SCRIPTS, PLANS, AND KNOWLEDGE

Roger C. Schank and Robert P. Abelson

Yale University
New Haven, Connecticut USA

"Of what a strange nature is knowledge! It clings to the mind, when it has once seized on it, like a lichen on the rock,"

- Frankenstein's Monster
(M. Shelley, *Frankenstein or the Modern Prometheus,* 1818)

Abstract

We describe a theoretical system intended to facilitate the use of knowledge In an understanding system. The notion of script is introduced *to* account for knowledge about mundane situations. A program, SAM, is capable of using scripts to understand. The notion of plans is introduced to account for general knowledge about novel situations.

I. Preface

In an attempt to provide theory where there have been mostly unrelated systems, Minsky (1974) recently described the work of Schank (1973a), Abelson (1973), Charniak (1972), and Norman (1972) as fitting into the notion of "frames." Minsky attempted to relate this work, in what is essentially language processing, to areas of vision research that conform to the same notion.

MInsky's frames paper has created quite a stir in AI and some immediate spinoff research along the lines of developing frames manipulators (e.g. Bobrow, 1975; Winograd, 1975). We find that we agree with much of what Minsky said about frames and with his characterization of our own work. The frames idea is so general, however, that It does not lend itself to applications without further specialization. This paper is an attempt to develop further the lines of thought set out in Schank (1975a) and Abelson (1973; 1975a). The ideas presented here can be viewed as a specialization of the frame idea. We shall refer to our central constructs as "scripts."

II. The Problem

Researchers in natural language understanding have felt for some time that the eventual limit on the solution of our problem will be our ability to characterize world knowledge. Various researchers have approached world knowledge in various ways. Winograd (1972) dealt with the problem by severely restricting the world. This approach had the positive effect of producing a working system and the negative effect of producing one that was only minimally extendable. Charniak (1972) approached the problem from the other end entirely and has made some interesting first steps, but because his work is not grounded in any representational system or any working computational system the restriction of world knowledge need not critically concern him.

Our feeling is that an effective characterization of knowledge can result in a real understanding system in the not too distant future. We expect that programs based on the theory we outline here and on our previous work on conceptual dependency and belief systems will combine with the MARGIE system (Schank et al., 1973 a; Riesbeck, 1975; Rieger, 1975) to produce a working understander. We see understanding as the fitting of new information into a previously organized view of the world. We have therefore extended our work on language analysis (Schank, 1973a; Riesbeck 1975) to understanding - an understander, like an analyzer, should be "bottom up" until it gets enough information to make predictions and become "top down." Earlier work has found various ways In which a word in a single sentence sets up expectations about what is likely to be found in the rest of the sentence. A single sentence and its corresponding conceptualizations set up expectations about what is to follow in the rest of a discourse or story. These expectations characterize the world knowledge that bears on a given situation, and it is these expectations that we wish to explore.

III. Scripts

A script, as we use It, Is a structure that describes an appropriate sequence of events in a particular context. A script is made up of slots and requirements about what can fill those slots. The structure is an Interconnected whole, and what is in one slot affects what can be in another. Scripts handle stylized everyday situations. They are not subject to much change, nor do they provide the apparatus for handling novel situations, as plans do (see section V).

For our purposes, a script is a predetermined, stereotyped sequence of actions that define a well-known situation. A script is, in effect, a very boring little story. Scripts allow for new references to objects within them just as if these objects had been previously mentioned; objects within a script may take "the" without explicit introduction because the script itself has already implicitly introduced them. (This can be found below, in the reference to "the waitress" in a restaurant, for example.)

Stories can invoke scripts In various ways. Usually a story is a script with one or more interesting deviations.

I. John went into the restaurant.
He ordered a hamburger and a coke.
He asked the waitress for the check and left.

II. John went to a restaurant.
He ordered a hamburger.
It was cold when the waitress brought it.
He left her a very small tip.

III. Harriet went to a birthday party.

t The work of the second author was facilitated by National Science Foundation Crant CS-35768.

She put on a green paper hat.
Just when they sat down to eat the cake, a piece of plaster fell from the ceiling onto the table.
She was lucky, because the dust didn't get all over her hair.

IV. Harriet went to Jack's birthday party.
The cake tasted awful.
Harriet left Jack's mother a very small tip.

Paragraph I is an unmodified script. It is dull. It would be even duller if all the events in the standard restaurant script (see below) were included.

Paragraph II is a restaurant script with a stock variation, a customer's typical reaction when things go wrong.

Paragraph III invokes the birthday party script, but something wholly outside the range of normal birthday parties occurs - the plaster falls from the ceiling. This deviation from the script takes over the initiative in the narrative until the problem it raises is resolved, but the birthday script Is still available in the indirect reference to the party hat and in the possibility that normal party activities be resumed later in the narrative. It seems natural for reference to be made to dust in the hair following the plaster's falling, which implies that there is a kind of script for falling plaster too. (This kind of script we call a vignette (Abelson, 1975a).) Notice that "the ceiling" refers to an uninteresting "room" script, which can be used for references to doors and windows that may occur. Thus it is possible to be in more than one script at a time.

Paragraph IV illustrates the kind of absurdity that arises when an action from one script is arbitrarily inserted Into another. That one feels the absurdity is an indication that scripts are in inadmlssable competition. It is conceivable that with adequate introduction the absurdity in paragraph IV could be eliminated.

With these examples, a number of Issues have been raised. Let us at this point give a more extensive description of scripts. We have discussed previously (Schank, 1975b) how paragraphs are represented in memory as causal chains. This work implies that, for a story to be understood, inferences must connect each input conceptualization to all the others in the story that relate to it. This connection process is facilitated tremendously by the use of scripts.

Scripts are extremely numerous. There is a restaurant script, a birthday party script, a football game script, a classroom script, and so on. Each script has players who assume roles in the action. A script takes the point of view of one of these players, and it often changes when it is viewed from another player's point of view.

The following is a sketch of a script for a restaurant from the point of view of the customer. Actions are specified in terms of the primitive ACTs of conceptual dependency theory (Schank, 1973b.

```
script:   restaurant
roles:    customer, waitress, chef, cashier
reason:   to get food so as to go up in pleasure
             and down in hunger

scene 1:  entering
          PTRANS self into restaurant
          ATTEND eyes to where empty tables are
          MBUILD where to sit
          PTRANS self to table
          MOVE sit down

scene 2:  ordering
          ATRANS receive menu
          MTRANS read menu
          MBUILD decide what self wants
          MTRANS order to waitress

scene 3:  eating
          ATRANS receive food
          INGEST food

scene 4:  exiting
          MTRANS ask for check
          ATRANS receive check
          ATRANS tip to waitress
          PTRANS self to cashier
          ATRANS money to cashier
          PTRANS self out of restaurant
```

In this script, the instruments for performing an action might vary with circumstances. For example, in scene 2 the order might be spoken, or written down wi th predesignated numbers for each item, or even (in a foreign country wi th an unfamiliar language) indicated by pointing or gestures.

Each act sequence uses the principle of causal chaining (Schank, 1973b,Abelson, 1973). That is, each action results in conditions that enable the next to occur. To perform the next act in the sequence, the previous acts must be completed satisfactorily. If they cannot be, the hitches must be dealt with. Perhaps a new action not prescribed in the script will be generated in order to get things moving again. This "what-if" behavior, to be discussed later, is an important component of scripts. It is associated with many of the deviations in stories such as paragraph IT.

In a text, new script information is interpreted in terms of its place in one of the causal chains within the script. Thus in paragraph 1 the first sentence describes the first action in scene 1 of the restaurant script. Sencence 2 refers to the last action of scene 2, and Sentence 3 to the first and last actions of scene A. The final interpretation of paragraph I contains the entire restaurant script, with specific statements filled in and missing statements (that he sat down, for example) assumed.

In paragraph II, the first two sentences describe actions in scenes 1 and 2. Part of the third sentence is in the script as the first action of scene 3, but there is also the information that the hamburger is cold. The fourth sentence ("He left her a very small tip") is a modification of the third action of scene 4. The modifier, "very small," is presumably related to the unexpected information about the "cold hamburger." Even a stupid processor, checking paragraph II against the standard restaurant script, could come up with the low-level hypothesis that the small size of the tip must have something to do with the temperature of the hamburger, since these two Items *of* Information are the only deviations from the script. They must be related deviations, because if they were unrelated the narrative would have no business ending with two such unexplained features.

Of course we do not want our processor to be stupid. In slightly more complex examples, adequate understanding requires attention to the na-

ture of deviations from the script. A smarter processor can infer from a cold hamburger that the INCEST in scene 3 will then violate the pleasure goal for going to a restaurant. The concept of a very small tip car stored with the restaurant script as a what-if associated with violations of the pleasure goal.

The general form for a script, then, is a set of paths joined at certain crucial points that define the script. For restaurants the crucial parts are the INGEST and the ATRANS of money. There are many normal ways to move from point to point. Ordering may be done by MTRANSing to a waiter or by selecting and taking what you like (in a cafeteria). Likewise the ATRANS of money may be done by going to the cashier, or paying the waitress, or saying, "Put it on my bill." There are also paths to take when situations don't go as planned. Paragraphs III and IV call up deviant paths in the birthday party script. All these variations Indicate that a script is not a simple list of events but rather a linked causal chain; a script can branch into multiple possible paths that come together at crucial defining points.

To know when a script is appropriate, script headers are necessary. These headers define the circumstances under which a script is called into play. The headers for the restaurant script are concepts having to do with hunger, restaurants, and so on In the context of a plan of action for getting fed. Obviously contexts must be restricted to avoid calling up the restaurant script for sentences that use the word "restaurant" as a place ("Fuel oil was delivered to the restaurant").

Scripts organize new inputs in terms of previously stored knowledge. In paragraph I, many itenfi. that are part of the restaurant script are added to the final interpretation of the story. We don't need to say that a waitress took the customer's order or that he ate the hamburger. These ideas are firmly a part of the story because the restaurant script requires them. In understanding a story that calls up a script, the script becomes part of the story even when it is not spelled out. The answer to the question "Who served John the hamburger?" seems obvious, because our world knowledge, as embodied in scripts, answers it,

What-Ifs

There are at least three major ways in which scripts can be thrown off normal course. One is distraction, interruption by another script, such as the plaster's falling from the celling. We will not pursue here an analysis of the conditions and consequences of distraction. The other two ways, obstacle and error, are intimately connected with what-if behavior. An obstacle to the normal sequence occurs when someone or something prevents a normal action from occurring or some usual enabling condition for the action is absent. An error occurs when the action is completed in an inappropriate manner, so that the normal consequences of the action do not come about.

In principle, every simple ACT in a standard script has potential obstacles and errors. We assume that, every time an obstacle or error occurs in a script that is being learned, the methods used to remove the obstacle or redeem the error are stored with the script as what-ifs. The result of many repetitions is that most of the common what-ifs are attached to the script.

Every obstacle has one or more characteristic what-ifs. In scene 2 of the restaurant script, if the waitress ignores the customer, he will try to catch her eye or call to her when she passes nearby. If he can't make out the menu or needs further information, he will ask the waitress. If she doesn't speak his language, he will attempt her language, or make gestures, or seek another customer to translate, or accept her suggestion of what to order. In scene 3, if the waitress does not bring the food, he will again try to catch her eye. If the food is not fit, he will send it back.

Errors have a slightly different character from obstacles but follow the same general rules. Receiving the menu is errorful if the waitress ATRANSes a printed sheet to the customer but it is yesterday's menu, or the breakfast instead of the dinner menu. Reading the menu may yield an error if the customer gets the wrong idea of what it says - say, he thinks filet mignon is a fish. Here it is up to the waitress to supply the what-if corrective. Deciding what to order may yield an error if the customer goes through the decision process but forgets the stored outcome. The what-if is to review the MBUILD ("Let's see now, what did 1 decide?") Giving the order to the waitress may be in error if she writes down something other than what the customer said, or omits a portion of it. The what-if is to repeat the order, asking the waitress whether she is sure she got all of It.

IV. The SAM Program

We have created a program that uses scripts to make inferences in domains it knows about and have added it to the MARGIE program reported at 31JCAI. SAM (Script Applier Mechanism) receives Conceptual Dependency structures as input (these are gotten from English by using Riesbeck's parser (1975)). SAM identifies the script to be used (by using Riesbeck's request format) and fills in the identified script with the new inputs as they are received. It makes the inferences that are necessary to complete a causal chain between input conceptualizations (see Schank (1973b) for a discussion of causal chains). The resultant complex structure can then be queried by input English questions or be used to make summaries and paraphrases of the initial story. SAM was written by Chris Riesbeck, Richard Cullingford and Wendy Lehnert. Goldman's generation program (1975) is used to map the answers of paraphrases into English. It was modified for this purpose by Gerald DeJong and W.J. Stutzman. An example of SAM's input and output is:

Input:

John went to a restaurant. The hostess seated John. The hostess gave John a menu. The waiter came to the table. John ordered lobster. John was served quickly. John left a large tip. John left the restaurant.

Questions and Answers:

Q: What did John eat?
A: LOBSTER.
Q: Who gave John the menu?
A: THE HOSTESS.
Q: Who gave John the lobster?
A: PROBABLY THE WAITER.
Q: Who paid the check?
A: PROBABLY JOHN.

Q: What happened when John went to the table?
A: THE HOSTESS CAVE HIM A MENU AND JOHN SAT DOWN.
Q: Why did John get a menu?
A: SO HE COULD ORDER.
Q: Why did John give the waiter a large tip?
A: BECAUSE HE WAS SERVED QUICKLY.

Paraphrase:

JOHN DECIDED HE WAS GOING TO GO TO A RESTAURANT. JOHN WENT TO A RESTAURANT. JOHN INDICATED TO THE HOST HE WOULD LIKE TO BE AT A TABLE. THE HOST INDICATED TO JOHN HE COULD GO TO A TABLE. JOHN WENT TO THE TABLE. THE HOST WENT TO THE TABLE. JOHN SAT DOWN IN THE CHAIR. JOHN GOT THE MENU FROM THE HOST. JOHN READ THE MENU. THE WAITER SAW JOHN WAS AT THE TABLE. THE WATTER WENT TO THE TABLE. JOHN ORDERED LOBSTER. THE CHEF PREPARED THE LOBSTER. THE WAITER GOT THE LOBSTER FROM THE CHEF. THE WAITER SERVED JOHN THE LOBSTER. JOHN ATE THE LOBSTER. JOHN INDICATED TO THE WAITER HE WOULD LIKE TO GET THE CHECK FROM HIM. JOHN GOT THE CHECK FROM THE WAITER. JOHN LEFT THE WAITER A LARGE TIP. JOHN PAID THE CHECK. JOHN LEFT THE RESTAURANT.

Summary:

JOHN WENT TO A RESTAURANT AND ATE LOBSTER.

This program rune on the PDP-10 at Yale. It currently has only a small amount of knowledge and a small vocabulary. But we feel encouraged that our script theory is workable because of the simplification in the inference process that has resulted from the use of scripts.

V, Plans

Plans are responsible for the deliberate behavior that people exhibit. Plans describe the set of choices that a person has when he sets out to accomplish a goal. In listening to a discourse, people use plans to make sense of seemingly disconnected sentences. By finding a plan, an understander can make guesses about the intentions of an action in an unfolding story and use these guesses to make sense of the story.

Consider the following paragraph:

John knew that his wife's operation would be very expensive.
There was always Uncle Harry ...
He reached for the suburban phone book.

How are we to make sense of such a paragraph? It makes no use of headers or the scripts they signal. It would be unreasonable to posit a "paying for an operation" script with all the necessary acts laid out as in our restaurant script. But, on the other hand, the situation is not entirely novel, either. The problem of understanding this paragraph would not be significantly different if "wife's operation" were changed to "son's education" or "down payment on the mortgage." There is a general goal state in each case, raising a lot of money for a legitimate expense, and there is a generalized plan or group of plans that may lead to the goal state.

Plans start with one or more goals. A high-level goal is illustrated by the sequence:

John wanted to become king.
He went to get some arsenic.

A low-level goal is illustrated by:

John wanted to cut his steak.
He called to his wife in the kitchen.

A plan is a series of actions that will realize a goal. Often in order to realize one goal another must be decided on and a plan drawn up to achieve it. In the first example above, a goal to attain power is reduced to a goal to get arsenic. High-level goals are more interesting and we have concentrated on them first.

We define a "deltact" as an action or a group of actions that leads to a desired state. Deltacts constitute subplans that are pursued because of their intended effects. There are five deltacts in the present system:

ΔAGENCY - a change in obligation to do something for somebody
ΔCONT - a change in the control of an object
ΔKNOW - a change in what an actor knows
ΔPROX - a change in the proximity relations of objects and actors
ΔSOCCONT - a change in social control over a person or a situation

There is also a set of lower-level deltacts (Abelson, 1975a). Plans are made up of deltacts. When a collocation of deltacts is used often enough, it becomes a script.

A plan includes a set of planboxes, lists of actions that will yield state changes and the preconditions for these actions, along with a set of questions for choosing the appropriate planbox.

For instance, the TAKE plan has the goal of enabling the taker to do something with an object, whatever is generally done with it. To TAKE something you must be close to it, so either the object and the taker must be in the same location or the taker must use a subplan ΔPROX. Either no one else must have CONTROL of the object or at least there must be no bad consequences in the taker's attempt to PTRANS the object to himself. The TAKE plan calls a PTRANS of the object if all the preconditions are positive.

But if, say, someone else CONTROLS the object, a plan for the taker's gaining CONTROL must be called. This subplan is ΔCONT. ΔCONT has a set of planboxes attached to it. These planboxes define a deltact Just as inferences define a primitive ACT. A planbox is a list of primitive ACTs that will achieve a goal. Associated with each ACT are Its preconditions, and a planbox checks them. A set of positive conditions allows the desired ACT. Negative conditions call up new planboxes or deltacts that have as their goal the resolution of the negative state.

Preconditions fall into three classes. A controlled precondition can be fixed when it is negative by doing an ACT. A negative uncontrolled precondition cannot be fixed, and another planbox must be tried. Negative mediating preconditions can be altered but require plans of their own to change. Mediating preconditions usually refer to the willingness of other parties to participate in plans. Further details on planboxes appear in Schank (1975c).

To see why an understander needs plans, consider the following sequence:

Willa was hungry.
She took out the Michelin Guide.

Most readers understand that Willa was using the

Michelin Guide to find a good restaurant. But if the first sentence were subjected to straightforward inference (a la Rieger, 1975), predicting that Willa Is likely to do something to enable herself to INGEST fc the second sentence would seem to answer this prediction only in the weird interpretation that she will eat the Michelin Guide. An understander will reject this in favor of any better path that it can find. The first sentence will be analyzed for any goal that might generate a plan. "Hungry" is listed in the dictionary as indicating the need for a plan to do a ACONT of food. One *means* for gaining control of food is a restaurant. An enablement for this means is going to a restaurant, which requires APROX. This in turn requires knowing where you are going, which may require AKNOW.

In the dictionary, all books are listed as means of satisfying AKNOWs and the Michelin Guide is listed as a book. To complete the processing of this sequence it would, of course, be necessary to have the information that the Michelin Guide lists restaurants. Without thi6 information, the sequence might be as nonsensical as "Willa was hungry. She took out Introduction to Artificial Intelligence."

With the information that the Michelin Guide is a source of knowledge about restaurants, we know why the second action was done and can predict future actions. We have transformed a seemingly disconnected sequence into one that provides the expectations that are so vital to understanding. If the next sentence is "Willa got in her car," we will know that the plan is being effected. By using what we know about cars (that they are instruments of PTRANS) and the script for restaurants (that is starts with a PTRANS), we can make the inference that Willa is on her way to a restaurant. Some restaurant header would still be required to initiate the restaurant script in its full glory.

The procedure of taking out the Michelin Guide when hungry, while seemingly novel, could conceivably be routine for a certain individual in a certain context. If we know that Willa is a gourmet tourist staying in Paris who enjoys going to a different restaurant every evening, then the procedure of looking in the Guide might become part of her restaurant script. For her there is a scene before scene 1 in which she ATTENDS to the Guide, MBUILDs a choice, and MTRANSes a reservation. A routinized plan can become a script, at least from the planner's point of view.

VI. Conclusion and Prognosis

It is clear that in order to understand one needs knowledge. Knowledge is a potentially unwieldy thing, so what we must do is determine the types of knowledge that there are and find out how to apply them. The SAM system is a first step at adding struetured knowledge to the MARGIE system (Schank et al, 1973). We are currently building up our knowledge base by adding more scripts to SAM. In addition we are adding a plan component PAM. These two programs should bring us up to the level of understanding simple stories about a large range of known domains.

But what about complex stories? Is the kind of understanding that humans exhibit on real stories likely to resemble the mechanisms to be found in SAM?

When a person reads a 300 page novel he does not (unless he is very unusual) remember all the conceptualizations stated in the story in the form of a giant causal chain. Rather he remembers the gist of the book. Maybe 5 or 10 pages of summary could be extracted from him after reading the book. Previously we have said that Conceptual Dependency Theory will account for memory for gist of sentences. But it cannot be seriously proposed that this Is all that is needed for gist of long and complex stories. Some other explanation must be given.

In a recent experiment, Abelson [1975b] showed that people remember stories better when they are asked to take some particular point of view (of one of the participants or of an observer in a particular place), and that what they remember is contingent on which point of view they had. The ramifications of this experiment for a theory of language understanding have to be that when people have a clue of what to forget they do better at remembering. In other words, good forgetting is the key to remembering. Likewise, If we want to build programs that remember, we had best teach them how to forget. One method of forgetting is simply not noticing levels of detail that are there. This can be done by treating the instruments for an action at a different level than the main ACTs that they explain. When looking at a story at one level of detail we would not see the level of detail underneath it unless specifically called upon to do so.

For example, consider the sentence, "John went to New York by bus." We have previously represented this sentence by a simple ACT (PTRANS), and an instrumental act (PROPEL). But it must be realized that as with any other script, questions could be answered about this sentence that were not specifically in it. Subjects all seem to agree that the answer to "Was there a bus driver?" is "Yes" and to "Did John pay money to get on the bus?" is "Probably." This seems to Indicate that the instrument of John's PTRANS is, in actuality, the entire bus script.

Should we, as understanders, go so far as to place the entire bus script in what we obtain from understanding the above sentence? The answer seems obvious. You don't want to do all that unless you need to, but you want to have quick access to It in case you need to.

Consider the following story:

> John wanted some cheesecake. He decided to go to New York. He went to New York by bus. On the bus he met a nice old lady who he talked to about the prices in the supermarket. When he left the bus he thanked the driver for the ride and found the subway to go to Lindy's. On the subway he was reading the ads when suddenly he was robbed. He wasn't hurt though and he got off the train and entered Lindy's and had his cheesecake. When the check came, he said he couldn't pay and was told he would have to wash dishes. Later he went back to New Haven.

Ideally, our representation of this story should account for the fact that hearers of this story invariably forget the sentences about talking to the old lady and the bus driver, but alwayB remember the mugging, its consequence of dishwashing, and the main goal of going to New York to eat cheesecake.

We propose to represent stories therefore in the following way: There will be a causal chain connecting the main events of the story. (Here the PTRANS to the restaurant, the INGEST, and the PTRANS back home.) Underneath each of these main events will be the instrumental script that underlies each of them. (The bus script, the subway script, and the restaurant script.) These scripts will be "forgotten" to be reconstructed later, with the exception that any event that occurred within them that was not predicated by them will be placed on a "weird list" to be specially remembered.

The final representation of a story will consist of the events connected directly to the goals and plans to realize those goals made by the participants. These goals will be tied to the events that actually occurred and to the weird events and their consequences. Thus four lists replace our original (and growing) causal chain. An event list (with script events left out); a goal list; a plan list; and a weird list. What these lists do is help us forget. And of course forgetting helps us remember.

There are two ways in which this occurs: by omission and by prototyping. Events which enter none of the four lists (such as the conversation with the old lady), are dropped entirely. (More precisely, they are retained only until the constructed final representation is transferred from working memory to long-term memory. Anything not in this final representation is lost.) Also, the event list and plan list are condensed by using pointers to prototypes. The details are thus "normalized" (Bartlett, 1932); what is remembered is that a normal plan for satisfying such-and-such goal was used, including normal enactments of appropriate scripts. The function of the weird list is to mark the interesting departures from these normalities.

What we are saying then is that one of the major issues in Artificial Intelligence research much be the creation of the theory of forgetting. It simply is not possible to assume that people do, or that machines should, remember everything they encounter. in listening to a speaker, reading a book, or engaging in a conversation, people could not possibly remember everything they are told verbatim. In attempting to get the gist of a sequence, they must employ what we call forgetting heuristics. As part of these forgetting heuristics, are heuristics that search out items of major importance. The selection of these major items is the key to forgetting. We don't really wish to assert that people couldn't possibly remember everything they hear. Rather we wish to find a procedure that will let us see only the major items, yet also find, with some difficulty, the thoughts or statements that underlie them, and the ideas that underlie those, and so on.

Thus, the key to understanding must be, in order to facilitate search among what has been understood, an organization of the new information, in such a fashion as to seem to forget the unimportant material and to highlight the important material. Forgetting heuristics must do this for us. So the first task before us is to establish what the most significant items in a text are likely to be, and then to establish the heuristics which will extract and remember exactly those items.

References

Abelson, R.P. (1973). The structure of belief systems. In R.C. Schank and K.M. Colby (eds.). Computer models of thought and language. San Francisco: Freeman.

Abelson, R.P. (1975a). Concepts for representing mundane reality in plans. In D. Bobrow and A. Collins (eds.), Representation and understanding: Studies in cognitive science. New York: Academic.

Abelson, R.P. (1975b). Does a Story Understander need a point of view? In R.C. Schank and B. Nash-Webber (eds.), Using Knowledge to Understand, in Proceedings of the Conference on Theoretical Issues in Natural Language Processing.

Bartlett, F. (1932). Remembering. Oxford University Press.

Bobrow, D. (1975). Dimensions of representation. In D. Bobrow and A. Collins (eds.). Representation and understanding: Studies in Cognitive Science. New York: Academic.

Charniak, E. (1972). Towards a model of children's story comprehension. AI TR-266, Mass. Institute of Technology, Cambridge, Mass.

Goldman (1975). Conceptual generation. In R. Schank, Ed., Conceptual Information Processing. North Holland Publishing, Amsterdam.

Minsky, M. (1974). Frame-systems. AI Memo. Mass. Institute of Technology, Cambridge, Mass.

Norman, D. (1972). Memory, knowledge, and the answering of questions. Center for Human Information Processing Memo CHIP-25. Univ. of California at San Diego.

Rieger, C. (1975). Conceptual memory. In R. Schank, ed., Conceptual Information Processing. North Holland Publishing, Amsterdam.

Riesbeck, C. (1975). Conceptual analysis. In R. Schank, ed., Conceptual Information Processing. North Holland Publishing, Amsterdam.

Schank, R. (1973a). Identification of conceptualizations underlying natural language. In Schank and Colby (eds.). Computer Models of Thought and Language. W.H. Freeman Press.

Schank, R. (1973b). Causality and reasoning. Technical Report #1. Istituto per gli studl Semantici e Cognitivi. Castagnola, Switzerland.

Schank, R. (1975a). The Role of Memory In Language Processing. To appear in C. Cofer and R. Atkinson (ede.). The Nature of Human Memory. W.H. Freeman Press.

schank, R. (1975b). The structure of episodes in memory. In D. Bobrow and T. Collins (eds.), Representation and understanding: Studies in cognitive science. New York: Academic.

Schank et al. (1973). R. Schank, N. Goldman, C. Rieger, and C. Riesbeck. MARGIE: Memory Analysis Response Generation and Inference on English. Proceedings of the 3IJCAI.

Schank, R. (197. , Using Knowledge to Understand. Tn R. Schank and B. Nash-Webber (eds.), Proceedings of the Conference on Theoretical Issues in Latural Language Processing.

Winograd, T. (1972). Understanding Natural Language. Academic Press.

Winograd, T. (1975). Frame Representations and the Declarative/Procedural Controversy. In D. Bobrow and A. Collins (eds.), Representation and Understanding: Studies in Cognitive Science. New York: Academic.