AD-767 716

COGNITIVE ORGANIZATION AND LEARNING

Donald A. Norman

California University

Prepared for:

Office of Naval Research
Advanced Research Projects Agency

August 1973

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

AD 767 716

COGNITIVE ORGANIZATION AND LEARNING

DONALD A. NORMAN

Contract Number: N00014-69-A-0200-6045

ARPA Order Number: 2284

Task Authority Identification Number: NR 154-360

Program Code Number: 3D20

Effective Date of Contract: 1 February, 1973

Contract Expiration Date: 31 January, 1974

Amount of Contract: $100,437

Organization: University of California, San Diego

Principal Investigator: Donald A. Norman

[Phone (714) 453-2000 X 2018]

Scientific Officer: Dr. Marshall Farr, Director
Personnel and Training Research Programs
Office of Naval Research

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency, the Office of Naval Research, or the U.S. Government.

This research was supported by the Advanced Research Projects Agency and the Office of Naval Research of the Department of Defense and was monitored by ONR under Contract No. N00014-69-A-0200-6045.

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
U S Department of Commerce
Springfield VA 22151

37

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Cognitive Organization and Learning		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Donald A. Norman		8. CONTRACT OR GRANT NUMBER(s) N00014-69-A-0200-6045
9. PERFORMING ORGANIZATION NAME AND ADDRESS Prof. Donald A. Norman Center for Human Information Processing University of California, San Diego/La Jolla,Ca.92037		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Personnel and Training Research Programs N00014 Office of Naval Research Arlington, VA.22217		12. REPORT DATE August, 1973
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Cognitive organization, learning, memory, computer languages, problem solving, education, cognitive psychology, semantic networks, natural language processing, interactive teaching.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

When one learns complex material, the important thing appears to be the ability to understand the material. Once understanding occurs, then learning and remembering follow automatically. The conventional psychological literature says little about the processes involved in the learning of complex material--material that takes weeks, months, or even years to be learned. Yet, most adult learning is of this form. This paper examines some of the issues that are involved in this type of learning, issues such as the nature

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (Abstract)
of understanding, the types of hypotheses that subjects bring to bear on the learning process, and the types of processes that need to be studied in order to understand the psychology of learning.

Cognitive Organization and Learning[1]

Donald A. Norman
University of California, San Diego
La Jolla, California 92037, USA

Memory and Learning

Just what happens when a complex subject matter is learned? Despite progress in unravelling the structures of sensory and short-term memory and despite tantalizing starts towards the representation of semantic material, we still have little understanding of just what happens in learning.

One thing concerns me about our understanding of the memory process. When I learn new material—such as the content of the papers presented at this symposium—almost none of this learning requires the kind of attentive rehearsal processes so well studied in the psychological literature on short- and long-term memory. I listen. I understand. And that is that. No rehearsal. No formal attempt to categorize or organize. Simply understanding. If I fail to understand, then I will also fail to remember. I have been examining the learning of complex material in an attempt to determine the psychological processes that are acting to ensure proper retention. (Complex materials are those that require days or weeks to acquire.)

Again, I find that the traditional acquisition processes which I have heretofore been studying are of little importance. In this paper, I make a step towards a description of the mechanisms that do operate in these situations.

One thing seems to be clear: in order to learn material for later use, it helps to have that material organized in an appropriate manner. If this is done, or if the process of organizing the material is part of the learning task, then the permanent acquisition of that material appears to be done reasonably effortlessly. Otherwise, effort is needed to ensure its permanent retention. The psychological literature shows many examples of the efficient learning that can occur when organizational processes accompany exposure to material--even when no formal attempt is made to retain the material.

Two Types of Rehearsal

In the earlier theories of the way that information was transferred from short-term to long-term memory, it was thought that rehearsal played an important role (for example, the model of Waugh & Norman, 1965). Today, it appears that the argument is not so simple. Rehearsal of an item appears to help strengthen its memory representation, but in ways that need not be useful for later recall. Craik & Watkins (in press), Bjork (in press), and Woodward, Bjork & Jongeward (in press) have shown how repeated rehearsal of some material need not lead to an increase in the ability of a subject to recall that material. More important, they have found it necessary to distinguish between two types of rehearsal processes: one that seems primarily effective in maintaining the item within short-term memory, the other that seems designed to aid in its later retrievability. Let me call these two forms rehearsal for maintenance and for accessibility.

The paradigmatic experiment looks like this (after Woodward et al.): A subject is shown a list of words, some of which he is going to be asked to remember for a future test, others of which will not be tested, but he is not told which is the case until some interval of time has passed from that item's presentation (nothing else is presented in that interval, however). Thus, during that interval, the subject can either choose to attempt to learn the item presented or can instead simply choose to maintain it in STM, waiting for the appropriate signal to decide what to do next.

Woodward et al. showed that the interval of time that passed between the initial presentation and the signal did not affect the ability of the subject to recall the material at a later test, whereas the length of the interval that occurred after the signal and before the presentation of the next item did make a difference. This fact alone would seem to suggest that during the initial period, subjects maintained the item in STM without transferring it to LTM. This is not completely correct, however, because when the subjects were tested on a recognition test, the earlier interval did make a difference.

Thus, we appear to have a rehearsal process that strengthens the memory trace as measured by a recognition measure, but that has no effect on memory as measured by a recall measure.
Any theory that postulates a role for rehearsal as a method for strengthening memory representation is partly correct, but fails to recognize that subjects can perform different kinds of processes during rehearsal. Moreover, it fails to take into account the distinctions between memory structures that are tested by tests that use recall techniques and by tests that use recognition.

Endel Tulving (see Tulving & Madigan, 1970) has long argued that it is important to distinguish between
the strength of an item in memory and its accessibility
That is, information could very well be present in a memory, but simply not easily accessible, especially by simple recall tests. The evidence collected by Woodward et al. supports this view by demonstrating that a subject's renearsal activities seem able to affect memory strength independently of memory accessibility.

Levels of Processing

Craik (1973) and Craik & Lockhart (1972) have argued that it is important to consider the depth of processing that memory information undergoes. To a large extent, they claim, the depth of processing determines the later retrievability of information: there is a strong correlation between how deeply an item is processed and the normal interpretation of the memory stage at which an item is represented. Thus, about the least meaningful amount of processing that an item can undergo is to be recognized, and this stage of pattern recognition is usually considered to be synonymous with short-term memory storage. Acoustic features are abstracted at this level. When an item is processed in some way that reflects its semantic characteristics, it also tends to be retained for longer periods of time. This argument was so compelling for Craik and Lockhart that they suggested a new interpretation of the literature on short-term and long-term memory. Perhaps, instead of several discrete types of memory, there is a single, unitary system, one in which items presented to it receive differing levels of processing and thereby give the appearance of different memory stages. I disagree with this part of their argument. Nonetheless, it isn't necessary to accept their entire argument to realize the importance of their analysis of the role that depth of processing plays. Consider this implication, however: the more deeply an item is processed, the more likely it is that it will be retained in memory.

The Learning of Complex Material

Most contemporary research on memory has concentrated on the study of relatively simple information. When we come to the study of how a person learns a complex body of material, we find almost nothing. Yet, in our normal adult lives, it is the learning of complex material that occupies most of our time. We must often study a topic for days, weeks, or even years before claiming to have mastered it. Some people are unable to learn some topics, even though others who appear equally able do so readily.

I would like to know what goes on within the head of the learner. To do so, it is clear that I must study topics related to the organization of material and to depth of processing. One interesting point about the learning of complex material is that often the problem is not really one of memory. Rather, the material is either understood (and therefore learned) or it is not. Often, the learner will struggle with a concept until he comes to understand it: at that point he claims to have learned it.

Why does memory acquisition appear to play such a major role in theories of psychology and yet a relatively minor role in much actual learning of complex topics? What is meant by "understanding"? These would appear to be the important questions that must be faced if ever we are to claim to understand human cognitive processes.

Note that it is possible to reformulate the learning of the traditional material studied in memory research.
When subjects are able to devise encoding strategies for the material, then they find that retention becomes easy, almost automatic. Many of the mnemonic systems in use by those who practice the art of memory make use of the fact that given a suitable encod-

ing, no real effort need be exerted to remember material. In the experimental laboratory, numerous investigators have shown that the discovery of a sensible encoding strategy makes memory for the items very easy. A good example of this is Bower's demonstration of the ease with which subjects could learn words once they were able to put them into sentences (see Bower, 1970, for a review on mnemonics). The classic study is probably the one that demonstrated that with suitable mnemonics, as many as 500 paired associates could be learned essentially completely with a single presentation of each pair (actual recall was 496 words correct: Wallace, Turner & Perkins, 1957). In these studies, as in all uses of mnemonic devices, the trick is to discover the appropriate mnemonic: once discovered, the learner can simply pass on to the next item with little or no further processing.

I will illustrate the problem of learning by considering two particular problems. First, I consider the teaching of a computer programming language. I choose this topic only because it is convenient, it seems to be at about the right level of difficulty to guide our studies. It is very simple. Someone who knows computer programming but who has never seen this particular language can usually learn it simply by reading the list of commands, a process that may take less than five minutes. Yet, students who have no background in programming sometimes take weeks to learn it. Thus, the topic is well defined, it is simple, yet it is complex enough to be a good vehicle for study.

Second, I consider a problem of rule induction from the traditional literature on problem solving: Luchins' (1942) water jug problem. This provides a typical paradigm for learning, even though it is not normally viewed in that light.

FLOW. The programming language called FLOW was developed by Professor Jeffrey Raskin at the University of California, San Diego for use in teaching computer programming to students in the visual arts, students who both had little knowledge of science and mathematics and who also disliked these subjects. FLOW is unique in several ways. First, it has been designed to simplify the process of entering information into the computer. At any point in the program, only the typewriter keys which lead to legal commands are operative. When a key which would lead to an illegal character is depressed, it has no effect. In addition, by a system called "typing amplification," whenever the user has typed a sufficient number of characters that the computer can unambiguously interpret which command is meant, the entire command appears on the screen without waiting for the student to finish. Thus, by these two features, the most common problems for the beginner are eliminated: typing errors and difficulty with the keyboard. In addition, we have modified the system to add several other useful features for our studies.[2]

The command set of FLOW is illustrated in Table 1. In this

Insert Table 1 about here

table, the part of the command that the student must type is underlined. The language is essentially self-explanatory, except perhaps for the commands that refer to "IT." IT is the name of a pointer that refers to a single letter in a string of text (the text is always the "TEXT IS ..." statement that was encountered most recently in the stream of processing). When first invoked, the IT pointer refers to the first letter of the text. Each time the command GET IT is used, the pointer moves one letter to the right

Table 1

The FLOW Language

The student only has to type the underlined letters. (Some commands fall into more than one category, and so they are repeated.)

CONTROL STATEMENTS

If it is 'E' go to 235

If counter is 42 go to 240

Go to 10

Stop

COUNTER CONTROL

Make counter zero

Add one to counter

Decrease counter by one

If counter is 7 go to 290

Print counter

TEXT MANIPULATION

Text is 'THE HOUSE IS RED.'

Get it

If it is 'E' go to 235

Print it

PRINT STATEMENTS

Print 'THAT IS CORRECT:'

Print return

Print counter

Print it

SYSTEM COMMANDS

Run

Walk

List

New

Help

(Escape)

Backspace -- Line numbers

along the text string. (A text string is assumed to contain an indefinite number of blanks at its right, so that when repeated use of the GET IT command runs out of letters on the text string, it will then continually point at a space.)

For example, a program to count the number of times the letter "E" occurs in a sentence looks like this:

```
 10 Text is "THIS IS A SAMPLE SENTENCE."
 20 Make counter zero
 30 Get it
 40 If it is "." go to 200
 50 If it is "E" go to 100
 60 Go to 30
100 Add one to counter
110 Go to 30
200 Print "The number of E's is"
210 Print counter
220 Stop
```

This simple language allows many fundamental properties of programming to be taught while maintaining a simple structure. The concepts of simple conditionals can be taught, as well as simple text manipulation. A pointer is present. Algorithms and loops or iterations can be taught. Some things cannot be done, such as letting program statements be called by other parts of the program (subroutines, co-routines, recursion). Any program that requires more than one pointer or more than one counter at a time cannot be performed. But despite these limitations, once the student has learned FLOW, he is in a good position to move on to any of the more advanced algebraic languages such as ALGOL, BASIC or FORTRAN.[3]

Teaching FLOW

The basic language contains a set of only 12 different commands. In addition, there are 5 system commands, as well as some simple conventions used for typing line numbers and correcting lines. The program just illustrated requires the use of only 9 different commands (including two different forms of the PRINT commands). In addition, the student should know at least 3 system commands or concepts: How to LIST a program, how to RUN a program, how to type in statements and how to correct errors. Clearly, however, there is more to understanding the language than these simple lists of concepts.

Learning as Problem Solving

In learning to program FLOW, the subject has to learn the commands by acquiring them into long-term memory. Yet the command structure is perhaps the easiest part of the learning task, and perhaps the least essential. Simple memory acquisition thus plays almost no role in the learning of this process. If we examine what a subject must do in order to solve a typical program--say the program just illustrated--we see that it is more like problem solving than anything else: the subject needs to learn how to solve programming problems.

Now we are faced with a new issue: what is involved in learning how to solve a problem? Indeed, what is involved in simply solving a problem? For one thing, to solve a problem, a subject must learn to put together rules and processes, not simply static concepts. One reason why our theories of memory are of such little use in the study of this topic is that the memory theories deal with static concepts and not with processes. Thus before continuing the discussion of FLOW, it makes sense to examine problem solving.

Problem Solving

To solve a problem at least two separate components are needed. First, the basic structure of the problem and of the allowable operations must be learned. Second, the operations must be combined in the appropriate way for getting to the desired goal. The learning of a board game or a card game (chess, gin rummy, go, bridge, gomoku) is a good example. Usually, an adult can be taught the rules of a new board or card game relatively easily, often with a single recitation of the rules. Moreover, once having heard those rules, the beginning player can make intelligent moves: he does not simply select moves randomly from the set of legal moves. Thus, some basic strategies do not have to be taught. The formal learning of the game structure may require only a short period of time. The learning of appropriate strategies that transform the initial strategies of a beginning player to the more advanced ones of an expert is both difficult and time consuming: the time is measured in years. There is very little research on this type of learning. All that I know about the stages in this learning process comes from the work by Eisenstadt & Kareev (1973). Although Newell & Simon (1972) have treated problem solving the most thoroughly of any modern investigators, they say very little about the learning of problem solving skills.

The Water Jug Problem. We can perhaps get an idea of what skills are involved in problem solving (and, therefore, in computer programming) by examining one of the classic studies in the literature: Luchins' (1942) studies of the water jug problem.

Here is a statement of a typical problem:

> You are given three jugs of water:
> one holds exactly 21 gallons,
> one holds exactly 127 gallons,
> and one holds exactly 3 gallons.
> Your task is to determine how
> you can end up with exactly 100
> gallons.

To solve this problem, the subject needs to know (or assume) that water is available from an unlimited source. He must also know what it means to fill and empty a container. In the original experiment by Luchins, this particular problem followed one that involved only two containers and in which the experimenter had explained the answer if the subject had not gotten it after 2 minutes. (The answer to this three jug problem was given if the subject could not discover it in 3 minutes.) After these two problems were presented, the experimental series of 9 similar problems followed.

To solve the problem, the subject must first understand the operations that are available, including the conditions for their application and the results that will occur. Notice that the operations themselves are not stated as a part of the problem.[4] Basically there are three types of operations: to pour from one container to another; to fill a container; and to empty a container. These are described in Table 2.[5]

Insert Table 2 here

No single one of the operations of Table 2 will solve the problem: to reach a solution, the basic operations must be combined in some way. To do this requires a meta-operation, a statement about what operations are permitted on the operations themselves. Finally, the meta-operations may yield a combination of operations that yield the answer. Call this sequence a strategy. In the actual experiment conducted by Luchins, after he gave his subjects this three jug problem, he gave them four more, where each of the five problems could be solved by the application of the same strategy:

* Fill the largest jar. (Empty the other two.)
* Pour from the largest jar to the medium jar.
* Pour from the largest jar to the smallest jar.
* Empty the smallest jar.
* Pour from the largest to the smallest.

The desired amount is now left in the largest jar.

The problem I am interested in is how subjects were able to derive that sequence. To Luchins, that was not of much interest. Rather, he wanted to demonstrate the fact that once a strategy had been developed, the subject would follow it. Thus, after the five three jug problems, he asked them to solve this one:

You are given a 23, a 49, and a 3 gallon container.
Your task is to get 20 gallons.

Table 2

Possible Operations for the Water Jug Problem

Let x be the name of a container. Then C(x) represents the capacity of container x and W(x) represents the amount of water contained in x.

Let $W_f(x)$ be the final state of W(x) and $W_i(x)$ the initial state.

Clearly,

Both C(x) and W(x) ≥ 0

and W(x) $\leq$ C(x).

Operation	Initial Conditions	Final Conditions
Pour from A to B.	$W_i(A) > 0$ $W_i(B) < C(B)$	If $W_i(A) < C(B) - W_i(B)$, $W_f(A) = 0$ $W_f(B) = W_i(B) + W_i(A)$ Otherwise, $W_f(A) = W_i(A) - [C(B) - W_i(B)]$
Fill A.	$W_i(A) < C(A)$	$W_f(A) = C(A)$
Empty A.	$W_i(A) > 0$	$W_f(A) = 0$

Meta-operations for Combining Operations

1. Any operation may follow any other operation if the proper preconditions are met.
2. Any operation that leads to a state that was previously present is not permitted.

Subjects invariably follow the strategy they had developed previously, emptying the 49 gallon container once into the 23 gallon one, then emptying it twice into the 3 gallon one, thus getting 20 gallons (49 - 23 - 3 - 3). Rarely does a subject who has experienced the entire sequence of problems simply pour from the 23 gallon container to the 3 gallon one.

To Luchins, this was the detrimental effect of set or *einstellung*. I believe it to illustrate quite a different point: the efficient use of strategies. I suspect that the development of a new strategy by application of meta-rules requires more effort than the following of a previously determined strategy. If this is true, then the subjects were minimizing their mental effort by solving this new problem simply by applying a strategy that was known to be successful.

The solution of a problem requires a number of different steps and concepts. The problem must be understood. The various permissible operations must be determined, and they in turn must be understood. A set of meta-operations must be applied to form a strategy. Finally, if there is to be any learning, successful strategies must be recognized and used again, perhaps by adding newly developed strategies to the set of operations that are permissible.

Learning FLOW

Now let us return to the examination of the computer programming language. To program, it is necessary that the student acquire some knowledge of how commands may be combined. Before this can be done, the student must understand what computer languages and programs are. The student must understand the problem he is to solve, he must know the specific commands of the language, and he must understand the meta-operations that characterize the art of programming. Finally, he must acquire specific strategies for specific classes of problems.

The difficulties of learning to program are almost completely divorced from the formal difficulty of "learning" or of acquisition into memory. The hard task is the task of understanding: once that is accomplished, memory follows automatically.

Propositional and Algorithmic Knowledge

In his recent studies of learning, Greeno has considered issues similar to those discussed here (see Greeno, 1973). One issue concerns the relationship between general factual knowledge (such as the statement that a jug can contain water) and algorithmic knowledge (such as the strategy that solves the water jug problem). The distinction between the two forms is not very clean, and in fact it is possible to argue that algorithms are simply collections of factual statements: in an algorithm, the statements convey information about operations that can be performed. As Greeno himself puts it:

> "Factually, it seems clear that the structures in semantic memory cannot be neatly partitioned into some that are algorithmic and others that are propositional in the more general, conceptual sense. I feel quite sure that I have both propositional and algorithmic knowledge about physical density, and I am not proposing that in such cases we postulate two cognitive structures, one for the algorithm and the other for conceptual understanding. However, there are also some relatively clear cases. My knowledge about integration by parts is almost purely algorithmic, except for the conceptual knowledge that I have about integrals generally. I can remember the algorithm (sometimes) and carry out the operations, but I do not have much understanding of why it works, and I do not predict at all reliably when it will be helpful."
>
> (Greeno, 1973, pp. 114-115)

But even if the distinction is not clear, it is still useful. Consider the FLOW language: To be able to solve the problem of printing the number of E's in a sentence, the student must understand both conceptual and algorithmic ideas. He must know the conceptual knowledge carried by the commands (see Table 1). In addition, he must be aware of the ways in which these statements are entered into the computer, and how the program is actually executed. He must also have certain types of algorithmic knowledge, for example:

* How to do tests for proper conditions.
* How to do loops, or iterations.
* How to properly sequence instructions.

Student Presumptions

I have already remarked on the fact that even a naive learner starts with certain basic assumptions and predetermined strategies. Thus, upon learning the rules of a new game, the adult player plays sensibly even on his first attempt. This beginning knowledge may not be a virtue. One major difficulty in teaching a topic is to overcome students' prior concepts. When students have difficulty in acquiring a concept, it often means they are attempting to acquire the wrong one. Consider these examples:

* The instructor in the normal university course on FLOW (Professor Raskin) studiously avoids telling the students that what they are doing has anything to do with computers until long after they have learned to program (the word "program" is not used either). Otherwise, he says, a student who dislikes science may invoke the one firm conceptual statement of knowledge he has about computers: He can't understand them. This is a self-fulfilling prophecy.

* Students may sometimes struggle with the keyboard for long periods of time, evidently failing to understand very simple concepts.

* One student spent considerable time constructing the symbol for the letter "O" (oh) out of a left and right parentheses. The reason: the computer typewriter (a General Electric Terminet) which was used to print the teaching manual for FLOW has a gap in the O at the top and bottom in order to distinguish it from a zero. Clearly, this attribute of the problem was thought to be important by the student even though it was of absolutely no interest to us.

These examples show something of the nature of the hypotheses that students bring to bear on the learning task. In some sense, these prior hypotheses are at the crux of the problem. To the student, the task is very much one of concept formation. He must somehow grasp the concepts that are involved in the situation. No matter that the experimenter is on his side, trying to help. The subject still must hypothesize and test, hopefully managing to separate the irrelevant from the relevant.

These errors of students are comical at first, but they are serious. They indicate that the students are struggling to perform the appropriate task, but that they have grasped at the incorrect concepts. The problem is akin to that faced in the concept formation literature when incorrect hypotheses often cause a subject to fail to get the appropriate classification chosen by the experimenter (See Trabasso & Bower, 1968; or Levine, 1971). We could argue, in fact, that this is a problem in attention, with the student selecting for attention in the learning situation inappropriate attributes of the situation. Thus, the student who is attempting to print a symbol by combining two parentheses is working efficiently at the wrong aspect of the problem, one that in concept formation terms is simply called an irrelevant dimension.

We see that part of what is going on in the learning of complex tasks is akin to the processes of concept formation. Memory does not play much of a role in the understanding of either complex tasks or concepts, although it is important in allowing the subject to determine just what attributes of the situation are relevant. Once the appropriate concept is discovered, then it must be remembered, of course, but this learning is probably more like the simple all-or-none learning discussed by Bjork (1968) for the acquisition of simple strategies than it is of the more laborious type of learning by repeated trials that is so often discussed in the literature on short- and long-term memory.

The concepts involved here, of course, differ considerably from those normally studied within the concept formation literature. Moreover, the learning situation differs from the other in that the experimenter is desperately attempting to help the student acquire the relevant concepts. Nonetheless that analogy seems meaningful.

What Should We Teach?

If conventional theories of learning and memory have taught us one thing, it is that factual knowledge is difficult to learn: it becomes a lesson in paired-associate learning. Functional knowledge is different. Once the function is understood, the knowledge appears with relative ease. it is derived, not memorized. The functions are mnemonic devices, and so it is function that we should be teaching.

There still remains the question of why: Why is functional knowledge easier to learn than factual knowledge? There is one reasonably obvious possible reason. A function has more constraints on its possible relations to other concepts than does a list of concepts.

Earlier (Norman, 1973), I made the argument that new material is learned most efficiently when integrated within the network of old knowledge by means of a web structure rather than a linear process. That is, the learning should add the new material by attaching a conceptual framework within memory to the old, previously acquired material. Then that framework should be elaborated upon, filling in the rough web with finer structures until a complete network has been acquired. By these successive stages of elaboration, new material is always firmly established within the framework created by the old. With more conventional teaching processes, learning occurs in a linear fashion, so that each new piece of information depends upon the knowledge of the succeeding new information. In this case, the structure is not sufficiently redundant, and if one link fails, then so must all that follows (see Figures 1 and 2).

Insert Figures 1 and 2 about here

In part, what functional learning might do is tie concepts with one another. Rote learning by temporal association is like linear learning; functional learning is web learning.

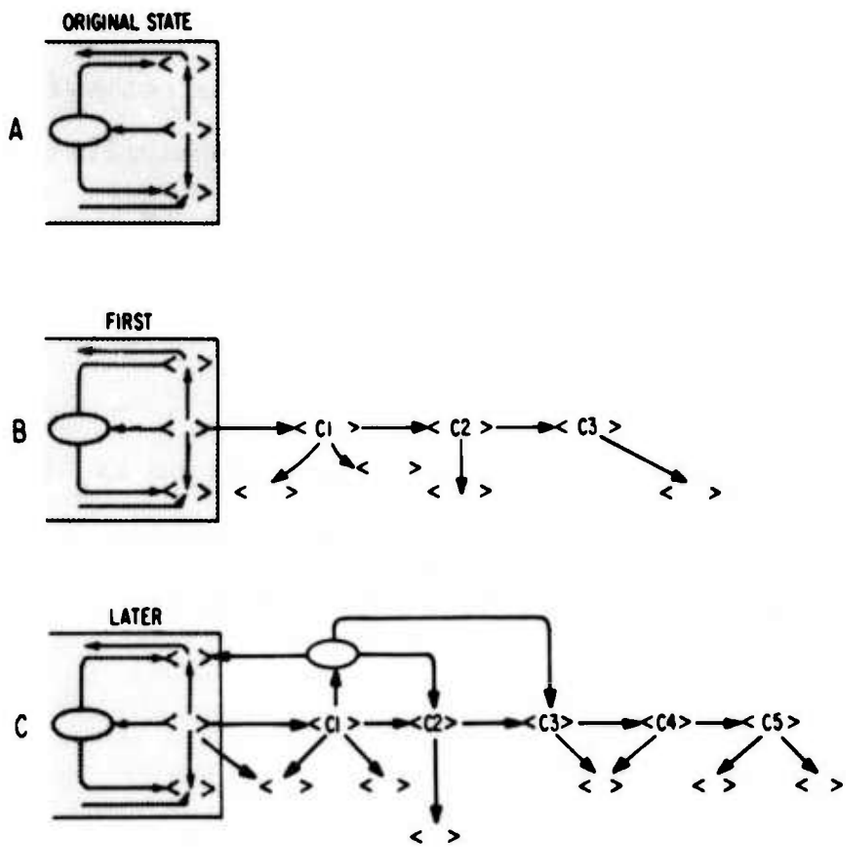

Figure 1. Linear Learning. New knowledge is added to the previous structure in a linear order. Lose one link, and all the rest is lost. Reprinted from Norman (1973) with permission of the publisher.

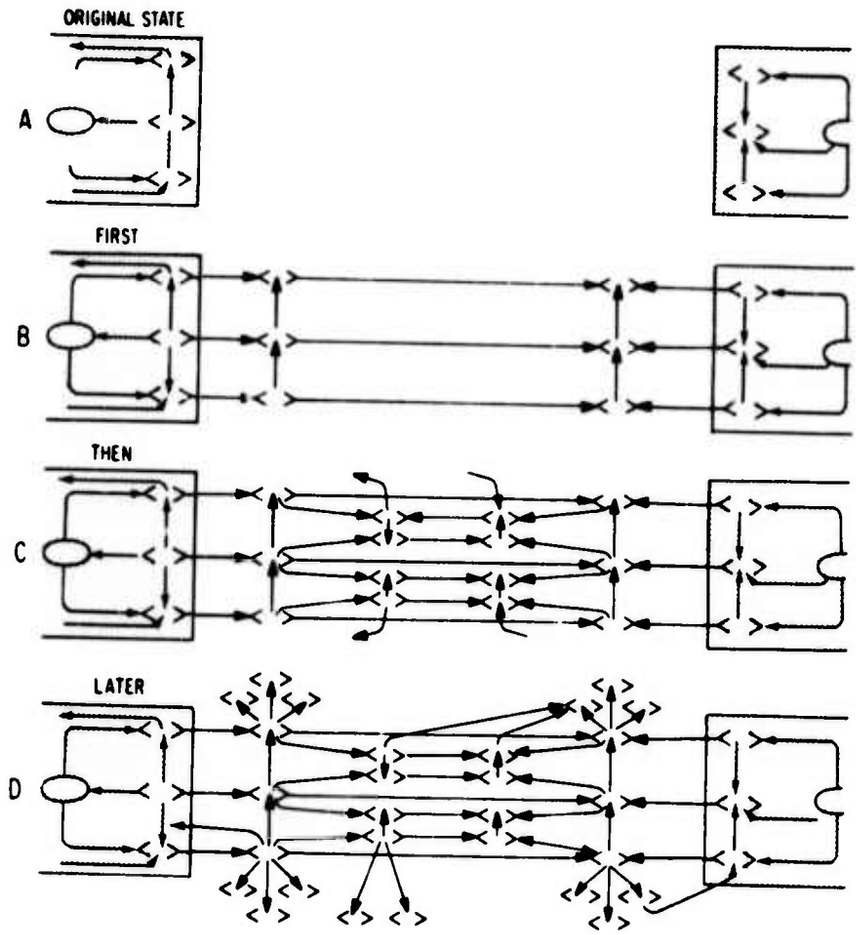

Figure 2. Web Learning. First form a coarse net of concepts, each well connected to previous knowledge. Then build upon that to construct a firm, integrated web. This procedure is not so easy to perform as that of linear learning, but it should yield superior results. Reprinted from Norman (1973) with permission of the publisher.

The Near Miss

One very important contribution to the study of learning is provided by Winston (1973). What is an arch? A tower? A pedestal? Winston worried about how a system could come to learn these concepts simply by observing structures built from blocks while being told what they were. Suppose the task is to learn an "arch." The scheme he devised is that of noticing (nearly) all relations among the blocks, and using that set to define an arch. But suppose the next example fulfilled the relations and yet was not an arch. This is the critical learning step, and for Winston, the nature of these examples was very important. The learning trials should consist of a carefully selected set of positive examples along with negative examples that were near misses. The learner forms the appropriate concept by comparing his acquired structure with that for the near miss, noticing the critical distinctions that cause the example to lie outside the definition.

Winston derived a working example of how concepts might be learned (his system is a working computer program). His most important contribution, however, might be in his consideration of the importance of the training-sequence, and in particular, the importance of the near miss. What Winston appears to have done is to provide teaching techniques that make clear just which attributes of the problem are the relevant ones. Thus, he combines an active learning process with subtle guides for the direction that the learner must follow.

Toward a New Learning Theory

We must learn to characterize the strategies that are acquired by students. When a subject is engaged in a learning situation, he brings to bear not only his knowledge of the subject matter, but also his idea of the expectations of the situation. He has some overall concept of the situation he is in, he has a concept of the performance expected of him, and he has some idea of the appropriate types of operations he can perform. Finally, he has to have some idea of the basic commands or operations available to solve the problems put before him.

Clearly new knowledge must be well integrated within old if it is to be acquired easily and effectively. But this statement hides much. For one, the knowledge consists often of processes and routines, not static concepts. The memory theories developed in recent years say little or nothing of how anything other than concepts are acquired. For another, the fact that subjects maintain hierarchical levels of strategies and meta-strategies means that the description of the integration of knowledge will have to account for the different levels of activity that is involved.

Two aspects of recent studies of memory seem especially relevant. First is the type of activities that subjects perform during the accessibility form of rehearsal which help that material become available for both later recognition and recall. (A comparison with what subjects do in the maintenance form of rehearsal would be productive.) Second is continued study of the notion of "depth of processing," most especially with the aim of determining why depth is so often correlated with good retention.

New knowledge seems easier to acquire when it is learned as a process than when it is learned as a collection of facts. All these features: of hypotheses, of process learning, of strategies, of incorporating new within old must be studied in order that we can learn about the process of learning.

Footnotes

1. The research reported here was supported by the National Science Foundation Grant GB32235X and by the Advanced Research Projects Agency and the Office of Naval Research of the Department of Defense and was monitored by ONR under Contract No. N00014-69-A-0200-6045.

2. The experimental studies are carried out by me and Donald Gentner. The system programming has been performed by Mark Wallen.

3. To the reader who believes this to be too simple a language to take seriously, I urge him to attempt these two problems.
   a. Print "yes" if the last two letters of the text are "ie" or "ei"; print "no" otherwise.
      The program should print "yes" for dei, die, diie and diei, and "no" for died, dice and deii.
   b. Print the first word that has an "E". If the text is "This is a sample sentence", the program should print "sample".

4. It is interesting to study now a subject is able to expand the statement of the problem into the set of conditions necessary to solve it. A reasonable amount of world knowledge is required. In my informal experiments, young children--around 7 to 8 years--tend to disallow the operation of emptying a container onto the ground. They tend to add another presupposition to the problem: either that water cannot be wasted, or that "backup" is not allowed.

5. Clearly there are alternative ways of stating these operations. The rules could be more general, with the three rules stated here subsumed under a single "pour" operation. They could also be more specific, so that for the three jug problem there would be three "fill" operations, three "empty" operations, and six "pours." Although the form does not matter, the content does: the knowledge in Table 2 must be known in order to solve the problem.

References

Bjork, R. A. All-or-none subprocess in the learning of complex sequences. Journal of Mathematical Psychology, 1968, 5, 192-195.

Bjork, R. A. Rehearsal as a short-term holding operation in memory. In press.

Bower, G. H. Analysis of a mnemonic device. American Scientist, 1970, 58, 496-510.

Craik, F. I. M. A "level of analysis" view of memory. In P. Plimer, L. Krames and T. M. Alloway (Eds.), Communication and Affect: Language and Thought. New York: Academic Press, 1973.

Craik, F. I. M. & Lockhart, R. S. Levels of processing: A framework for memory research. Journal of Verbal Learning and Verbal Behavior, 1972, 11, 671-684.

Craik, F. I. M. & Watkins, M.J. The role of rehearsal in short-term memory. Journal of Verbal Learning and Verbal Behavior, in press.

Eisenstadt, M. & Kareev, Y. Towards a model of human game playing. Proceedings of the Third International Conference on Artificial Intelligence. Stanford, California, 1973. (Also available as La Jolla, California: Center for Human Information Processing, University of California, Technical Report 33.)

Greeno, J. G. The structure of memory and the process of solving problems. In R. L. Solso (Ed.), Contemporary Issues in Cognitive Psychology: The Loyola Symposium. Washington, D.C.: Winston, 1973 (Distributed by Halsted Press, John Wiley and Sons).

Levine, M. Hypothesis theory and nonlearning despite ideal S-R-reinforcement contingencies. Psychological Review, 1971, 78, 130-140.

Luchins, A. S. Mechanization in problem solving. Psychological Monographs, 1942, 54(6), Whole number 248.

Newell, A. & Simon, H. A. Human Problem Solving. Englewood Cliffs, New Jersey: Prentice-Hall, 1972.

Norman, D. A. Memory, knowledge, and the answering of questions. In R. L. Solso (Ed.), Contemporary Issues in Cognitive Psychology: The Loyola Symposium. Washington, D.C.: Winston, 1973 (Distributed by Halsted Press, John Wiley and Sons).

Trabasso, T. & Bower, G. H. Attention in Learning. New York: Wiley, 1968.

Tulving, E. & Madigan, S. A. Memory and verbal learning. Annual Review of Psychology, 1970, 21, 437-484.

Wallace, W. H., Turner, S. H. & Perkins, C. C. Preliminary studies of human information storage. Philadelphia: Institute for Co-operative Research, The Moore School of Electrical Engineering Technical Report, The University of Pennsylvania, 1957.

Waugh, N. C. & Norman, D. A. Primary memory. Psychological Review, 1965, 76, 89-104.

Winston, P. Learning to identify toy block structures. In R. L. Solso (Ed.), Contemporary Issues in Cognitive Psychology: The Loyola Symposium. Washington, D.C.: Winston, 1973 (Distributed by Halsted Press, John Wiley and Sons).

Woodward, A. E., Jr., Bjork, R. A. & Jongeward, R. H., Jr. Recall and recognition as a function of primary rehearsal. Journal of Verbal Learning and Verbal Behavior, in press.