AD-A062 974 YALE UNIV NEW HAVEN CONN DEPT OF COMPUTER SCIENCE F/G 5/2
MEMORY ORGANIZATION FOR NATURAL LANGUAGE DATA-BASE INQUIRY.(U)
SEP 78 J L KOLODNER N00014-75-C-1111
UNCLASSIFIED RR-142 NL

1 OF 1
AD
A062974

END
DATE
FILMED
2-79
DDC

(6) MEMORY ORGANIZATION FOR NATURAL LANGUAGE DATA-BASE INQUIRY.

(11) Sep[redacted]78

(9) Research Report. #142

(12) 43p.

(10) Janet L. Kolodner

(14) RR-142

(15) N00014-75-C-1111

407 051

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

78 12 27 019

The work here 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.

MEMORY ORGRANIZATION FOR NATURAL LANGUAGE DATA-BASE INQUIRY

September 1978

Research Report #142

Janet L. Kolodner

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER #142	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Memory Organization for Natural Language Data-Base Inquiry		5. TYPE OF REPORT & PERIOD COVERED Technical 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Janet L. Kolodner		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-1111
9. PERFORMING ORGANIZATION NAME AND ADDRESS Yale University - Department of Computer Science Artificial Intelligence Project New Haven, Connecticut 06520		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209		12. REPORT DATE September 1978 13. NUMBER OF PAGES 38
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research Information Systems Program Arlington, Virginia 22217		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this report is unlimited

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

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

18. SUPPLEMENTARY NOTES

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

Conceptual information	data-bases	inference
Knowledge organization	question answering	knowledge structures
Natural language processing	scripts	memory

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

A natural language data-base query system, called CYRUS, is described. It stores and retrieves biographical information about people, making use of knowledge structures such as scripts and role themes in order to store the information in memory and guide the question answering process.

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

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

-- OFFICIAL DISTIRUBTION LIST --

Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12 copies
Office of Naval Research Information Systems Program Code 437 Arlington, Virginia 22217	2 copies
Office of Naval Research Code 102IP Arlington, Virginia 22217	6 copies
Advanced Research Projects Agency Cybernetics Technology Office 1400 Wilson Boulevard Arlington, Virginia 22209	3 copies
Office of Naval Research Branch Office - Boston 495 Summer Street Boston, Massachusetts 02210	1 copy
Office of Naval Research Branch Office - Chicago 536 South Clark Street Chicago, Illinois 60615	1 copy
Office of Naval Research Branch Office - Pasadena 1030 East Green Street Pasadena, California 91106	1 copy
Mr. Steven Wong Administrative Contracting Officer New York Area Office 715 Broadway - 5th Floor New York, New York 10003	1 copy
Naval Research Laboratory Technical Information Division Code 2627 Washington, D.C. 20375	6 copies
Dr. A.L. Slafkosky Scientific Advisor Commandant of the Marine Corps Code RD-1 Washington, D.C. 20380	1 copy

Office of Naval Research 1 copy
Code 455
Arlington, Virginia 22217

Office of Naval Research 1 copy
Code 458
Arlington, Virginia 22217

Naval Electronics Laboratory Center 1 copy
Advanced Software Technology Division
Code 5200
San Diego, California 92152

Mr. E.H. Gleissner 1 copy
Naval Ship Research and Development
Computation and Mathematics Department
Bethesda, Maryland 20084

Captain Grace M. Hopper 1 copy
NAICOM/MIS Planning Board
Office of the Chief of Naval Operations
Washington, D.C. 20350

Mr. Kin B. Thompson 1 copy
Technical Director
Information Systems Division
OP-91T
Office of the Chief of Naval Operations
Washington, D.C. 20350

Advanced Research Project Agency 1 copy
Information Processing Techniques
1400 Wilson Boulevard
Arlington, Virginia 22209

Professor Omar Wing 1 copy
Columbia University in the City of New York
Department of Electrical Engineering and
Computer Science
New York, New York 10027

Office of Naval Research 1 copy
Assistant Chief for Technology
Code 200
Arlington, Virginia 22217

MEMORY ORGANIZATION FOR NATURAL LANGUAGE DATA-BASE INQUIRY*

Janet L. Kolodner

July, 1978

Artificial Intelligence Project
Computer Science Dept.
Yale University
New Haven, CT. 06520

INTRODUCTION

The CYRUS system is a natural language data-base query system containing biographical information about Cyrus Vance, Secretary of State of the United States. It also contains additional information associated with Vance that may be needed in order to answer questions about him, such as current events and limited biographical information about people he has been in close contact with. Questions input in English are answered by accessing conceptual structures containing this information.

Humans are good processors of conceptual information. Upon hearing a natural language utterance, they extract both the implicit and explicit content of the utterance, combine this with contextual information to further refine the content, integrate the concepts involved with what they already know, and make the appropriate answer. Later, they can retrieve the content of that utterance from memory and use it to understand something else they hear, making the appropriate assumptions and inferences, and supplying an appropriate answer.

*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.

CYRUS is an attempt to model the way human memory might be organized to do the task just described. It is designed to extract both meaning and intent from natural language input, and use that input either to update its memory or to answer a question. It is the organization of memory that is of primary importance in describing the task of adding new information or retrieving information that is already there.

This paper addresses the problems of knowledge organization and retrieval of information in the data-base. The conceptual structures used to represent, organize, and retrieve information are described, as well as the procedures used to answer natural language questions. Specific event information about the persons represented in the data-base must be stored; so must world events, general world knowledge, and rules about inference and reference resolution. The first half of the paper will describe and explain some of the structures used for organizing this knowledge; the second half will show how CYRUS makes use of these structures.

A sample question-answering dialogue with the system follows. It illustrates some of the information stored in the data-base, some of the types of questions the system can answer, and some of the reference problems it can solve.

Q1: Who is Cyrus Vance?
A1: SECRETARY OF STATE OF THE UNITED STATES.
Q2: How did Vance become Secretary of State?
A2: HE WAS APPOINTED BY PRESIDENT CARTER.
Q3: Who is Grace Vance?
A3: CYRUS VANCE'S WIFE.
Q4: Do they have any kids?
A4: YES, FIVE.
Q5: How old is Vance?
A5: 61.
Q6: Where was he born?

A6: CLARKSBURG, WEST VIRGINIA.
Q7: Who are his parents?
A7: AMY ROBERTS VANCE AND JOHN CARL VANCE.
Q8: Where did he go to college?
A8: YALE UNIVERSITY.
Q9: Where was Vance last week?
A9: IN THE UNITED STATES.
Q10: Where is he now?
A10: IN ISRAEL.
Q11: Why did he go there?
A11: TO DISCUSS ARAB-ISRAELI RELATIONS WITH THE EGYPTIAN AND ISRAELI FOREIGN MINISTERS.
Q12: How long will he be there?
A12: SIX DAYS.
Q13: Where will he go afterwards?
A13: TO EGYPT.

MOTIVATION FOR CYRUS

In any data-base, a major problem is how to represent the information to be stored and how to systematically organize and inter-relate that information so that the data-base can be efficiently updated and queried. In a very large data-base, some information might not be explicitly in the data-base, but could be there implicitly as an inference from something explicit. In order to retrieve information that is only implicitly in the data-base, it is necessary to efficiently organize both the information in memory and the procedures which access and perform inferences on that information.

A second problem of data-base systems is specifying a query language. Natural language is the most useful language for querying a data-base in that it does not require the questioner to have any special training. It makes the data-base accessible to everyone. Communicating to a data-base in natural language, whether for updating or for retrieval of information, requires that the system understand the meaning, intent, and other implications of the natural language utterance. Inferences that must be made from a natural language utterance must be organized systematically to avoid an exponential explosion upon implementation.

In the CYRUS system, we have used instantiations of conceptual frames and knowledge structures (such as scripts) to represent and organize specific events in memory. These knowledge structures are also used to organize inferences necessary for natural language dialogue with the data-base, for both the updating and the retrieval processes.

A data-base meant to be updated and queried in natural language should be organized taking into account the peculiarities of natural language. Most data-bases have not been designed in this way, and have dealt with natural language inquiry only through natural language front ends specific to each data-base. Such front-end interface is typically added on after the data-base has been designed, thus ignoring the fact that efficient natural language inquiry requires a data-base specifically designed for that purpose. CYRUS is a data-base designed for natural language updating and querying. Both the nature of the information in the data-base and the wide range of possibilities in natural language dialogue have been taken into account in designing CYRUS.

In understanding natural language, it is necessary to use contextual information and knowledge about the intent of the speaker. Because the implications and intent of an utterance are all included in its meaning, its internal representation must include implications and intent. CYRUS does this by using conceptual representations, including Conceptual Dependency (CD), scripts, and role themes (Schank and Abelson, 1977). These structures, along with a new conceptual structure called an era, are used to represent and organize information in memory, and also to guide the processing of new information -- either to update memory or to answer questions.

Lexical representations, which use English words to represent and retrieve information, do not offer the flexibility needed to deal with conceptual knowledge. For example, suppose a lexical data-base had the information "John has a red car," and it needed to answer the question "Who can drive to work tomorrow?" A data-base representation based on words could answer that question only by linking the words "car" and "driving" in a way that is not generalizable to other domains. A conceptual data-base, on the other hand, would use the meaning of the question and the inferences associated with that meaning to answer the question. It would not be concerned that the words do not match, because the representations in the data-base and the internal representation of the question would not be based on words, but rather on meaning. A conceptual data-base would construct a canonical meaning representation of the question to which inferences could be efficiently associated. Different utterances with the same meaning would be associated with the same representation and inferences. A conceptual data-base must be able to make inferences from the meaning of a question -- it must efficiently figure out the intent of the question and the correct place in the data-base to look for the answer. It should also be generalizable enough to add arbitrary new domains easily.

A number of relational data-bases and question-answering systems that use symbolic logic, deductive reasoning, and statistical methods have been written (for examples, see Petrick (1975), Waltz (1978), and McSkimin and Minker (1977)). While all of these systems have produced working models, they have not been developed as generalized systems for dealing with conceptual information, and have been deficient in one or more of the problem areas mentioned above.

In addition to the interest in improving data-base design, CYRUS was also motivated by ongoing work at Yale. SAM (Cullingford, 1978) and FRUMP (DeJong, 1977) are two programs that understand newspaper stories by using scripts. SAM reads in detail, using knowledge of a script to fill in missing details of the story. FRUMP skims news stories from the UPI wire and produces summaries. In working with these systems, it became apparent that we would benefit from a general memory model which would know outside information that SAM did not know, but which could be consulted by SAM. For example, in stories about VIP's, SAM needed to know certain information about these VIP's -- their names, functions, family relations, where they were from, etc. -- in order to understand the stories. However, there is no reason why SAM, which is a script specialist, should have to store that information. Such information should be stored in a separate memory module which SAM, or any other program needing that information (such as FRUMP or PAM (Wilensky, 1978) or the conceptual analyzer (Riesbeck, 1975)) could query.

CYRUS was designed to integrate information produced by FRUMP (which processes input from the UPI wire) into a data-base that could be continually updated. Vance was chosen as the subject of the memory model because he is in the news so often and will provide a large number of news updates.

STRUCTURES IN CYRUS FOR ORGANIZING EVENTS

Representations of events in CYRUS take the form of scripts and Conceptual Dependency representations. Scripts and CD are used to represent events in memory, but they are not representationally adequate for organizing all the biographical information we know about someone. Events in memory must be organized in a way that makes both the retrieval of an event and the addition of a new event fast and efficient. Information in memory must be organized so that memory can be searched in the most optimal way. In answering a question about Vance's law career, we don't care about his experiences in high school; we want to be able to look directly in the right place in the data-base to get the answer. An era is a type of memory structure designed for organizing biographical events in memory.

Eras represent time spans in a person's life that are characterized by one outstanding occupational, familial, or social role. For instance, a person's four years in high school constitutes an era. His primary role is that of high-school student; his primary activity during that time is attending high school every day (or doing the high-school script every day). Periods of going to college or professional school, being in the military service, being a politician or businessman, or being a parent are other examples of eras.

A role theme (Schank and Abelson, 1977) is the information suggested by a person's occupational, social, family, or situational role. Examples of role themes are: student, businessperson, professional, politician, parent, husband, and wife. For example, if we know that someone is a student, then we assume that person goes to classes during the day, does homework, is probably not very wealthy, probably does not work full time but may work part time, that he or she has a course of study, may be training for a particular type of job, and so forth. The student role theme contains all of that information.

A role theme may have sub-role-themes associated with it. For instance, one sub-role-theme of the student role theme is high-school student, another is college student, and another is graduate student. If we know somebody is a graduate student, we assume student role theme information about him, and in addition we assume role-theme information specific to graduate students: they are usually supported by fellowships, they keep odd hours, they may be expected to do some teaching, they do independent research, they have a strong interest in their field of study, and they are preparing for a job in that field. We assume role theme information when we hear that a person is living in that role and we have no other contradictory knowledge about that person's involvement in the role. If a person deviates from the stereotypic role theme, then we must use our specific knowledge of his unique situation for understanding.

Eras contain and organize all the biographical information related to a specific role theme in a person's life, and are named for that particular role theme. Since many role themes occur simultaneously in a person's life, many eras can exist simultaneously. For example, a married businessperson occupies at least two separate role themes -- during the day at work, he or she is in the businessperson role theme, and at home with the family, he or she is in the husband or wife (or possibly also the parent) role theme. Events associated with work would be stored in an occupational era for that person, while those associated with being in the spouse role theme would be stored in a family era. In CYRUS, eras are defined by major occupational (including educational), family, and social role themes. Thus, each person has parallel sequences of occupational, family, and social eras, with each era characterized by a particular occupational, family, or social role theme.

USING ROLE THEMES AND ERAS IN UNDERSTANDING

Role theme and era knowledge is useful in retrieving information from memory. Suppose we want to answer the question "Has Vance ever won an election?" Script, role theme, and era knowledge are all used in answering this question. The election script, mentioned in the question, is usually done by someone in a political role theme, and would therefore be found in a political era. Processing in the election script would tell us to look in Vance's political eras for the answer. Events during those portions of his life can be searched for the answer. Political role themes found in his political eras can be searched to see if one required an election. In this case, the political role themes that Vance has assumed have been advisor to the president, representative at peace talks, and Secretary of State. We can look at the enablement or entry conditions for each of those role themes to see if any were elected positions. In this case, none were. We have not found an instance of Vance winning a political election.

However, further processing in the election script tells us that people can win elections in school or social organizations, and that we should therefore look in eras when Vance was in school, or in social eras, to find an instance of him winning an election. A similar procedure for these two era types provides the answer.

Knowledge about eras and role themes is also used in adding information to the data-base and in interpreting information retrieved from the data-base. For example, knowing that Vance is a political dignitary tells us that he will do things to advance his political career. Suppose we want to add to the data-base that Vance went to a party with other cabinet members. Going to a party is normally a social event that belongs in a social era, but in this case, further investigation of the event shows that the other people invited were political colleagues rather than social friends of Vance. One rule we have about career role themes is that socializing with colleagues is something career people do. We would therefore put the event in Vance's political career era. Later, if asked why he went to the party, we can use role theme information from the era in which the event is stored to answer that he went to the party to maintain political career relationships.

Role themes and eras are important because they tell us how to structure knowledge, and they also give us expectations that tell us how to process knowledge both for question-answering and for memory-updating. Representations, storage processes, and retrieval mechanisms are closely related through scripts, role themes, and eras.

ORGANIZING INFORMATION ABOUT PEOPLE

Since CYRUS holds primarily biographical information about people, an important knowledge structure in this data-base is the person frame. The person frame organizes all the information we may have about a person. For people we know very well, the person frame is very densely filled; for people we don't know as well, it is more sparse. Persons have names, parents, spouses, children, occupations, appearances, attitudes, histories, etc., which are all represented in the person frame. One of the most important properties of a person (perhaps the most important in this system) is his history or life sequence. Figure 1 (p. 15) gives an example of some of the properties included in a typical person frame. These are the properties that usually interest a typical friend. Therefore, information such as office address and social security number are not included in the person frame, while name, spouse's name, occupation, and hair color are.

A person's life cycle can be represented by sequences of eras. Sequences of eras are viewed as the personal time line for the person in question. Upon hearing about a person, we make assumptions about that person based upon what we know about possible era sequences. We know that he had an early childhood beginning at birth, that he had a family and certain family relationships, and, if he is in a western culture, that he went to elementary school, high school, etc. Upon hearing about a person who is a lawyer, we combine our knowledge about era sequences and enablements for becoming a lawyer in order to recognize that college and law school were also eras in his occupational sequence of eras.

An era includes the role theme of the person during that time span, time pointers to other parallel era sequences, and the list of biographical events that happened during that era and were related to the associated role theme. The list of biographical events contains CD representations for events and sequences of events as well as instances of scripts. All of the biographical information concerning an era can be found in its list of events. However, some information is more important than other information, and for reasons of efficiency of retrieval, that information is represented redundently in other lists attached to the era. Important information includes: people the actor is in contact with in this role theme and his relationships to those people; related activities and hobbies the person is involved with while in this role theme; and information about how this person differs from the stereotypical role theme. All of the information in these lists of people, activities, and so forth, can also be found by looking through the list of events. However, retrieval becomes more efficient since retrieval of important data is easier than retrieval of more trivial data, certainly a desireable feature in a general retrieval system. These additional lists contain the important information that is brought to mind in thinking about an era, and thus encode information that is most likely to be queried. In addition, some events in the list of events are tagged as being important events, i.e., those most likely to be remembered first in thinking about the era and those most likely to be asked about.

The same redundancy occurs with respect to the person frame. Most of the properties attached to the person frame, including current occupation and family relationships, can be found by looking at the person's life sequence. However, current information is likely to be accessed more than past information, so it is represented a second time in the person frame. This enables more relevant information to be retrieved more easily.

ORGANIZATION AND REPRESENTATION OF EVENTS IN CYRUS

A person frame contains pointers to all of the information we know about a person. In CYRUS, a person frame for a particular person is represented using a token. A token is an atom representing an object and has a list of the object's properties attached to it. A person token has attributes of the person frame attached to it as well as a marker specifying that it represents a person.

A simplified verson of the token for Cyrus Vance appears in figure 1. Anything with prefix "HUM" is a pointer to another person token. Anything with prefix "LOC" is a pointer to a location token, and anything with prefix "CON" is a conceptualization (an event, a state, an instance of a script, era, or role theme, or a sequence of eras).

```
HUM1: CLASS #PERSON
      FIRSTNAME CYRUS
      MIDDLENAME ROBERTS
      LASTNAME VANCE
      GENDER *MASC*
      OCCUPATION CON27
      PROFESSION CON26
      BIRTHDATE TIM1
      BIRTHPLACE LOC1
      FATHER HUM11
      MOTHER HUM12
      WIFE HUM4
      KIDS (HUM5 HUM6 HUM7 HUM8 HUM9)
      EYECOLOR BROWN
      HAIRCOLOR GREY
      LIFE CON2
```

figure 1

Biographical events about a person are referenced through the property LIFE on the person token. The property LIFE points to the person's sequences of eras. Figure 2 shows part of Vance's occupational sequence.

```
CON20: CLASS #ERA
       VALUE ($$PROFSCHOOL ACTOR HUM1)
       DURATION (ORDER (3 YEARS))
       ROLETHEME CON21
       EVENTS (CON22 CON23 ...)
CON21: VALUE (RT-LAWSCHOOLSTUDENT ACTOR HUM1 SCHOOL ORG1)

CON25: CLASS #ERA
       VALUE ($$CAREER ACTOR HUM1)
       DURATION (ORDER (10 YEARS))
       ROLETHEME CON26
       EVENTS (CON28 CON29 ...)
CON26: VALUE (RT-LAWYER ACTOR HUM1 FIRM ORG2)

CON30: CLASS #ERA
       VALUE ($$POLITICAL-CAREER ACTOR HUM1)
       ROLETHEME CON27
       EVENTS (CON32 CON33 ...)
CON27: VALUE (RT-SEC-OF-STATE ACTOR HUM1 COUNTRY POL1)
CON32: VALUE ((<=> ($VIPVISIT ACTOR HUM1 DESTINATION POL5 ...)))
```

figure 2

Events can also be related causally through causal connectives and temporally through relational time links. Each event in memory has attached to it one or more time specifications relating it temporally to other events in memory. Most events do not have real time specified, but have a fuzzy time or time relative to some other event. Only the most important events, or some few events where real time is known, have real time specified, such as birth and high-school graduation. Some typical specifications would be "when I was 15 years old" or "2 years after we moved to Oklahoma" (represented, of course, in CD). If needed, real time, time intervals, and durations can be quickly computed using these relational times.

Events are related causally through causal links pointing to other states or events in memory. These are both forward and backward links, and can cross era boundaries and sequences of eras. These links are necessary in answering questions concerned with causality. A sequence of events is related causally by each event in the sequence pointing to the next and previous events in the sequence. Direct causality such as enablement, reason, or general causality are represented with both backward and forward causal links. Usually, in a sequence of events, only the first or last event in the sequence is important enough to be in the EVENTS lists attached to each era. However, whenever one of those events is retrieved, the whole sequence will be retrieved, and the question answerer will have the option of answering with the whole sequence or any part of it, depending upon the question type and what the user has already been told.

ANSWERING QUESTIONS

Question answering is one of the primary functions of CYRUS. Question answering involves more than simple retrieval of information. Inference, knowledge of causality, knowledge of intent, and general world knowledge are all part of the retrieval process. In addition, the question answerer must have some understanding of what the questioner already knows, and must know when a question makes sense and is therefore legitimate. (Norman (1972), Collins, et al. (1975)).

Questions are represented using CD, script, and role theme representations, i.e., the same representations used in memory. In order to understand questions, a question answerer must interface with a conceptual analysis program which parses from English into CD. The program CYRUS uses is based on one written by Chris Riesbeck (1975). Similarly, in order to output answers, a question answerer must interface with a generation program that translates from CD back into English. CYRUS's generator is modeled after one designed by Neil Goldman (1975). CYRUS itself works with Conceptual Dependency only and is designed to be language-independent. This means that if it interfaced with a parser or generator using a language other than English, it could understand questions or generate answers in that language. Inference, knowledge of causality and intent, scriptal and era knowledge, and processing knowledge are all inferred from internal representations, and therefore would remain the same regardless of which natural language was used.

Once the natural-language question has been parsed into its CD representation, question answering begins. The question must be understood and the memory must be searched for an appropriate answer. The question-answering process has four phases. When people ask questions, they use pronouns and other references to previous dialogue. Thus, in the first phase, these references must be resolved using focus, context, and structures of previous questions and answers. Also in this phase, tokens used in the question are identified as permanent tokens the memory recognizes. In the second phase, the category of the question is determined to discover the type of information being asked for, such as enablement conditions, motivation behind an event, when an event occurred, etc. Determining the question category helps in interpreting the intent of the question. For example, determining that "Who is Cyrus Vance?" is an Identification question tells us that the answer requires a meaningful identifying characterization of Cyrus Vance. The first two phases can be thought of as understanding the question. In the third phase, memory is searched for the answer. In the fourth phase, the answer retrieved from memory is further interpreted so that the generator can give a good natural language answer. Finally, the generator generates the answer in natural language. The phases are divided this way only to make discussion of the question-answering process easier. There is no claim that people process questions in phases such as these. Most of the discussion that follows will deal with interpreting questions for intent (i.e., understanding the question), searching memory for the answer, and specifying which information belongs in the generated response.

RETRIEVAL AND RESPONSE IN CYRUS

Scripts, role themes, and eras play an important role in CYRUS. Besides being used for representation and organization of knowledge in memory, they are also used to organize and guide question answering and memory updating. Many problems of inference and general world knowledge are handled through the processing defined by these knowledge structures. They provide a way of organizing inferences and providing expectations that guide the processing of knowledge. After using inferences to determine that a question is about a particular script or era, the question answerer can consult processing information organized under that structure to find out where in memory to look for an answer. Determining that a question is asking about a particular script or era narrows the search by providing information about both which era(s) to look in and where in the selected era(s) to look. Scripts and eras thus provide ways of determining if a question is legitimate, and help to infer the intent of a question.

The CYRUS system uses the same basic question categories and similar heuristics for each category as the question answerers (Lehnert, 1978) for the SAM (Cullingford, 1978) and PAM (Wilensky, 1978) systems, with appropriate changes to take care of the organization and the knowledge structures of CYRUS. Some of these question types are more applicable to CYRUS than others, and some categories have been subdivided because of their specificity in this system. The important categories for questions in CYRUS are:

1. Identification
2. Feature Specification
3. Enablement
4. Instrumental/Procedural
5. Concept Completion
6. Time
7. Place
8. Verification
9. Duration
10. Motivational
11. Occurrence
12. Result Orientation

Question categories are determined by examining the CD representation of the question. After the category is determined, the question concept or main point of the question is extracted and the retrieval process begins. The question categories will each be described briefly in explaining the retrieval process. For more information about them see Lehnert (1978).

After references in the question have been resolved and the question category has been found, the question can be answered. Processing is different for each question category, but there are similarities in processing based upon the memory organization and interaction of the question answerer with the memory. Processing relies on script structures, era structures, knowledge about people and objects, and combinations of these things. Processing will be explained by describing in general the information each question type is looking for, and then showing specific examples of how the question answerer interacts with the memory representations.

1. Identification questions

Identification questions ask for further specification of a token, and most often look like "Who is X?" or "What is Y?" An important part of answering these questions lies in inferring the intent of the question. A person asking this type of question wants to know the most relevant identifying feature of the person or object being asked about. This could be different each time the question is asked, depending upon the context. For instance, Mrs. Jones is Mary's mother to Mary and her friends, but she is Johnnie's teacher to Johnnie and his mother. In the domain of CYRUS, the relevant identifying characteristic of a person is usually the person's occupation if it is an identifying occupation, otherwise it will be the person's relation to an important person, his function, or whatever else will identify him. If a name is not mentioned in the question, it is usually assumed that the questioner is asking for the name of the person being described. CYRUS answers these questions by identifying the token asked about in the question, and then extracting the relevant information from its person frame. Therefore, in answering questions such as "Who is Vance's wife?" or "Who is Vance's father?" where the name is not specified, CYRUS answers with "Grace Vance" or "John Vance". In the question "Who is Vance's father?", "Vance's father" is represented as the male parent of Vance. Rules for identifying a parent follow:

1. If the child is known, then get the appropriate parent from his property list.
2. If the child is not known, search the list of known people for the person with the property of being the parent of the designated person.

Thus, when asked the question, "Who is Vance's father?", CYRUS will look on the Cyrus Vance token, and find the name of his father. If it had not been there, then CYRUS would have done a memory search to see if any person listed had that property. The processing for that question is as follows:

```
Who is Vance's father?
(QUESTION IS ((ACTOR (#PERSON KIDS (HUM1)
        GENDER *MASC*) EQUIV (*?*))))
(QUESTION TYPE IS identification)
(SEARCHING TOKEN HUM1 FOR FATHER)
(ANSWER IS HUM11)
John Vance
```

When the name is specified, as in "Who is Cyrus Vance?", it is up to the question answerer to pick out the best identifying feature of the token, as explained above. In this case, CYRUS will answer "Secretary of State of the United States," since his occupation is Vance's best identifying feature. If it were asked "Who is John Vance?", it would answer "Cyrus Vance's father," his relation to an important person, and therefore the most meaningful way of identifying him. A person is identified by determining his most distinguishing property. A partial set of rules for answering Identification questions follow.

If the name is mentioned in the question, then get the most distinguishing feature of the person as follows:

1. If the person's occupation is distinguishing, then identify him by his occupation.

2. If the person is related to an important person, then identify him using that relation.

3. If the person has a function important to what is being discussed, then identify him by his function.

To answer some Identification questions, CYRUS must consult information encoded in scripts or role themes. For instance, in order to identify Cyrus Vance as Secretary of State, CYRUS had to consult the script for Secretary of State to find out if it was a distinguishing occupation. Some of the information it has about Secretary of State is that it is an important political position appointed by the President, and there is only one of them in a country. This is enough information to tell CYRUS that it is indeed a distinguishing occupation and thus should be used to answer the question. In order to identify John Vance as Cyrus Vance's father, CYRUS had to consult family information from the person frame to get the list of family relations. Since CYRUS knows that father is a family relation, it found who John Vance was father of, and thus determined that this was his distinguishing feature. Some intermediate output from CYRUS follows showing the flow of control in answering those questions.

```
Who is Cyrus Vance?
(QUESTION IS ((ACTOR HUM1 EQUIV *?*)))
(QUESTION TYPE IS identification)
(SEARCHING TOKEN HUM1 FOR DISTINGUISHING FEATURE)
(SEARCHING SECRETARY OF STATE SCRIPT FOR DISTINGUISHING FEATURE)
(ANSWER IS ((ACTOR HUM1 EQUIV (#PERSON OCCUPATION CON27))))
Secretary of State of the United States

Who is John Vance?
(QUESTION IS ((ACTOR HUM11 EQUIV *?*)))
(QUESTION TYPE IS identification)
(SEARCHING TOKEN HUM11 FOR DISTINGUISHING FEATURE)
(SEARCHING FAMILY RELATIONS FOR IMPORTANT RELATIONS)
(ANSWER IS ((ACTOR HUM11 EQUIV (#PERSON GENDER *MASC* KIDS
(HUM1)))))
Cyrus Vance's father
```

2. Feature Specification questions

Feature Specification questions ask for a feature of a person or object. They are answered by looking at the token for the person or object in question, and then locating the appropriate feature. Some of the same processes used for answering Identification questions are used to identify the token being asked about. For instance, if the question was "What color were Vance's father's eyes?", the procedures for answering Identification questions would be used to locate the token for Vance's father, where information about his eye color would be stored (if it were known). Some features carry additional information. For instance, in answering "How old is Vance?", we would invoke the following rules:

1. Get the AGE feature from the actor's token.

2. If there is no AGE feature, find out if the person is alive by seeing if his DEAD feature is marked. If so, answer, "Actor is not alive", or, if the time of the actor's death is known, then answer "Actor died at time specified."

3. Otherwise, try to find out when the actor was born as follows and subtract it from the present date.

   a. Find his birthdate by looking at the BIRTHDATE feature of his token or checking his BIRTH script which is at the beginning of his EARLYCHILDHOOD era.

   b. If no birthdate is found, look at the timeline. If there is a date for high-school graduation, subtract 18 from that to get his approximate birthdate. Otherwise, if there is a date for his graduation from college, subtract 22 from that to get his approximate birthdate. Otherwise, if there is a date for him entering his CAREER era, then approximate his birthdate at between 18 and 25 years before that. Otherwise, if he has a marriage date, then approximate his birthdate as between 21 and 28 years before that.

This whole procedure is invoked when someone asks CYRUS for a person's age. In answering "How old is Vance?", the answer is found by calculating it from his birthdate, and intermediate printout from CYRUS is as follows:

```
How old is Cyrus Vance?
(QUESTION IS ((ACTOR HUM1 IS (AGE VAL (*?*)))))
(QUESTION TYPE IS feature specification)
(PROPERTY AGE NOT FOUND, CALCULATING AGE)
(FOUND BIRTHDATE)
(ANSWER IS ((ACTOR HUM1 IS (AGE VAL (61)))))
Sixty-one.
```

```
How old is Vance's father?
(QUESTION IS ((ACTOR HUM11 IS (AGE VAL (*?*)))))
(QUESTION TYPE IS feature specification)
(PROPERTY AGE NOT FOUND, CALCULATING AGE)
(FOUND DEATH)
(ANSWER IS ((CON ((ACTOR HUM11 IS (DEAD VAL (T))) AND
          ((ACTOR HUM11 IS
                    (AGE VAL (APPROX VAL (85))))))))
He is not alive, but would be about 85.
```

3. Enablement and Instrumental/Procedural questions

Enablement questions ask for enabling conditions for events. Instrumental/Procedural questions ask for instrumentality or how something was accomplished. Many times the distinction is blurry. For instance, the questions "How did Vance become Secretary of State?" and "How did Vance become a lawyer?" would be categorized as Instrumental/Procedural questions, but could also be interpreted as Enablement questions without changing the answer. They are asking the same things as "How was Vance able to become Secretary of State?" and "How was Vance able to become a lawyer?", and we would expect the same answers. This is taken care of in CYRUS by the script and role theme structures. The representations for "How did Vance become a lawyer?" and "How was Vance able to become a lawyer?" both make reference to the lawyer script. In answering the questions, the first thing we want to do is reference the lawyer script and see if there is any special processing it tells us to do. In this case, the lawyer script tells us to look at the enablement conditions of the lawyer role theme, which says to do the following:

> If the question type is Enablement or Instrumental/Procedural and if the procedure is being asked about a particular person, then find out which law school that person went to by getting the name from his LAWSCHOOL script, which is in the role theme of his PROFSCHOOL era, and answer "He went to X law school." If there is no name found, then answer "He went to law school." If there is no LAWSCHOOL script or PROFSCHOOL era found, then check that the person's PROFESSION or OCCUPATION is lawyer. If so, then answer "He went to law school." If not, then answer "He is not a lawyer." If the question is not about a particular person, then answer "by going to law school." Also, if he is a lawyer then find which state he passed the bar exam in by looking at the beginning of his CAREER era or the end of his PROFSCHOOL era for the event. If you find it, add "he passed the bar in X state" to the initial answer, else add "he passed the bar exam".*

Thus, the answer to "How did Vance become a lawyer?" would be "He went to Yale Law School and passed the New York bar." Notice that if the question had been "How does one become a lawyer?", then it could have been answered by the same procedure and the answer would have been "by going to law school and passing the bar." The same holds for "How did Vance become Secretary of State?" and "How was Vance able to become Secretary of State?" The structure of the script that holds information about being Secretary of State is consulted for special processing instructions.

```
How did Vance become Secretary of State?
(QUESTION IS ((<=> ($SEC-OF-STATE ACTOR HUM1) INST (*?*))))
(QUESTION TYPE IS instrument/procedural)
(SEARCHING SEC-OF-STATE SCRIPT STRUCTURE FOR ANSWER)
(SEARCHING ENABLEMENT FOR SEC-OF-STATE ROLE THEME)
(ANSWER IS ((<=> ($APPOINT APPOINTER HUM15 APPOINTEE HUM1 POSITION
                 (RT-SEC-OF-STATE ACTOR HUM1 COUNTRY POL1)))))
He was appointed by President Carter.
```

*Note that answers shown in the rules above and any rules that follow are only a paraphrase of the internal CD representation. They are meant to get across the content of the answer, and are not necessarily what would be generated by the generator.

4. Concept Completion, Time and Place Questions

Concept Completion questions ask for the missing component in a question. For example, "Who went to the Middle East with Vance?" asks for the group, and "Where did Vance go?" asks for the place. Time and Place questions are more specific categories of Concept Completion questions, requiring additional processing. Some of these questions have the added complexity of also dealing with scripts or other knowledge structures. For example, the questions "When was Vance born?" and "Where was Vance born?" are Time and Place questions respectively. They also ask for information from the BIRTH script. Processing algorithms in the BIRTH script follow for Time and Place questions.

If the question type is TIME, then

1. Get the BIRTHDATE of the actor.

2. Otherwise, look at his BIRTH script, located as the first event in his EARLYCHILDHOOD era. Get the TIME off his BIRTH script.

3. Otherwise, answer the question "How old is the actor?" by following the algorithm specified above. Subtract his age from the present date to get his date of birth.

4. Otherwise, answer "I don't know."

If the question type is PLACE, then

1. Get the BIRTHPLACE of the actor.

2. Otherwise, look at his BIRTH script, located as above. Get the PLACE off his BIRTH script.

3. Otherwise, answer the question "Where did actor live in his EARLYCHILDHOOD era?". Answer "probably" and that place.

4. Answer "I don't know."

```
When was Cyrus Vance born?
(QUESTION IS ((<=> ($BIRTH ACTOR HUM1)) TIME (*?*)))
(QUESTION TYPE IS time)
(SEARCHING BIRTH SCRIPT FOR ANSWER)
(SEARCHING FOR BIRTHDATE OF ACTOR)
(ANSWER IS TIM1)
March 27, 1917

Where was Cyrus Vance born?
(QUESTION IS ((<=> ($BIRTH ACTOR HUM1)) PLACE (*?*)))
(QUESTION TYPE IS place)
(SEARCHING BIRTH SCRIPT FOR ANSWER)
(SEARCHING FOR BIRTHPLACE OF ACTOR)
(ANSWER IS LOC1)
Clarksburg, West Virginia

When was Grace Vance born?
(QUESTION IS ((<=> ($BIRTH ACTOR HUM1)) TIME (*?*)))
(QUESTION TYPE IS time)
(SEARCHING BIRTH SCRIPT FOR ANSWER)
(SEARCHING FOR BIRTHDATE OF ACTOR)
(CALCULATING AGE OF ACTOR)
(ANSWER IS TIM100)
Approximately 1920.
```

Other questions about time are not answered quite so easily. For instance, "When was Vance working in Cyprus?" could be narrowed down through inferences to his career era or his political career era, but then all events in those eras would have to be searched for the answer. When a matching event was found, the question answerer would have to determine how to answer the question -- with respect to world events, with respect to other things going on in Vance's life at the time, with respect to other events in Vance's political life, or with real time. Often the time representation attached to the event solves that problem. For instance, the time slot for Vance's Cyprus visit includes the date and a pointer to the major world event that was going on. If there is no specific date, however, the time response will have to be determined by the inferred intent of the question. A question about Vance's political career would be answered with respect to the major world event going on at the time or in relation to other political appointments he has had. A question about his family life would be answered in terms of his career since that is what we (and therefore CYRUS) focus on in thinking about Vance. Time questions asked about people other than Vance would be answered in terms of the part of that person's life that is considered most important or in terms of the person the questions are focused on. Thus, a good answer to the question "When did Vance's father die?" would be "When Vance was five years old." "When did Vance get married?" would be answered by "Soon after he became a lawyer." It is the intent of the question that determines how it should be answered. In answering questions about time in CYRUS, questions are usually answered with respect to the occupational era sequence.

Questions about place are answered in a similar way to time questions. As in the example above, if the match found to the event in the question does not have a place attached, then the time atom and the residence list on the person's token can be used to determine place. Any script or era mentioned in the question is always consulted first, however, before doing any general searches or calculations. Answering questions about place requires knowledge of intent and importance of activities. In answering the question "Where did Vance go to law school?", the law-school student script is consulted first. Processing rules from that script tell CYRUS that the question is asking for an identification of the law school Vance attended. It is therefore more appropriate to answer "Yale" than to answer "in New Haven". In answering the question "Where was Vance last week?", it must be determined that the best answer is the country he was in last week. A number of correct answers could be given, including "in an airplane", "in a hotel", "in Israel", and "in the universe", but only the third one is the relevant answer. Place must be determined in terms of an important activity Vance was doing, in this case a visit to Israel. Similarly, in answering the question "Where is Vance now?", the same type of processing must be done. If there is no information in the data-base about where Vance is at a particular time, then knowledge coming from the role theme he occupies at that time is used to supply an answer. For example, CYRUS has enough information about Secretaries of State to know that if he is not on a trip, then the Secretary of State is probably in his nation's capital, or at least in his own country. In Vance's case, this is the United States, and CYRUS can therefore answer "in the United States". Knowledge of Vance as somebody who travels a lot (since he is Secretary of State) tells CYRUS that the best way to answer questions asking for his whereabouts is usually to name the country he is in.

5. Verification questions

Verification questions are yes/no questions. In answering these questions, sometimes it is enough to search for a match in memory after the search space has been narrowed. However, most of the time that is not enough. More than a simple yes or no is needed to answer these questions, so further interpretation of the question is needed to determine what else is being asked for. For instance, if asked "Does he have any children?", an appropriate reinterpretation is: "Does he have any children, and if so, how many or what are their names?" If asked, "Is he married?", an appropriate way to reinterpret the question is: "Is he married, and if so, to whom?" In CYRUS, that information comes from knowledge about family relationships which can be found in the person frame. When a Verification question is asked about a variable in a script or a slot in the person frame, it is assumed that there should be some further specification about that variable. When a Verification question is asked about the existence of an instance of some script, further information should be given about that script. Processing family relationships is as follows:

> If the question type is Verification, and the unknown is KIDS, and the initial answer is "yes", then get the list of kids if it is known else the number of kids and append their names if known and there are less than four, else append their number to the initial answer.
>
> If the question type is Verification and the unknown part is spouse, and the initial answer is "yes" then append the answer to "Who did the actor marry?" if known to the initial answer, else append the answer to "When did the actor marry?" to the initial answer.

In a question such as "Has he ever been to France?", inferences off the primitive act PTRANS help to answer the question. Reinterpreting this question using available inferences, CYRUS understands the question as "Has he ever taken a trip to France?" and knows to look for instances of the TRIP script. Algorithms for processing in the TRIP script follow:

1. If the question type is verification and the initial answer is "yes", then append the answer to "When was the trip?" to the initial answer.

2. If the trip was part of a CAREER era or MILITARY SERVICE era, then append the answer to "What was the purpose of the trip?" to the answer. Otherwise, if the trip was before marriage, then append the answer to "Who else went on the trip?" to the answer.

If an answer is not found through checking instances of the TRIP script, then by using other inferences, the question can be reinterpreted as "Has he ever lived in France?", and the residence list of the person can be checked. If France were on the residence list, then CYRUS would include in the answer the information that the trip was for the purpose of residing in France.

6. Duration questions

Duration questions are those questions that ask for calculation of time duration. The questions "How long has Vance been Secretary of State?" and "How many years has it been since Vance became Secretary of State?" are both answered by finding the time when he began being Secretary of State, and subtracting it from the present date. "How long was Vance in the Mid-East?" is answered by getting the start time and the end time of his trip and subtracting. Notice that these are both the same calculation. In the first question, the end time was now. The rule for answering Duration questions is:

1. get the start time
2. get the end time
3. subtract the start time from the end time.

Many times, getting the start time or the end time of an event involves making a time calculation. For example, in answering "How long has it been since Vance graduated law school?", the question "When did Vance graduate law school?" (a Time question) must be answered to get the start time. If the data-base did not explicitly have that date, then the approximate time of Vance's graduation would be calculated as explained in the section (4) above by making inferences from when he graduated college or when he was born. Thus, processing associated with scripts and eras can be used for calculating start and end times of events and eras. Some scripts, such as the VIPVISIT script, have start and end times as script variables. When that is the case, Time questions do not have to be answered to calculate start and end times.

```
How long was Vance in Israel?
(QUESTION IS ((ACTOR HUM1 IS (*LOC* VAL POL1))
        TIME ((BEFORE *NOW* X)) DURATION (*?*)))
(QUESTION TYPE IS duration)
(MATCHED QSTAT TO $VIPVISIT)
(SEARCHING $VIPVISIT FOR ANSWER)
(CALCULATING DURATION FROM ARRTIME AND DEPTIME)
(ANSWER IS ((*DAYS* 6)))
Six days.
```

7. Occurrence, Result Orientation and Motivational questions

Occurrence questions ask what followed after an event, as in "What happened when Vance went to the Mid-East?" These questions are answered by finding the event mentioned in the question and following the causal chain from that event. Any forward causal chain connections are retrieved in answering these questions.

Result Orientation questions ask what resulted from a particular event, as in "What resulted from Vance's Mid-East visit?" These questions are answered by finding the event in the question and following LEADTO, REASON, INITIATE, and ENABLE links in the causal chain. The difference between processing for these questions and Occurrence questions is that Occurrence questions can also get their answers from NEXT links.

Motivational questions ask for the motivation or reason behind doing an act, as in "Why did Vance go to the Mid-East?" As in answering Occurrence and Result Orientation questions, CYRUS finds a match in the event list of the appropriate era, and then looks at the causal chain coming from that event. In the case of Motivational questions, CYRUS follows all COMESFROM links from the matched event to see if any of the COMESFROM links point to events which could have motivated the matched event. A COMESFROM link points backwards to events that are causally connected by REASON, INITIATE, or LEADTO connectives. An event that has a REASON or INITIATE connective pointing to the matched event will answer a Motivational question. In addition, some scripts usually have goals attached as script variables. The VIPVISIT almost always has a goal attached and VIPVISIT processing would retrieve it.

For any of the question types discussed above, if the answer cannot be found by the methods specified, then further inferences can be used. For example, the questions above are all instances of the VIPVISIT script, and knowledge of the VIPVISIT script tells CYRUS how to do further processing as follows:

> If the question type is Occurrence or Result Orientation, and an answer has not been found from the causal chain, then look in the world events timeline at the time corresponding to the trip, and see if anything significant happened in the part of the world where the actor was at that time. If so, use that for the answer.
>
> If the question type is Motivational, and an answer has not been found from the causal chain or attached to the script instance, then look at the world events timeline at the time corresponding to the trip, and see if there were any problems going on in that part of the world at that time. If so, use that as the answer.

Using the world events timeline to answer questions means that it must be continuously updated along with the rest of the data-base in order to answer questions correctly.

If all of these methods fail, Motivational questions can be answered by looking at role theme information. If the event in question is a typical one for the role theme the person is involved in, then CYRUS can use the reasons associated with that act in the role theme (if they exist). If there are no reasons associated, and it is an act typically done by somebody in the role theme, then it can use the default answer "because person x is in role theme y". For example, if asked "Why did Vance go to England?", CYRUS would determine that it was a VIPVISIT instance. If it could not find any reason for the visit, then its knowledge of the political dignitary role theme would tell CYRUS that political dignitaries normally go on VIPVISITs in order to improve relations between countries or for good will. Since CYRUS knows relations with England don't need improving, it assumes he went on a goodwill mission, and answers accordingly.

CONCLUSIONS

CYRUS is a data-base system designed for natural language inquiry. Natural language inquiry requires a conceptual data-base, since it is the intent of the questions, not necessarily the individual words used in asking the questions that is important. Question answering using conceptual information involves much more than the mere retrieval of information. Knowledge must be stored in a meaningful way so as to help with the retrieval process.

CYRUS makes use of a theory of human memory organization in order to store and retrieve information from its data-base. Memory is organized through knowledge structures called scripts, role themes, eras, and person frames. The question answering process consists of interpreting the question by finding the correct question category, and processing the question using inferences organized in these knowledge structures. This model of memory organization and retrieval has proven to be sufficient for dealing with knowledge of events and simple causations in the domain chosen. It can be easily extended to include other domains by adding new script and role theme structures, and new processing rules to the memory, without changing anything else in the system.

CYRUS is small, and there has been no need to refine low-level indexing schemes, such as organization of events within the eras. In creating a larger system, minimizing search within eras will become important. When FRUMP begins updating CYRUS from the UPI wire, there will be a large number of events put into Vance's current political career era. At that time, problems of indexing within eras, updating and debugging false or partially true information, deciding when to forget low-level details of stories, and representing and handling more complex time and time relations will need to be addressed in more detail. These problems, and others, are currently being worked on, and their solutions will be included in a complete CYRUS system, including both question answering and updating modules, which should be running within the next year.

ACKNOWLEDGEMENTS

I'd like to express my gratitude to Roger Schank, Wendy Lehnert, and the rest of the Yale AI Project, especially Warren Odom, for their help, ideas, and encouragement on this project.

REFERENCES

Collins, A., Warnock, E. H., Aiello, N., and Miller, M. L. (1975). Reasoning from incomplete knowledge. In Bobrow, D. G. and Collins, A., ed. Representation and Understanding. Academic Press, Inc., New York.

Cullingford, R. (1978). Script application: computer understanding of newspaper stories. Ph. D. thesis. Research Report #116. Department of Computer Science. Yale University, New Haven, CT.

DeJong, G. F. (1977). Skimming newspaper stories by computer. Research Report #104. Department of Computer Science. Yale University, New Haven, CT.

Goldman, N. (1975). Conceptual generation. In R. C. Schank, ed. Conceptual Information Processing. North Holland, Amsterdam.

Lehnert, W. G. (1978). The Process of Question Answering. Lawrence Erlbaum Press, Hillsdale, N.J.

McSkimin, J. R. and Minker, J. (1977). The use of a semantic network in a deductive question-answering system. In Proceedings of the 5th International Joint Conference on Artificial Intelligence - 1977. Cambridge, MA.

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

Petrick, S. R. (1975). On natural language based query systems. IBM Research Report RC 5577. Thomas J. Watson Research Center, Yorktown Heights, N.Y.

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

Schank, R. C. (1975). Conceptual Information Processing. North Holland, Amsterdam.

Schank, R. C. and Abelson, R. P. (1977). Scripts, Plans, Goals, and Understanding. Lawrence Erlbaum Press, Hillsdale, N.J.

Waltz, D. L. (1978). An English language question-answering system for a large relational database. In Communications of the ACM, July, 1978. ACM, New York.

Wilensky, R. (1978). Understanding goal-based stories. Ph. D. Thesis. Research Report (forthcoming). Department of Computer Science. Yale University, New Haven, CT.