DOCUMENT RESUME

ED 087 474 IR 000 220

AUTHOR	Grignetti, Mario C.; Warnock, Eleanor H.
TITLE	Mixed-Initiative Information System for Computer-Aided Training and Decision Making. Final Report.
INSTITUTION	Air Force Systems Command, L.G. Hanscom Field, Mass. Electronic Systems Div.; Bolt, Beranek and Newman, Inc., Cambridge, Mass.
REPORT NO	ESD-TR-73-290
PUB DATE	15 Sep 73
NOTE	70p.
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	Computer Assisted Instruction; Computer Oriented Programs; *Computer Science; *Decision Making; *Information Networks; *Information Systems; *Networks; Program Descriptions
IDENTIFIERS	Advanced Research Projects Administration; ARPA; ARPANET; ARPA Network; *NET SCHOLAR

ABSTRACT

A description of the NET-SCHOLAR system, an on-line aid for naive users of the Advanced Research Projects Administration (ARPA) Computer Network, is provided. The discussion focuses upon the system's representation and handling of functional and procedural information and its ability to deal with action verbs, all within the context of the ARPA network. Section I presents a brief introduction to the system and Section II a series of tutorial questions, the answers to which clarify the need to work with action verbs. These verbs are then classified and a taxonomy of questions constructed, providing the basis for the next phase of the discussion. Section III describes the method used to encode the meaning of operations, procedures, etc., while Section IV presents a predecessor of NET-SCHOLAR used temporarily while NET-SCHOLAR was being built. Section V describes NET-SCHOLAR itself, with emphasis upon the system's English comprehension capabilities. The description is illustrated with actual protocols, and a summary and conclusions are presented. (Author/PB)

ESD-TR-73-290

MIXED-INITIATIVE INFORMATION SYSTEM FOR COMPUTER-AIDED TRAINING AND DECISION MAKING

Mario C. Grignetti
Eleanor H. Warnock
Bolt, Beranek and Newman, Inc.
50 Moulton Street
Cambridge, MA 02138

15 September 1973

Approved for public release; distribution unlimited.

Prepared for

DEPUTY FOR COMMAND AND MANAGEMENT SYSTEMS
HQ ELECTRONIC SYSTEMS DIVISION (AFSC)
L. G. Hanscom Field, Bedford, Massachusetts 01730

LEGAL NOTICE

When U.S. Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

OTHER NOTICES

Do not return this copy. Retain or destroy.

This technical report has been reviewed and is approved for publication.

Sylvia R. Mayer
SYLVIA R. MAYER, GS-14
Project Scientist

Sylvia R. Mayer
SYLVIA R. MAYER, GS-14
Task Scientist

FOR THE COMMANDER

Robert W. O'Keefe
ROBERT W. O'KEEFE, Lt Col, USAF
Director, Information Systems Technology
Deputy for Command and Management Systems

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-73-290	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (*and Subtitle*) MIXED-INITIATIVE INFORMATION SYSTEM FOR COMPUTER-AIDED TRAINING AND DECISION MAKING		5. TYPE OF REPORT & PERIOD COVERED Final Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Mario C. Grignetti Eleanor H. Warnock		8. CONTRACT OR GRANT NUMBER(s) F19628-72-C-0163
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bolt, Beranek and Newman, Inc. 50 Moulton Street Cambridge, MA 02138		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS PE 62702F PROJECT 2801 TASK 2801.04.1
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy for Command and Management Systems Hq Electronic Systems Division (AFSC) L G Hanscom Field, Bedford, MA 01730		12. REPORT DATE 15 September 1973 13. NUMBER OF PAGES 67
14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)		15. SECURITY CLASS. (*of this report*) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

Approved for public release; distribution unlimited.

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

18. SUPPLEMENTARY NOTES

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

Computer-assisted instruction, Artificial intelligence, Man-computer interaction, natural language comprehension, semantic networks, procedural knowledge representation, on-line aids.

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

We describe a SCHOLAR system, NET-SCHOLAR, designed to help people use the ARPA Computer Network and the facilities available through it. NET-SCHOLAR is capable of answering questions dealing with descriptional, functional, and procedural information. Previous SCHOLAR systems dealt only with static information, within the context of Geography. The report includes actual on-line protocols.

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

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

Preface

The work described in this report is the result of a joint effort involving several staff members at BBN. Mrs. Nelleke Aiello wrote the output routines that convert data base representations into English text. Miss Eleanor Warnock designed and implemented the English Comprehension System that is the back bone of NET-SCHOLAR. She also wrote Section 5 of the report. Dr. Allan Collins provided insight and ideas. Without his knowledge and long experience in working with SCHOLAR systems, the present effort would not have succeeded. The late Jaime R. Carbonell originated this work and was its principal investigator until his untimely death, in February 1973. Mr. Grignetti worked under the supervision of Dr. Carbonell, and in February 1973 became principal investigator. He wrote the first 4 Sections of this report. Other staff members that participated were Gregory Harris and Jaime G. Carbonell Jr.

TABLE OF CONTENTS

Section 1 INTRODUCTION

This is the Final Report required under Contract No. F19628-72-C-0163, an effort to develop an on-line aid for naive users of the ARPA Computer Network. We have named this aid NET-SCHOLAR, since it is a SCHOLAR-like system [Carbonell, 1970; Carbonell and Collins, 1970, 1972] oriented towards helping people how to use the ARPA NETwork (ARPANET) [Heart et al, 1970; Crocker, 1972] and to make use of the computer facilities available through it. The original SCHOLAR system, oriented towards tutoring the geography of South America, has been renamed GEO-SCHOLAR.

The most important difference between a SCHOLAR system designed for the tutoring of Geography and a SCHOLAR system aimed at helping users of a computer network is the latter's necessity to deal with procedural and functional information. This, in turn, requires that the system be able to handle action verbs.

Most of the questions that a student asks in the context of Geography can be asked comfortably without using action verbs, because such questions deal mostly with static information. But when the subject becomes how to use the ARPANET, such a restriction is no longer acceptable. It becomes imperative to be able to handle questions such as 'How do I save a file in BBN-TENEX' or 'How does one edit text with TECO.' The work reported in this document is therefore centered around the representation and handling of functional and procedural information, within the context of the ARPA network.

What are the characteristics of this type of information? To gain some insight into this question, we conducted a number of tutoring sessions, involving a knowledgeable user of the ARPANET tutoring a naive (with respect to computer networks) computer user. The results are described in Section 2. The paramount importance of being able to handle action verbs, operations, purposes, and procedures became absolutely clear. Verbs were classified and a taxonomy of questions was constructed, providing the basis for the next phase of our work.

In Section 3 we describe our method for encoding the meaning of actions, operations, purposes, procedures, etc. which is based on the use of a suitably modified case grammar, 'a la' Fillmore [Fillmore, 1968].

Section 4 describes a predecessor of NET-SCHOLAR. This temporary system (TMPNET-SCHOLAR) is not able to comprehend questions involving action verbs, but once a question is paraphrased accordingly, it produces an answer that is comparable, in its use of verbs, to that of the target system. TMPNET-SCHOLAR came into being because we needed a system with which to test and shake down the data base while the English comprehension routines of NET-SCHOLAR were being built.

In Section 5, we describe NET-SCHOLAR proper, with its English comprehension capabilities developed to the point of understanding simple questions (no subordinate clauses, for example) involving action verbs. The description is illustrated with actual protocols. A Summary and Conclusions Section closes the Report.

As a closing note, we would like to emphasize that NET-SCHOLAR is by no means a finished product, and that in our followup work we will continue to improve the capability of SCHOLAR systems to interact naturally with people.

Section 2 TUTORING PROTOCOLS

2.1 Introduction

In order to gain some insight into what kind of information a naive user would need to know in order to make intelligent use of the ARPANET, we decided to conduct some actual tutoring sessions in which a user familiar with TENEX but almost totally ignorant about computer networks was tutored in an interactive manner by another user well versed and knowledgeable on the ARPANET.

The format of the sessions was left entirely free. We asked each of two tutors to teach two naive members of BBN's staff (naive in the sense that they had never used any computer network before) how to use the ARPANET so that they might be able to accomplish some reasonable goal, like being able to talk to several hosts or to copy files through the network. The tutor and the student sat in front of a teletype and the conversation was taped and transcribed.

It is important to give an idea of what transpired during these tutorial sessions. To this end we present first a digest of the most important topics covered, how the tutor chose to present them and the questions that arose. The digest summarizes about 52 pages of transcript. Then we describe how we analyzed these transcripts and finally we present a classification of questions that users are likely to ask when they learn how to use the ARPANET.

2.2 Digest of one of the tutoring sessions

General description of the ARPANET

Topological aspects

Sites and hosts (local and remote).

Network Control Programs (NCP) and their function.

TELNET, the NCP of TENEX.

Operational description of TELNET.

Tutor logs in to BBN's TENEX system and calls TELNET.

LOCAL and REMOTE commands modes.

Tutor states their existence and gives a general idea of what they do.

Points out that LOCAL command mode is often called COMMAND mode.

Command recognition and other features designed to facilitate use.

Tutor shows use of abbreviations, help features, and how TELNET refuses to accept illegal characters.

CONNECTION.TO Command

Tutor demonstrates by connecting to SRI-ARC, a TENEX system located at Stanford Research Institute. This system is the home of the Network Information Center (NIC).

Logging in to a foreign host

Tutor logs in using appropriate identification and password. A mask is printed out, to hide the password instead of the expected echo expression. More about this later. Calls the NLS subsystem to demonstrate how to access information stored in the NIC.

Question: How do you break the connection?

Tutor demonstrates: First logs out of foreign host (student is reassured that this will not log out of local host). Then, types escape character (↑Z) and issues the disconnect command. Points out that disconnecting without logging out would cause the job to remain detached.

The question of echos

Referring to the problem of masked passwords, the tutor talks about FULL DUPLEX and HALF DUPLEX, and about LOCAL ECHO MODE and REMOTE ECHO MODE.

Sets the echo mode to LOCAL.

The function of the escape character (↑Z)

i) Returns control to TELNET

ii) Sets the command mode to COMMAND

iii) Returns to REMOTE command mode upon terminating a command.

Talking to two hosts at the same time

The tutor proposes this as a further demonstration of the use of the CONNECT command.

Questions: Did you disconnect from SRI yet? There is a difference between logging out and disconnecting, right? How many connections can you have?

Tutor connects to RAND-SRC (the TENEX system at RAND Corporation) and logs in. Then he returns to TELNET (in BBN) and connects to BBN (through the network). This is confusing to the student, so the tutor describes two commands that may help in situations such as the present: LIST CONNECTIONS, RETRIEVE CONNECTION (sets user in the desired connection), and NET STATUS (it looks to see which connections are going out of each socket)*

Question: What is a socket?

Description of sockets (socket numbers)

Tutor explains what they a e by analogy with jobs. "Job numbers are just a way of talking about (identifying) particular interactions. In the same manner, socket numbers are the way hosts have of making sure data is channeled right." Points out that when a socket in a host is made to correspond with another socket in another (or the same) host, a connection is made. The subject is complex and there is evidence of confusion in the student's mind.

Question: I still do not understand what you mean by socket. Why do you need four sockets for a connection? Why are socket numbers supposed to apply to job numbers or user numbers?

The tutor proposes to clear up confusion by way of example-- she is going to start two programs (each in a different host) that will talk to each other. This exercise proves to be useful but many doubts remain.

---

*A new command (WHERE.AM.I) has been added recently to TELNET to cope with situations such as the one described.

Tutor and student agree on adjourning and reconvening the following day.

2.3 Analysis of transcripts

As a first pass in analyzing the transcripts, we set out to investigate how important action verbs are in dealing with this kind of subject matter, the extent to which they are used, and in what context. By action verbs we mean all verbs except the verb <u>have</u> and the linking verb <u>be</u> (no other linking verb appeared on the transcripts).

We began by extracting all the action verb phrases that appeared in one of the transcripts, and we catalogued them as follows: for each new verb phrase that we encountered we made an entry on a card headed by the main constituent of the verb phrase, a classification of its type, and a back pointer to where it occured in the transcript. For example, the first time the verb phrase "have to remember" occurred, we entered it on a card headed "remember", and classified it as an 'aux-Vrb, Pres' verb phrase meaning that its Tense is present and that the phrase is constructed with 'have to ' as an Auxiliary (or Modifier). Thereon, all repetitions of this same verb phrase were ignored.

In this manner we catalogued approximately 280 different verb phrases of 148 different main constituents (action verbs). There are at least two reasons for this relatively high number of verbs. Firstly, verbs were indexed according to their meaning, and not according to their spelling. For example, we had 2 different cards headed by the verb call -- to call [summon] a program (as in "Call TELNET"), and to call [name] a file, a connection, or a host, (as in "The host of Stanford Research Institute is called SRI-ARC") -- and 5 cards headed by the verb go -- they convey the same meaning as disappear ("it is gone"), traverse ("bytes going across the network"), explain ("we could go into why this is so") , exit or leave ("we must go out"), and undergo ("The NCP's go through a hand-shake procedure").

A second reason for the high number of different verbs encountered was the presence of synonyms or quasi-synonyms such as send and transmit; type in, give, issue, say, tell, (a command to the computer system); start, begin, initiate; stop, finish, end, terminate, complete; etc.

All this prompted us to attempt building a taxonomy of the verbs used in the transcript, to see if some kind of structure or hierarchy could be uncovered. To do this in a manner as free as possible, we just laid out the cards on a large table and set out to group them into clusters and then the clusters into superclusters, etc. At the beginning our criterion for grouping was completely unformulated -- they were just grouped, somehow. After several trials some criteria began to emerge -- there appeared to be physical action verbs, (denoting operations, functions and commands) and mental action verbs (addressing mental actions, intellectual functions, and states of mind). Ignoring the overlaps between these two main categories, let us designate the first as action verbs and the second as mind verbs. Within each of these main categories there were several levels of generality, in such a way that a verb in a high level somehow conveyed to us part of the meaning of several verbs in the level below. A frequently occurring feature was the presence of opposite pairs, like save vs. retrieve, start vs. stop, allow vs. limit, come vs. go, remember vs. forget, etc.

Without attempting to justify the procedure in any other ground than those alluded to above, we present below a classification of the verbs extracted from the transcript. We represent levels by degree of indentation, so that verbs with greater specificity of meaning (lesser generality) are indented more than verbs with less specific meaning. In order to save vertical space, we have strung out horizontally verbs that should be considered at the same level -- the criterion is that all verbs on a line have the degree of generality represented by the indentation of the first verb in the line. We have also enclosed in parenthesis synonyms, quasi-synonyms, and antonyms. Where appropriate, comments have been introduced to explain the criterion followed for that particular class.

Classification of action verbs

Verbs that address the action or are used in circumlocutions about the action
(DO, PERFORM, EXECUTE)

Verbs that create new wholes using parts
MAKE, (COMPOSE vs. SEPARATE)

Intent modifiers
HELP, TRY, UNDERGO, ORDER, COMMAND, USE

Affecting the state or course of the action
(HAPPEN, TAKE PLACE)
((START, BEGIN, INITIALIZE) vs. (STOP, FINISH, END, COMPLETE))
((INTERRUPT, BREAK) vs. CONTINUE)
(WAIT, TAKE TIME)
((ALLOW, LET) vs. LIMIT)

Specific action verbs

MOVE

((COME, ARRIVE) vs. (GO, LEAVE)), RETURN

PUT, LEAD vs. FOLLOW

Alluding to possession

(GIVE vs. TAKE), (RID vs. HOLD), KEEP, SET

Transmission of information

COMMUNICATE, INTERACT

((ASK, (ASK FOR, REQUIRE)), (GET, OBTAIN))

TALK ABOUT

((CALL, TALK TO, CONNECT) vs. DISCONNECT)

Applied to commands

(TYPE, TYPE IN) vs. TYPE OUT), PROVIDE, ISSUE GIVE, SAY, TELL)

Commands and their imperatives

(SAVE vs. RETRIEVE), (CALL IN, SUMMON)

((COPY, TRANSFER), ((SEND, TRANSMIT) vs. RECEIVE))

(LIST, PRINT OUT), RUN

((LOG ON, LOG IN) vs. (LOG OUT, GET OUT, EXIT))

SPECIFY, (CALL, NAME), ECHO

PERCEIVE

(SEE, LOOK), (HEAR, LISTEN TO)

Classification of Mind Verbs

THINK, REALIZE, MEAN

(LEARN, KNOW)

REMEMBER vs. FORGET)

DECIDE, (SHOW, DEMONSTRATE), RECOGNIZE

ESTABLISH

(ASSUME, SUPPOSE), (GUESS, MAKE UP)

ACCEPT, (THINK, BELIEVE)

((CONFUSE (CONCERN, WORRY)) vs. (EXPLAIN UNDERSTAND))

REFER, (CORRESPOND, PERTAIN)

COMPARE

RESEMBLE, DISTINGUISH, EQUATE

This initial classification helped us identifying an important class of representations for the encoding of verbs in a semantic network, as we shall see in Section 3.

2.4 The question of questions

In order for NET-SCHOLAR to be a valuable Question Answering System, it must be able to answer most of the questions posed in real life by most naive users of the ARPANET. Since the set of questions asked during the tutorial sessions provide a starting basis for the kind of information the system will have to process and interpret, we set out to extract and classify them. The following list of questions was put together by abstracting representative ones from the transcripts and by adding a few of our own.

LIST OF QUESTIONS

1. Is <host name> up?
2. Can I transfer files through the network?
3. How do I interrupt (break) a command in TELNET?
4. What is the command to insert a string using TECO?
5. How do I log in to a host using the network?
6. What is the meaning of "SETTINGS LOADED" in TELNET?
7. What is the purpose of the command ; Y FILENAME in TECO?
8. How do I read in a file for editing in TECO?
9. How do I edit at BBN?
10. Where am I now?

In terms of pragmatics, these questions can be classified in the following four types:

1. State of... question

Examples:

1a. Is SRI up?

1b. Where am I now?

2. Means or methods for performing an action

Prototype:

Does X	action verb
Can I	do (noun of action -object-prep phrase.
How do I	

Examples:

2a. Does ;Y read in files in TECO?

2b. Can I transfer files over the network?

2c. How does one interrupt a command in TELNET?

2d. How do I edit at BBN?

3. Addressing unnamed actions or methods

Prototype:

What is	command / method / way	action verb / related to / meaning of / intended but unnamed action	object-prep phrase / state of...
	Does X / Can I / How do I		

Examples:

3a. What is the command for inserting a string in TECO?

3b. How do I get out of remote mode in TELNET?

3c. How do I find out if SRI is up?

4. Purpose of...

Prototype:

What is	use of / meaning of / purpose of	-object-prep phrase.

Examples:

4a. What is the meaning of the comment "SETTINGS LOADED" in TELNET?

4b. What is the purpose of the command ;Y FILENAME in TECO?

Undoubtedly, there are many more types of questions that we have not included (some of them deliberately) in this preliminary classification. However, the object of this part of our work was to provide us with a framework, an initial basis from which to start and proceed to build a data base with the features that would enable NET-SCHOLAR to deal with these types of questions.

Section 3 FUNCTIONAL AND PROCEDURAL REPRESENTATIONS

3.1 Introduction

Most of the questions that students ask about the geography of South America can be asked comfortably without resorting to action verbs. This is because such questions deal mostly with static information, for which the use of action verbs is seldom necessary. For this type of questioning descriptive representations of knowledge about static reality -- such as are used in the data base of GEO-SCHOLAR -- are adequate.

The evidence presented in Section II however, points out the need for a capability to deal with functional and procedural information, in addition to static information, when the subject shifts from geography to how to use the ARPANET. This, in turn, requires a system capable of handling questions involving action verbs.

Talking about computer memories, for example, we may say that they are hardware devices, that they are a part of every computer , that they may have certain attributes (such as size, word length, access time, data transfer rate, etc.), that cores, drums, disks, etc. are some examples of memories, etc. This type of knowledge can be represented in a data base 'a la' GEO-SCHOLAR. But the facts, for example, that memories store data, or that computer users save files in disks, would not be representable in such data base. We need, therefore, to have a capability for representing functional and procedural knowledge* (what memories are used for and how they are used for that purpose), which in turn requires a capability for representing the meaning of action verbs.

In what follows we begin with a brief review of the classical grammar of action verbs, followed by a description of case grammar. [Fillmore, 1968]. This is necessary since Fillmore's ideas provided the theoretical framework in which our representation of verb meanings is based. Then we describe how we actually implemented these ideas and how we incorporated case-structure descriptions in our Semantic Network.

3.2 Action verbs

A commonly accepted way of defining verbs is that they are words used to assert or express action, state or

---

*A parallel study [Collins, et al, 1973] addresses the problem of how to teach procedural knowledge.

condition, or being.

Verbs that express condition or being, such as be, seem, grow, feel, become, appear, are also called linking verbs because they merely link the subject with a predicate complement -- these verbs subordinate their meaning to the function of connecting the predicate idea with the subject. One of these verbs, be, and the verb have in its possessive and relational meanings , are the only verbs needed by SCHOLAR in the geography application. Therefore, we can say that handling verbs is, generally speaking, an entirely new problem for SCHOLAR-like systems.

3.2.1 Syntactic descriptions

Let us begin by reviewing the fundamentals of classical or traditional grammar that are applicable to verbs. The verb related a subject (a being, force or instrument that instigates or causes the action or state identified by the verb) to a direct object that receives or is the consequence of such action. Sometimes the meaning of a verb is clear without a direct object. An example is:

s1 The computer crashed.

But most times a direct object is needed -- in general, action verbs are associated with direct objects, and the verbs are called transitive.

s2 FTP transfers files through the network

Some action verbs (give, being, show, ask, tell, etc.) are often followed by a second object (the indirect object) to name the person or thing for whom the action is performed.

s3 Show me a list of TELNET commands

We said before that verbs were words used to assert or express action, condition or state, or being. Actually, instead of a single word, we may have verb phrases, or phrasal verbs denoting different aspects, moods, and time of occurrence of the action (Ex. should have been told).

Verb phrases can be classified in eight possible forms that can be conveniently represented as follows [Chomsky 1965, page 43]

Verb Phrase → Aux + Verb

Aux → Tense (Modal) (Perfect) (Progressive)

The meaning of the notation is that Tense is always present, while Modality and Perfect or Progressive aspects may or may not be present. Time of occurrence of the action (effected in the past, future, extending into the future, etc.) as well as its possibility, its optional or doubtful character, its necessity, etc. can be very precisely encoded in this manner.

Certain verb forms are no verbs at all. Active and passive participles and participial phrases are often used as adjectives, gerunds are used as nouns, and infinitives can be used as nouns, adjectives or adverbs. Examples of active and passive participles used as adjectives are:

the rotating disk

the connecting link

the stored data

the used time

When participial phrases are used as adjectives, they can have modifiers and objects of their own.

s4 Linking computers throughout the continent, the ARPANET serves a valuable function.

Gerunds are ing forms of verbs that are used as nouns. In s5, the object is a gerund phrase.

s5 Try connecting to SRI-ARC

When infinitives are used as nouns, they can often be interchanged with gerunds.

s6 Try to connect to SRI-ARC

When used as adjectives or adverbs, infinitives can usually be interchanged with "for V-ing".

s7 Devices are used to transfer data

adjective phrase

(modifies devices)

s7a Devices are used for transferring data

The "to" may be elliptic after such verbs as help or make. Certain properties, features and additives that characterize the behavior of certain verbs must be taken into account in the syntactic descriptions. What follows is a "potpourri" of such facts. As such, it is not intended to be exhaustive but only illustrative of the complexity of the problem and of the wide range of idiosyncracies exhibited by common verbs.

Certain transitive verbs may or may not lend themselves to intransitiveness (i.e., object deletion). While read applies to a narrow range of objects and can be employed intransitively, keep applies to a much wider range of objects and cannot be used intransitively.

Certain verbs, such as own, denote an action that tends to prolong itself indefinitely in time. Since the progressive aspect denotes precisely that (i.e., continuing action), it is not surprising to realize that own does not lend itself to take progressive form -- indeed such form would be redundant.

Verbs can be classified according to the class of subjects and objects that they will take, such as abstractedness of subjects or animatedness of objects. For example, only time-intervals elapse, and only physical objects move, (not programs, sincerity, or chess). Other verb categorizations can be made according to a) whether or not they take a "like" predicate, such as in

s9 The Tenex Monitor handles the EXEC like an ordinary user program,

or, b) according to the type of Prepositional phrases they take, such as at, into, after, for, up following the verb look. In this case prepositions carry meaning; i.e., look at (direct one's gaze), look for (seek), look into (examine), etc.

3.2.2 Fillmore's case grammar

One way of encompassing many of the facts about verbs we just finished describing has been proposed by Fillmore [Fillmore, 1968]. Since we have adopted many of his ideas in the construction of our data base, a brief summary is therefore in order.

Fillmore takes the view that a simple sentence is constituted by a verb plus several noun phrases. Each noun phrase is associated with the verb in a particular case relationship, and each case relationship occurs only once (except where conjunction occurs).

The array of cases in any given simple sentence typifies the sentence, and verbs can be classified according to the sentence type they may be plugged in.

The case relationships are:

1) Agentive (A). The instigator (typically animated, but not necessarily so) of the action identified by the verb

2) Instrumental (I). The inanimate force or object that is the cause of the action

3) Dative (D). The animate being affected by the action

4) Factitive (F). The object or being <u>resulting</u> from the action of the verb

5) Locative (L). Where did the action happen. Also in what orientation

6) Objective (O,S). The thing that is affected by the action. It may be a subordinated sentence (S).

This is a non-exhaustive list -- Time and Benefactive cases appear to be necessary additions, and many others have also been proposed.

Subjects, objects, and prepositional phrases are formed by rules patterned after sentence types. For example, in an A, O active sentence, A is the subject and O is the object; in an I, O active sentence, I is the subject; in an A, I, O, active sentence, A is the subject, O the object, and I is appended in a prepositional phrase beginning with "with". If neither A nor I are present, the O is the subject.

Verbs and nouns are plugged into the case frame (array of cases) of a sentence according to their lexical features. Thus, for example, animated nouns can be A or D, but abstract nouns cannot be L. Lexical features for verbs are represented in a manner similar to sentence types. For example, the sentence

s1Ø Users edit text with TECO

has a case frame [A,O,I], and for the verb edit the list [(A), (O), (I)] represents that case frame. The parentheses denote options. Other examples are open [O, (L), (A)] with an obligatory O, and logout [D, (L) (A)].

In addition, verbs are classified:

a) by ideosyncratic choices of particular nouns as subjects or objects, overruling what is determined by the general rules,

b) by the choice of preposition associated with each case relationship, and

c) by the choice of complementizer (that, -ing, for - to - for verbs taking the S case).

As examples of a) above, the verbs belong, want, and think, are accepted by the frame [O,D] but belong takes O as subject while want and think take D as subject.

The prepositions generally associated with each case are as follows:

A--by

I--by if no A, else with

O,F--none

D--to

L--on, at, in, etc. depending on noun and verb.

Preposition deletion (and sometimes substitution) is a frequent grammatical transformation (e.g., deletion of by when the agentive is in the subject position).

3.3 Semantic networks for action verbs

Let us now see how the actual representation of actions and purposes may be implemented in the data base. The reader is assumed to be familiar with the structure and the format of the data base for GEO-SCHOLAR. [Carbonell, 1970; Carbonell and Collins, 1970, 1972]

We begin with representations of action verbs. At least 5 kinds of information appear to be necessary--syntactical, case-structural, procedural, and ontological information, and information about conditions and requirements. These categories apply to verbs in general, but not necessarily to each particular verb. Let us examine them.

1) SYNTAX Information

The heading (the unit's name) is the infinitive form of the verb, and is followed by a set of descriptors. The first descriptor is VRB, followed by synonyms or other verbs of closely similar meaning. This is the only syntactic descriptor incorporated in the version of NET-SCHOLAR to be described in Section 5. In future SCHOLAR systems of this type we may incorporate more descriptors. A second descriptor could be a list of the conjugation irregularities the verb may exhibit,if any.

1st descriptor 2nd descriptor

Example: WRITE ((VRB WRITE) ((CONJIRR WROTE WRITTEN)...)...

A third descriptor could be a list of case frames, 'a la' Fillmore, of the sentence types into which the verb can be plugged in.

Example:

WRITE [(...)(...)(OBJ AGENT (OPTION LOC))

meaning that OBJ and AGENT (Objective and Agentive) are required case-relationships, while the locative case relationship (LOC) may or may not be specified.

An important observation to be made at this point is the restriction in meaning of the verb <u>write</u> that is implicit in the choice of this second descriptor. We are concerned, in the context of tutoring naive users of the ARPANET, neither with producing novels or plays as writers do, which would make OBJ optional, nor with the process of forming letters or symbols, which would require the Instrument case-- we restrict the meaning of the verb to the setting down of information so as to make it possible to read, as when writing programs or writing files on magnetic tapes.

2) CASE Information

By this we mean a characterization of the noun phrases that, with the given verb, form sentences that are both meaningful and relevant. We do this by listing the nouns or the most general class of nouns that 'agree' with the given verb under the various case relationships.

Examples:

Under DELETE

(OBJ NIL FILE JOB DATA)

(AGENT NIL USER)

(INSTR NIL PROGRAM COMMAND COMPUTER/SYSTEM)

Observe that instead of listing CHARACTER, STRING, WORD, etc. that 'agree' with DELETE in the objective case relationship, we have used their SUPERC, namely DATA.

3) Conditions and Requirements

Sometimes, we want to represent certain conditions and/or requirements that must prevail so that the action identified by a verb may take place.

For example, the computer system may recognize a command after the user has typed in the first few characters of the command, i.e.: the computer prints the rest of the command. In order for this to take place, the user must have typed in enough characters so that the command can be uniquely identified. We represent this type of information as follows:

Under RESPONSE (Under RECOGNITION MODE)

```
(PRINT NIL ( AGENT NIL COMPUTER/SYSTEM)
           (OBJ NIL (COMMAND NIL WHOLE))
           (CONDITION NIL
                  (TYPE NIL (AGENT NIL USER)
                        (OBJ NIL
                             (CHARACTERS NIL SUFFICIENT)))))
```

4) Procedural information

When we say

"The computer system recognizes the intended command for the user by means of the recognition mode."

we specify an instrument (recognition mode) for the action of the computer recognizing a command for the benefit of the user. However, naming the instrument may not be enough for a user to go ahead and use effectively the information. He may need some "how to" or procedural information, to amplify the instrumental information. Such procedural information can be paraphrased (continuing our example),

"by the user's typing the initial string of the command, followed by altmode*"

---

*Typing the Altmode character, activates the recognition mode of TENEX.

We represent such information as follows:

```
Under PURPOSE (Under RECOGNITION MODE)
(RECOGNIZE NIL
     (AGENT NIL COMPUTER/SYSTEM)
     (BENEF NIL USER)
     (INSTR NIL RECOGNITION/MODE)
     (OBJ NIL (COMMAND NIL INTENDED))
     (PROCEDURE NIL
          (TYPE NIL
               (AGENT NIL USER)
               (INSTR NIL RECOGNITION/MODE)
               (OBJ NIL ($SEQ
                    (STRING NIL INITIAL
                    (OF NIL COMMAND))
                    ALTMODE/CHARACTER)))))
```

The attribute $SEQ specifies that the actions listed under it must be performed successfully and successively, in the given order.

5) Other Ontological Information

The key component of this part of the representation is a more general verb of which the given verb is an instance, embedded in the standard SUPERC format of SCHOLAR. At a lower level of relevancy there is a definition of the action.

Examples:

Under WRITE

(SUPERC NIL COPY (STORE (I 3) (ON NIL MEMORY PERIPHERAL)))

Under INSERT

(SUPERC NIL EDIT (PUT (I 3) WITHIN))

Meaning can also be carried by verbs conveying the opposite action and by nouns or adjectives closely related to the verb.

3.4 Semantic networks for functions and purposes

In addition to the descriptive information on the various components of the ARPANET and its hosts, which is handled in a manner very similar to the one in the geography data base, there are items describing what the unit does, or what it is used for, or what are its purposes. This information is stored under a special attribute that we shall call PURPOSE, and is represented by a list of action verbs, each followed by sublists of cases and characterizations of noun phrases. Within each PURPOSE there may be a PROCEDURE that explains how the indicated purpose, function, or operation can be accomplished.

Example:

Under TECO, (the most commonly used editing language in TENEX)

```
(PURPOSE NIL (EDIT NIL
     (AGENT NIL USER)
     (OBJ NIL SYMBOLIC/DATA)
     (INSTR NIL TECO)
     (PROCEDURE (I 2)
          (($L DELETE INSERT) NIL
          (AGENT NIL USER)
          (OBJ NIL STRING)
          (INSTR NIL TECO\COMMAND]
```

The information encoded above could be paraphrased in several ways. From the information directly under PURPOSE we could derive:

s11 TECO edits symbolic data

s12 Users edit symbolic data with (or using or by means of) TECO

The information under PROCEDURES could be paraphrased as follows:

s13 To edit, TECO commands delete and insert strings.

s14 To edit, users delete and insert strings with (or using or by means of) TECO commands.

These sentences are constructed in the same manner as s11 and s12-- the only difference is the affixation of the explicative "to edit,".

In Section 5, we shall present more examples of encodings of the general types we have described.

Section 4 BUILDING THE DATA BASE

In the previous Sections we demonstrated the importance of being able to handle action verbs in order to deal effectively with procedural information, and we proposed a method for representing the meaning of action verbs. This entailed the design and implementation of an essentially new SCHOLAR system, having the ability to comprehend questions formulated with action verbs, and the building of a data base of information about the ARPANET. The system that emerged from this work is called NET-SCHOLAR and is described in Section 5.

Building a data base covering significant aspects of the ARPANET is in itself a large, time consuming undertaking. Not only a large amount of information must be gathered, digested, condensed, and represented within the structural constraints of semantic networks, but the interdependency of the units of information that constitute the data base make the task of building it a highly interrelated one. It was therefore essential to have a system of routines with which to exercise the data as the semantic network was built, without waiting for the NET-SCHOLAR routines to be available. The solution that we adopted was to develop a temporary, provisional system that we called TMPNET-SCHOLAR.

TMPNET-SCHOLAR consists of slightly modified GEO-SCHOLAR routines and a data base that is easily transformable into a NET-SCHOLAR data base.

Consider, for example, the representation of

"The CTRL-A command deletes the last character in the input string"

In the NET-SCHOLAR data base, such information is represented as follows:

```
Under CTRL-A/COMMAND
     (PURPOSE NIL
          (DELETE NIL
               (INSTR NIL CTRL-A/COMMAND)
               (OBJ NIL (CHARACTER NIL LAST))
               (LOC NIL (IN NIL INPUT/STRING))))
```

while in the TMPNET-SCHOLAR data base it appears as

```
     (PURPOSE NIL
          (DELETE NIL
               (CHARACTER NIL LAST)
               (IN NIL INPUT/STRING)))
```

TMPNET-SCHOLAR can not handle action verbs on input but gives the appearance of such ability in output. One can not ask "What does CTRL-A do?" or "What does CTRL-A delete," but one can ask "What is the purpose of CTRL-A" and get back the answer "The purpose of CRTL-A is to delete the last character in the input string". TMPNET-SCHOLAR can not even begin to compare, in usefulness and in freedom of expression, with what is achievable with NET-SCHOLAR. But with its help we were able to shake down a fairly sizable amount of information and construct a data base that, with relatively moderate effort, was converted to the NET-SCHOLAR format once the routines had been implemented.

The following protocol, obtained on-line with the TMPNET-SCHOLAR, gives an idea of its capabilities and limitations.

WHAT IS THE ARPANET?*

ARPANET IS A COMPUTER NETWORK.

TELL ME ABOUT COMPUTER NETWORKS

COMPUTER NETWORKS ARE A NETWORK OF THE INTERCONNECTED COMPUTER SYSTEMS.
THE STRUCTURE IS GRAPH LIKE.
THE NODES ARE THE COMPUTER SYSTEMS.
THE INTERMEDIATE NODES ARE THE MESSAGE PROCESSORS.
THE TERMINAL NODES ARE THE HOSTS.
THE LINK IS THE DATA LINES.

WHAT ARE THE HOSTS OF THE ARPANET?

THE MAIN HOSTS ARE BBN-TENEX, SRI-ARC, MULTICS, RAND-RCC, AND UTAH-10.

TELL ME SOMETHING ABOUT BBN-TENEX.

BBN-TENEX IS A COMPUTER SYSTEM AND A HOST.
THE COMPUTER IS THE MODIFIED PDP-10.
THE CORE OF THE MEMORY IS PAGED.
THE OPERATING SYSTEM IS TENEX.
THE MAIN SUBSYSTEM IS THE EXEC.
THE NETWORK CONTROL PROGRAM IS TELNET.

WHAT IS THE PURPOSE OF THE EXEC?

THE PURPOSE OF THE EXEC IS TO ENABLE THE USER TO INTERACT WITH TENEX.

TELL ME SOMETHING ABOUT THE COMMANDS OF THE EXEC.

THE KINDS OF THE COMMANDS OF THE EXEC ARE EDITING, ACCESSING, FILE HANDLING, AND PROGRAM CONTROLLING.

WHAT IS AN EDITING COMMAND?

(JUST A MINUTE ...) THE EDITING COMMAND IS TO EDIT.
THE PURPOSE IS TO CHANGE TEXT.

*

WHAT ARE THE EDITING COMMANDS OF THE EXEC?*

THE EDITING COMMANDS OF THE EXEC ARE THE CTRL-A COMMAND, THE CTRL-W COMMAND, THE CTRL-R COMMAND, AND THE CTRL-X COMMAND.

WHAT IS THE PURPOSE OF THE CTRL-A COMMAND?

THE PURPOSE OF THE CTRL-A COMMAND IS TO DELETE THE LAST CHARACTER IN THE INPUT STRING.

WHAT IS THE PROCEDURE?

THE PROCEDURE IS TO TYPE "CTRL" AND "A" AND SIMULTANEOUSLY.

WHAT IS THE RESPONSE?

THE RESPONSE IS TO PRINT "\" FOLLOWED BY THE DELETED CHARACTER.

TELL ME MORE ABOUT THE COMMADS\S\DNDS OF THE EXEC.

THE COMPONENTS OF THE COMMANDS OF THE EXEC ARE THE KEYWORD, THE ARGUMENTS, THE NOISE WORDS, AND THE SUBCOMMANDS.

THE INPUT MODES ARE THE LONG INPUT MODE, THE SHORT INPUT MODE, AND THE RECOGNITION MODE.

TELL ME ABOUT THE NOISE WORDS

THE NOISE WORDS ARE INSERTED FOR CLARITY.
THEY ARE ENCLOSED WITHIN PARENTHESES.
THEY ARE NOT ACTIVE COMPONENTS.

WHAT IS THE PURPOSE OF THE NOISE WORDS?

THE PURPOSES OF THE NOISE WORDS ARE TO FACILITATE THE USAGE OF THE COMMAND AND TO CUE THE USER ABOUT THE NEXT COMPONENT.

WHAT IS AN INPUT MODE?

AN INPUT MODE IS A MODE.
IT IS A WAY TO TYPE THE INPUT STRING.

TELL ME ABOUT THE RECOGNITION MODE OF THE EXEC.

THE RECOGNITION MODE OF THE EXEC IS BEST FOR THE INEXPERIENCED USERS.

THE PROCEDURE IS TO TYPE THE INITIAL STRING OF THE KEYWORD FOLLOWED BY THE ALTMODE COMMAND, EACH ARGUMENT FOLLOWED BY THE ALTMODE COMMAND, AND THE TERMINATOR.

THE RESPONSES ARE EITHER TO PRINT THE WHOLE COMPONENT AND THE NOISE WORDS IF THE USER TYPED THE SUFFICIENT CHARACTERS OR TO PRINT THE BELL CHARACTER TO PROMPT THE USER FOR MORE CHARACTERS.

WHAT IS THE PURPOSE OF THE ALTMODE COMMAND?

THE PURPOSE OF THE ALTMODE IS TO ACTIVATE THE RECOGNITION MODE OF THE EXEC.

TELL ME MORE ABOUT THE COMPONENTS OF THE COMMANDS OF THE EXEC.

THE FIRST COMPONENT OF THE COMMANDS OF THE EXEC IS THE KEYWORD.
IT IS THE NAME OF THE COMMAND.
THE DEFAULT IS "RUN".
THE NOISE WORDS ARE INSERTED FOR CLARITY.
THEY ARE ENCLOSED WITHIN PARENTHESES.
THE SUBCOMMANDS ARE OPTIONAL.
THEY ARE NOT ESSENTIAL.
THE STRUCTURE IS THE SEQUENCE OF FIELDS.

TELL ME MORE ABOUT THE STRUCTURE OF THE COMMANDS.

THE STRUCTURE OF THE COMMANDS IS THE SEQUENCE OF THE FIELDS.

EACH FIELD IS EITHER THE KEYWORD OR THE ARGUMENT FOLLOWED BY THE NOISE WORD.

WHAT IS THE CALLING PROCEU\UDURE FOR THE EXEC?

WAIT A MINUTE. I'M CHECKING MY FILES.

OK. LET'S CONTINUE.

THE CALLING PROCEDURE OF THE EXEC IS TO TYPE THE CTRL-C COMMAND.

TELL ME ABOUT THE NETWORK CONTROL PROGRAM OF BBN-TENEX.

THE NETWORK CONTROL PROGRAM OF BBN-TENEX IS TELNET.

TELL ME ABOUT TELNET.

TELNET IS A NETWORK CONTROL PROGRAM.
IT IS A SUBSYSTEM OF TENEX.
THE MAIN PURPOSES ARE TO MAKE THE CONNECTIONS AND TO PROVIDE THE INFORMATION ABOUT THE ARPANET TO THE USER.
THE MAIN COMPONENT IS THE COMMAND INTERPRETER.
THE ESCAPE CHARACTER IS CTRL-Z INITIALLY.
THE KINDS OF THE COMMANDS ARE ACCESSING, ECHO CONTROLLING, AND INFORMING.
THE INPUT MODE IS LOCAL AND REMOTE.

WHAT IS THE ENTERING PROCEDURE?

THE ENTERING PROCEDURE IS TO TYPE "TELNET" FOLLOWED BY THE TERMINATOR.

WHAT IS THE CONNECTING PROCEDURE?

THE CONNECTING PROCEDURE IS TO ISSUE THE ACCESSING COMMANDS.

TELL ME ABOUT THE ACCESSING COMMANDS.

THE ACCESSING COMMAND IS THE CONNECTION.TO COMMAND AND DISCONNECT\COMMAND.

WHAT IS THE PURPOSE OF THE CONNECTION.TO COMMAND?

THE PURPOSE OF THE CONNECTION.TO COMMAND IS TO CONNECT TO THE REMOTE HOST.

WHAT IS THE PROCEDURE?

THE PROCEDURE IS TO TYPE THE NAME OF THE HOST FOLLOWED BY THE TERMINATOR.

WHAT IS A REMOTE HOST?

THE REMOTE HOST IS AT THE REMOTE SITE.

TELL ME ABOUT THE COMMAND INTERPRETER OF TELNET.

THE PURPOSE OF THE COMMAND INTERPRETER IS TO HANDLE THE COMMANDS TYPED BY THE USER.
THE MODES ARE FANCY AND QUERY.
THEY ARE FREELY INTERMIXED.

TELL ME ABOUT THE QUERY MODE.

THE PURPOSE OF THE QUERY MODE IS TO FIND THE POSSIBLE CONTINUATION OF THE COMMAND.
THE RESPONSE IS ALL THE POSSIBLE CONTINUATIONS.
THE PROCEDURE IS TO TYPE "?".

TELL ME ABOUT THE FANCY MODE.

THE PURPOSE OF THE FANCY MODE IS TO PROCESS ONLY THE CORRECT CHARACTERS.

TELL ME MORE ABOUT THE FANCY MODE.

THE PURPOSE OF THE FANCY MODE IS TO PROCESS ONLY THE CORRECT CHARACTERS.
THE INCORRECT CHARACTERS ARE NOT ECHOED.
THE ECHO IS THE BELL CHARACTER INSTEAD.
THE COMMAND IS NOT ABORTED.

WHAT IS THE PURPOSE OF REMOTE MODE?

I'M AFRAID I CAN'T ANSWER YOUR QUESTION.

*TELL ME ALL YOU KNOW ABOUT R\RTHE REMOTE MODE OF TELNET *

I DON'T KNOW ENOUGH TO ANSWER YOU, SORRY.

TELL ME ALL ABOUT THE INPUT MODES OF TELNET.

THE INPUT MODES OF TELNET ARE LOCAL AND REMOTE.
THE PURPOSE OF THE LOCAL INPUT MODES IS
TO ACTIVATE THE COMMAND INTERPRETER.
IT IS TO ENABLE THE USER TO TALK TO TELNET.
THE ENTERING PROCEDURES ARE EITHER TO ISSUE
LOCAL.MODE\COMMAND OR TO TYPE THE ESCAPE CHARACTER.
THE PURPOSE OF THE REMOTE INPUT MODES IS
TO ENABLE THE USER TO TALK TO THE HOST CONNECTED.
THE ENTERING PROCEDURE IS TO ISSUE EITHER
THE CONNECTION.TO COMMAND OR THE
REMOTE.MODE COMMAND.

*

Section 5 DESCRIPTION OF NET-SCHOLAR

NET-SCHOLAR is a computer program to answer questions about the ARPA network. Like TMPNET-SCHOLAR, it is a SCHOLAR-type system with a data base in the form of a semantic network. It is different in that it has an ability to handle verbs and verb relations in understanding the user's questions and in formulating answers. Verbs are crucial in this system because most of what a user wants to know about the network is procedural in nature.

A case grammar representation (described in Section 3) is used for verbs. Cases, of course, do not have a one-to-one correspondence to surface-structure placement in sentences. For instance, in the sentence "The Ctrl-A command deletes a character", the Ctrl-A command is the instrument in the deleting, and in the sentence "I can delete a character with the Ctrl-A command", the Ctrl-A command is again the instrument, in spite of the fact that it is the subject in the one sentence and the object of a preposition in the other.

Some sample pieces of data base are shown in Figure 1. The DELETE section under CTRL-A\COMMAND gives information about what the Ctrl-A command deletes, using the standard cases of AGENT (filled by the noun "user"), INSTRument (filled by Ctrl-A command), OBJect (last character), and LOCative (input string). Similarly, the ENTER part of COMPUTER\SYSTEM tells how to enter a computer system, even giving a complicated PROCEDURE. (Notice that the procedure, in its turn, can have verbs, with their cases, embedded within it.) Purposes, conditions, side-effects, etc., are also stored in this framework.

In the verb entries (i.e., the DELETE entry itself, not the DELETE part of CTRL-A\COMMAND), the attribute "CASES" of the verb tells what kinds of concept nouns fit into the roles of agent, instrument, etc. This information is very general and is used internally in the processing of a question--in particular, in the case assignment--and is not directly used to print out an answer.

```
COMPUTER\SYSTEM
   (((CN COMPUTER\SYSTEM))
    (SUPERC NIL SYSTEM)
    (SUPERP NIL COMPUTER\CENTER)
    [ENTER (I 2)
           (AGENT NIL USER)
           (INSTR NIL ARPA\NETWORK)
           (OBJ NIL COMPUTER\SYSTEM)
           (PROCEDURE NIL
                     ($SEQ (CALL NIL (AGENT NIL USER)
                                     (OBJ NIL TELNET))
                           [TYPE NIL (AGENT NIL USER)
                                     (OBJ NIL (NAME NIL
                                                     (OF NIL
                                                  COMPUTER\SYSTEM]
                           (LOGIN NIL (AGENT NIL USER)
                                      (INSTR NIL LOGIN\COMMAND)
                                      (LOC NIL (TO NIL
                                                  COMPUTER\SYSTEM]
    (EXAMPLES (I 4)
              ($EOR MULTICS BBN-TENEX RAND-RCC SRI-ARC UTAH-10))

CTRL-A\COMMAND
   (((XN CTRL-A\COMMAND CTRL-A))
    (SUPERC NIL EDITING\COMMAND)
    (SUPERP NIL EXECUTIVE)
    (PURPOSE NIL (DELETE NIL (AGENT NIL USER)
                             (OBJ NIL (CHARACTER NIL LAST))
                             (INSTR NIL CTRL-A\COMMAND)
                             (LOC NIL INPUT\STRING)))

DELETE
   (((VRB DELETE))
    (SUPERC NIL EDIT)
    (CASES (I 6 B)
           (AGENT NIL USER)
           (OBJ NIL DATA FILE JOB)
           (INSTR NIL PROGRAMMING\LANGUAGE PROGRAM
                 COMPUTER\SYSTEM JSYS EDITING\COMMAND COMMAND]

ENTER
   (((VRB ENTER))
    (CASES (I 6 B)
           (AGENT NIL USER)
           (INSTR NIL COMMAND SUBSYSTEM COMPUTER\NETWORK)
           (OBJ NIL COMPUTER\SYSTEM OPERATING\SYSTEM]
```

Figure 1
Some Partial Data Base Entries in NET-SCHOLAR

```
WHAT COMMAND DELETES A CHARACTER
((NP INSTR (WHADJ WHAT) (CN COMMAND))
 (VP (VRB DELETE +S))
 (NP OBJ (DET A) (CN CHARACTER)))

HOW DO I ENTER SRI-ARC
((WHADV HOW)
 (AUX DO)
 (NP AGENT (PRN I))
 (VP (VRB ENTER))
 (NP OBJ (XN SRI-ARC)))

WHERE IS DATA STORED
((WHADV WHERE)
 (AUX BE +S)
 (NP OBJ (CN DATA))
 (VP (VRB STORE +PAST)))

TELL ME ABOUT THE TENEX EXEC CTRL-A COMMAND
((VP (VRB TELL\ME\ABOUT))
 (NP OBJ (DET THE) (XN TENEX) (XN EXECUTIVE) (XN
 CTRL-A\COMMAND)))

WITH WHAT PROGRAM CAN I ACCESS THE NETWORK
((PRP WITH)
 (NP INSTR (WHADJ WHAT) (CN PROGRAM))
 (AUX CAN)
 (NP AGENT (PRN I))
 (VP (VRB ACCESS))
 (NP OBJ (DET THE) (CN COMPUTER\NETWORK)))
```

Figure 2
Sentences After Parsing and Case Assignment

Net-Scholar's processing of a question is divided into four parts--parsing, case assignment, retrieval, and sentence-generation.

The first step is parsing. The parser is somewhat unsophisticated, but it is adequate for the purpose. It takes the input and builds a tree structure for the sentence, based on a restricted English grammar. It currently handles only simple constructions, e.g., no relative clauses. Noun phrases, though, are allowed to be somewhat complex, with adjectives, nouns, and prepositional phrases modifying the noun head. Some examples of parsed sentences are in Figure 2.

Case assignment takes the parsed sentence, finds the main verb, and figures out the relation of each noun phrase to it. The output looks like a parse tree, with the addition of a case label at the beginning of each noun pharse (NP) expression. In the first sentence in Figure 2, "what command" has been labelled as an instrument, and "a character" is an object.

Case assignnent bases its decisions mostly on semantics and heavily uses the match-on-superc subroutine. Match-on-superc compares any two concepts and decides whether they can be the same (e.g., character and information), or whether the two are distinct (e.g., character and computer). It goes up the superconcept chain (from character to data to information) for each of the two concepts and sees whether there is an intersection of the two chains. If there is no intersection, the two concepts are distinct; if one is directly on the superconcept chain of the other, then the two concepts coincide. If there is an intersection further up the chain (character and word both have the superconcept data), it is more complicated and further checks must be made. If the subroutine can't prove that they are different, it concludes that it doesn't know.

In assigning cases, a match-on-superc is tried between the noun head (the noun which the NP is about) in each NP of the sentence, and each noun in each case under the CASES of the verb in the data base. If there is a match--e.g., between "character" in the sentence and "data" in the OBJ case under DELETE--the case assignment routine takes note of the case (OBJ) and the word that matched (data) and continues on to try the others. A weight is also assigned based on the goodness of the match. For instance, "character" would match "character" perfectly, but a match with "data" is slightly less good, since characters are data but so are a lot of other things.

In addition to match-on-superc, case assignment uses syntactic clues, such as the presence of certain prepositions ("with" for an instrument) or the noun's position as surface object or surface subject of the sentence. It also uses match-on-superp (match on superpart, similar to match-on-superc), on locative and instrumental cases only. Eventually, all the semantic and syntactic possibilities are considered, their weights are compared, and the best case assignment is selected.

Now the sentence is ready for the retrieval process. For retrieval, the problem in a question like "What command deletes a character" is to find information somewhere in the data base that answers the question. The retrieval routine searches under each noun in the sentence ("command" and "character"), under the things that matched in the case assignment, examples of those things, and various other likely places, until it finds what it wants. At each place, it looks for the appropriate verb ("delete") and, if it finds it, matches (with match-on-superc and match-on-superp) the case nouns stored there against the case nouns in the sentence. For example (Figure 1), if it finds "delete" under "Ctrl-A\command", it matches "Ctrl-A\command" against "command" and "character" against "character". Here the match is good, and Net-Scholar answers as shown in the first example in Figure 3. If the match had been bad--e.g., if the information had been about deleting a word instead of about deleting a character--then the search would have continued.

WHAT COMMAND DELETES A CHARACTER

THE CTRL-A COMMAND DELETES THE LAST CHARACTER IN AN INPUT STRING.

HOW DO I ENTER SRI-ARC

THE USER ENTERS A COMPUTER SYSTEM USING THE ARPA NETWORK,
PROCEDURE: FIRST HE CALLS TELNET;
THEN HE TYPES A NAME OF A COMPUTER SYSTEM;
THEN HE LOGS IN TO A COMPUTER SYSTEM USING THE LOGIN COMMAND.

WHAT DOES TELNET DO

TELNET PROVIDES TO A USER INFORMATION ABOUT THE ARPA NETWORK;
OR IT MAKES CONNECTIONS WITHIN THE ARPA NETWORK.

CAN I ENTER BBN-TENEX WITH CTRL-C

THE USER ENTERS BBN-TENEX USING THE CTRL-C COMMAND.
PROCEDURE: HE TYPES A CTRL AND A C SIMULTANEOUSLY.

HOW DO YOU TRANSFER A FILE OVER THE NETWORK

FTP TRANSFERS FILES OVER THE ARPA NETWORK.

WHERE IS THE NETWORK INFORMATION CENTER

Figure 3
Actual Questions With Answers by NET-SCHOLAR

THE NETWORK INFORMATION CENTER IS AT SRI-ARC.

WHAT IS THE PURPOSE OF A COMPUTER SYSTEM

A COMPUTER SYSTEM PROCESS DATA AND PROGRAMS AUTOMATICALLY.

WHERE IS DATA STORED

THE USER STORES PROGRAM OR DATA IN FILES.

Figure 3(continued)
Actual Questions With Answers by NET-SCHOLAR

That is the basic procedure, but there are a lot of special things to handle. For instance, in the question "What command deletes a character", I don't want to be told "A command deletes a character", but rather "The Ctrl-A command deletes a character". That is, "command" is not an adequate match for "what command" and something more specific is needed. (This is a peculiar property of WH words, like "what".) Each question type has its own idiosyncracies about the way it wants retrieval to handle it. For example, a "how" question is asking for an instrument or a procedure, and a "tell me about" question just wants a slice of data base information about something.

Retrieval, of course, also has the task of evaluating complex noun phrases. This may involve the straight-forward searching for an attribute under an object, or the applying of any of a number of inferences.

When the information to answer a question has been found, all that remains is for the sentence-generation routine to put it into sentence form and print it out. Basically, the routine finds the primary verb, orders the cases for that verb, adjusts the subject and verb to be singular or plural, and puts in the necessary articles, prepositions, etc. When the piece of information is complex and embedded, several sentences may be made, as in the second example in Figure 3.

```
(CTRL-A\COMMAND NIL (DELETE NIL
                    (OBJ NIL (CHARACTER NIL LAST))
                    (INSTR NIL CTRL-A\COMMAND)
                    (LOC NIL INPUT\STRING]
```

"The CTRL-A command deletes the last character in an input string."

Figure 4
Example of Input and Output of Sentence Generation

In Figure 4, there is a sample piece of information and the sentence produced from it. DELETE is a regular verb in the cases it takes, and the elements present are ordered: INSTR + VERB + OBJ + LOC. If an AGENT had also been present, a different order would have been used. To the ordered list of elements, articles are added and modifiers are placed, as in "the last character", prepositions are added, "in an input string", the verb is made to agree, "deletes", and finally the sentence is printed "The Ctrl-A command deletes the last character in an input string."

Section 6 SUMMARY AND CONCLUSIONS

6.1 Summary.

We have constructed a SCHOLAR system, NET-SCHOLAR, capable of dealing with functional and procedural information within the context of the ARPA network. Previous SCHOLAR systems (referred to as GEO-SCHOLAR) dealt only with static information, within the context of the geography of South America.

NET-SCHOLAR incorporates several basic changes with respect to GEO-SCHOLAR, particularly in terms of handling routines and semantic network representations.

The most important class of new elements incorporated to the semantic network are action verbs. Following the work of Fillmore, our representation of the meaning of verbs is based on Case structures -- a characterization of the noun phrases that "agree" with the action verb under various Case relationships (such as Agent, Instrument, Object, etc) in the formation of meaningful and relevant simple sentences. An example would be "People (as Agents) activate programs (as Objects) with program activation commands (as Instruments)". Particular purposes or actions related to the various topics or objects represented in the data base incorporate instantiations of the nouns characterized globally in the Case structure (i.e. "Users call the EXEC with CTRL-C"). Procedural information is incorporated as means to achieve a purpose or action. Sometimes the specification of an Instrument is enough to teach a user how to perform a certain action (i.e.: call the EXEC). In most instances however, another action or sequence of actions must be specified. The following dialogue is self explanatory:

-How do I call the EXEC?

-You call the EXEC with CTRL-C

-How?

-By typing CTRL-C

-How?

-By depressing the CTRL key and the C key simultaneously.

In terms of handling routines, the most important new features of NET-SCHOLAR are the English Comprehension and the retrieval package.

The English Comprehender is designed to understand action sentences having a simple structure, that is sentences incorporating a main action verb but no subordinated sentences. We say that an action sentence is understood by NET-SCHOLAR when all its noun phrases have been assigned Case relationships with respect to the main verb. An example is "Users (as Agents) call the EXEC (as

Object) with CTRL-C (as Instrument)." The main new feature of the NET-SCHOLAR retrieval package is its ability to find a particular instantiation of an action verb in the data base, that matches a certain set of Case relationships.

An important class of questions that the new routines are designed to answer, can be conceived as sentences that point out a missing Case relationship with respect to the main verb. In this case, the retrieval routines of NET-SCHOLAR are designed to find an occurrence in the data base of the verbs such questions use, for which the nouns assigned to the various Cases by the English Comprehender are acceptable under such an occurrence. It all happens as if, for example, the question "How do I call the EXEC" were paraphrased as "Find an occurrence of the verb 'call' in the data base, for which 'I' is an acceptable Agent and 'EXEC' an acceptable Object, and give me the Instrument."

6.2 Conclusions

The present NET-SCHOLAR satisfies the design requirements outlined above, and to that extent it has been successful. Much remains to be done, however, to make systems of the NET-SCHOLAR type more useful to people. In the following we outline some of the areas in which we feel more work is necessary.

1) Further work in English comprehension is necessary to enable the system to understand hypotheses, causal explanations, and descriptions of situations as offered by users--unlike factual questions, these often involve complex sentence structures that are beyond the powers of the present system.

Also, the system would have a much more natural feel for its users if it were able to deal effectively with anaphoric references (pronoun references, for example) and with paraphrases (different ways of saying the same thing). The present NET-SCHOLAR is almost powerless in this regard.

2) In NET-SCHOLAR we have represented the meaning of actions and procedures by describing them in terms of other actions and procedures. This conceptual representation carries implicit with it the existence of an external "understander" that is capable of carrying out and executing the actions described in the procedural information. What we would like to add is a representation of actions as simulation programs, so that we may obtain the consequences of such actions by executing the simulation and examining the resultant computation. In this manner, we could avoid deducing the distant

side effects of an action, which is often very difficult, by actually simulating the action and observing the result.

3) The NET-SCHOLAR system as it now exists is only able to answer questions. To make NET-SCHOLAR a really effective CAI system it must be given the ability to present information and ask questions. Since sophisticated tutorial facilities now exists in GEO-SCHOLAR, similar facilities should be incorporated in NET-SCHOLAR. These facilities should be augmented with tutorial strategies that are well-suited to teaching procedural knowledge, such as teaching by instantiation and demonstration (the teacher presents examples and works them out), and teaching by doing (asking the student to perform a task that encompasses the procedural knowledge being taught).

In our follow on work, we will develop a system to help naive users of the NLS (On Line System) and we shall tackle some of these problems.

REFERENCES

Carbonell, Jaime R. and Collins, Allan M. "Mixed-Initiative Systems for Training and Decision-Aid Applications," ESD-TR-70-373, November 1970.

Carbonell, Jaime R. "AI in CAI: An Artificial-Intelligence Approach to Computer Assisted Instruction," IEEE Trans. on Man-Machine Systems, MMS-11, New York, pgs. 190-202, December 1970.

Collins, Allan M., et al., "Semantic Inferential Processing by Computer," Proceedings of the International Congress of Cybernetics and Systems, Oxford, England, August 1972.

Collins, Allan M., et al., "Improving Interactive Capabilities in Computer Assisted Instruction", BBN Report No. 2631, August 1972.

Chomsky, Noam, <u>Aspects of the Theory of Syntax</u>, MIT Press, 1965.

Crocker, Stephen D., et al., "Function-oriented Protocols for the ARPA Computer Network," SJCC 1972, pgs. 271-279.

Fillmore, Charles J. "The Case for Case," in <u>Universals in Linguistic Theory</u> (eds.) Bach and Harms, Holt, Rinehart, and Winston, 1968.

Heart, F., et al., "The Interface Message Processor for the ARPA Network," SJCC 1970, pgs. 551-567.