Bolt Beranek and Newman Inc.
Artificial Intelligence

BBN Report No. 2833
Job No. 11489

SEMANTIC NETWORKS

Contract N00014-70-C-0264, NR 348-027

Final Report

Allan M. Collins
Eleanor H. Warnock

May 1974

Submitted to:
Messrs. Marvin Denicoff & Gordon Goldstein
Information Systems
Mathematical & Information Sciences Division
Office of Naval Research, Code 437
Department of the Navy
Washington, D.C. 20360

BBN Report No. 2833
Job No. 11489
A.I. Report No. 15

SEMANTIC NETWORKS

Final Report

Allan M. Collins

Eleanor H. Warnock

Design and Programming of the work described in the report were carried out by:

Nelleke Aiello

Jaime G. Carbonell

Susan M. Graesser

Mark L. Miller

Joseph J. Passafiume

TABLE OF CONTENTS

ABSTRACT

Our work on semantic networks under contract N00014-70-C-0264 with the Office of Naval Research, Information Sciences Division involved three distinct areas: inferences, map displays, and English comprehension. The inference strategies implemented in SCHOLAR include different types of deductive, negative, and functional inferences. The graphics package allows users to ask questions and give commands in English to control SCHOLAR's map display, which is tied into the semantic network on South American geography. With partial support from this contract, we also developed an English Comprehension System, utilizing a data base on the ARPA network. Unlike geography, most questions about the ARPA network pertain to actions and procedures, which involve complicated English sentence structure, and hence necessitate sophisticated parsing and retrieval strategies. This report describes our work in each of these three areas.

ANNOTATED BIBLIOGRAPHY OF PAPERS PREPARED FOR THE PROJECT

Carbonell, Jaime R., "Artificial Intelligence and Large Interactive Man Computer Systems." Proceedings of 1971 IEEE Systems, Man, and Cybernetics Conference, Anaheim, California (October 1971).

This is an early paper describing the SCHOLAR system for mixed-initiative man-computer dialogues. SCHOLAR can ask questions, evaluate student answers, and answer student questions. It does this in a fairly natural subset of English. Unlike conventional CAI systems, it generates its questions and answers during the dialogue from its semantic network of knowledge. The paper describes SCHOLAR from a systems point of view. It discusses semantic networks and irrelevancy, context, question selection, answer analysis, input comprehension, information retrieval, and English output generation. It gives guidelines for the subsequent development of a graphic facility for map interactions.

Collins, Allan M., Carbonell, Jaime R. and Warnock, Eleanor H., "Semantic Inferential Processing by Computer." Proceedings of the International Congress of Cybernetics and Systems, Oxford, England (August 1972)

This paper briefly discusses some inferences under development in the SCHOLAR system: deductive, negative, and functional inferences. The procedures use SCHOLAR's data base of South American geography but they are essentially context independent.

Carbonell, Jaime R. and Collins, Allan M., "Natural Semantics in Artificial Intelligence." Proceedings of Third International Joint Conference on Artificial Intelligence, Stanford, California (August 1973). Also, to be reprinted in American Journal of Computational Linguistics.

This paper discusses human semantic knowledge and processing. One major section discusses the imprecision, the incompleteness, the openendedness, and the uncertainty of people's knowledge. The other major section discusses strategies people use to make different types of deductive, negative, and functional inferences, and the way uncertainties combine in these inferences.

Collins, Allan, Warnock, Eleanor H., Aiello, Nelleke, and Miller, Mark L., "Reasoning from Incomplete Knowledge." To appear in D.G. Bobrow and A. Collins (Eds.), Knowledge, Understanding, and Dialogue, Academic Press, New York (1975).

This paper discusses dealing with incomplete knowledge. One part describes in detail SCHOLAR's sophisticated set of procedures for finding contradictions. The other major part talks about functional knowledge: how people use it, how it may be represented in a computer, and how reasoning with it may be taught.

INTRODUCTION

In this section we will discuss, very briefly, some pertinent aspects of SCHOLAR, a mixed-initiative instructional system. More detailed discussions are provided in Carbonell (1970, 1971). Three data bases currently exist: one is about geography of South America, and another about the ARPA network and a third about a text-editing system called NLS. SCHOLAR's knowledge about a subject matter is in the form of a semantic network of facts, concepts, and procedures. This is a modified and extended network a la Quillian (1969) and has a rich internal structure with a well-defined syntax.

Dialogue with SCHOLAR takes place in a subset of English, that is restricted mainly by SCHOLAR's limited syntactic capabilities. In tutorial fashion, the system uses its semantic network to generate the material it presents, the questions it asks, and the corrections it makes. At any time SCHOLAR accepts questions and requests from the student, and generates responses based on its semantic network, making calculations and inferences of different types that it has been programmed to handle.

The dialogue is unanticipated, and is dependent on the student's responses, questions, and requests. Figure 1 presents a sample protocol of an interaction with SCHOLAR (the person types after the asterisk and returns control to the system with a second asterisk).

WHAT IS THE LANGUAGE OF ARGENTINA?

PROBABLY SPANISH

VERY GOOD.

APPROXIMATELY WHAT IS THE AREA OF ARGENTINA?

TELL ME SOMETHING ABOUT PERU

PERU IS A COUNTRY.
IT IS LOCATED IN SOUTH AMERICA.
THE CAPITAL OF PERU IS LIMA.

THE AREA OF ARGENTINA IS 1,100,000 SQUARE MILES

APPROXIMATELY TRUE, BUT A MORE EXACT VALUE IS 1,200,000.

THE EASTERN BORDERING COUNTRY OF PERU IS _ _ _ _ _ _.

CATHOLICISM

WRONG.

CATHOLICISM IS NOT A COUNTRY BUT A RELIGION.
THE CORRECT ANSWER IS: BRAZIL

WHERE IS PERU?

WHAT IS THE MOST IMPORTANT DIFFERENCE BETWEEN PERU AND PARAGUAY?

THE PRIMARY DIFFERENCE IS:
THE POPULATION OF PERU IS LARGER BY A FACTOR OF 7.8.

Figure 1. A Sample Dialogue Between SCHOLAR and a Student.

Figure 2 shows some excerpts from SCHOLAR's semantic network on geography. Properties, none of which are obligatory, can have as values single words (usually English words defined elsewhere in the network), numbers, different types of lists, and other properties. Attributes are usually English words, but there is a set of special attributes for important relations, like SUPERC (for superconcept or superordinate, e.g., Lima is a city and a capital), SUPERP (for superpart, e.g., Lima is a part of Peru and South America), SUPERA (for superattribute, e.g., fertile refers to soil and soil refers to topography), APPLIED/TO (color applies to things, and capital to countries and states), CONTRA (for contradiction, e.g. barren contradicts fertile and democracy contradicts dictatorship), case-structure attributes like AGENT and INSTRUMENT (Fillmore, 1968), and various others.

The entry for location under Peru in Figure 2 illustrates an important aspect of SCHOLAR's semantic network called embedding. Under the attribute location there is the value South America plus several subattributes, among which is bordering countries. But under bordering countries there are subattributes like northern and eastern, some of which have several values. Embedding describes the ability to go down as deep as necessary to describe a property in more or less detail.

In the data base there are also tags, such as the (I 0) after location and the (I 1) after bordering countries. These tags are called importance or irrelevancy tags (I-tags), and they vary from Ø to 6. The lower the tag, the more important the piece of information is. The tags add up as you go down through lower embedded levels. One of the ways SCHOLAR uses I-tags is to decide what is relevant to say at any given time.

In the rest of this report, we will discuss our work in SCHOLAR on inference, map displays, and English comprehension.

```
CAPITAL
   SUPERC (I 0) CITY
                PLACE (I 0)
                      OF (I 0) GOVERNMENT
   APPLIED\TO (I 4) COUNTRY STATE
   EXAMPLES (I 2) ($EOR BUENOS\AIRES LIMA MONTEVIDEO
        BRASILIA GEORGETOWN CARACAS BOGOTA QUITO
        SANTIAGO ASUNCION LA\PAZ WASHINGTON)


 FERTILE
   CONTRA (I 0) BARREN
   SUPERA (I 0) SOIL


 PERU
   SUPERC (I 0) COUNTRY
   SUPERP (I 1 B) SOUTH\AMERICA
   LOCATION (I 0)
      IN (I 0)
         SOUTH\AMERICA (I 0) WESTERN
      ON (I 0)
         COAST (I 0)
               OF (I 0) PACIFIC
      LATITUDE (I 4)
           RANGE (I 0) -18 0
      LONGITUDE (I 5)
           RANGE (I 0) -82 -68
      BORDERING\COUNTRIES (I 1)
                NORTHERN (I 1) ($L COLOMBIA ECUADOR)
                EASTERN (I 1) BRAZIL
                SOUTHEASTERN (I 1) BOLIVIA
                SOUTHERN (I 2) CHILE
   PEOPLE (I 2)
      POPULATION (I 0)
         APPROX (I 0) 11000000
      LANGUAGE (I 2)
           ($L PRINCIPAL OFFICIAL) (I 0) SPANISH
           INDIAN (I 2) ($L QUECHUA AYMARA)
   CAPITAL (I 1) LIMA
   CITIES (I 2)
        PRINCIPAL (I 0) ($L LIMA CALLAO AREQUIPA TRUJILLO CHICLAYO
                            CUZCO)
```

FIGURE 2. Three Partial Entries from SCHOLAR's Geography Data Base

INFERENCES

We have programmed modules in SCHOLAR to handle various types of deductive, negative, and functional inferences in answering questions. The deductive inferences include superpart, superordinate, and various special purpose computations. The negative inferences include a sophisticated contradiction module and the ability to infer that a question does not make sense (the APPLIED/TO test). Work on functional inference includes an ability to make rough calculations about climate from altitude and location information, and a general purpose routine for drawing positive and negative analogies. This section describes the various modules in each of these domains of inference.

Deductive Inferences

Reaction-time experiments (Collins & Quillian, 1969, 1970) support the notion that people use deductive inferences to answer even quite simple questions like "Can a wren fly?" There are now two kinds of deductive inferences in SCHOLAR, superpart and superconcept inferences.

Superpart Inferences. Properties that refer to such attributes as climate, topography, and language, transfer from an object to parts of that object. This is shown in examples 1 and 2 of Figure 3. In example 1, hot refers to temperature, which in turn refers to climate, but there is nothing stored about the climate of Asuncion. Because climate transfers on the part relationship, SCHOLAR looks for climate under Asuncion's superparts. The temperature is stored as subtropical under eastern Paraguay, the region that Asuncion is part of. Because hot is an acceptable

(1) *IS ASUNCION HOT?*

YES.

(2) *WHAT IS THE LANGUAGE OF BRASILIA?*

THE LANGUAGE IN BRASILIA IS PORTUGUESE.

(3) *IS IT TRUE THAT THE LLANOS IS GRASSY*

YES.

(4) *WHAT IS THE COLOR OF THE ATLANTIC?*

THE COLOR OF THE ATLANTIC IS BLUE.

FIGURE 3. Examples of Outputs from Deductive Inferences in SCHOLAR.

substitute for subtropical, SCHOLAR answers "Yes." Again, in example 2, Portuguese is retrieved, not from Brasilia, but from Brasilia's superpart, Brazil.

Superconcept Inferences. Properties of the superordinate are generally true for instances. In examples 3 and 4 of Figure 3, the superpart inference would not work, and so SCHOLAR searches the superordinates of Llanos and Atlantic. The Llanos is a savanna, and SCHOLAR finds that the terrain of a savanna is grassy. Similarly, the Atlantic is an ocean and the color stored with ocean is blue. If another color were peculiar to the Atlantic, it could be stored with the Atlantic, and the superordinate inference would be precluded.

The Comparative Module. This module answers questions about which of several things is smaller or larger, or longer, or higher. Three examples (1-3) can be seen in Figure 4.

Example 1 requires finding which is bigger of two things. Since they are countries, the module takes bigger to mean area, and computes which has the larger area. In Example 2 the question about the second largest city in Argentina also uses the comparative module. The size of a city is assumed to mean population size, and so the module compares the population size for all the cities stored under Argentina (there are nine), and computes which is second largest. This involves looking up each city because the population is not stored with the city names under Argentina. The same module will find the longest river in South America, or the fourth highest mountain in the Andes or the smallest country in South America.

(1) *WHICH IS BIGGER, BRAZIL OR URUGUAY?*
IT IS BRAZIL.

(2) *WHAT IS THE SECOND LARGEST CITY IN ARGENTINA?*
IT IS CORDOBA.

(3) *WHAT IS THE LANGUAGE OF THE CAPITAL OF THE LARGEST COUNTRY IN SOUTH AMERICA*

THE LANGUAGE OF THE CAPITAL OF BRAZIL IS PORTUGUESE.

(4) *COMPARE THE ACONCAGUA AND THE ORINOCO.*

THERE ARE NO SIGNIFICANT SIMILARITIES BETWEEN THE ACONCAGUA AND THE ORINOCO.

THE DIFFERENCES ARE:

THE ACONCAGUA IS A MOUNTAIN.
BUT
THE ORINOCO IS A RIVER.

THE ACONCAGUA IS IN ARGENTINA AND CHILE.
BUT
THE ORINOCO IS IN VENEZUELA.

(5) *GIVE ME A COMPARISON BETWEEN THE AREAS OF PARAGUAY AND CHILE.*
THE AREA OF CHILE IS LARGER BY A FACTOR OF 1.3.

(6) *WHAT ARE ALL THE SIMILARITIES BETWEEN BOLIVIA AND PERU?*
THE SIMILARITIES ARE:

THEY ARE BOTH COUNTRIES.
THEY ARE IN SOUTH AMERICA.
THE AREAS OF BOLIVIA AND PERU ARE EQUAL.
THE AREA OF EACH IS APPROX 465,000 SQUARE MILES.

(7) *WHAT IS THE MOST IMPORTANT DIFFERENCE BETWEEN BRASILIA AND RIO*
THE PRIMARY DIFFERENCE IS:

BRASILIA IS IN CENTRAL BRAZIL.
BUT
RIO DE JANEIRO IS IN EASTERN BRAZIL.

FIGURE 4. Examples of Outputs from Comparative and Comparison Modules in SCHOLAR.

Example 3 shows the comparative module in conjunction with a superpart inference. The example also illustrates that the questioner need not ask simply for the largest, or the longest, or second highest something; he can ask for a property of one of these. In the example, the questioner asks about the language of the capital of the largest country. The comparative module first calculates the largest country (i.e. Brazil). Then its capital is determined internally (Brasilia) by looking for the capital under Brazil. Language is not stored with Brazilia, but language transfers on the superpart relation. A superpart of Brasilia is Brazil, and the language stored under Brazil is Portuguese. Hence, this example illustrates a complicated set of embedded operations to determine the answer.

The Comparison Module. This module compares any two entries in the data base to find their similarities and differences. It looks for these in order of importance as determined by I-tags. In Figure 4, examples 4 through 7 illustrate different outputs by the comparison module.

Examples 4 and 5 show two kinds of basic comparisons between objects. In example 4, the module looks for similarities and differences between the two objects named (i.e. Aconcagua and the Orinoco). Finding a similarity consists of finding the same attribute under both objects with the same value (within a tolerance of 10% for numerical values). Finding a difference consists of finding the same attribute with contradictory values for the two objects. In this case there are no similarities, but there are differences on two attributes, the superordinate and location. Example 5 shows a comparison between two objects with respect to an attribute (i.e. area) specified in the question.

The module finds that the two numerical values are not within the 10% tolerance, so it computes the ratio of the two and gives that as an answer. This is the kind of answer the module gives for any difference in numerical values.

Examples 5 and 7 show how the module handles questions relating only to similarities or only to differences. Example 6 asks for all the similarities between Bolivia and Peru, and three are found. When the module finds a similarity in numerical values, as it did with areas in Example 6, it gives the value for one of the objects in addition to pointing out they are equal. In example 7, the most important difference between Brasilia and Rio is determined by looking at attributes in the order of their I-tag values until one is found with contradictory values. Brasilia and Rio both have the same superordinate, so the most important difference occurs in their location. It is possible to ask for the two most importance differences (or similarities) or simply the primary differences.

Other Computations. There are two other modules that have been programmed but not yet integrated in SCHOLAR. We have written a subroutine that can find the sum or average on some dimension for a set of objects. For example, it can compute the average population of all the countries in South America or the combined area of the two large savanna regions in South America.

Another effort consisted in writing a subroutine that would compute the attributes "large," or "tall," or "small" for an object in a given context. If the system is asked "Is Montevideo very large?", it will consider its size with respect to whatever superparts Montevideo has. If it finds both Uruguay and South

America, it should respond something like, "It has about one million people. It is only the eighth largest city in South America. But it is the largest city in Uruguay." The module to do the necessary computations is ready, but the necessary constructions to produce the English output are not.

Negative Inferences.

People do not store most things that are not true, for example that Mexico has no king. Therefore, answering "No" to a question normally requires an inference. The problem with negative inferences is to decide when to say "No" and when to say "I don't know." If one is willing to say "No" simply whenever something is not stored, then the only problem is that many answers will be wrong. We will describe several kinds of negative inferences in SCHOLAR.

Match Procedures in SCHOLAR. If you ask people whether Buenos Aires is in Brazil, they tell you "no," because it is in Argentina. They also tell you that the reason Buenos Aires isn't a country is because it is a city. In both cases the negative response involved is based on the distinction between two similar objects, in one case Brazil and Argentina, and the other, city and country. In SCHOLAR this basic comparison process is handled by two subroutines called Match-on-Superordinate and Match-on-Superpart We can explain how these subroutines work in terms of the various outcomes of Match-on-Superordinate shown in Figure 5.

(1) *IS SANTIAGO A PLACE?*
YES.
MORE SPECIFICALLY, SANTIAGO IS A CITY.

(2) *IS A CITY A CAPITAL?*
SOMETIMES.
MORE GENERALLY, A CITY IS A PLACE.

(3) *IS THE PAMPAS A LAKE?*
NO.
THE PAMPAS IS A PLAIN, NOT A LAKE.

(4) *IS BOLIVIA A CAPITAL?*
NO.
BOLIVIA IS A COUNTRY, NOT A CAPITAL.

(5) *IS IT TRUE THAT RECIFE IS THE CAPITAL OF URUGUAY?*
NO.
RECIFE IS IN BRAZIL, NOT IN URUGUAY. THE CAPITAL OF URUGUAY IS MONTEVIDEO.

(6) *IS SAO PAULO A PORT?*
I DON'T KNOW IF SAO PAULO IS A PORT, BUT IT IS A CITY.

FIGURE 5. Examples of Outputs from Match-on-Superordinate Subroutine.

Match-on-Superordinate is basically an intersection routine (Quillian, 1968). For the sake of efficiency it does not search properties other than the superordinate unless an intersection is found. Its function is to decide if X can be Y for any two entries in the data base.

The first two examples in Figure 5 show different kinds of "Yes" outcomes. The first example involves a simple superordinate inference, and the second shows the outcome when Y is an X. SCHOLAR does not now distinguish between the two kinds of superordinates involved here (Santiago is an instance of city, whereas a capital is a type of city), but it easily could.

The next four examples show how the subroutine deals with the problem of when to say "No" and when to say "Don't know." Its basic strategy is to try to find some basis for saying "No," and only if it fails does it conclude "Don't know." If it fails to find a contradiction, some other subroutines may be called to look for a less certain basis for saying "Yes" or "No."

The third example shows that if there is no common superordinate of X and Y, a reasonable response is "No." In the example, the top-level superordinate for Pampas is "place", and for lake is "body-of-water," so the superordinate chains do not intersect (if they did, then another outcome would distinguish them).

The last three examples illustrate different outcomes when an intersection occurs. In the fourth example, Bolivia has the superordinate country, and a capital has the superordinate city, and both of these have the superordinate place. But in the data base under place, country and city are marked as mutually exclusive kinds of places (by a $EOR for exclusive-or), so the routine concludes "No."

The next example illustrates the case where the two objects, in this case Recife and Montevideo, have a common superordinate, but are not on a $EOR list together. In this case, they have a distinguishing property in that they are located in different places. This is determined by the Match-on-Superpart subroutine which answers the question "Is X part of Y?". Match-on-Superpart works like Match-on-Superordinate, but is more complicated, because it is necessary to find a mismatch at two mutually exclusive things with the same superordinate (e.g. two regions, two oceans) in order to say "No." People frequently give a distinguishing property such as the difference in location as a reason for saying that two things are not the same. This observation led to the test for a distinguishing property in the Match-on-Superordinate subroutine.

The last example shows the failure to find any basis for a contradiction. A port can be a city and Sao Paulo is a city, and they are not stored on a $EOR list nor are there any distinguishing properties between them. So there is no contradiction. At present this leads to a "Don't know" response. It would be appropriate at this point to try a probabilistic inference, such as a lack-of-knowledge inference (see Carbonell & Collins, 1973) or a functional inference.

Contradictions. The two Match subroutines form the basis for a sophisticated module that finds a variety of contradictions. The module is used both for answering true/false questions and for evaluating student answers to questions. The strategy is to say "No" only if the module can find a basis for distinguishing the correct value from the student's value. If it can find a basis for distinguishing the two, it gives this as a reason for saying "No." Otherwise it prints out whatever relevant information it has stored.

Figure 6 shows examples of various outcomes from the contradiction module. Though the examples show the flexibility of the module only to a limited degree, the basic strategies that are used (other than for numerical values) are illustrated by these examples.

The first two examples illustrate the two basic ways to come into a match procedure, in this case Match-on-Superpart. Example 1 is a question about the location of Sao Paulo, so the module looks under Sao Paulo to see if Chile is stored as the location. The location of Sao Paulo is in fact stored as Brazil and the Brazilian Highlands. So the module asks if Chile is disjoint from Brazil or the Brazilian Highlands. Match-on-Superpart finds that Chile and Brazil are stored together on an exclusive-or ($EOR) list of countries so the module concludes that Sao Paulo cannot be in Chile.

Example 2 is interpreted as a question about the cities of Chile, rather than about the location of Sao Paulo. The first thing SCHOLAR does (before calling the module) is to look under the cities of Chile to see if Sao Paulo is there. It is not, so this module looks for information under Sao Paulo that could lead to

(1) *IS SAO PAULO IN CHILE?*

NO. SAO PAULO IS NOT IN CHILE BUT IN BRAZIL.

(2) *IS SAO PAULO A CITY IN CHILE?*

NO. SAO PAULO IS IN BRAZIL, NOT IN CHILE. THE CITIES OF CHILE ARE SANTIAGO, VALPARAISO, CONCEPCION, ANTOFAGASTA, ARICA, VINA DEL MAR, PUNTA ARENAS, AND IQUIQUE.

(3) *IS SAO PAULO IN NORTHEASTERN BRAZIL?*

NO. SAO PAULO IS NOT IN NORTHEASTERN BRAZIL BUT IN THE BRAZILIAN HIGHLANDS.

(4) *IS GUYANA IN BRAZIL?*

NO. GUYANA AND BRAZIL ARE DIFFERENT COUNTRIES.

(5) *IS THE PURUS A TRIBUTARY OF THE RIO DE LA PLATA?*

I DON'T KNOW IF A TRIBUTARY OF THE RIO DE LA PLATA IS THE PURUS. THE TRIBUTARIES OF THE RIO DE LA PLATA ARE THE PARANA AND THE URUGUAY RIVER.

(6) *IS THE ACONCAGUA A TRIBUTARY OF THE RIO DE LA PLATA?*

NO. THE CORRECT ANSWER IS THE PARANA AND THE URUGUAY RIVER. THE ACONCAGUA IS A MOUNTAIN, NOT A TRIBUTARY.

(7) *IS PERU THE NORTHERN BORDERING COUNTRY OF ARGENTINA?*

NO. THE NORTHERN BORDERING COUNTRIES OF ARGENTINA ARE PARAGUAY AND BOLIVIA.

FIGURE 6a. Outputs by the Contradiction and Answer-Evaluation Module.

(8) *IS VALPARAISO THE CAPITAL OF CHILE?*

NO. ONE OF THE PRINCIPAL CITIES OF CHILE IS VALPARAISO. THE CAPITAL OF CHILE IS SANTIAGO.

(9) THE EASTERN AND CENTRAL PLAIN OF ARGENTINA IS ---.

CHACO

NO. THE NORTHERN PLAIN OF ARGENTINA IS THE CHACO. THE EASTERN AND CENTRAL PLAIN OF ARGENTINA IS THE PAMPAS.

(10) WHAT ARE THE REGIONS OF CHILE?

CHACO, CENTRAL VALLY AND ATACAMA.

THE ATACAMA DESERT AND THE CENTRAL VALLEY ARE CORRECT. YOU MISSPELLED CENTRAL VALLEY. THE CHACO IS IN ARGENTINA, PARAGUAY, AND BOLIVIA, NOT IN CHILE. AN OTHER REGION OF CHILE IS THE SOUTHERN ANDES.

(11) *ARE THE ANDES A REGION IN CHILE?*

WELL, YOU'RE ALMOST RIGHT. MORE SPECIFICALLY, A REGION OF CHILE IS THE SOUTHERN ANDES.

(12) *IS PORTUGUESE THE LANGUAGE OF URUGUAY?*

NO. THE LANGUAGE OF URUGUAY IS SPANISH.

(13) *IS WHEAT A PRODUCT OF PERU?*

I DON'T KNOW IF A PRODUCT OF PERU IS WHEAT. THE PRODUCTS OF PERU ARE COTTON, CATTLE, FERTILIZER, RUBBER, COPPER, AND PETROLEUM.

(14) *IS BUENOS AIRES TROPICAL?*

NO. BUENOS AIRES IS TEMPERATE.

FIGURE 6b. Outputs by the Contradiction and Answer-Evaluation Module.

a contradiction. Since Sao Paulo is a city, Match-on-Superordinate, which is called first, does not find a contradiction. But Match-on-Superpart does find a contradiction between Sao Paulo's superpart, Brazil, and Chile as in the previous example. In this case the module prints out both the contradiction and the information it has about the cities of Chile.

Examples 3 and 4 illustrate two other possible results of a call to Match-on-Superpart. Example 3 is different from Example 1 in that the mismatch occurs at two regions, the Brazilian Highlands and Northeastern Brazil, rather than at two countries. The two regions are stored on a $EOR list of mutually exclusive regions. Notice that the fact that Sao Paulo is in Brazil could not be used to say "No" in this case. Example 4 shows what happens when the mismatch occurs at two countries both of which were mentioned in the student's question. In such a case the appropriate response is to point out that they are different countries.

Example 5 shows the "Don't Know" outcome when there is a list of values that is incomplete. In this case the module can find no basis for saying the student's value is not correct. This is because the Purus is like one of the correct values (the Parana) in that both are rivers in Brazil. Thus the module cannot rule out the Purus using either of the two Match subroutines The Purus is in fact a tributary of the Amazon, but the module does not know how to use its information to say "No." (If the information were stored in different form, it might.) So it indicates its ignorance, and points out what it knows about the tributaries of the Rio de la Plata.

Examples 6 and 7 are variants of Example 5. Example 6 shows what happens if there is a basis for rejecting the student's value. In this case Aconcagua is a mountain, and the superordinate chains of mountains and tributaries do not intersect, so Match-on-Superordinate concludes that they are distinct. Example 7 shows that if there is an exhaustive ($EX) list stored, as with the northern bordering countries of Argentina, then this is grounds for saying no. This is true even though Peru is a country in South America, just like Paraguay and Bolivia.

Examples 8 and 9 illustrate what happens when the student's value appears elsewhere under the object in the question. In Example 8 Valparaiso appears as a city under Chile, so this is pointed out to the student. In Example 9, the student named the wrong plain in Argentina (i.e. the Chaco) in answer to a question by SCHOLAR. The module found the information about the Chaco stored under Argentina and gave this to distinguish the two plains.

Example 10 illustrates the flexibility of the module for handling lists. The module tries to match each of the student's values to one of the stored values. Atacama is another name for Atacama Desert so this matches first. "Central Vally" matches on spelling correction to the Central Valley, so this pair is matched. Chaco doesn't match Southern Andes, and in fact the Chaco's location, which is an exhaustive ($EX) list of countries, produces a mismatch with Chile. Here the Chaco is distinguished by naming the countries it is in rather than by giving its location within Argentina, as in the previous example. The module also adds the fact that the student left out the Southern Andes in the answer.

Example 11 illustrates the qualified "Yes" that the module gives when the student's value is less specific than the value stored. In this case the Andes is the superpart for Southern Andes. The same outcome occurs if the student's value is a superordinate of the stored value. If the inverse relation holds, the module would give the same qualified "Yes", and "more generally" would replace "more specifically."

Example 12 and 13 illustrate the "uniqueness assumption" made by the module. By the uniqueness assumption, we mean that the module assumes that if there is only one value stored, it is unique or exhaustive. Thus in Example 12, there is only one language stored for Uruguay, so the module assumes that it is the only language. This is in contrast to Example 13, where there is a list of products stored. The module assumes that a list is not exhaustive unless it is marked as such (by a $EX). With single values, the module assumes exhaustiveness unless the value is marked as inexhaustive (with a $L). As an example of the marked case, suppose there were only one value stored for the products of Guyana (e.g. bauxite). This would be stored as an inexhaustive list that happens to have only one value. It is not really appropriate to say "The product of Guyana is bauxite" because there are probably other products. It is better to say "The principal product of Guyana is bauxite" in such a case. It turns out that most lists are inexhaustive, whereas most single values are exhaustive, so the smaller class is marked in each case.

Example 14 shows how a contradiction can occur in conjunction with a superpart or superordinate inference. In this case no climate information is stored with Buenos Aires. Since climate transfers on the part relation, Buenos Aires' superparts, Argentina and the Pampas, are searched for climate information.

It is found that the climate of the Pampas is temperate. Because this value does not match the student's value, it is sent off to the contradiction module to determine the relation between the two values. Match-on-Superordinate determines that tropical and temperate are contradictory, because tropical appears on the CONTRA list stored under temperate. Alternatively they both might be stored on an exclusive-or ($EOR) list under temperature. Thus, contradiction between values, such as between temperate and tropical, can be stored with each value or more globally with one of their superordinates.

In general, whenever any inference or computation module comes up with a difference between two values, it turns these over to the contradiction module to decide on a match, a mismatch, or a "Don't know" outcome. This modularity allows different inferences to combine in a flexible way.

Inferring that a question does not make sense. Examples 1 and 2 in Figure 7 illustrate how SCHOLAR infers that an attribute does not apply to a concept. This APPLIED/TO test, which is a type of selection restriction, is performed before any of the other inferences are tried. In Example 1, capital has country, state, and province on its APPLIED/TO list. The Amazon's superordinate river, is not among those three things, so SCHOLAR points out its difficulty.

In Example 2, tropical refers to (i.e. its SUPERA is) temperature, and temperature refers to climate. Climate does not apply to religion, which is Catholicism's superordinate. So SCHOLAR points out that it does not know what it means for climate to apply to a religion.

(1) *TELL ME ABOUT THE CAPITAL OF THE AMAZON.*

THE AMAZON IS A RIVER, AND I DON'T KNOW WHAT IT MEANS FOR A RIVER TO HAVE A CAPITAL.

(2) *IS CATHOLICISM TROPICAL?*

CATHOLICISM IS A RELIGION, AND I DON'T KNOW WHAT IT MEANS FOR A RELIGION TO HAVE A CLIMATE.

(3) *IS BUENOS AIRES BARREN?*

NO. BUENOS AIRES IS FERTILE.

(4) *IS FRANCE BARREN?*

I DON'T KNOW ANYTHING ABOUT THE TOPOGRAPHY OF FRANCE.

FIGURE 7. Examples of the APPLIED/TO Test and the Failure to Infer an Answer.

The APPLIED/TO test uses Match-on-Superordinate in comparing elements on the APPLIED/TO list to the object in the question. In Example 3, barren refers to soil and soil in turn to topography. Topography can apply to any place, so Match-on-Superordinate is used to decide if Buenos Aires is a place. Buenos Aires is a city, and cities are places, so the APPLIED/TO test is passed. The answer is based on a superpart inference like the one in Example 14 of Figure 6. Buenos is part of the Pampas and the Pampas is fertile, so SCHOLAR concludes the answer is "No."

Failure to infer an answer. Example 4 shows what happens if all the procedures above are tried and fail. As in the previous example, barren refers to soil and soil to topography. In the data base, nothing is stored under France, except its Superordinate, country, and its Superpart, Europe; and there is nothing about topography under either of these. So SCHOLAR explains its ignorance with a "Don't know" answer.

Functional Inferences

Functional relations can be used in a number of different ways. We have considered six such ways: (1) to make direct calculations (e.g., to estimate a place's climate from its latitude and altitude); (2) to make negative calculations (e.g., if a place has an altitude over a mile high it doesn't have a tropical climate even though it is on the equator); (3) to make positive analogies (e.g., if another place has a similar latitude and altitude, its climate is likely to be similar to that of the place we are interested in); (4) to make negative analogies (e.g., if another place has a quite different latitude or altitude, it is not likely that the place we are interested in would have a similar climate) (5) to answer "Why" questions (e.g., if asked why a place has a particular climate, it is because of its values for latitude, altitude, etc.); and (6) to answer "Why not" questions (e.g., if a place does not have a particular climate, it is because one or more of the values for latitude, altitude, etc. is wrong).

Our approach has been to write modules for each of these operations, that are independent of the particular functional relationships involved. That is to say, the functional knowledge should be part of the data base, and the strategies for making computations or analogies or answering "Why" questions should look at what is stored in the data base to determine what can in fact be inferred.

We began our work on functional inferences with the agricultural products and climate functions. The agricultural products of a place are mainly a function of the climate, rainfall, and soil fertility. Climate in turn is largely a function of latitude and altitude.

We developed a function that computes whether a place's climate is tropical, subtropical, temperate, or cold given values for latitude and altitude. We also developed a general purpose module to make positive and negative analogies, but it is currently limited by the data base to analogies about climate and agricultural products. This module is described below. Work on other functional relationships, negative calculations, and on answering "Why" and "Why not" questions is still under way.

Analogies. When functional information is incompletely specified or missing from the data base, it is not possible to do a direct calculation to answer a question about, say, the climate of a region. But if the relevant parameters are known, a sensible response can often be inferred by analogy with other cases. When the other case considered is similar in terms of the relevant functional determinants, the information is said to be derived by positive analogy. When there is a large difference in the relevant determinants, a negative analogy has been used. For example, wheat is not a likely product of the Atacama Desert, since its climate and soil are so unlike Uruguay, whose products include wheat. Positive and negative analogies require similar processing, and the two are performed by a single module.

The algorithm operates in the following manner. It generates a list ("XLIS") of possible analogous items by taking the list of examples stored under the superordinate of the object and selecting those with the desired property (e.g. a subtropical climate). It determines from the data base (under the entry for Climate, Agricultural Products, etc.) what factors the desired attribute depends on.

Each item on XLIS is then compared to the original object with respect to those factors. In computing the overall degree of match or mismatch, the different factors are weighted for their relative importance. For example, in the case of climate, which depends on latitude and altitude, very similar latitudes might compensate for slightly dissimilar altitudes. In testing for a match or mismatch, scalable attributes can be compared for their degree of match, whereas non-scalable attributes are either synonomous, contradictory, or unrelated. In order to make a positive analogy, the XLIS item with the highest degree of match must be above a certain criterion. For a negative analogy, the XLIS item with the highest degree of match must be below a different criterion; in other words, the best match should be a clear mismatch. If the reliability or certainty of the answer falls below a prespecified figure, then the routine gives up. The reliability can also be used to indicate in the printed response the amount of certainty in SCHOLAR's answer.

Figure 8 shows some examples of the output of the module. It can perform positive and negative analogies in answering "true/false" and "Why" questions about the agricultural products or climate of a place. The module, though completed, is not yet generally available in SCHOLAR.

(1) *IS THE CLIMATE OF BUENOS AIRES SUBTROPICAL?*

I DON'T KNOW THE CLIMATE OF BUENOS AIRES; HOWEVER, THE CLIMATE OF CARACAS IS SUBTROPICAL; SINCE BUENOS AIRES IS NOT ANALOGOUS TO CARACAS IN ALTITUDE AND LATITUDE, IT SEEMS UNLIKELY.

(2) *WHAT IS THE CLIMATE OF BUENOS AIRES?*

I DON'T KNOW THE CLIMATE OF BUENOS AIRES; HOWEVER, THE CLIMATE OF MONTEVIDEO IS TEMPERATE; SINCE BUENOS AIRES IS ANALOGOUS TO MONTEVIDEO IN ALTITUDE AND LATITUDE, IT SEEMS REASONABLE TO ASSUME:

THE CLIMATE OF BUENOS AIRES IS TEMPERATE.

(3) *ARE SHEEP AN AGRICULTURAL PRODUCT OF PATAGONIA?*

I DON'T KNOW THE AGRICULTURAL PRODUCT OF PATAGONIA; HOWEVER, THE AGRICULTURAL PRODUCT OF THE CHACO IS SHEEP; SINCE PATAGONIA IS NOT ANALOGOUS TO THE CHACO IN CLIMATE AND SOIL, IT SEEMS UNLIKELY.

(4) *IS CORN AN AGRICULTURAL PRODUCT OF THE RIO GRANDE?*

I DON'T KNOW THE AGRICULTURAL PRODUCT OF RIO GRANDE; HOWEVER, THE AGRICULTURAL PRODUCT OF THE PAMPAS IS CORN; SINCE RIO GRANDE IS ANALOGOUS TO THE PAMPAS IN CLIMATE AND SOIL, IT SEEMS REASONABLE TO ASSUME:

YES.

FIGURE 8. Examples of Outputs from Functional Analogies Subroutine.

MAPS IN SCHOLAR

We implemented a sophisticated map-handling capability in the geographic version of SCHOLAR. One of the major motives for this effort was to see how graphic structures could be mixed with symbolic (verbal) information in a semantic net. Maps are a convenient form of two-dimensional graphs. Also, people seem to use images of maps to answer questions about relative positions of places (e.g., "What countries do we fly over in a direct flight from Caracas to Buenos Aires?") and about relative sizes, and we wanted to give the computer an ability to process map information the way people do. The point was not to build a graphics system but to build an integrated mixed system that used both maps and English in its task, and that incorporated some important SCHOLAR features, such as unanticipated student input, importance, and semantic network.

The graphic data base contains information in a hierarchy of figures, for drawing coastlines, borders, rivers, cities, regions, etc. Each figure is made up of primitive sets of points and lines and/or calls to other figures. The structure happens to be not too different from the verbal part of the semantic network that holds the rest of SCHOLAR's knowledge, and the two kinds of information are stored in parallel.

There is an important interplay between the graphic and symbolic data. When a map of area, say, Brazil, is displayed, the contour stored with Brazil is put on the screen. But what about the cities and rivers of Brazil? They are not called directly by the graphic figure of Brazil. Rather, SCHOLAR looks at the symbolic information on Brazil, selects those things that are part (in the part-superpart sense) of Brazil, and adds some of them to the map. Because of the I-tags, SCHOLAR knows enough to add only those things that are most important. After all, in a course map, detail is irrelevant, and things displayed on a map of Brazil shouldn't all be present in a map of South America. The amount of detail may be increased by zooming in for a closer look at part of the map, or by requesting the addition of detail.

Figure 9 is a series of pictures of a session of "map" interactions. Notice that these are not simply "Give me a map" questions, but involve blinking, enlarging, backing away, and remembering previous maps.

We have tried to make SCHOLAR sensitive to the current situation. If the student asks to see a city, SCHOLAR will not just blow the city's symbol up to the size of the screen and display it, as it would with a country; instead, it will take an appropriate superpart (e.g., Brazil for the city Brasilia) to fill the screen and simply blink the city requested. SCHOLAR will also do different things depending on the current map being shown. If the student asks about Peru when he is looking at South America, he probably wants to know where Peru is in relation to the other countries, and SCHOLAR will blink Peru's border; if the student is looking at Uruguay, however, he clearly wants a map of Peru.

If SCHOLAR cannot figure out just what the student intended, it is likely to come back and ask for verification of what it thinks it should do. An example is the response to the command to blink Brasilia when Brasilia is not around, and it is not clear whether the student really wants to switch to Brazil. This responsive interactive quality is something that we think is important.

KYBD→TENEX

FIGURE 9a. Output from Map Interaction with SCHOLAR.

KYBD→TENEX

DISPLAY URUGUAY

THIS IS URUGUAY.

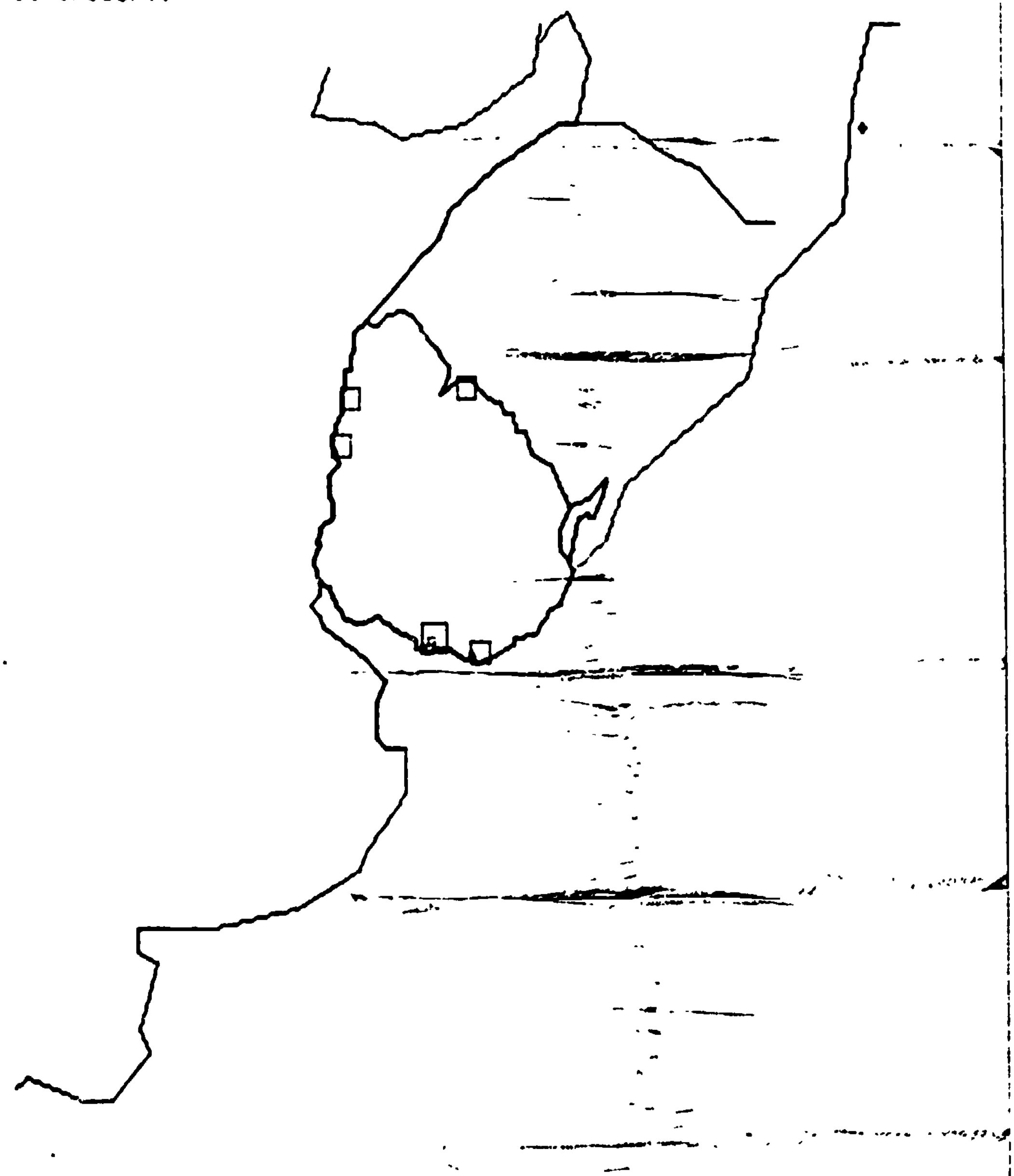

FIGURE 9b. Output from Map Interaction with SCHOLAR.

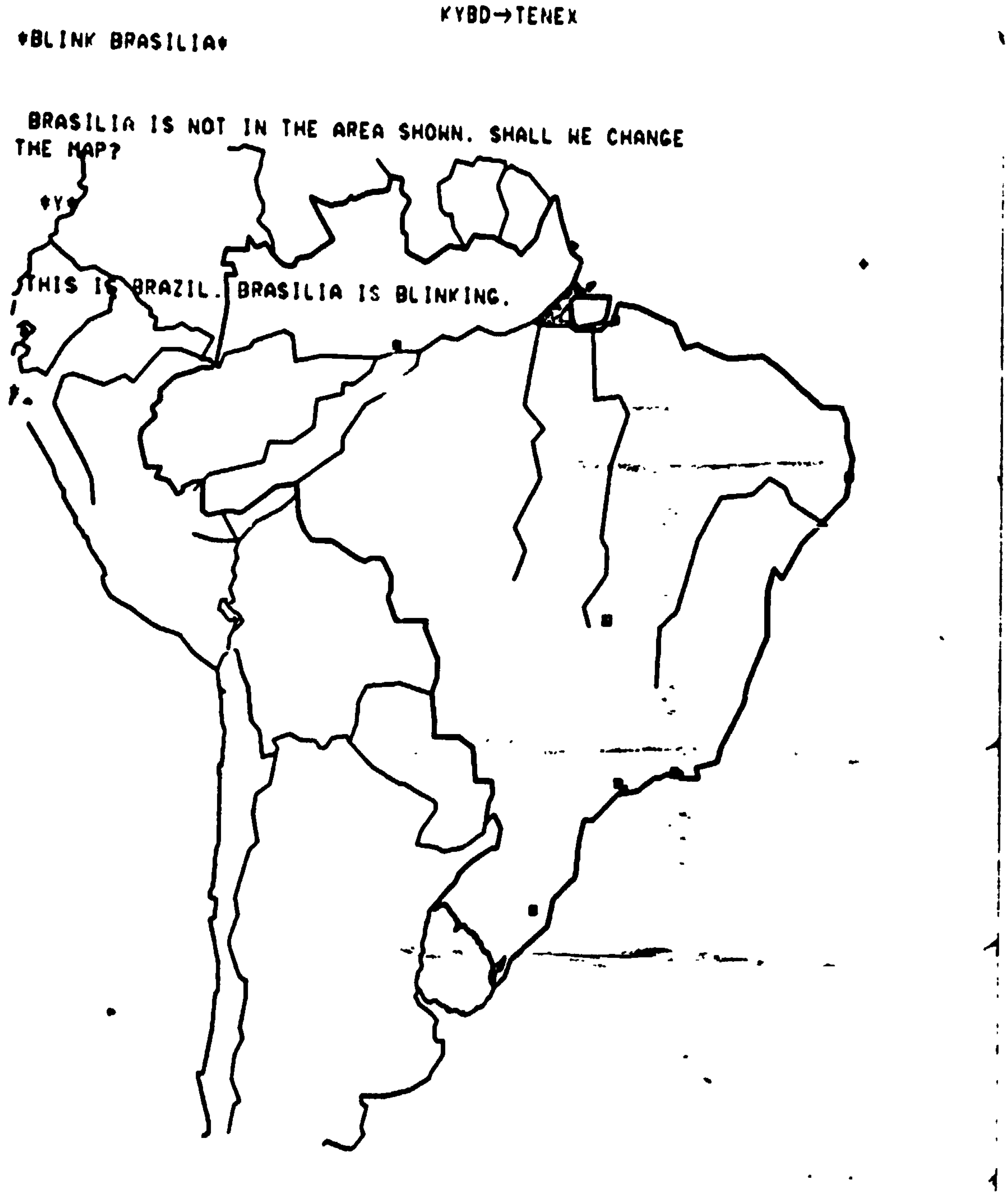

FIGURE 9c. Output from Map Interaction with SCHOLAR.

KYBD→TENEX

WHERE IS RIO?

RIO DE JANEIRO IS BLINKING.

GIVE ME A MAP OF CHILE

THIS IS CHILE.

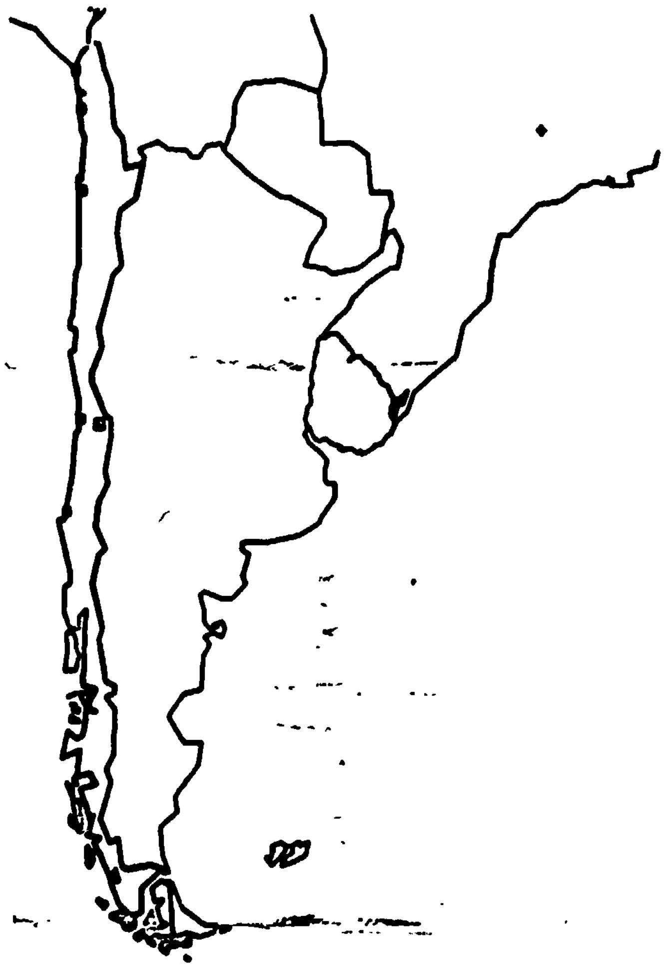

FIGURE 9d. Output from Map Interaction with SCHOLAR.

FIGURE 9e. Output from Map Interaction with SCHOLAR.

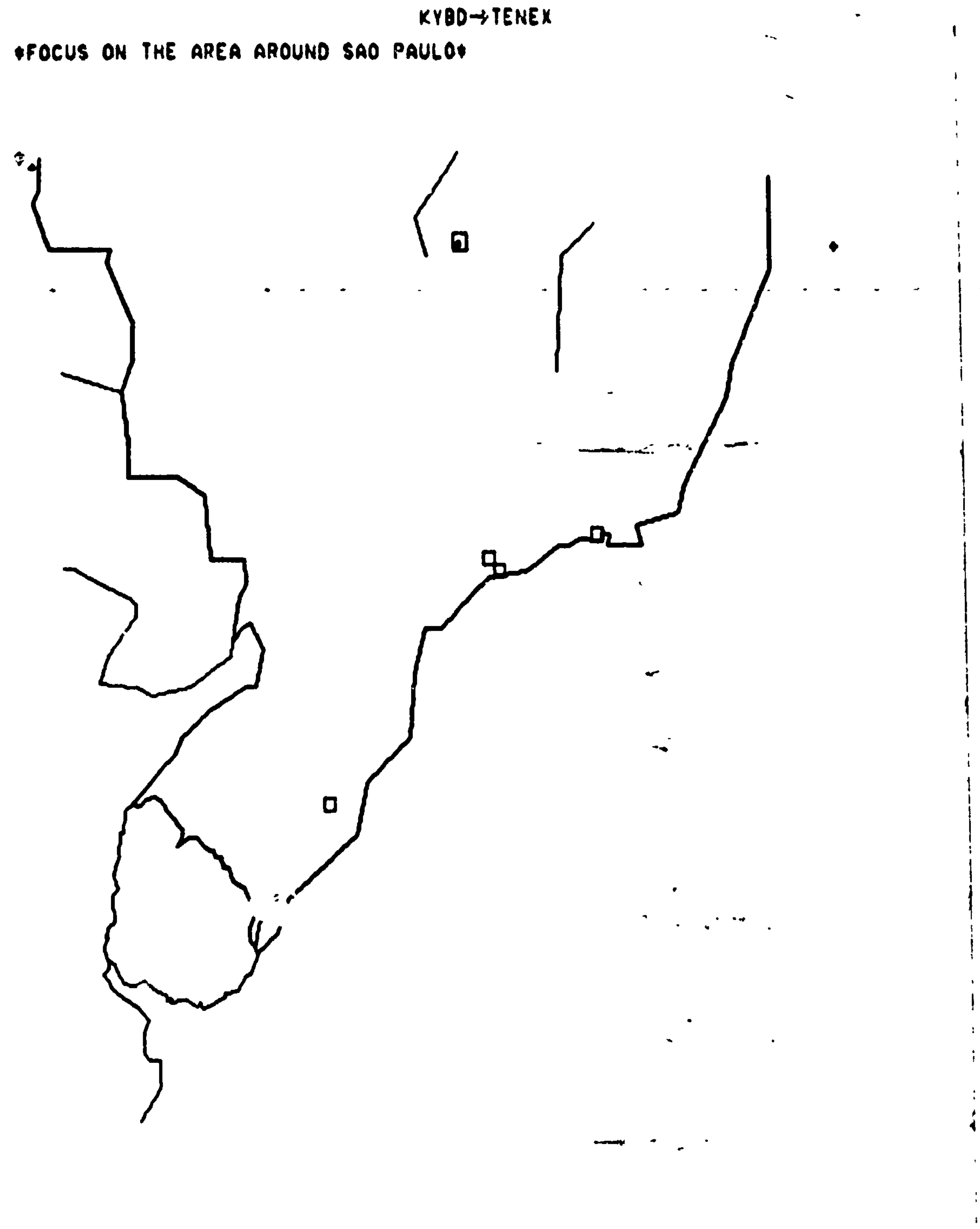

FIGURE 9f. Output from Map Interaction with SCHOLAR.

KYBD→TENEX

GO BACK TO URUGUAY

THIS IS URUGUAY.

SHOW ME THE CITIES OF URUGUAY

WE HAVE ALREADY DISCUSSED THIS.

MONTEVIDEO, SALTO, PAYSANDU, RIVERA, AND PUNTA DEL
ESTE ARE BLINKING.

*

*

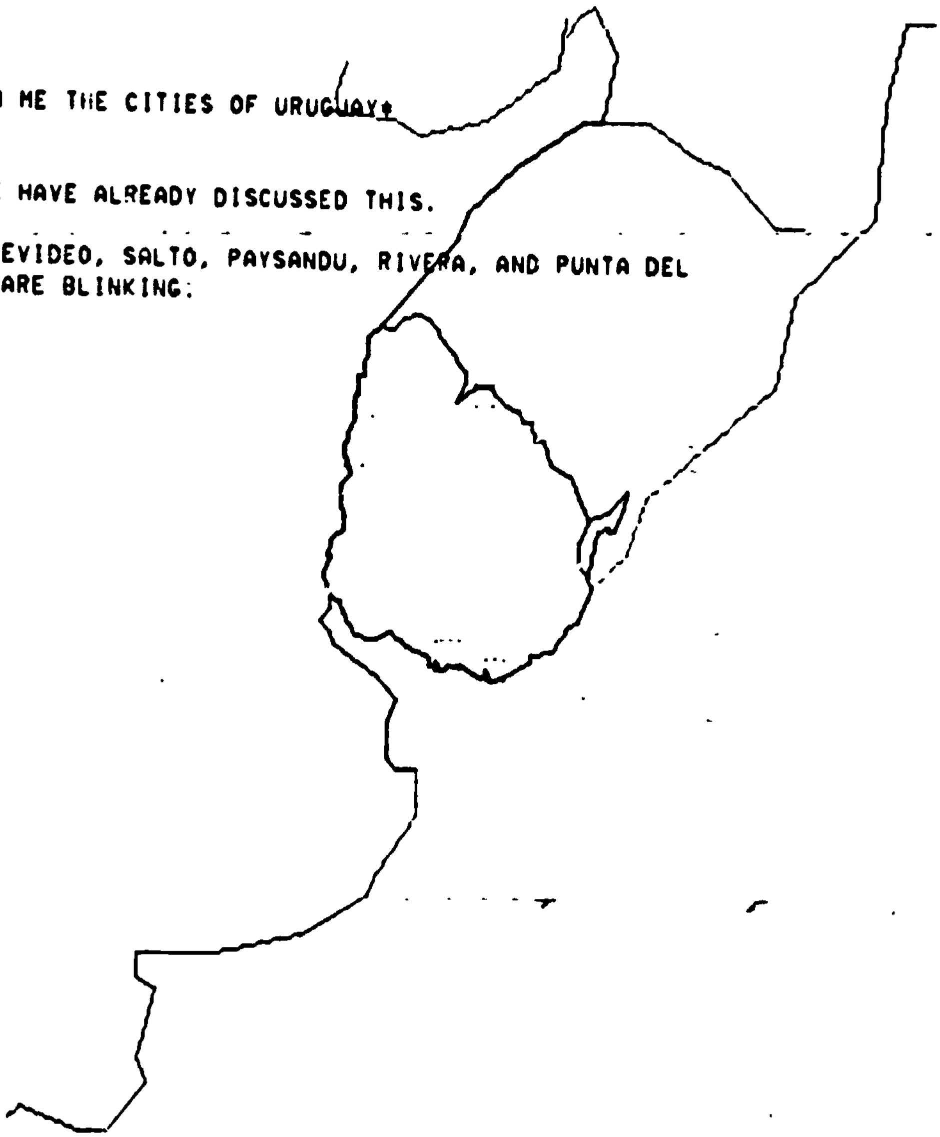

FIGURE 9g. Output from Map Interaction with SCHOLAR.

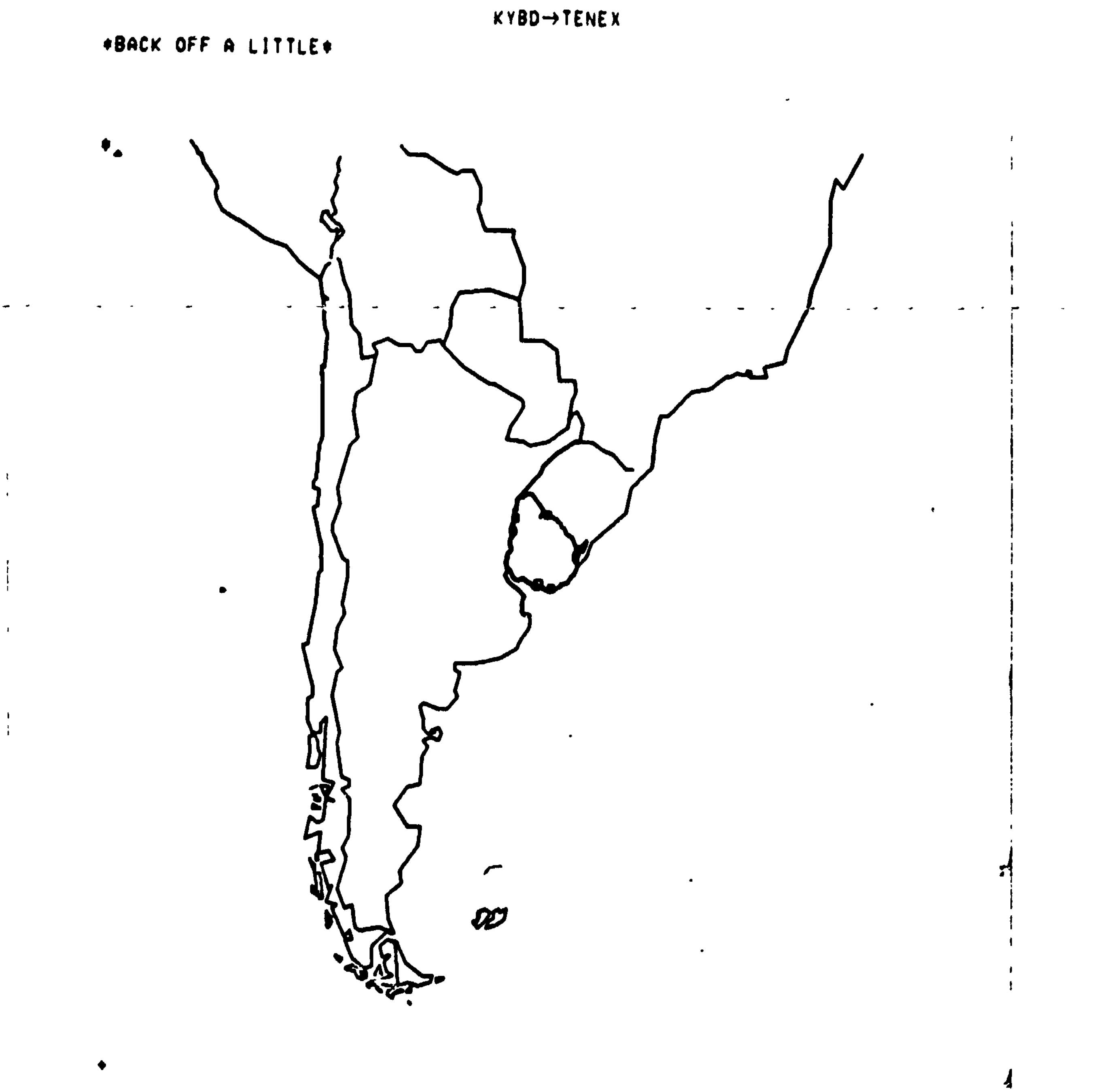

FIGURE 9h. Output from Map Interaction with SCHOLAR.

Interpretation of pen pointing similarly requires SCHOLAR to be conscious of the rest of the input. (The user can point at things on the screen by means of knobs or a "mouse." See Figure 10 for a protocol.) For example, if a city in a country is on a river, pointing at it would also hit the other two, so the verbal input is searched for hints, sometimes a direct word like "city," and sometimes a clue like "length." Then of all the things that are currently being displayed, only the cities are made target sensitive, and so the city is found, as was intended. The same thing happens if the user is careless and doesn't hit the city precisely: SCHOLAR knows that it wants to find a city and will look around to see if there is one nearby, rather than settling for the river or country.

Another problem comes up in a question like "What river is around here?" where the area on the map shows nothing. The student is asking what would be there if it were there. It is clearly impractical to search every item in the graphic data base to see if it could be the thing intended, so SCHOLAR first figures out what kind of thing it is looking for (here, rivers) and then searches its list of things in the area for rivers.

After SCHOLAR has decided what has been pointed at, it processes the rest of the question just as if the user had said the name instead of pointing.

SCHOLAR's map-handling ability has been developed to respond to direct requests and questions by the user. An obvious extension of this, which we are implementing under other support, is to have SCHOLAR ask questions and manipulate maps on its own initiative.

(1) *WHAT IS THIS?*

IT IS SAO PAULO.

(2) *IS THIS THE CAPITAL OF BRAZIL*

YES.

(3) *WHAT RIVER IS AROUND HERE*

IT IS THE AMAZON.

(4) *GIVE ME THE POPULATION OF THIS CITY*

THE POPULATION OF RIO DE JANEIRO IS APPROXIMATELY 4,700,000 PEOPLE.

FIGURE 10. Interaction with SCHOLAR Showing Dialogue Accompanying Pointing at Display.

We had started implementing the map-handling capability using our E&S (Evans and Sutherland) display processor. After repeated processor malfunctions, we decided to develop a second implementation in an IMLAC display processor. We designed the IMLAC display interface so that at the top level it would look like the E&S interface. Since the IMLAC is more restricted than the E&S in its hardware capabilities, many software routines were written to do the tasks previously done automatically by the hardware.

One of them clips the portions of the figure to be displayed which are outside the screen, eliminating the wrap-around generated by the IMLAC. It can also be told to display not what should be at the center of the screen, but one of the wrapped around images. This allows for the accessing of any part of a display of almost unlimited size.

Another development is the SIMHIT routine for figuring out what the user is pointing at. Given a point, it tests the point's coordinates against each figure that is a target candidate (figures that would be made "target sensitive" for hardware). If the figure under consideration is a line or a point, the routine finds out whether the user's point is within a certain small distance. If the figure is an area, the routine breaks the area into narrow trapezoids using an internal grid of hatched lines and determines whether the point lies in one of them.

The SIMHIT routine can also be used for calculations of things not necessarily related to the display. For instance, it could see whether a city is in a given country, or in general whether any two areas intersect. It turns out that the best procedure using the semantic network to answer such questions is cumbersome, and in some cases it cannot be certain of a "No" answer where a map intersection routine can. We might add that people often report using image processing to answer questions of this kind, such as "Is Algeria in Africa?" or "Is San Francisco in Nevada?".

The important contribution of the work on Map-SCHOLAR is the close integration of visual and semantic information. Because units in the maps (e.g. the Amazon, the Amazon delta, the border of Chile and Argentina) are also units in the semantic network, it makes it possible to refer to places either by pointing to them, by naming them, or both. To date, this capability has only been developed in answering questions and in responding to commands about maps. However, in future work we plan to exploit its potential for simulating human image processing, and for tutoring visual and semantic information in an integrated manner.

ENGLISH COMPREHENSION

Our work on English comprehension and representation of actions was done in NET-SCHOLAR, a SCHOLAR-type system that answers questions about the ARPA computer network. Like regular SCHOLAR, its data base is in the form of a semantic network, though the information is about the ARPA network instead of geography. Because most of what a user wants to know about the network is procedural in nature, verbs are crucial in this system, and it is a good environment for dealing with verbs and actions. We have developed in NET-SCHOLAR an ability to handle verbs and verb relations in understanding the user's questions and in formulating answers.

A case grammar representation is used for verbs. This is following Fillmore's (1968) usage of "cases" to refer to the semantic relations of nouns to a verb. 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 11. 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

```
CTRL-A\COMMAND
     SUPERC (I 0) EDITING\COMMAND
     SUPERP (I 0) EXECUTIVE
     PURPOSE (I 0)
          DELETE (I 0)
               AGENT (I 0) USER
               OBJ (I 0) CHARACTER (I 0) LAST
               INSTR (I 0) CTRL-A\COMMAND
               LOC (I 0) INPUT\STRING

DELETE
     SUPERC (I 0) EDIT
     CASES (I 6 B)
          AGENT (I 0) USER
          OBJ (I 0) DATA FILE JOB
          INSTR (I 0) PROGRAMMING\LANGUAGE PROGRAM
               COMPUTER\SYSTEM JSYS EDITING\COMMAND COMMAND

COMPUTER\SYSTEM
     SUPERC (I 0) SYSTEM
     SUPERP (I 0) COMPUTER\CENTER
     ENTER (I 2)
          AGENT (I 0) USER
          INSTR (I 0) ARPA\NETWORK
          OBJ (I 0) COMPUTER\SYSTEM
          PROCEDURE (I 0)
               ($SEQ CALL (I 0)
                          AGENT (I 0) USER
                          OBJ (I 0) TELNET
                     TYPE (I 0)
                          AGENT (I 0) USER
                          OBJ (I 0)
                             NAME (I 0)
                                OF (I 0) COMPUTER\SYSTEM
                     LOGIN (I 0)
                          AGENT (I 0) USER
                          INSTR (I 0) LOGIN\COMMAND
                          LOC (I 0)
                             TO (I 0) COMPUTER\SYSTEM)
     EXAMPLES (I 4)
          ($EOR MULTICS BBN-TENEX RAND-RCC SRI-ARC UTAH\10)

ENTER
     CASES (I 6 B)
          AGENT (I 0) USER
          INSTR (I 0) COMMAND SUBSYSTEM COMPUTER\NETWORK
          OBJ (I 0) COMPUTER\SYSTEM OPERATING\SYSTEM
```

FIGURE 11.   Some Partial Data Base Entries in NET-SCHOLAR.

within it. Purposes, conditions, side-effects, etc., are also stored in this framework.

NET-SCHOLAR's processing of a question is divided into four parts--parsing, case assignment, retrieval, and sentence-generation. 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 12.

Next, case assignment figures out the relation of each noun phrase to the main verb of the sentence. The output is the parse tree with the addition of a case label at the beginning of each noun phrase (NP) expression. In the first sentence in Figure 12, "what command" has been labelled as an instrument, and "a character" is an object.

Case assignment bases its decisions mostly on semantics. It uses the Match-on-Superordinate routine (described earlier), which compares two concepts to see if they could refer to the same thing. It tries matching the main noun in each noun phrase, against the nouns in the cases stored with 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.

```
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 12. Sentences after Parsing and Case Assignment.

In addition to Match-on-Superordinate, case assignment uses syntactic clues, such as the presence of certain prepositions ("with" for an instrument) or the noun's position in the sentence. It also uses Match-on-Superpart 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 the case nouns stored there against the case nouns in the sentence. For example in Figure 11, 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 13. 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. This is the basic procedure, though there are also a lot of special things to handle.

Retrieval 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.

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.

FIGURE 13. Actual Questions with Answers by NET-SCHOLAR.

When the information to answer a question has been found, all that remains is for the sentence-generation routine to put it into sentence for' and print it out. To make a sentence from a piece of data base, the routine finds the main verb, arranges t' cases in the appropriate order for that verb, adjusts the su t and verb to be singular or plural, and puts in the necessary articles, prepo itions, etc. When the piece of information is complex and embedded, several sentences may be made, as in the second example in Figure 13.

In Figure 14, there is a sample piece of information and the sentence produced from it. DELETE is a regular verb i 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."

```
CTRL-A\COMMAND (I 0)
     DELETE (I 0)
          OBJ (I 0)
               CHARACTER (I 0) LAST
          INSTR (I 0) CTRL-A\COMMAND
          LOC (I 0) INPUT\STRING
```

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

FIGURE 14. Example of Input and Output of Sentence Generation.

REFERENCES

J. R. Carbonell, "A.I. in CAI: An Artificial-Intelligence Approach to Computer-Assisted Instruction," IEEE Trans. on Man-Machine Systems, Vol. MMS-11, No. (December 1970)

J.R. Carbonell, "Artificial Intelligence and Large Interactive Man-Computer Systems," Proceedings of 1971 IEEE Systems, Man, and Cybernetics Conference, Anaheim, California (October 1971)

J.R. Carbonell and A.M. Collins, "Natural Semantics in Artificial Intelligence," Proceedings of Third International Joint Conference on Artificial Intelligence, Stanford, California (August 1974).

A.M. Collins and M.R. Quillian, "Retrieval Time from Semantic Memory," Journal Verb. Learn. Verb. Behav., Vol. 8, No. 2 (April 1969).

A.M. Collins and M.R. Quillian, "Facilitating Retrieval from Semantic Memory; The Effect of Repeating Part of an Inference," in A.F. Sanders (ed.) Attention and Performance III, North Holland Pub. Co., Amsterdam (1970).

C. Fillmore, "The Case for Case" in Bach and Harms (eds.), Universals in Linguistic Theory. Holt, Rinehart, and Winston,N.Y. (1968)

M.R. Quillian, "Semantic Memory" in M.L. Minsky (ed.) Semantic Information Processing, MIT Press, Cambridge, Mass. (1968).

M.R. Quillian, "The Teachable Language Comprehender: A Simulation Program and Theory of Language", CACM, Vol. 12, No. 8 (August 1969).

BEST COPY AVAILABLE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER BBN Report No. 2833	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SEMANTIC NETWORKS		5. TYPE OF REPORT & PERIOD COVERED FINAL REPORT covering 15 April 70 to 31 Dec 1973
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(S) Allan M. Collins Eleanor H. Warnock		8. CONTRACT OR GRANT NUMBER(S) No. N00014-70-C-0264
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bolt Beranek and Newman Inc. 50 Moulton Street Cambridge, Mass. 02138		10. PROGRAM ELEMENT PROJECT TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Information Systems, Mathematical & Information Sciences Division (cont.below)		12. REPORT DATE May 1974
		13. NUMBER OF PAGES 55
14. ~~MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)~~ Office of Naval Research, Code 437 Department of the Navy, Washington, D. C.		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this report is unlimited.

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

18. SUPPLEMENTARY NOTES

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

cognitive processes
inference
human memory
natural language understanding
question-answering systems
artificial intelligence
deduction
semantics
analogy
~~visual processing~~

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

Our work on semantic networks under contract N00014-70-C-0264 with Office of Naval Research, Information Sciences Division involved three distinct areas: inferences, map displays, and English comprehension. The inference strategies implemented in SCHOLAR include different types of deductive, negative, and functional inferences. The graphics package (cont.on back page)

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (continuation of ABSTRACT)

allows users to ask questions and give commands in English to control SCHOLAR's map display, which is tied into the semantic network on South American geography. With partial support from this contract, we also developed an English Comprehension System, utilizing a data base on the ARPA network. Unlike geography, most questions about the ARPA network pertain to actions and procedures, which involve complicated English sentence structure, and hence necessitate sophisticated parsing and retrieval strategies. This report describes our work in each of these three areas.