processes. This model is realized by a computer program called the Elementary Perceiver and Memorizer (EPAM).

The EPAM program is the precise statement of an information processing theory of verbal learning that provides an alternative to other verbal learning theories which have been proposed.[1] It is the result of an attempt to state quite precisely a parsimonious and plausible mechanism sufficient to account for the rote learning of nonsense syllables. The critical evaluation of EPAM must ultimately depend not upon the interest which it may have as a learning machine, but upon its ability to explain and predict the phenomena of verbal learning.

I should like to preface my discussion of the simulation of verbal learning with some brief remarks about the class of information processing models of which EPAM is a member.

a. These are models of mental processes, not brain hardware. They are *psychological* models of mental function. No physiological or neurological assumptions are made, nor is any attempt made to explain information processes in terms of more elementary neural processes.

b. These models conceive of the brain as an *information processor* with sense organs as input channels, effector organs as output devices, and with internal programs for testing, comparing, analyzing, rearranging, and storing information.

[1] Examples of quantitative (or quasi-quantitative) theories of verbal learning are those of Hull et al. (1940), Gibson, (1940), and Atkinson (1954).

c. The central processing mechanism is assumed to be serial; *i.e.,* capable of doing only one (or a very few) things at a time.

d. These models use as a basic unit the *information symbol; i.e.,* a pattern of bits which is assumed to be the brain's internal representation of environmental data.

e. These models are essentially *deterministic,* not probabilistic. Random variables play no fundamental role in them.

The Basic Experiment

Early in the history of psychology, the psychologist invented an experiment to simplify the study of human verbal learning. This "simple" experiment is the rote memorization of nonsense syllables in associate pairs or serial lists.

The items to be memorized are generally three-letter words having consonant letters on each end and a vowel in the middle. Nonsense syllables are chosen in such a way that the three-letter combinations have no ordinary English meaning. For example, CAT is not a nonsense syllable, but XUM is.[2]

In one basic variation, the rote memory experiment is performed as follows:

a. A set of nonsense syllables is chosen and the syllables are paired, making, let us say, 12 pairs.

b. A subject is seated in front of a viewing apparatus and the syllables are shown to him, one pair at a time.

c. First, the left-hand member of the pair (*stimulus item*) is shown. The subject tries to say the second member of the pair (*response item*).

d. After a short interval, the response item is exposed so that both stimulus and response items are simultaneously in view.

e. After a few seconds, the cycle repeats itself with a new pair of syllables. This continues until all pairs have been presented (a *trial*).

f. Trials are repeated, usually until the subject is able to give the correct response to each stimulus. There is a relatively short time interval between trials.

g. For successive trials the syllables are reordered randomly. This style of carrying out the experiment is called *paired-associates presentation.*

The other basic variant of the experiment is called *serial-anticipation presentation.* The nonsense syllables (say, 10 or 12 items) are arranged

[2] People will defy an experimenter's most rigorous attempt to keep the nonsense syllables association-free. Lists of nonsense syllables have been prepared, ordering syllables on the basis of their so-called "association value," in order to permit the experimenter to control "meaningfulness."

in a serial list, the order of which is not changed on successive trials. When he is shown the nth syllable, the subject is to respond with the $(n+1)$st syllable. A few seconds later, the $(n+1)$st syllable is shown and the subject is to respond with the $(n+2)$d syllable, and so on. The experiment terminates when the subject is able to correctly anticipate all of the syllables.

Numerous variations on this experimental theme have been performed.[3] The phenomena of rote learning are well studied, stable, and reproducible. For example, in the typical behavioral output of a subject, one finds:

a. Failures to respond to a stimulus are more numerous than overt errors.

b. Overt errors are generally attributable to confusion by the subject between similar stimuli or similar responses.

c. Associations which are given correctly over a number of trials sometimes are then forgotten, only to reappear and later disappear again. This phenomenon has been called oscillation.[4]

d. If a list x of syllables or syllable pairs is learned to the criterion; then a list y is similarly learned; and finally retention of list x is tested; the subject's ability to give the correct x responses is degraded by the interpolated learning. The degradation is called retroactive inhibition. The overt errors made in the retest trial are generally intrusions from the list y. The phenomenon disappears rapidly. Usually after the first retest trial, list x has been relearned back to criterion.

e. As one makes the stimulus syllables more and more similar, learning takes more trials.

The Information Processing Model

This section describes the processes and structures of EPAM.

EPAM is not a model for a particular subject. In this respect it is to be contrasted with the binary choice models of particular subjects which Feldman describes (1961*a*). The fact is that individual differences play only a small part in the results of the basic experiment described above.

It is asserted that there are certain elementary information processes which an individual must perform if he is to discriminate, memorize and associate verbal items, and that these information processes participate in all the cognitive activity of all individuals.[5]

It is clear that EPAM does not yet embody a complete set of such

[3] For an extended treatment of this subject, see Hovland, *Human Learning and Retention* in Stevens (1951).

[4] By Hull (1935). Actually he called it "oscillation at the threshold of recall," reflecting his theoretical point of view.

[5] Some information processing models are conceived as models of the mental

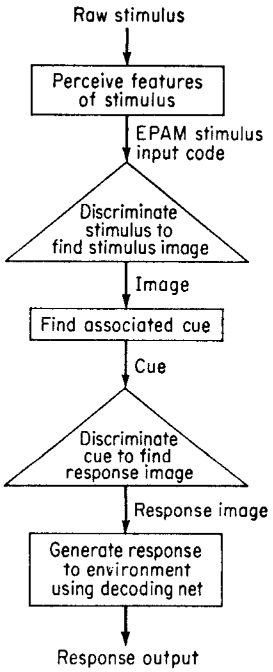

Figure 1. EPAM performance process for producing the response associated with a stimulus.

processes. It is equally clear that the processes EPAM has now are essential and basic.

OVERVIEW: PERFORMANCE AND LEARNING

Conceptually, EPAM can be broken down into two subsystems, a performance system and a learning system. In the performance mode, EPAM produces responses to stimulus items. In the learning mode, EPAM learns to discriminate and associate items.

The performance system is the simpler of the two. It is sketched in Fig. 1. When a stimulus is noticed, a *perceptual process* encodes it, producing an internal representation (an *input code*). A *discriminator* sorts the input code in a *discrimination net* (a tree of tests and branches) to find a stored *image* of the stimulus. A response *cue* associated with the image is found, and fed to the discriminator. The discriminator sorts the cue in the net and finds the response image, the stored form of the response. The response image is then decoded by a *response generator* letter by letter in another discrimination net into a form suitable for output. The response is then produced as output.

The processes of the learning system are more complex. The discrimination learning process builds discriminations by growing the net of tests and branches. The association process builds associations between images by storing response cues with stimulus images. These processes will be described fully in due course.

---

function of particular subjects; *e.g.*, Feldman's Binary Choice Model (1959). Others treat the general subject as EPAM does. Still others are mixed in conception, asserting that certain of the processes of the model are common for all subjects while other processes may vary from subject to subject; *e.g.*, the General Problem Solver of Newell, Shaw, and Simon (1959*a*). Alternatively, information processing models may also be categorized according to how much of the processing is "hard core" (*i.e.*, necessary and invariant) as opposed to "strategic" (*i.e.*, the result of strategy choice by control processes). I suggest the obvious: that models of strategies for information processing will tend to be models of particular subjects. As exemplars, Lindsay's Reading Machine (1960), a "hard-core" model, treats the general subject; Wickelgren's model of the conservative focusing strategy in concept attainment (Wickelgren, 1962; Bruner, Goodnow, and Austin, 1956), a pure strategy model, can predict only the behavior of particular subjects.

The succeeding sections on the information processing model give a detailed description of the processes and structures of both systems.

INPUT TO EPAM: INTERNAL REPRESENTATIONS OF EXTERNAL DATA

The following are the assumptions about the symbolic input process when a nonsense syllable is presented to the learner. A *perceptual system* receives the raw external information and codes it into *internal symbols*. These internal symbols contain descriptive information about features of external stimuli. For unfamiliar 3-letter nonsense symbols, it is assumed that the coding is done in terms of the individual letters, for these letters are familiar and are well-learned units for the adult subject.[6] The end result of the perception process is an internal representation of the nonsense syllable—a list of internal symbols (*i.e.*, a list of lists of bits) containing descriptive information about the letters of the nonsense syllable. Using Minsky's terminology (1961*a*), this is the "character" of the nonsense syllable.

I have not actually programmed this perception process. For purposes of this simulation, I have assigned coded representations for the various letters of the alphabet based on 15 different geometrical features of letters. For purposes of exploring and testing the model, at present all that is really needed of the input codes is:

a. that the dimensions of a letter code be related in some reasonable way to features of real letters.

b. that the letter codes be highly redundant, that is, include many more dimensions than is necessary to discriminate the letters of the alphabet.

To summarize, the internal representation of a nonsense syllable is a list of lists of bits, each sublist of bits being a highly redundant code for a letter of the syllable.

Given a sequence of such inputs, the essence of the learner's problem is twofold: first, to *discriminate* each code from the others already learned, so that differential response can be made; second, to *associate* information about a "response" syllable with the information about a "stimulus" syllable so that the response can be retrieved if the stimulus is presented.

DISCRIMINATING AND MEMORIZING: GROWING TREES OF IMAGES

I shall deal with structure first and reserve my discussion of process for a moment.

Discrimination Net. The primary information structure in EPAM is the

[6] The basic perception mechanism I have in mind is much the same as that of Selfridge (1955) and Dinneen (1955), whose computer program scanned letters and perceived simple topological features of these letters.

discrimination net. It embodies in its structure at any moment all of the discrimination learning that has taken place up to a given time. As an information structure it is no more than a familiar friend: a sorting tree or decoding network. Figure 2 shows a small net. At the terminals of the net are lists called *image lists,* in which symbolic information can be stored. At the nodes of the net are stored programs, called *tests,* which examine characteristics of an input code and signal branch left or branch right. On each image list will be found a list of symbols called the *image.* An image is a partial or total copy of an input code. I shall use these names in the following description of net processes.

Net Interpreter. The discrimination net is examined and altered by a number of processes, most important of which is the *net interpreter.* The net interpreter sorts an input code in the net and produces the image list associated with that input code. *This retrieval process is the essence of a purely associative memory: the stimulus information itself leads to the retrieval of the information associated with that stimulus.* The net interpreter is a very simple process. It finds the test in the topmost node of the tree and executes this program. The resulting signal tells it to branch left or branch right to find the succeeding test. It executes this, tests its branches again, and repeats the cycle until a terminal is found. The name of the image list is produced, and the process terminates. This is the discriminator of the performance system which sorts items in a static net.

Discrimination Learning. The discrimination learning process of the learning system grows the net. Initially we give the learning system no discrimination net but only a set of simple processes for growing nets and storing new images at the terminals.

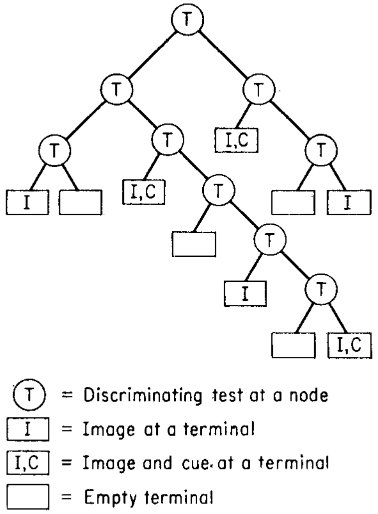

Figure 2. A Typical EPAM discrimination net.

To understand how the discrimination and memorization processes work, let us examine in detail a concrete example from the learning of nonsense syllables. Suppose that the first stimulus-response associate pair on a list has been learned. (Ignore for the moment the question of how the association link is actually formed.) Suppose that the first syllable pair was DAX-JIR. The discrimination net at this point has the simple two-branch structure shown in Fig. 3. Because the syllables differ in their first letter, Test 1 will probably be a test of some characteristic on which the letters D and J differ. No more tests are necessary at this point.

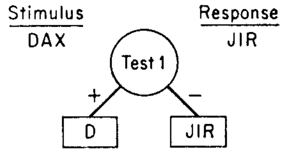

Figure 3. Discrimination net after the learning of the first two items. Cues are not shown. Condition: no redundant tests added. Test 1 is a first-letter test.

Notice that the image of JIR which is stored is a full image. Full response images must be stored—to provide the information for *producing* the response; but only partial stimulus images need be stored—to provide the information for *recognizing* the stimulus. How much stimulus image information is required the learning system determines for itself as it grows its discrimination net, and makes errors which it diagnoses as inadequate discrimination.

To pursue our simple example, suppose that the next syllable pair to be learned is PIB-JUK. There are no storage terminals in the net, as it stands, for the two new items. In other words, the net does not have the discriminative capability to contain more than two items. The input code for PIB is sorted by the net interpreter. Assume that Test 1 sorts it down the plus branch of Fig. 3. As there are differences between the incumbent image (with first letter D) and the new code (with first letter P) an attempt to store an image of PIB at this terminal would destroy the information previously stored there.

Clearly what is needed is the ability to discriminate further. A match for differences between the incumbent image and the challenging code is performed. When a difference is found, a new test is created to discriminate upon this difference. The new test is placed in the net at the point of failure to discriminate, an image of the new item is created, and both images—incumbent and new—are stored in terminals along their appropriate branches of the new test, and the conflict is resolved.[7] The net as it now stands is shown in Fig. 4. Test 2 is seen to discriminate on some difference between the letters P and D.

The input code for JUK is now sorted by the net interpreter. Since Test

[7] With the processes just described, the discrimination net would be grown each time a new item was to be added to the memory. But from an informational processing standpoint, the matching and net-growing processes are the most time-consuming in the system. In general, with little additional effort, more than one difference can be detected, and more than one discriminating test can be added to the net. Each redundant test placed in the net gives one "empty" image list. At some future time, if an item is sorted to this empty image list, an image can be stored without growing the net. There is a happy medium between small nets which must be grown all the time and large nets replete with redundant tests and a wasteful surplus of empty image lists. Experimentation with this "structural parameter" has been done and it has been found that for this study one or two redundant tests per growth represents the happy medium. However, I would not care to speak of the generality of this particular result.

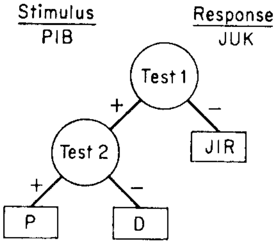

Figure 4. Discrimination net of Fig. 3 after the learning of stimulus item, PIB. Test 2 is a first-letter test.

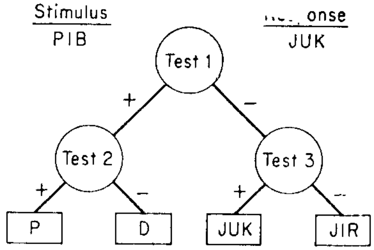

Figure 5. Discrimination net of Fig. 4 after the learning of the response item, JUK. Test 3 is a third-letter test.

1 cannot detect the difference between the input codes for JUK and JIR (under our previous assumption), JUK is sorted to the terminal containing the image of JIR. The match for differences takes place. Of course, there are no first-letter differences. But there are differences between the incumbent image and the new code in the second and third letters.

Noticing Order. In which letter should the matching process next scan for differences? In a serial machine like EPAM, this scanning must take place in some order. This order need not be arbitrarily determined and fixed. It can be made variable and adaptive. To this end EPAM has a *noticing order for letters of syllables,* which prescribes at any moment a letter-scanning sequence for the matching process. Because it is observed that subjects generally consider end letters before middle letters, the noticing order is initialized as follows: first letter, third letter, second letter. When a particular letter being scanned yields a difference, this letter is promoted up one position on the noticing order. Hence, letter positions relatively rich in differences quickly get priority in the scanning. In our example, because no first-letter differences were found between the image of JIR and code for JUK, the third letters are scanned and a difference is found (between R and K). A test is created to capitalize on this third-letter difference and the net is grown as before. The result is shown in Fig. 5. The noticing order is updated; third letter, promoted up one, is at the head.

Learning of subsequent items proceeds in the same way, and we shall not pursue the example further.

ASSOCIATING IMAGES: RETRIEVAL USING CUES

The discrimination net and its interpreter associate codes of external objects with internal image lists and images. But the basic rote learning experiment requires that stimulus information somehow lead to response information and a response. The discrimination net concept can be used for the association of internal images with each other (*i.e.,* response with stimulus) with very little addition to the basic mechanism.

An association between a stimulus image and a response image is accomplished by storing with the stimulus image some of the coded information about the response. This information is called *the cue.* A cue is of the same form as an input code, but generally contains far less information than an input code. A cue to an associated image can be stored in the discrimination net by the net interpreter to retrieve the associated image. If, for example, in the net of Fig. 3 we had stored with the stimulus image the letter J as a cue to the response JIR, then sorting this cue would have correctly retrieved the response image. *An EPAM internal association is built by storing with the stimulus image information sufficient to retrieve the response image from the net at the moment of association.*

The association process determines how much information is sufficient by trial and error. The noticing order for letters is consulted, and the first-priority letter is added to the cue. The cue is then sorted by the net interpreter and a response image is produced. It might be the wrong response image; for if a test seeks information which the cue does not contain, the interpreter branches left or right randomly (with equal probabilities) at this test.[8] During association, the selection of the wrong response is immediately detectable (by a matching process) because the response input code is available. The next-priority letter is added to the cue and the process repeats until the correct response image is retrieved. The association is then considered complete.

Note two important possibilities. First, by the process just described, a cue which is really not adequate to guarantee retrieval of the response image may by happenstance give the correct response image selection during association. This "luck" usually gives rise to response errors at a later time.

Second, suppose that the association building process does its job thoroughly. The cue which it builds is sufficient to retrieve the response image *at one particular time,* the time at which the two items were associated. If, at some future time, the net is grown to encompass new images being added to the memory, then a cue which previously was sufficient to correctly retrieve a response image may no longer be sufficient to retrieve that response image. In EPAM, association links are "dated," and ever vulnerable to interruption by further learning. Responses may be "unlearned" or "forgotten" temporarily, not because the response information has been destroyed in the memory, but because the information has been temporarily lost in a growing network. If an association failure of this type can be detected through feedback from the environmental or ex-

[8] This is the only use of a random variable in EPAM. We do not like it. We use it only because we have not yet discovered a plausible and satisfying adaptive mechanism for making the decision. The random mechanism does, however, give better results than the go-one-way-all-the-time mechanism which has also been used.

perimental situation, then the trouble is easily remedied by adding additional response information to the cue. If not, then the response may be more or less permanently lost in the net. The significance of this phenomenon will perhaps be more easily appreciated in the discussion of results of the EPAM simulation.

RESPONDING: INTERNAL AND EXTERNAL

A conceptual distinction is made between the process by which EPAM selects an internal response image and the process by which it converts this image into an output to the environment.

Response Retrieval. A stimulus item is presented. This stimulus input code is sorted in the discrimination net to retrieve the image list, in which the cue is found. The cue is sorted in the net to retrieve another image list containing the proposed response image. If there is no cue, or if on either sorting pass an empty image list is selected, no response is made.

Response Generation. For purposes of response generation, there is a fixed discrimination net (decoding net), assumed already learned, which transforms letter codes of internal images into output form. The response image is decoded letter by letter by the net interpreter in the decoding net for letters.

THE ORGANIZATION OF THE LEARNING TASK

The learning of nonsense symbols by the processes heretofore described takes time. EPAM is a serial machine. Therefore, the individual items must be dealt with in some sequence. This sequence is not arbitrarily prescribed. It is the result of higher order executive processes whose function is to control EPAM's focus of attention. These *macroprocesses,* as they are called, will not be described or discussed here. A full exposition of them is available in a paper by Feigenbaum and Simon (1962).

Stating the Model Precisely: Computer Program for EPAM

The EPAM model has been realized as a program in Information Processing Language V (Newell et al., 1961*e*) and is currently being run both on the Berkeley 7090 and the RAND 7090. Descriptive information on the computer realization, and also the complete IPL-V program and data structures for EPAM (as it stood in October, 1959) are given in an earlier work by the author (1959).

IPL-V, a list processing language, was well suited as a language for the EPAM model for these key reasons:

a. The IPL-V basic processes deal explicitly and directly with list *structures.* The various information structures in EPAM (e.g., discrimination net, image list) are handled most easily as list structures. Indeed, the discrimination is, virtually by definition, a list structure of a simple type.

b. It is useful in some places, and necessary in others, to store with some symbols information descriptive of these symbols. IPL-V's description list and description list processes are a good answer to this need.

c. The facility with which hierarchies of subroutine control can be written in IPL-V makes easy and uncomplicated the programming of the kind of complex control sequence which EPAM uses.

Empirical Explorations with EPAM

The procedure for exploring the behavior of EPAM is straightforward. We have written an "Experimenter" program and we give to this program the particular conditions of that experiment as input at the beginning of an experiment. The Experimenter routine then puts EPAM *qua* subject through its paces in that particular experiment. The complete record of stimuli presented and responses made is printed out, as is the final net. Any other information about the processing or the state of the EPAM memory can also be printed out.

A number of simulations of the basic paired-associate and serial-anticipation experiments have been run. Simulations of other classical experiments in the rote learning of nonsense syllables have also been run. The complete results of these simulation experiments and a comparison between EPAM's behavior and the reported behavior of human subjects will be the subject of a later report. However, some brief examples here will give an indication of results expected and met.

A. STIMULUS AND RESPONSE GENERALIZATION

These are psychological terms used to describe the following phenomenon. If X and X′ are similar stimuli, and Y is the correct response to the presentation of X; then if Y is given in response to the presentation of X′, this is called stimulus generalization. Likewise, if Y and Y′ are similar responses, and Y′ is given in response to the presentation of X, this is called response generalization. Generalization is common to the behavior of all subjects, and is found in the behavior of EPAM. It is a consequence of the responding process and the structure of the discrimination net. For those "stimuli" are similar in the EPAM memory whose input codes are sorted to the same terminal; and one "response" is similar to another if the one is stored in the same local area of the net as the other (and hence response error may occur when response cue information is insufficient).

B. OSCILLATION AND RETROACTIVE INHIBITION

We have described these phenomena in an earlier section.

Oscillation and retroactive inhibition appear in EPAM's behavior as consequences of simple mechanisms for discrimination, discrimination learning, and association. They were in no sense "designed into" the behavior. The appearance of rather complex phenomena such as these gives one a little more confidence in the credibility of the basic assumptions of the model.

These two phenomena are discussed together here because in EPAM they have the same origin. As items are learned over time, the discrimination net grows to encompass the new alternatives. Growing the net means adding new tests, which in turn means that more information will be examined in all objects being sorted. An important class of sorted objects is the set of cues. Cue information sufficient at one moment for a firm association may be insufficient at a later moment. As described above, this may lead to response failure. The failure is caused entirely by the ordinary process of learning new items. In the case of oscillation, the new items are items within a single list being learned. In the case of retroactive inhibition, the new items are items of the second list being learned in the same discrimination net. In both cases the reason for the response failure is the same. According to this explanation, the phenomena are first cousins (a hypothesis which has not been widely considered by psychologists).

In the EPAM model, the term *interference* is no longer merely descriptive—it has a precise and operational meaning. The process by which later learning interferes with earlier learning is completely specified.

C. FORGETTING

The usual explanations of forgetting use in one way or another the simple and appealing idea that stored information is physically destroyed in the brain over time (*e.g.*, the decay of a "memory trace," or the overwriting of old information by new information, as in a computer memory). Such explanations have never dealt adequately with the commonplace observation that all of us can remember, under certain conditions, detailed and seemingly unimportant information after very long time periods have elapsed. An alternative explanation, not so easily visualized, is that forgetting occurs not because of information destruction but because learned material gets lost and inaccessible in a large and growing association network.

EPAM forgets seemingly well-learned responses. This forgetting occurs as a direct consequence of later learning by the learning processes. Furthermore, forgetting is only temporary: lost associations can be reconstructed by storing more cue information. EPAM provides a mechanism for explaining the forgetting phenomenon in the absence of any information loss. As far as we know, it is the first concrete demonstration of this type of forgetting in a learning machine.

Conclusion: A Look Ahead

Verification of an information processing theory is obtained by simulating many different experiments and by comparing in detail specific qualitative and quantitative features of real behavior with the behavior of the simulation. To date, H. A. Simon and I have run a number of simulated experiments. As we explore verbal learning further, more of these will be necessary.

We have been experimenting with a variety of "sense modes" for EPAM, corresponding to "visual" input and "written" output, "auditory" input and "oral" output, "muscular" inputs and outputs. To each mode corresponds a perceptual input coding scheme, and a discrimination net. Associations across nets, as well as the familiar associations within nets, are now possible. Internal transformations between representations in different modes are possible. Thus, EPAM can "sound" in the "mind's ear" what it "sees" in the "mind's eye," just as all of us do so easily. We have been teaching EPAM to read by association, much as one teaches a small child beginning reading. We have only begun to explore this new addition.

The EPAM model has pointed up a failure shared by all existing theories of rote learning (including the present EPAM). It is the problem of whether association takes place between symbols or between tokens of these symbols. For example, EPAM cannot learn a serial list in which the same item occurs twice. It cannot distinguish between the first and second occurrence of the item. To resolve the problem we have formulated (and are testing) processes for building, storing, and responding from chains of token associations (Feigenbaum and Simon, 1962).