12

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

(12)

This work was supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored under the Office of Naval Research under contract N00014-75-C-1111.

Question Answering in a Story Understanding System

Wendy Lehnert

Research Report #57

December 1975

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: Research Yale Computer Science Report #57
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): Question Answering in a Story Understanding System
5. TYPE OF REPORT & PERIOD COVERED: Technical Report
6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Wendy Lehnert
8. CONTRACT OR GRANT NUMBER(s): N00014-75-C-1111
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Department of Computer Science, Yale University, 10 Hillhouse Avenue, New Haven, Connecticut 06520
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: RR-57
11. CONTROLLING OFFICE NAME AND ADDRESS: Advanced Research Projects Agency, 1400 Wilson Boulevard, Arlington, Virginia 22209
12. REPORT DATE: December 1975
13. NUMBER OF PAGES: 56
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Office of Naval Research, Information Systems, Arlington, Virginia 22217
15. SECURITY CLASS. (of this report): Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is 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)

artificial intelligence
computer understanding
semantics
computational linguistics
question answering

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

Computational question answering is discussed as a demonstration of understanding for computer programs which read stories. A story understanding program is described which uses the question answering model proposed. Theoretical issues are presented and techniques for expanding the question answering model are suggested.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE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)

Table of Contents

1. Introduction

This paper is concerned with a computational model for question answering. The ideas proposed here will be discussed on two levels:

(1) as a theory of human question answering, and

(2) in terms of implementing computerized question answering.

The problems addressed here are those encountered in the specific task of story understanding where question answering is used as a demonstration of comprehension.

> John went to New York by bus. On the bus he talked to an old lady. When he left the bus, he thanked the driver. He took the subway to Leone's. On the subway his pocket was picked. He got off the train and entered Leone's. He had some lasagna. When the check came, he discovered he couldn't pay. The management told him he would have to wash dishes. When he left, he caught a bus to New Haven.
>
> Q1: Why did John go to New York?
> A1: Because John wanted to go to Leone's.
>
> Q2: How did John get to Leone's?
> A2: John took a bus to New York and then he took the subway to Leone's.
>
> Q3: Did anything unusual happen on the bus?
> A3: I don't remember anything.
>
> Q4: Did anything unusual happen on the subway?
> A4: A thief picked John's pocket.
>
> Q5: What did John order?
> A5: John ordered lasagna.
>
> Q6: Did anything unusual happen at the restaurant?
> A6: John discovered that he couldn't pay the check and so he had to wash dishes.
>
> Q7: Why couldn't John pay the check?
> A7: Because John did not have any money.
>
> Q8: Why didn't John have any money?
> A8: Because a thief picked John's pocket.
>
> Q9: How did John get back to New Haven?
> A9: John went from Leone's by subway and then he went to New Haven by bus.

Q10: How was John able to take the bus to New Haven?
A10: Probably John had a ticket.

The above story and ensuing dialogue are actual input and output from a computer program called SAM. SAM is designed to input English stories about a particular set of subject domains, and generate English responses to questions about what it read. SAM can also produce paraphrases of the input stories in either English or Chinese, but the problem of paraphrase will not be pursued here.

The ultimate test of any story understanding system is whether or not it can answer questions. While the capacity to paraphrase and summarize are desirable goals, such abilities are not the best demonstration of understanding. For any story, reasonable paraphrases and summaries can be made which do not exhibit any of the inferences necessary to real understanding. "John went to a restaurant, but on the way he was pickpocketed, and so he ended up washing dishes." This is a good summary for the Leone's story, but it relies on the reader's ability to make inferences and draw connections. The causality between having your pocket picked and washing dishes in a restaurant requires roughly five inferences:

1) Having your pocket picked often results in having no money.

2) People usually go to a restaurant to eat.

3) If you eat in a restaurant, you are expected to pay.

4) You need money in order to pay for things.

5) If you can't pay for a meal that you've eaten in a restaurant, the management may force you to wash dishes.

Any system which does not make these inferences has a dubious understanding

of the Leone's story. If a paraphrase program generates a single sentence summary, there is no way of knowing what inferences had been made by examining the output alone. Question answering provides a much clearer demonstration of comprehension whenever inferences must be made to supply missing information because questions can always be asked concerning specific inferences.

The problem of question answering is highly related to problems of memory representation and organization. It is therefore impossible to discuss question answering per se without reference to some model of memory. The next section describes briefly the fundamental memory models which underly the SAM system and which will be incorporated in further development of the current question answering model.

II. Scripts and plans

World knowledge and its organization in memory is a critical problem in natural language processing. Scripts and plans are theoretical devices which have been proposed as models of human memory organization. A vast amount of mundane world knowledge appears to be encoded in people in the form of scripts and plans. These same constructs are being exploited as a means of organizing world knowledge in a computer.

Scripts are memory units which contain information about situations or activities frequently encountered. Scripts describe the expectations involved in extremely mundane situations such as going to a restaurant, shopping in a grocery store, or stopping at a gas station. People acquire scripts through experience and use them both operationally (as in actually going to a restaurant) and cognitively (as in understanding stories about restaurants). When you go to a restaurant, you have certain expectations about finding a table, ordering, being served, eating, getting a check, paying the check, etc. These are so ingrained that you probably don't have to spend much conscious processing time on them. Most likely you only think about them when they fail or deviate from your expectations. If you hear that John went to a restaurant and ordered a hamburger, you will infer that he ate a hamburger unless you hear something to the contrary. You weren't told that he ate a hamburger; you used your scriptal knowledge of restaurants to make the inference. While scriptal knowledge must vary from person to person according to variations of experience, there are quite a few standard scripts which will be held in common as a cultural norm. Most people will have the same restaurant script since restaurants are highly standardized.

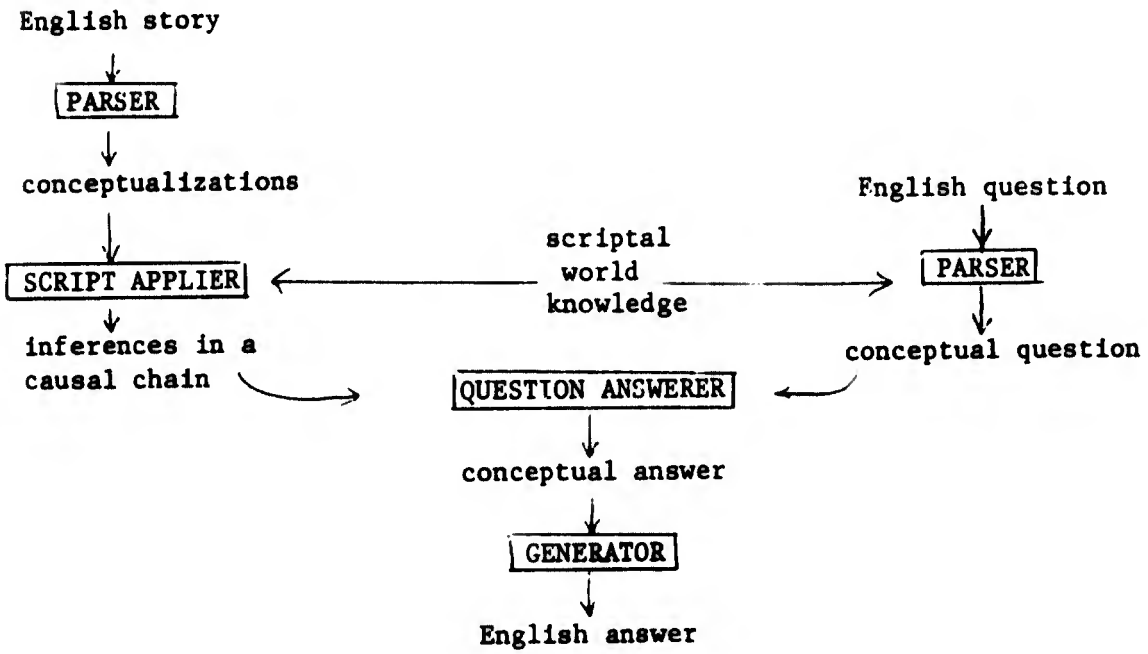
Planning structures are used when scripts do not apply or fail to contain sufficient information. While scripts are tightly bound to well specified situations, the same plan can be invoked in a variety of settings. For example, suppose you are trying to find a friend's house in San Francisco and you have the address but you've never been in San Francisco before. There is clearly no script for this but you nevertheless know what sorts of things to do. You might invoke a plan which says to wander randomly until you hit the right street, but a better plan would entail knowledge acquisition. You need to find out where the street is. So you consult appropriate knowledge sources. If you have a map you look at it. If you don't have a map you might go about finding one, or you may opt for another knowledge source and try asking people if they can tell you. If you've asked ten people to no avail you might give up on finding it yourself and call your friend so he can tell you where he is or perhaps come and rescue you. The principles involved in this process are very general by nature. The same planning structures could be used for finding a particular office in the Pentagon, or finding a book in a library (modulo the possibility of being rescued by the book). Plans are extremely general procedures which are adaptable to a number of situations and are used when there is no standard routine to follow.

Plans and scripts are related in that plans may give birth to scripts. If I invoke the same plans for getting stores to cash my checks, and these plans are always successful, I will have a script after a while. Should my script fail at some time, I will have to revert back into planning mode. But as long as the Park Avenue address and the AMA membership card work, I

will try them first. What originated as an inform/reason plan, evolved into a script due to repeated successes. For a more complete discussion of scripts and plans, see ([1], [2], [4], [8]).

III. An overview of question answering

The SAM system may be thought of in terms of four distinct phases or modules. The parser translates English stories or questions into Conceptual Dependency representations. Conceptual Dependency is a language independent meaning representation which decomposes actions into a small set of primitive acts [6]. When the parser has translated a story, the script applier is then called to generate inferences. This is where scriptal world knowledge is used to understand the story. The script applier structures its inferences into a causal chain which can then be used by the question answerer. Answers generated by the question answerer are in Conceptual Dependency and must be given to the generator for translation into English. For a more detailed account of the entire SAM system see [9]. While scriptal world knowledge is used primarily by the script applier, there are times when the parser, question answerer, and generator all need to access scriptal knowledge directly in the course of their processing.



The question answering process itself may be viewed in two parts. The interpretive phase takes a question in Conceptual Dependency and categorizes it in terms of particular question types. Once the question type is identified, the response phase executes appropriate searches of memory for an answer. These memory searches vary somewhat for different question types. For the most part, the question answering process consists of interpretation followed by response. But there are questions where the response phase calls the interpretor for more information before an answer can be found. This happens when the focus or emphasis of a question is critical.

In addition to interpretation and response phases, the overall question answering process is influenced by an intentionality factor. It may be desirable to have a system which returns minimally correct answers, thereby leaving the questioner the option of asking for more information if needed. On the other hand, it may be preferable to have a system which returns elaborate answers intended to communicate as much relevant information as possible. Such variations in the mode of question answering can be described as shifts in intentionality.

While intention, interpretation, and response provide an intuitively natural decomposition of question answering, it is important to realize that any such division is artificial. The categorization suggested here is neither valid from the perspective of human cognition nor is it particularly useful from a programming point of view. It has been made solely to facilitate discussion, in much the same way that languages are often described in terms of syntax and semantics.

IV. Intentionality

When people engage in question answering dialogues, they may answer questions in a number of different ways. Answers to questions can be general or detailed, honest or misleading, sarcastic or straightforward, etc. While it is not clear why anyone would want a sarcastic or deceptive question answering system, it is desirable to control the level of detail with which questions are answered. Initially, such a consideration may appear to be quite sophisticated and not very pressing. But in fact, the issue of intentionality is encountered in something as fundamental as a yes or no question.

The interesting thing about yes or no questions is that they are hardly ever answered with a simple yes or no. In most civil conversation, people naturally elaborate their negative responses with an explanation or correction. People usually don't put others in the position of having to 'grill' them for information.

Q: Does Montana have a significant cattle industry?

A: No, there are a number of sheep ranches, but most of the land is too barren to support cattle.

is a slightly more natural conversation than

Q: Does Montana have a significant cattle industry?
A: No.

Q: Why not?
A: Most of the land is too barren to support cattle.

Q: Do they raise any livestock there?
A: Yes.

Q: What?
A: There are a number of sheep ranches.

So when writing a question answering program which will answer yes or no questions, one is immediately forced to decide whether or not elaborations on negative responses are desired.

Since the question answering system being developed here is intended to demonstrate the comprehension of a story understander, it is desirable that the answers communicate as much information as possible. This being the case, the current system is designed to incorporate a totally communicative intentionality. That is, the system endeavors to respond to questions with as much relevant information as possible, regardless of what was explicitly asked. In particular, negative answers to yes or no questions are always more than a simple 'no'.

So the intentionailty in a question answering system designed to demonstrate story understanding is fixed and maximal in terms of information to be conveyed. In another task, such as computer aided instruction, it may be useful to be able to alter the intentionality of the question answerer as more or less information is desired in the responses. The aspect of intentionality is important in a general question answering model and must be investigated further as question answering is developed in a variety of tasks.

V. Interpretation

A. Question Types

The first phase of interpretive processing determines the general question type. This preliminary analysis is based on procedural considerations as opposed to metaphysical leanings. When faced with the problem of categorizing all possible questions which might be asked about a story, there are many possible ways to slice the analysis. Since this approach to the problem is grounded by a desire to write a computer program which will execute the analysis, a question is categorized according to the various procedures which will be called to answer the question. The typing of questions described here is not intended to be a complete or definitive system. It is merely a first pass at the problem. There are five basic question types which have been implemented in SAM:

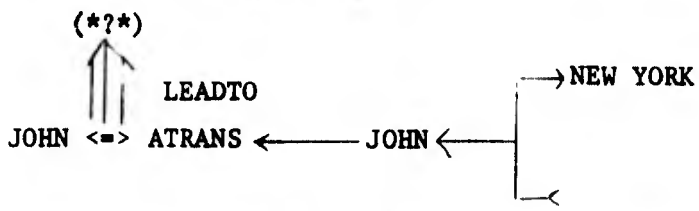
1) why questions
2) how questions
3) yes or no questions
4) occurrence questions
5) component questions

This is a fairly intuitive categorization. The first three types are self explanatory. An occurrence question asks about some general act or sequence of acts, as in questions which begin with "What happened when..." or "Did anything unusual happen when..." To a large extent component questions are essentially fill-in-the-blank type questions. "Who hurt John?" or "What did John eat?" etc. They refer to a question which describes a conceptualization which has an uninstantiated component.

This categorization is based on the types of conceptual representations which underlie English questions. If the category names are taken too literally, confusion will arise. Why isn't a did-anything-unusual-happen question a yes or no type instead of an occurrence type? Why shouldn't "how long did it take?" be typed as a how question? To avoid this confusion, questions must be analyzed purely on the basis of their conceptual representations, not on the level of English wording. Each question type has a distinctive form of Conceptual Dependency representation.

1. WHY QUESTIONS

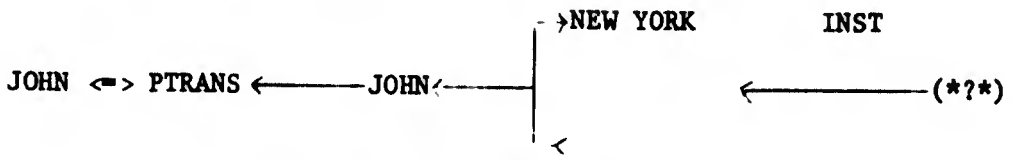
Why did John go to New York?



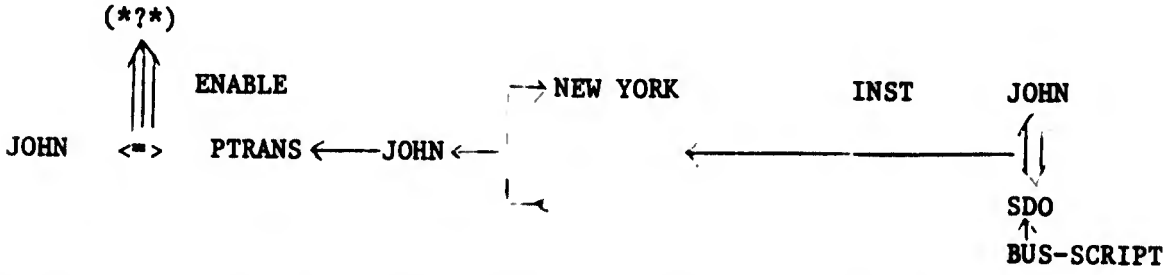
This question is conceptually equivalent to asking what caused John to go to New York. Why questions always embody some sense of causality. A why question can be answered in terms of motivation, goal-orientation, physical causation, or any number of causal variants. This general situation is represented by a highly flexible causal link, LEADTO. A why question is conceptually a causal chain: there are two complete conceptualizations linked by a very non-specific causality. The leading conceptualization is unknown and the consequent conceptualization is specified.

2. HOW QUESTIONS

How did John get to New York?

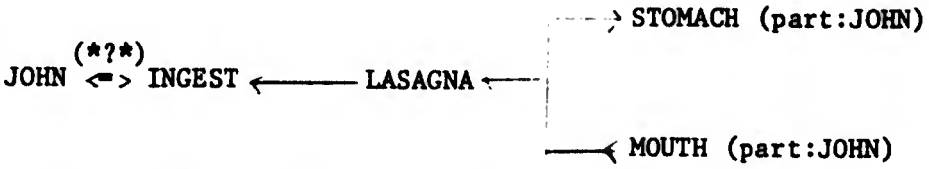


How could John take the bus to New York?



How questions consider the act in question in terms of either instrumentality or enabling conditions. In general, how questions are identified by having either an unknown filler for the role INST, or by having an unknown conceptualization as the causal antecedent of an ENABLE link.

3. YES OR NO QUESTIONS

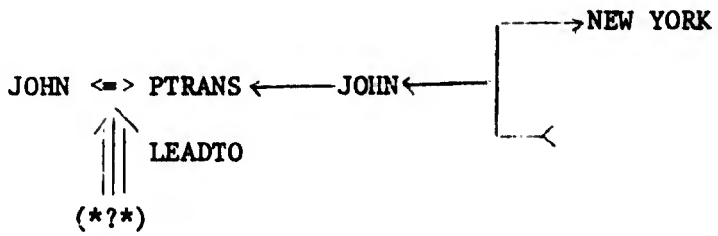


Yes or no questions seek to establish the truth of a statement. In Conceptual Dependency, the modifier MODE is given the value NEG to indicate when the concept did not take place. The default interpretation

of an undefined MODE is that the concept did take place. (This saves us from having to define MODE as POS whenever the corresponding concept is an actuality.) So the obvious representation of yes or no questions is to assign to MODE the value *?* to indicate that the validity of the concept is questioned.

4. OCCURRENCE QUESTIONS

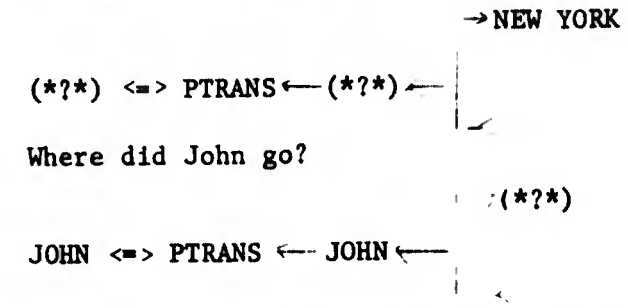
What happened when John went to New York?



This is conceptually equivalent to asking what was caused by John's going to New York. Similar to why questions, the general representation is a causal chain of two conceptualizations joined by the loose connective LEADTO. This time the consequence of the causality is unknown. Some occurrence questions are very script bound in meaning and must resort to a representation in terms of scripts. For example, "Did anything unusual happen on the bus?" asks for an occurrence which is considered to be unusual in the context of a standard bus script. 'Unusual' here must be represented with respect to the bus script. Conceptual representations with scriptal references are now being developed.

5. COMPONENT QUESTIONS

Who went to New York?



Component questions occur when one atomic component of a conceptualization is unknown. The atomic restriction differentiates component questions from the other question types where a complete conceptualization is unknown within a larger conceptualization (as in why questions).

B. Question Statements and Focus

The second phase of interpretive processing determines the question statement. A question statement is the conceptualization underlying the original question. Extraction of the question statement is initially straightforward. A common exercise in elementary grammar is to take a question and make a statement out of it. Does John go to New York? ... John goes to New York. This is essentially what a question statement is. By removing the interrogative aspect of the question, the remaining conceptualization conveys some statement of fact. Getting a question statement is almost this easy, but not quite.

Q1: Why did John fly to Siberia?

Q2: Why did John crawl across the street?

In Q1 interest is focused on the destination. (Why Siberia of all places?) In Q2 interest is focused on the act of crawling. (Why not walk like most

people?) General world knowledge about the remoteness of Siberia and usual methods of crossing streets is needed to establish the focus of these questions. If we answered A1 and A2 with ...

A1: Because it was too far to walk.

A2: Because he wanted to get to the other side.

one might say that we missed the point of the questions. Somehow the interpretive processing must be able to identify the most interesting aspect of a question statement. We must know where to focus attention. Correct identification of focus is tantamount to understanding the intent of a question. Consider the question:

Did John give Mary the book?

It is not difficult to imagine different story contexts where each of the following answers might be the most appropriate response.

A1: No, Dave gave Mary the book.

A2: No, John sold Mary the book.

A3: No, John gave Rita the book.

A4: No, John gave Mary the painting.

A5: No, John will give it to her tomorrow.

These answers differ by shifts of interpretive focus:

F1: actor (John)

F2: action (giving)

F3: recipient (Mary)

F4: object (book)

F5: time of action (did)

So understanding the point of a question often requires consideration

of the question context as well as the use of general world knowledge. It is also the case that in spoken language, the focus of a question may be signalled by stress intonations.

Although establishing focus is an interpretive problem, it is not part of the initial interpretive phase. Focus is something we don't want to deal with unless we have to. Suppose the answer to 'Did John give Mary the book?' is 'yes'. Then the response phase will find the conceptualization in memory and return yes. It is not necessary to identify a focus in this case. It is only when the response phase cannot find John giving Mary the book, that focus becomes necessary. In this event, the initial response is 'no', and the response phase should go on to search memory for a correction. The correcting conceptualization will be very close to John giving Mary the book, but it will differ by an actor, action, recipient, object, or time factor. If the interpreter can establish a focus, the memory search will not have to check for all possible variants, but will expect to find a particular type of variant: perhaps one where only the actor slot will differ. So we rely on the response phase to recognize problems in focus and to call the interpreter to resolve them. Thus far, one script-based technique has been implemented to establish focus. This mechanism will be outlined in Section VI.

VI. Response

A. The Static/Dynamic Distinction

There are two procedural classifications of memory retrieval. We will label them static and dynamic responses. The static response refers to locating and returning relevant information from the memory representation which was generated at the time the text was read. A dynamic response occurs when the relevant information is not contained in the original memory representation but must be actively reasoned from general world knowledge and inferencing in conjunction with the original representation. A static response relies on information which must have been embedded in the memory representation at the time the story was read. Failure to do so would indicate a failure to understand the story. A dynamic response, in contrast, deals with details and ramifications which can be ignored without loss of comprehension. To illustrate the difference, consider the following story:

> John took a train trip to Miami. He got on the train at Penn Station and took a nap until Philadelphia. When he woke up it was time for dinner so he went into the dining car. Over dinner he met someone from his hometown and they stayed up half the night talking. The train was delayed the next day in Georgia but John didn't care if he arrived later than expected. He enjoyed the trip down and decided to take the train back as well.

In this story, the question "How did John get to Miami?" requires a static response, but "How long did the trip take?" requires dynamic processing. If you did not know, upon reading the story, that John was

traveling to Miami by train, then one would seriously question your understanding of the story. On the other hand, it does not seem necessary to know the duration of the trip in order to feel that the story makes sense.

An answer to the second question can be gotten by looking back over the memory representation for clues concerning time elapsed. You would probably think for a moment, determine that only one overnight was described, and taking into account your personal knowledge about the overall speed of long distance trains as well as knowledge of the distance between New York and Miami, you would probably answer "About a day or so." Since nothing in the story indicated an unusual or unexpected time factor, and no information about the time involved is needed in order to understand the story, you would not generate a time interval in your memory representation. But you can reason a likely answer on the basis of 'clues' and general world knowledge.

While it is evident that such distinctions must occur in human processing, it is not so easy to know where the line is drawn. The division between static and dynamic is only as sharp as our subjective sense of what it means to 'understand' a story. Very fuzzy indeed. Until a more precise notion of 'story understanding' can be described, I will rely soley on intuition when a line must be drawn, and hope there is general agreement over such decisions.

B. Memory Representation

Procedural response techniques are dependent upon the overall memory structures used to represent the story. Three levels of memory representation have been developed to some detail: causal chains, scriptal structures, and planning mechanisms ([2], [4], [7]). The response techniques for memory retrieval must know what to look for at each level and how to find it. This paper describes the script-based and causal chain-based response techniques used by SAM. Techniques for retrieval from planning mechanisms are being developed but have not evolved enough to warrent discussion.

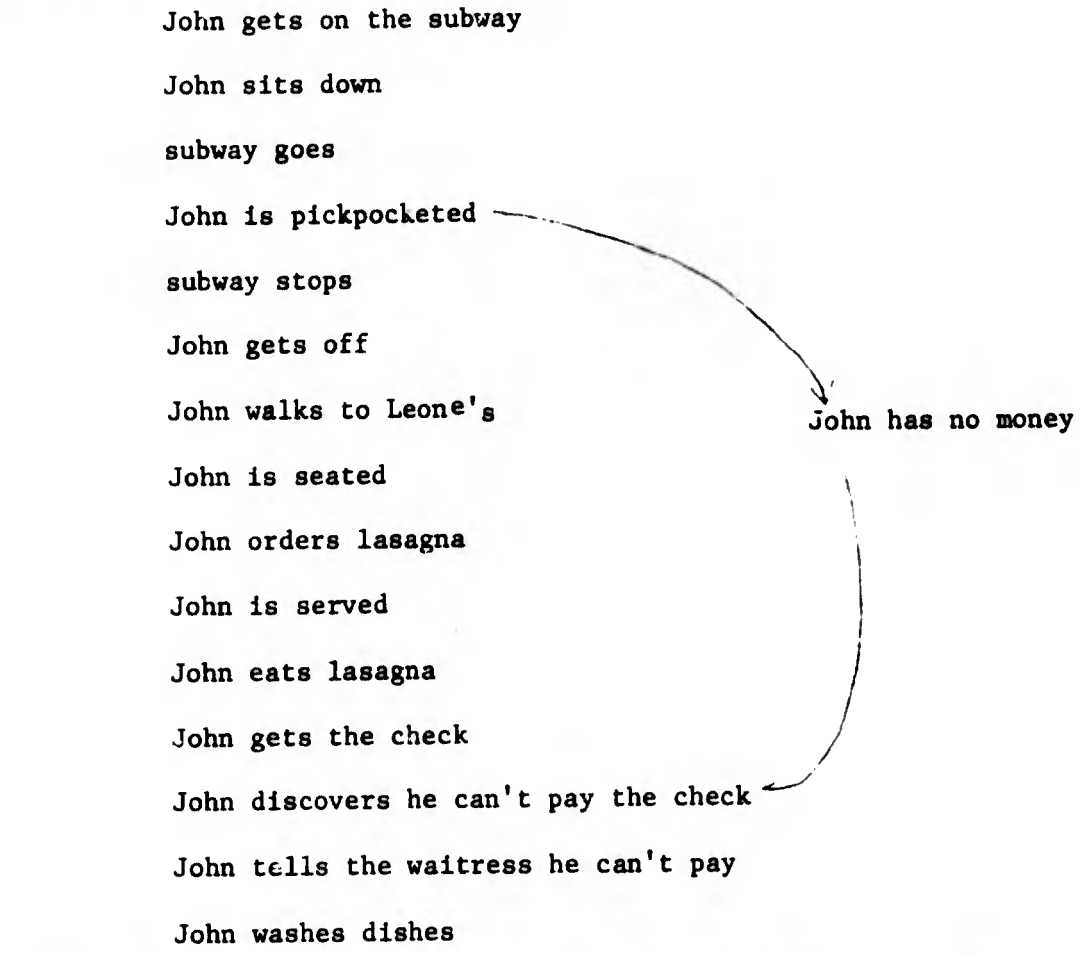
When a story representation is purely scriptal, two levels of memory representation are generated: a causal chain and a scriptal structure.

1. CAUSAL CHAINS

A causal chain is a low level representation in which individual conceptualizations are linked by causal relationships. The main path generated to connect input conceptualizations is called the primary path. Conceptualizations in the primary path are either input conceptualizations from the story, or scriptal inferences made by the script applier [2]. When an input concept does not relate to the primary chain by script-defined causalities, a general inference mechanism is called to connect the current concept with some previous concept in the memory representation by short circuiting the primary path. To illustrate this, consider the Leone's story again:

> John went to New York by bus. On the bus he talked to an old lady. When he left the bus, he thanked the driver. He took the subway to Leone's. On the subway his pocket was picked. He got off the train and entered Leone's. He had some lasagna. When the check came, he discovered he couldn't pay. The management told him he would have to wash dishes. When he left, he caught a bus to New Haven.

Part of the causal chain memory representation for this story looks something like this: (in actuality the primary path would contain many more inferences)

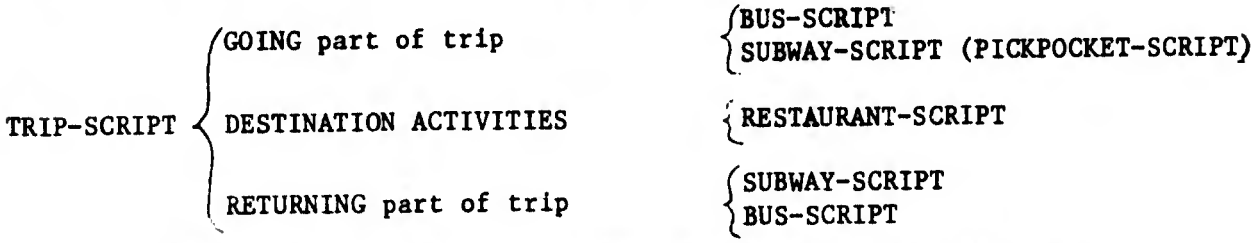


The primary path traces the chronological development of the story. When we reach the point where John discovers he can't pay, an explanation must be found since this is not an expected occurrence in the restaurant script. The only way to account for such a turn of events is by generating a branch of inferences off of the primary path [2]. In this case the branch is very simple, consisting of one inference which ties together

the pickpocketing with not being able to pay. For a detailed account of causal chains, see [7].

2. SCRIPTAL STRUCTURES

The other aspect of memory representation is a higher level structure which comes into play when a story contains more than one script. This script structure representation describes how the various scripts are related to each other. For example, a general trip script will embed various travel scripts which may themselves be nested or sequential, as well as scriptal descriptions of the destination activities. Scripts are generally related sequentially or by nesting, and some scripts (like the trip script) predict scriptal relations within themselves. The scriptal structure for the Leone's story looks like:



The trip script is divided into three components: going, destination activities, and returning. In the Leone's story, going becomes a sequence of two script activities: John went to New York by bus, and John took a subway to Leone's. The destination activity is eating at Leone's, and the returning activities are taking the subway from Leone's (by inference) and taking the bus back to New Haven. Within the first instantiation of the subway script is nested an occurrence of the pickpocket script. Not shown is the capability to recognize unusual, unexpected, or surprising activities within the context of an active script. For example, the

pickpocking is tagged as an unusual occurrence in the context of subways. As progress is made in processing longer stories, a theory of what things get forgotten will have to be developed. Recognition of strange occurrences will clearly be critical to the success of any theory of forgetting.

C. Retrieval from Scriptal Structures

Q1: How did John get to Leone's?
A1: John went to New York by bus and then he went to Leone's by subway.

Q2: Did anything unusual happen to John on the subway?
A2: A thief picked John's pocket.

In actual memory searches, the structural representation is searched before the causal chain representation. This is out of programming considerations more than anything else, since the higher level representation is smaller and therefore requires less time to search. However, it does seem appealing to work from a general level down to a more specific level on intuitive grounds. Intuitively, it seems that one should first be aware of a general act (like John going to New York) before considering all the details involved (like buying a bus ticket, finding a seat on the bus, etc.) I suspect that people try to work from summaries whenever possible, descending to varying levels of detail only as needed.

The specific search techniques vary depending on the question type and the top level script organization. The response program first identifies the highest scriptal structure of the story. In the Leone's story a trip script is embedded at the highest level. Once this top level script is identified, a procedure specific to that script is called to identify and

extract relevant data. If nothing is found on the highest level, procedures specific to the next level of scriptal structure are called, and so forth.

To be more concrete, Q1 enters the trip response program as a question type (how) and a question statement (John went to Leone's). A procedure is called for searching the top level script structure which in this case is a trip-script function. This function behaves differently for each question type. Given a how question, it first accesses the script summaries for all scripts found under the going, destination, and returning parts of the trip script structure. These script summaries are checked against the question statement for a match. When it finds John going to Leone's by subway, a match is made, and the process is programmed to return all the going activities up to and including the matched one. A concatenation function provides conjunctions so the generator can tack the English translations together in a single sentence. So the final response appears as "John went to New York by bus and then John went to Leone's by subway."

Q2 enters the trip response program the same way. This time the question type is occurrence and the question statement is a pointer to instances of the subway script. For this question type, it is not necessary to examine the script summaries of the going, destination, and returning activities as before. This time it is only necessary to locate instantiations of the subway script and examine these. A search of the entire script structure is made, returning pointers to each instance of a subway script (no restrictions were made about which subway

ride). Once these are collected, it is easy to determine whether or not things occurred while the script was active which were not part of the script and which had some degree of interest. (The script applier creates a record of such things as a part of its processing). We find one such occurrence in the first instantiation of the subway script, and return the scriptal summary of the strange occurrence: A thief picked John's pocket.

The important thing to realize about retrieval from scriptal structures, is that the techniques are determined by both the question type and the particular overall structure. When a response is not completed on the top level of the structure, procedures corresponding to the next level are activated, etc. So retrieval functions are designed for each scriptal structure and as such, serve to incorporate information about those structures which is not present in the static memory representation. There is nothing in the static data base of the trip script which identifies how John got to Leone's per se. This knowledge is pieced together at the time the question is asked.

Of course one can always argue about what form data must manifest in order to say that a particular piece of information is present. The static memory representation knows that the going part of the trip consisted of John taking a bus to New York followed by John taking the subway to Leone's. Since the act of getting somewhere is usually construed to entail all of the travel necessary (taking a bus and then taking the subway) as opposed to just the final segment (taking the subway), the response procedure is programmed to pick up all of the going activities. So the static/dynamic

distinction is fuzzy from the computer's point of view as well as the human's.

D. Retrieval from Causal Chains

If no response is generated from the processing of scriptal structures, the program resorts to retrieval off of the causal chain representation. To illustrate the retrieval techniques, we will consider a specific story and outline the procedures used to answer some sample questions. The following story and the questions are input to SAM. The answers discussed are answers SAM returns and the procedures described are those used by SAM.

> John went to a restaurant and the hostess gave him a menu. When he ordered a hotdog the waitress said that they didn't have any. So John ordered a hamburger instead. But when the hamburger came, it was so burnt that John left.

The causal chain representation looks something like this: (modulo a lot of states and acts)

John enters restaurant

John is seated

John gets a menu from the hostess

John orders a hotdog

(I1) The waitress tells John they don't have any hotdogs

(R1) John orders a hamburger

Waitress serves John the hamburger

(I2) The hamburger is burnt

John gets angry

(R2) John leaves the restaurant

1. WHY QUESTIONS

Q1: Why did John order a hamburger?
A1: Because the waitress told John they did not have any hotdogs.

Q2: Why was John angry?
A2: Because the hamburger was burnt.

Q3: Why didn't John eat a hotdog?
A3: Because the waitress told him they did not have any hotdogs.

When the script applier instantiates a path through the restaurant script, it flags certain occurrences as being mildly interesting. These are actions or situations which are occasionally encountered in the course of a script but are not strongly expected. These occurrences generally come in interference/resolution pairs. Since they are semi-standard situations, they have predictable consequences and are identified in causal templates, e.g., being informed that the kitchen cannot prepare what you want is semi-standard in the restaurant script. The standard resolution to this is to order something else or to leave. Since there is a fair degree of predictability involved, the script applier can recognize interference/resolution pairs when they occur and it flags them accordingly. Note that script interferences always signal predictable resolutions. It is not clear what one might do if they ran into Richard Nixon at a restaurant (ask for an autograph? leave in disgust?) so this is not a script interference recognized by the restaurant script. There is no predictable resolution for Nixon. All the script applier could do in that case would be to flag the meeting as an unusual occurrence, and make no predictions about the outcome. In the sample story we have two

interference/resolution pairs. Being told that there are no hotdogs is resolved by ordering a hamburger. Seeing that the hamburger is burnt is resolved by leaving.

Resolution search. When the response program gets a why-question type, it first examines the interference resolution pairs to see if the question statement matches any of the resolutions. If it finds one, the corresponding interference will explain the resolution. Thus we answer Q1:

Why did John order a hamburger?
Because the waitress told him they didn't have any hotdogs.

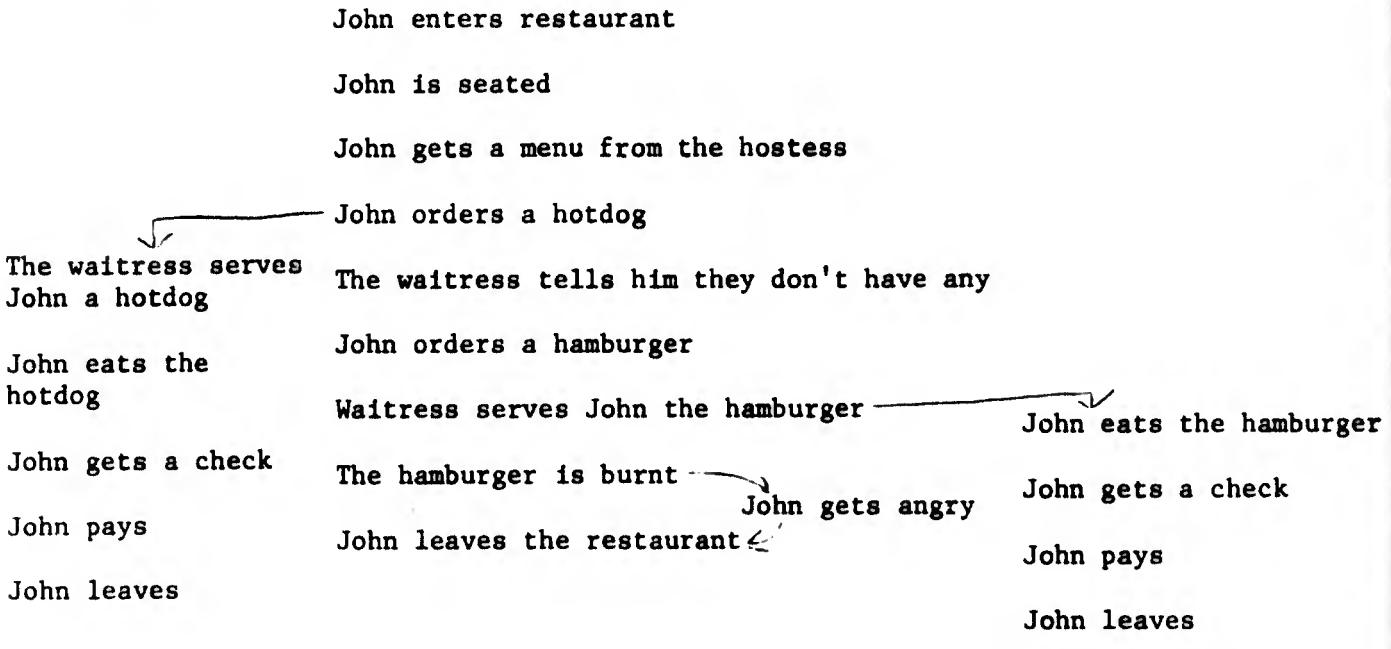
Inference search. But suppose a why question does not correspond to a resolution concept. We next look to see if the question statement is an inference branching off the primary path. If so, we follow the causality back to the preceding concept. This is what happens in Q2:

Why was John angry?
Because the hamburger was burnt.

Dynamic processing. Now suppose the question statement is not found among the resolutions or the inferences. Then we check to see if the question statement's MODE modifier evaluates to NEG. If the question statement is negative, then the act did not occur and will not be found in the memory representation. Yet the question presumably makes sense. The only way it can make sense to ask about something which didn't happen is if there was somehow a possibility (or probability) that it would have happened had things not taken the turn they did. That is, the act in question might have occurred had something not interferred.

The script applier can easily generate paths through the script which were not instantiated (they did not occur) but which might have happened

had something else not intervened. To see these ghost paths, we simply point to the places in the causal chain where interferences or unusual things occurred, and we call the script applier to generate default paths through the script from the points just prior to these non-standard occurrences. For example, a ghost path is generated from the concept prior to the waitress telling John that they have no hotdogs. This ghost path completes a path through the restaurant script which makes its predictions as though it never saw any more input after John ordered a hotdog. Similarly, when we see that the hamburger is burnt, a ghost path is generated from the preceding concept where the waitress serves John the hamburger.



Now we examine these ghost paths for our negative question statement. When the corresponding (non-negated) concept is found, we follow the ghost path back up to its branch off the primary path, and return the

interference or unusual occurrence right after the branch. Hence, Q3 is answered:

> Why didn't John eat a hotdog?
> Because the waitress told John they did not have any hotdogs.

In the event that a match can be made in more than one of the ghost paths, we trace the most recent (latest in the story) path back.

Generation and examination of ghost paths represents a dynamic response technique. When people store a conceptual structure in memory, it seems likely that they store only the information which cannot be inferred from other data. In particular, knowing that a particular sequence of events occurred, enables one to infer that conflicting or mutually exclusive events did not occur. It is difficult to believe that people store facts like Paris is not in Germany. It's enough to know that Paris is in France. The same principle is being applied here to scriptal data. Given one instantiation of a situational script, it is not necessary to store in memory all other possible instantiations with a negation flag. It is enough to store the instantiation which occurred and to be able to recognize events which didn't occur by generating ghost paths when needed for a question.

Since the ghost path technique conveys information about standard expectations and where such expectations break down, it is much more powerful than an approach based on recognition of mutually exclusive events. Simply knowing that something could not have happened is weak understanding unless there are causal relationships explaining why it didn't or couldn't happen. And in fact there are situations which suggest that recognition of mutually exclusive events could never work. If you are told that John ate a hamburger and then asked if he ate a hotdog, it looks

like you could discern that one rules out the other. But what if John orders a hamburger, can't get it, and just leaves? Now if asked whether or not John ate a hamburger, there is no corresponding concept to set this against unless you wish to store non-events such as John not eating.

It is true that ghost paths do not carry information concerning everything that didn't happen. I would claim however that they do contain everything you need to know (or not know). There is an issue involved in the task of question answering concerned with what questions make sense with respect to a given story. Clearly there are always questions which would never be asked and which would fail to make sense if asked. For example, in this last story, it would not seem relevant to ask if John is blond or if John voted Republican in '68. An intelligent program should know when a question is coming off the wall. Acceptable questions deal with inferences and expectations triggered by the story input. A program which has access to all the inferences and expectations a story generates will be able to recognize and process reasonable questions. Ghost paths provide a record of failed expectations and thereby serve to tie relevant non-events back to the story line.

2. COMPONENT QUESTION

What did the waitress serve John?
The waitress served John a hamburger.

The processing for component questions is very simple. The question statement for a component question is a conceptualization with some atomic component unknown. Usually the answer will be found by a search of the

primary path. The question statement is matched against the causal chain concepts, with a wild card match against the unknown question component. If no match is made in the primary path, inferences must be searched. Occasionally, more than one concept must be picked up as would be the case if we asked what John ordered (a hotdog and then a hamburger). Complications with component questions have not been investigated since so many of them appear to succumb to the above procedure.

3. OCCURRENCE QUESTIONS

What happened when John ordered the hotdog?
The waitress told John they didn't have any hotdogs.

What happened when John ordered the hamburger?
The waitress gave the order to the cook.
The cook prepared the hamburger.
The cook gave the hamburger to the waitress.
The waitress served John the hamburger.

What-happened-when question are almost as straightforward as component questions. For one of these questions to be reasonable, the act referred to must have taken place, and so is present in either the primary path or in the inference branches. Once the act in question is located, we simply run down the chain collecting conceptualizations until we find one that was explicitly mentioned in the input. From this list we try to delete terribly uninteresting conceptualizations by eliminating states (these can always be inferred from the acts surrounding them). The answers to the above questions contain conceptualizations not in the causal chain outlined on page 26. This is only because the causal chain on page 26 is not the complete chain generated by SAM but only a partial chain for illustration

purposes.

Cutting off the chain at the nearest input conceptualization is a fairly arbitrary policy. It is worth pointing out that this response was designed primarily to demonstrate the inferences which SAM makes. As a model of human response it is very poor since people never mention things which can be inferred but tend to bring up only those things which are of interest or significant consequence. There is also evidence that after hearing a story, people tend to confuse their inferences with what they actually heard. So while it is very easy for a computer to keep track of what was explicitly stated, it is not a natural human capacity and we would not want our model dependent on such an ability.

4. YES OR NO QUESTIONS

Q1: Did John order a hotdog?
A1: Yes.

Q2: Did John get angry?
A2: Probably.

Q3: Did the waitress give John a menu?
A3: No, the hostess gave John a menu.

Q4: Did John eat a hotdog?
A4: No, the waitress told John they did not have any hotdogs.

Q5: Did John pay the check?
A5: No, John was angry because the hamburger was burnt and so he left.

Given a yes or no question, the first processing phase tries to match the question statement against concepts in the primary path. If this fails, it examines inference branches off the primary path. If a match is found in the primary path, a positive response is made and the answer yes is returned. If a match is found on an inference branch, a slightly weaker

response of probably is returned to signal the small degree of doubt associated with inferences off the primary path. This accounts for the answers to Q1 and Q2.

It is not difficult to come up with counterexamples where a positive response should occur for question statements not found in the primary path or on inference branches. These cases seem to require some manner of dynamic processing which has yet to be defined.

Yes or no questions were discussed to some extent under section IV. For the reasons cited there, SAM's processing elaborates negative responses. A negative response occurs whenever the question statement is not found in the primary path or on an inference branch. Then the question must deal with some expectation which did not come through. To understand how the processing procedes from here, we must first discuss the idea of script constants and script variables.

Script-based identification of focus. In every script there are a set of very strong expectations. When we hear that John went to a restaurant, we expect certain activities to have taken place. For example, we expect that he sat down, he got a menu, he ordered, he ate, and he must have paid the check. These acts are called script constants. If our expectations regarding a script constant are violated, we want to be able to account for the variation. So if we hear that John did not pay the check, we want to know why not. Some explanation is expected or sought.

Within each of the expected script constants, there is often room for a certain amount of variation. We know that John must have gotten a menu, but it is not clear where it comes from. He might get it from the waitress,

or the hostess, or it may be sitting on the table and he picks it up himself. The source of the menu is a script variable. Naturally what John orders and eats are script variables, as well as who brings him the check (it could be the waitress or it might be the hostess). Some script variables take default values in the absence of explicit information. For example, I would assume that the waitress brings the check unless I hear otherwise.

So suppose a question statement is not found in the primary path or inference paths of the causal chain representation. This is the case when we ask if the waitress gave John a menu. We then check the question statement against a list of script constants to find out how badly the question statement expectation has been violated. If the act in question is a pure script constant, containing no variables, and therefore having no possible variations, then we need to find out why the act did not take place. On the other hand, if the script constant embodies a script variable, then we check back against the primary path in the causal chain. This time we know not to try to match the variable component, but to settle for anything in the chain that agrees with the question statement modulo the script variable.

Using focus: Question statements with script variables. The processor takes the question statement of the waitress giving John a menu, and it finds the script constant of John getting a menu with a variable actor component. It then goes back to the causal chain and finds John getting a menu from the hostess (having left the actor component open to a wild card match). Having achieved this match, we answer Q3:

Did the waitress give John a menu?
No, the hostess gave John a menu.

By using the script constant/variable technique to answer such questions we have momentarily dropped back to an interpretive mode. What we are really doing by identifying script constants and variables is establishing the question focus. In answering Q3 as above we have actually identified the focus of the question to be on 'waitress'. Suppose the focus had been put on 'menu'. Then a reasonable answer would be "No, the waitress gave John his meal", or "No, the waitress gave John the check". The elaboration is made by answering a component question with an unknown actor. "Who gave John a menu?" Identification of focus enabled us to ask the most appropriate component question. By using the predictive powers of the restaurant script, we were able to identify the appropriate focus and thereby return the most natural response.

But suppose we can't find the question statement in the causal chain even knowing not to match the variable component. This will happen when we try to answer if John ate a hotdog. We look for John eating something but we can't find even that much. At this point a very strong expectation has been violated and we demand to know why John didn't eat anything. Once again ghost paths are generated and examined to account for failed expectations. We take the question statement and try to find it in a ghost path. When we find it we trace back to the branching point off the primary path and return the non-standard concept just following the branch. We are essentially turning the question into a why question and asking why something didn't happen (why didn't John eat a hotdog?). The processing is the same as

before for why questions with negative modes in the question statement. This is how Q4 is answered:

> Did John eat a hotdog?
> No, the waitress told him they didn't have any hotdogs.

Using focus: Question statement which are script constants. Finally we consider the case where the question statement corresponds to a script constant containing [illegible] variables. John paying the check is a pure script constant without variables. By not [illegible] this pure script constant in the causal chain representation, our expectations have failed, and we need to ask why. Why didn't John pay the check? As in the last case, we are back to a why question and the processing procedes as before. Generate ghost paths to find the interfering or unusual event that explains it all. When we ask why didn't John pay the check, we find John paying the check in both ghost paths. The processor however takes the first match it finds, working from the end of the story back to the beginning. The best answer here picks up not only the interference under the branch point, but mental state changes and the resolution as well. This slightly more involved response occurs only when the interference involved causes a mental state change. So Q5 is answered:

> Did John pay the check?
> No, John was angry because the hamburger was burnt and so he left.

Interpretation-response interface. In answering a yes or no question which takes a negative response, we have seen the first instance of interpretation/response interfacing. The processing first moves through interpretation to determine the question type and question statement. We

then enter the response phase. When the response processing has determined that the initial answer is NO, we return to interpretive analysis to find the focus of the question. Once we have a focus we can resume response processing to produce an elaboration beyond the initial response NO. So when the response phase needs more information it asks for further interpretation. In general, it's best to avoid a lost of interpretive processing unless the response phase indicates a need for it.

VII. Expanding the model

A. The Generation/Selection Paradigm

The problem of question answering is intimately connected to the problem of memory representations and organization. This should be apparent from the description of procedural response techniques. Thus far, story understanding programs have used script-based memory representation. A theory of memory representation beyond scripts is now being developed which uses planning structures and other constructs, but none of this has been incorporated in a computer program. Computer story understanding based on plans is now being researched and this constitutes the next step for question answering programs. While a precise description of plan-based memory representation is not quite here, it is still possible to investigate the general principles which question answering will utilize in conjunction with such representations.

In order to pursue problems specific to question answering which are not issues of memory representation, I am currently working on what I call a generation/selection model of question answering. In this model, each question generates a number of feasible answers and a selection procedure picks out the best one. This is a convenient model because it makes a clean cut between memory representation and question answering. The generation end of the model embodies all the problems of memory representation while the selection part of the model is pure question answering. I am not proposing that this is the way human cognitive processing works, although it is the case that people are very quick to defend their answers in the face of alternative responses. The generation/selection model merely serves to make issues of question answering

accessible while the memory representation problems are being researched.

Given a set of possible answers to a question, the selection process works primarily by characterizing the various answers. The characterizations used are designed to indicate the relative merits of each answer. What makes an answer good or bad? When is one answer preferred over another? To establish criteria for judging answers, we must consider the overall dynamics of a question answering situation.

The first premise in general question answering is that the questioner is asking a question because he does not know the answer. Of course there are variations on this, as when someone asks to confirm their own beliefs or to hear another opinion, but people answering questions cooperatively generally operate on the assumption that they are providing information which is being sought because it is unknown. Even in a test situation where the answerer knows he is merely demonstrating his own knowledge, he still must answer as though he is explaining something to a naive questioner. In his processing he simulates a situation where he must provide presumably unknown information.

As anyone experienced in the fine art of test-taking knows, knowing what can be assumed and what must be spelled out is critical to success. This same problem exists in general question answering. But unlike the test-taking situation, deciding what is a given and what is not given usually occurs at levels of thought processing which never enter the arena of consciousness. If someone asks me why computers are smart enough to play chess but can't understand simple English, I will immediately assume that this person has a minimal or nil acquaintance with the nature of AI

research, but at the same time must have some knowledge of AI achievements in order to have asked the question in the first place. These inferences about the questioner's knowledge state are made very quickly, and probably without my thinking about it consciously. People in the position of answering questions must assess what is being asked for; what does the questioner know, and what doesn't he know. How can his question be answered in a manner which will satisfy him? If a child asks me what a computer is I will not answer in terms of CPU's and memory and peripherals. If a college student asks the same question I might. Sometimes we are incapable of answering questions effectively because we do not know how to assess or address certain knowledge states. This is why brilliant people can be such poor teachers. Only rarely are we consciously focused on the inferences we make about a questioner's knowledge state.

In question answering based on story understanding, a question about the story indicates that the questioner must know something about the story in order to have asked the question in the first place. So our answer should not return information which must have been known in order to ask the question. Such answers do not convey new information. If a question asks something that is explicitly in the story, it is safe to assume that the questioner does not remember that particular part of the story. In this case the answer is a straightforward retrieval of the story information. But when a question involves inferences about things not explicitly in the story, the problem of what can be assumed and what is of primary interest becomes unavoidable.

The best answers are those which convey the most information in the

most efficient way. In the Leone's story, we are answering "Why did John wash dishes?" with "Because John did not have any money." This is a very poor answer since it does not convey any information the questioner could not have figured out for himself. A patron washing dishes in a restaurant is so scripty that being unable to pay the check is highly suspect right away. Given that John couldn't pay the check, anyone would infer immediately that John had no money. A much better answer would be "Because a thief picked John's pocket on the subway." This answer conveys a lot of information. By inference it tells us that John had no money and therefore John couldn't pay the check. It conveys an entire causal chain which starts at the pickpocketing and ends with John washing dishes. And since part of the causal chain (John having no money and not being able to pay) is easily inferred, this answer has transfered information very efficiently. An answer which conveys content by inference is preferable to one which spells out such inferences explicitly. Given three possible answers:

Q: Why did John wash dishes at the restaurant?

A1: Because he couldn't pay the check.
A2: Because he had no money.
A3: Because he had been pickpocketed on the subway.

How do we select the best one? How can we characterize these answers to arrive at A3 as the best answer?

B. The Answer Selection Process

The following system for answer selection is designed to pick the best possible answer given a set of answers to a question based on some

story. Since this procedure does not reflect a complete theory of answer selection, it should not be difficult to find exceptional cases where the proposed model fails. This answer selection model is being presented here merely to indicate what directions must be investigated.

1. DEFINITIONS

The DATA CONTEXT (DC) is the input story to which a question refers. Answers to questions can be characterized as being INDEPENDENT OF DATA CONTEXT (IDC) if they are answers one might reasonably guess without having read any story in connection with the question.

Q: Why did John see a doctor?
A: John was sick. (IDC)

Q: Why did John fall asleep?
A: John was tired. (IDC)

Answers which are derived from the data context by being explicit in the data context are EXPLICITLY DEPENDENT (ED) on the data context. Answers which are derived from the data context by inferences on the data context are IMPLICITLY DEPENDENT (ID) on the data context.

Examples:

EX1: One morning John noticed that his dog was having trouble walking. That afternoon he took it to the vet.

Q: Why did John take his dog to the vet?
A: It was sick or injured. (IDC)
A: It was having trouble walking. (ED) ***
A: He wanted to make it well. (ID)

EX2: One day John broke the mainspring of his watch. The dentist who lived next door fixes old watches as a hobby. So John

took his watch to the dentist.

Q: Why did John take his watch to the dentist?
A: He had broken the mainspring. (ED)
A: The dentist fixes old watches. (ED) ***
A: He wanted it fixed. (ID)

EX3: One day John broke the mainspring of his watch. The dentist next door fixes old watches for a hobby. John called the dentist.

Q: Why did John call the dentist?
A: He wanted to make an appointment. (IDC)
A: He had broken the mainspring of his watch. (ED)
A: The dentist fixes old watches. (ED) ***
A: He wanted to get his watch fixed. (ID)

EX4: One day John broke the mainspring of his watch. He took it to the jeweler's to see if they would buy it.

Q: Why did John take his watch to the jeweler's?
A: It was broken. (IDC) (ED)
A: He had broken the mainspring. (ED)
A: He wanted to see if they would buy it. (ED) ***

EX5: While walking home, John realized he didn't have his umbrella with him. He remembered last having it at the restaurant he went to for lunch. John walked over to the restaurant.

Q: Why did John go to the restaurant?
A: He wanted to get something to eat. (IDC)
A: He wanted to find his umbrella. (ID) ***
A: He last had his umbrella at the restaurant. (ED)

EX6: John got three F's on his report card. He decided not to show it to his parents.

Q: Why didn't John want to show his report card to his parents?
A: His grades weren't good enough. (IDC)
A: John was afraid that they would be angry. (ID) ***
A: John had gotten three F's on it. (ED)

EX7: John was on the side of a highway when a large truck skidded off the road. At the scene of the accident John noticed an oil slick covering the pavement. When he saw a taxi approaching the spot at high speed, he waved his arms frantically, trying to signal the driver.

Q: Why was John waving at the taxi?
A: John wanted a ride. (IDC)
A: John wanted to prevent it from skidding off the road. (ID) ***
A: John wanted to stop it. (IDC)
A: John was trying to signal the driver. (ED)

2. MORE DEFINITIONS:

A1 is a CAUSAL ANTECEDENT of A2 iff it makes sense to say "A2 because A1".

A2 is an INTENTIONAL CONSEQUENT of A1 iff

a) A2 is of the form "X wanted to ... {C1} ..."
b) A1 is of the form "X ... {C2} ..."
c) if makes sense to say "X ... {C2} ... in order to ... {C1} ..."

A2 is a PLAN COMPONENT of A1 iff

a) A1 is of the form "X wanted ... {C1} ..." or "X needed ... {C1} ..."
b) it makes sense to say "... {A1} ... and X knew that ... {A2} ..."

Two answers are CONSISTENT iff one can readily be inferred from the other in the given data context, e. g., in EX5, finding his umbrella and wanting something to eat are not consistent answers. In EX3, the watch being broken and John wanting to get it fixed are consistent answers.

A1 is a MOTIVE ORIENTED answer iff

a) A1 is not of the form "X wanted..."
b) A1 describes an activity or state which strictly precedes the activity or state of the question statement in time.

3. THE SELECTION RULES:

The following rules are to be applied in succession. At the end of RULE 3, a tentative answer (TA) has been picked. The tentative answer may be changed by RULES 4-6. The tentative answer become the final answer only after the application of RULE 6 is completed.

RULE 1: An IDC is the preferred answer only if there are no ED's or ID's.

RULE 2: An ID is preferred over ED's only when the question has at least one IDC and the ID is not consistent with the IDC's.

RULE 3: Given a choice of ED's, first eliminate those ED's which are also IDC's (these are the least interesting answers). Next test to see if there are any motive oriented ED's. Eliminate these. If there is still a choice, resort to rule 3a.

RULE 3a: Given a choice of ED's, first eliminate those ED's which are also IDC's. Next test each ED by removing its associated data statement from the data context.

The ED in question is the best answer if:

a) the revised DC makes no sense, or
b) the revised DC generates an ID which is not consistent with the ED in question.

NOTE ON RULES 4-6: Do not consider IDC's as possible replacements for the TA.

RULE 4: If the TA is the causal antecedent of another answer, replace the TA with the other answer.

RULE 5: If another answer is the intentional consequent of the TA, replace the TA with the other answer.

RULE 6: If another answer is a plan component of the TA, replace the TA with the other answer.

4. SELECTION RULES IN ACTION

ILLUSTRATION OF RULE 2:

In EX5, we have A1: Because he want to find his umbrella.
A2: Because he last had his umbrella at the restaurant.
A3: Because he wanted to get something to eat.

A3 is an IDC. A1 is an ID and A2 is an ED. A1 is the best answer since A1 and A3 are not consistent.

ILLUSTRATION OF RULE 3:

In EX3, we have ED1: Because he had broken the mainspring of his watch.
ED2: Because the dentist fixes old watches.

Since ED1 is motive oriented, ED2 is the preferred answer.

ILLUSTRATION OF RULE 3a:

In EX4, we have ED1: Because he had broken the mainspring.
ED2: Because he wanted to see if they would buy it.

The revised DC wrt ED1 is: John took his pocketwatch to the jeweler's to see if they would buy it.

This generates: A1: Because it was broken (IDC)
A2: To see if they would buy it. (ED)

neither a) nor b) of the test hold here.

The revised DC wrt ED2 is: One day John broke the mainspring of his watch. He took it to the jeweler's.

This generates: A1: Because it was broken. (IDC) (ED)
A2: Because he had broken the mainspring. (ED)
A3: Because he wanted it fixed. (ID)

here A3 is not consistent with ED2, b) holds in this case, and so

we choose ED2 as the answer.

In EX2, we have ED1: Because he had broken the mainspring.
ED2: Because the dentist fixes old watches.

The revised DC wrt ED1 is: The dentist next store fixed old watches as a hobby. John took his watch to the dentist.

This generates: A1: Because he had broken it. (ID)
A2: Because he wanted it fixed. (ID)

neither a) nor b) hold here.

The revised DC wrt ED2 is: One day John broke the mainspring of his watch. John took his watch to the dentist.

Since this DC makes no sense, a) holds in this case, and so we choose

ED2 as our answer.

ILLUSTRATION OF RULE 4:

In EX6 we have A1: John had gotten three F's on it.
A2: John was afraid that they would be angry.

A1 is the TA. A1 is a causal antecedent of A2 since it makes sense to say "John was afraid that they would be angry because he had gotten three F's on it."

So A2 replaces A1 as the TA.

ILLUSTRATION OF RULE 5:

In EX7 we have A1: John was trying to signal the driver.
A2: John wanted to prevent him from skidding off the road.

A1 is the TA. A2 is an intentional consequent of A1 since it makes sense to say "John was trying to signal the driver in order to prevent him from skidding off the road. So A2 replaces A1 as the TA.

ILLUSTRATION OF RULE 6:

In EX3 we have A1: John wanted to get his watch fixed.
A2: The dentist fixes old watches.

A1 is the TA. A2 is a plan component of A1 since it makes sense to say "John wanted to get his watch fixed and he knew that the dentist fixes old watches." So A2 replaces A1 as the TA.

C. The Theory Behind the Selection Rules

We need to be able to justify the rules of the selection process in terms of a theory of text comprehension. There is one basic principle underlying all the hypotheses suggested.

PRINCIPLE: Given that the system is communicative, the best possible answer to a question is one which conveys (either explicitly or by inference) the maximal amount of relevant and interesting information without including extraneous information.

Rule 1: Since IDC answers are by definition those which require no data outside of the question itself and general world knowledge, these are clearly the least interesting of all possible answers. Aside from being boring, they will often be wrong in the presence of alternative answers.

RULE 2: This is based on the assumption that a body of text will only contain explicitly information which is absolutely necessary to correct comprehension or information which has a high interest value.

RULE 3: This is based on a conjecture that if both a motive and a goal were important or unusual enought to be included in the text, then the motive can probably be inferred from the goal, i.e., it is impossible to have an unusual motive with a usual goal since motives generally predict goals and a goal can only be consider usual in the context of some motive.

RULE 3a: This is clearly an attempt to determine which piece of information is more necessary to the comprehension process. Note that in the examples presented, this rule need never be called since RULE 3 covers all examples with multiple ED's. Since the implementation of the rule is so involved, there are probably better rules which will filter out ED's in the event that RULE 3 fails. This rule should only be activated as a last resort.

RULE 4: The effect is more important that the cause. And a cause can usual be inferred from its effect.

RULE 5: Intentions are more important than their corresponding actions.

RULE 6: Intentions are more important than their corresponding actions.

While these conjectures are each susceptible to counterexamples, by proposing such a system and examining where it breaks down, we can hopefully zero in on a set of principles which will define the types of processing a

question answering program must incorporate.

D. The Theory Behind a Generation/Selection Model

There are two theoretical stands one may take on a generation/selection model. It may be argued that such a model is a valid model of human question answering. This position would claim that when asked a question, people generate on some unconscious level, a number of possible answers to the question from which the best answer is selected. Alternatively, one may argue that people do not go through such processing at the time a question is asked, but that some form of equivalent processing is incorporated in the generation of memory representations.

On one hand it seems that the types of processing involved in a generation/selection model (the inferencing and explanation seeking) is necessary to comprehension, and would therefore have to occur at the time a story is understood, not at the time a question is asked. So perhaps all of the processing is embedded in the generation of a memory representation. On the other hand, if you ask someone something about a story, and upon hearing their answer, ask them why they didn't answer in some other way, people seem to be very quick to explain why their answer is better. So either generation/selection is a valid model of human cognition, or, a selection process can be applied very quickly to a given memory representation at the time one is asked to argue for their answer.

To the extent that one believes human cognitive processing must be optimally efficient, it makes sense to side with the viewpoint of incorporation in memory representation. This is simply because it would be inefficient for people to repeat any processing which had to be done at the time of

comprehension again at the time of question answering. If a process is completed once, and the results are coded in a memory representation, then it seems more reasonable to access the information encoded in that representation rather than process the same thing over again.

But even if the incorporation viewpoint is favored, it will still be useful to develop a generation/selection model in order to find out what must be included in the memory representation generated at the time of comprehension. So we may think of the generation/selection model as a way to find out what the memory representation must contain. We do not need to defend it as a model of human processing. The theory behind a generation/selection model will become theory behind any memory representation which embodies the end results of generation/selection processing.

VIII. Conclusions

Research in computational question answering seems to be evolving from a viewpoint which approaches the question answering problem as basically information retrieval. The predominant attitude toward question answering reasons that if you want a data base query system which is reliable over some user population, then you are primarily concerned with finding something that works; you do not need a program which as a 'human' understanding of the information it is processing. The best known system typifying this approach is Woods' LSNLIS program [12] which answers questions about the geological properties of lunar rock and soil samples. LSNLIS does not attempt to model human understanding. If you asked it whether or not any of the moon rocks were bigger than a breadbox, LSNLIS could not answer since all of its knowledge is purely technical. Of course Woods does not care if it can't answer that question because the intended user of LSNLIS is a geologist who would never ask such a question.

Question answering over non-technical information is very difficult because it requires 'human' processing of information as opposed to a strict retrieval of information. Many attempts have been made to base question answering systems on symbolic logic, deductive reasoning, and statistical methods. Partial surveys of such systems can be found in [11] and [13]. But none of these systems can handle mundane, common sense problems like understanding the causality between having your pocket picked and having to wash dishes in a restaurant. The sort of understanding needed for mundane problems relies on human inferencing.

We have seen how SAM incorporates a theory of human memory organization in order to make necessary inferences about what it reads. The only other

question answering program which attempts to model human memory is Scragg's LUIGI (see [10]). LUIGI answers questions about the preparation of food in a kitchen by simulating the processes of preparation. The underlying premise is that certain types of knowledge are organized under specific procedures. This is very close to our theory of scripts which claims that certain types of knowledge are organized under specific routines.

The theory of question answering I am developing is a theory of human thought process and interaction. The theory is therefore concerned with the cognitive processes people use when they engage in question answering. As the theory evolves it is accompanied by the development of a computational model (a computer program) which functions as an investigative tool. Whenever the program fails to simulate a human response, it has indicated where the theory is deficient, inconsistent, or wrong. Of course computer verification of cognitive theories does not constitute proof that the theories tested are in fact the way people do it. However it does indicate when you at least have a possible model of human thought processes.

But whether one cares about how people do it or not, it is becoming more and more apparent that the research in natural language processing must be guided by a consistent body of theory. The ad hoc systems designed for specific microworlds have just not been extendible to any general levels of competence. Initial efforts in natural language processing were founded on the premise that computers are computationally superior to people and therefore there must be a machine-optimal way to process natural language which will undoubtedly be better than the way people do it. Perhaps there is. But until somebody figures out a way to get a handle on it, it makes sense to approach the problem by asking how people manage it. It seems that one is forced to worry about how people do it whether the intrinsic

interest is there or not.

So the approach to computational question answering which I have presented is thoroughly based on computer simulation of human processes. As such, it is concerned with problems of human memory organization as well as problems of human interaction. The general question answering situation is much more than mere memory retrieval. It involves social interaction dynamics and an assessment of the questioner's knowledge state. Donald Norman [5] has examined some of the human processing involved in general question answering, but such issues are generally thought to be more appropriate to psychology than computer science. Just as language understanding pulls in a necessity for general world knowledge, question answering as an interactive process cannot avoid the dynamics of social interactions.

On the other hand, there is a great deal to be done before we need to seriously worry about social psychology. Memory representation is currently demanding primary attention and will continue to occupy a position of critical importance for quite some time. Since question answering is a simple test of computational understanding, the evolution of memory representation will be guided and parallelled by the development of question answering techniques. As such, question answering is not only a desirable goal by itself, but is more importantly an integral tool in the development of natural language processing systems.

References

[1] Abelson, R. P. Concepts for Representing Mundane Reality in Plans. In D. G. Bobrow & A. M. Collins (eds.), Representation and Understanding. New York: Academic Press, 1975.

[2] Cullingford, R. E. An Approach to the Representation of Mundane World Knowledge: The Generation and Management of Situational Scripts. American Journal of Computational Linguistics (in press).

[3] Lehnert, W. What Makes SAM Run? Script-Based Techniques for Question Answering. Proceedings for Theoretical Issues in Natural Language Processing, Cambridge, MA, 1975.

[4] Meehan, J. R. Using Planning Structures to Generate Stories. American Journal of Computational Linguistics (in press).

[5] Norman, D. A. Memory, Knowledge, and the Answering of Questions. Center for Human Information Processing: Technical Report 25. University of California, San Diego, 1972.

[6] Schank, R. C. Conceptual Information Processing. New York: American Elsevier, 1975.

[7] Schank, R. C. The Structure of Episodes in Memory. In D. G. Bobrow & A. M. Collins (eds.) Representation and Understanding. New York: Academic Press, 1975.

[8] Schank, R. C. & Abelson, R. P. Scripts, Plans, and Knowledge. Proceedings for the 4th International Joint Conference on Artifical Intelligence, Tbilisi, U. S. S. R., 1975.

[9] Schank, R. C., et al. SAM - A Story Understander. Department of Computer Science Research Report 43. Yale University, New Haven, CT, 1975.

[10] Scragg, G. W. Answering Questions about Processes. San Diego: University of California (Thesis), 1974.

[11] Simmons, R. F. Natural Language Question Answering Systems: 1969. Communications of the ACM, 1970, 13, 15-30.

[12] Woods, W. A., Kaplan, R.M. & Nash-Webber, B. The Lunar Sciences Natural Language Information System: Final Report. Bolt, Beranek and Newman Report No. 2378, Bolt, Beranek and Newman, Inc., Cambridge, MA. June 1972.

[13] Petrick, S. R. On Natural Language Based Query Systems. IBM Research Report RC 5577, Thomas J. Watson Research Center, Yorktown Heights, New York, August, 1975.

DISTRIBUTION LIST

Defense Documentation Center
Cameron Station
Alexandria, Virginia 22314

Office of Naval Research
Information Systems Program
Code 437
Arlington, Virginia 22217

Office of Naval Research
Code 102IP
Arlington, Virginia 22217

Office of Naval Research
Branch Office, Boston
495 Summer Street
Boston, Massachusetts 02210

Office of Naval Research
Branch Office, Chicago
536 South Clark Street
Chicago, Illinois 60605

Office of Naval Research
Branch Office, Pasadena
1030 East Green Street
Pasadena, California 91106

New York Area Office
715 Broadway - 5th Floor
New York, New York 10003

Naval Research Laboratory
Technical Information Division
Code 2627
Washington, D.C. 20375

Dr. A.L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps (Code RD-1)
Washington, D.C. 20380

Office of Naval Research
Code 455
Arlington, Virginia 22217

Office of Naval Research
Code 458
Arlington, Virginia 22217

Naval Electronics Laboratory Center
Advanced Software Technology Division
Code 5200
San Diego, California 92152

Mr. E.H. Gleissner
Naval Ship Research & Development Center
Computation and Mathematics Department
Bethesda, Maryland 20084

Captain Grace M. Hopper
NAICOM/MIS Planning Branch (OP-916D)
Office of Chief of Naval Operations
Washington, D.C. 20350

Mr. Kin B. Thompson
Technical Director
Information Systems Division (OP-91T)
Office of Chief of Naval Operations
Washington, D.C. 20350

Advanced Research Projects Agency
Information Processing Techniques
1400 Wilson Boulevard
Arlington, Virginia 22209

Professor Omar Wing
Columbia University in the City of New York
Dept. of Electrical Engineering & Computer Science
New York, New York 10027