UNIVERSITY of PENNSYLVANIA

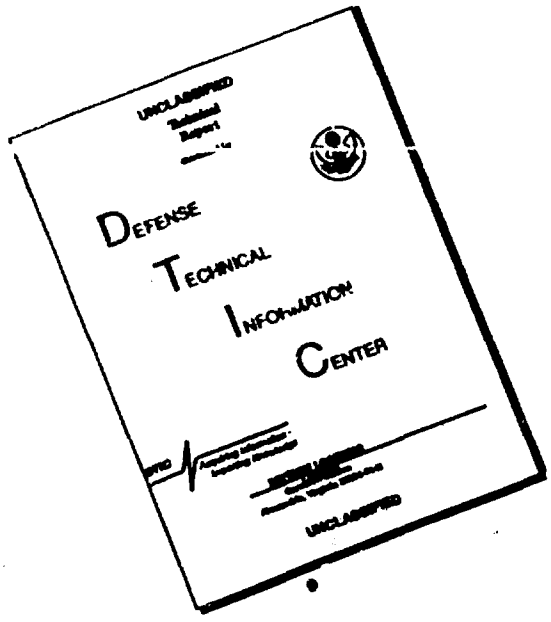
The Moore School of Electrical Engineering

PHILADELPHIA, PENNSYLVANIA 19104

AF 49(638)-1421

16

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

University of Pennsylvania
THE MOORE SCHOOL OF ELECTRICAL ENGINEERING
Philadelphia, Pennsylvania

REAL ENGLISH PROJECT REPORT

by

Harvey Cautin
and
Edward Regan

November, 1967

The work described in this paper has been supported by the Air Force Office of Scientific Research, Information Sciences Directorate (system studies) and by the Army Research Office-Durham (implementation tasks).

The Moore School Information
Systems Laboratory

University of Pennsylvania
THE MOORE SCHOOL OF ELECTRICAL ENGINEERING
Philadelphia, Pennsylvania

The Moore School Information Systems Laboratory

University of Pennsylvania

Principal Investigator
Morris Rubinoff

Participating Faculty

Pier L. Bargellini
John W. Carr, III
Aravind K. Joshi
James F. Korsh
George E. Rowland
Richard F. Schwartz
Warren D. Seider

Student Staff

S. Bergman
S. Bruckner
H. Cautin
T. Closs
J. Crowley
D. Deily
A. Eliasoff
I. Ellner
B. Everloff
M. Fogel
W. Franks
L. Haynes
G. Ingargiola
T. Johnson
A. Libove
J. Lucas
S. Mitrani
T. Purdom
E. Regan
S. Soo
V. Stein
D. Stone

University of Pennsylvania
THE MOORE SCHOOL OF ELECTRICAL ENGINEERING
Philadelphia, Pennsylvania

REAL ENGLISH PROJECT REPORT

The goal of Real English is to develop an information retrieval system with man-machine communication through a teletypewriter enabling the two parties to converse in English. The system should be able to perform, among other things, the functions of a librarian, i.e., 1) to inform the user of the system structure, 2) to teach the user to get information, and 3) to aid in clarifying vague terms or ideas. Also, the system should be able to update itself (i.e., add information to, or delete information from, the data file which is made up of documents, index terms, synonyms, their expansions, classification tables, and a thesaurus) and to keep track, by means of statistics, of how the system is being used so as to improve its performance. The user is free to use any dialogue he chooses. It is the responsibility of the system to determine with which part of the system the user is trying to communicate. The request is then translated into a command capable of performing the task. The set of such commands comprise the Symbolic Command Language of the Real English system.

The Real English system flowchart is shown in Figure 1. An explanation follows:

The user enters his message (1)* through a remote teletypewriter. A proof reading and erasing mechanism is provided to enable the user to change or correct his message.

---

* The numbers (i) refer to the numbered boxes of Figure 1.

This message along with a grammar and a word dictionary (i.e., a listing of words with their grammatical categories along with additional information) is used as input to the syntax analyzer (2). This analyzer attempts to parse the sentence into grammatical strings. The output parse is in the form of a tree which gives the following information:

1. type of sentence - interrogative, imperative, declarative.
2. index terms - It is anticipated that index terms will not be placed in the dictionary. Therefore, upon recognition of an unlisted word, a man-machine dialogue is initiated to determine if the word is an index term; if it is not, then a synonym must be provided. Note is made of index term locations since all such words have the same grammatical entry.
3. sentence components - The string names which comprise the syntax of the sentence are established along with their interrelationships. These string names will aid in locating the specific parts of speech, e.g., verb, adjective, etc.
4. word analysis - Each word of the sentence is given its grammatical category and its location in its defining string.

Assuming that a unique parse is obtained with no homographic ambiguity (i.e., no word appears with more than one possible meaning) the sentence is semantically interpreted to find its mode of inquiry (4) and the specific command being summoned. The sentence is then operated upon by that command routine which restyles the user's message into a form recognizable to the specific command of the symbolic command language.

The request is then executed (7) and the user again gains control (1).

In the event of homographic ambiguity, communication will be set up with the user in an effort to resolve the ambiguity (8). The system might supply both definitions (assuming it is doubly ambiguous) to the user and have him decide which is the intended meaning. Alternatively the system might present a set of broader or narrower or related terms for each possible meaning and again have the user make a decision based on this information.

When more than one parse is obtained (3) another type of ambiguity arises which will be settled through computer-directed man-machine dialogue. One solution is to interpret each parse and then let the user say which interpretation (i.e., which command) was intended. There is a good chance that although more than one parse is obtained, they will have the same semantic interpretation as far as command execution is concerned. For example, in the term "steel mill", it is immaterial whether "steel" is considered an adjective or a modifying noun.

If no parse is obtained (5), it is possible that the user is shortening his request assuming that the system knows the intended context of his message. For example, the following dialogue may take place (the primed numbers indicate the system's response).

1. Give me everything written by Allen.

1'. We have 4 references.

2. How about Wilson?

2'. No references.

3. And Stevens?

3'. 1 reference.

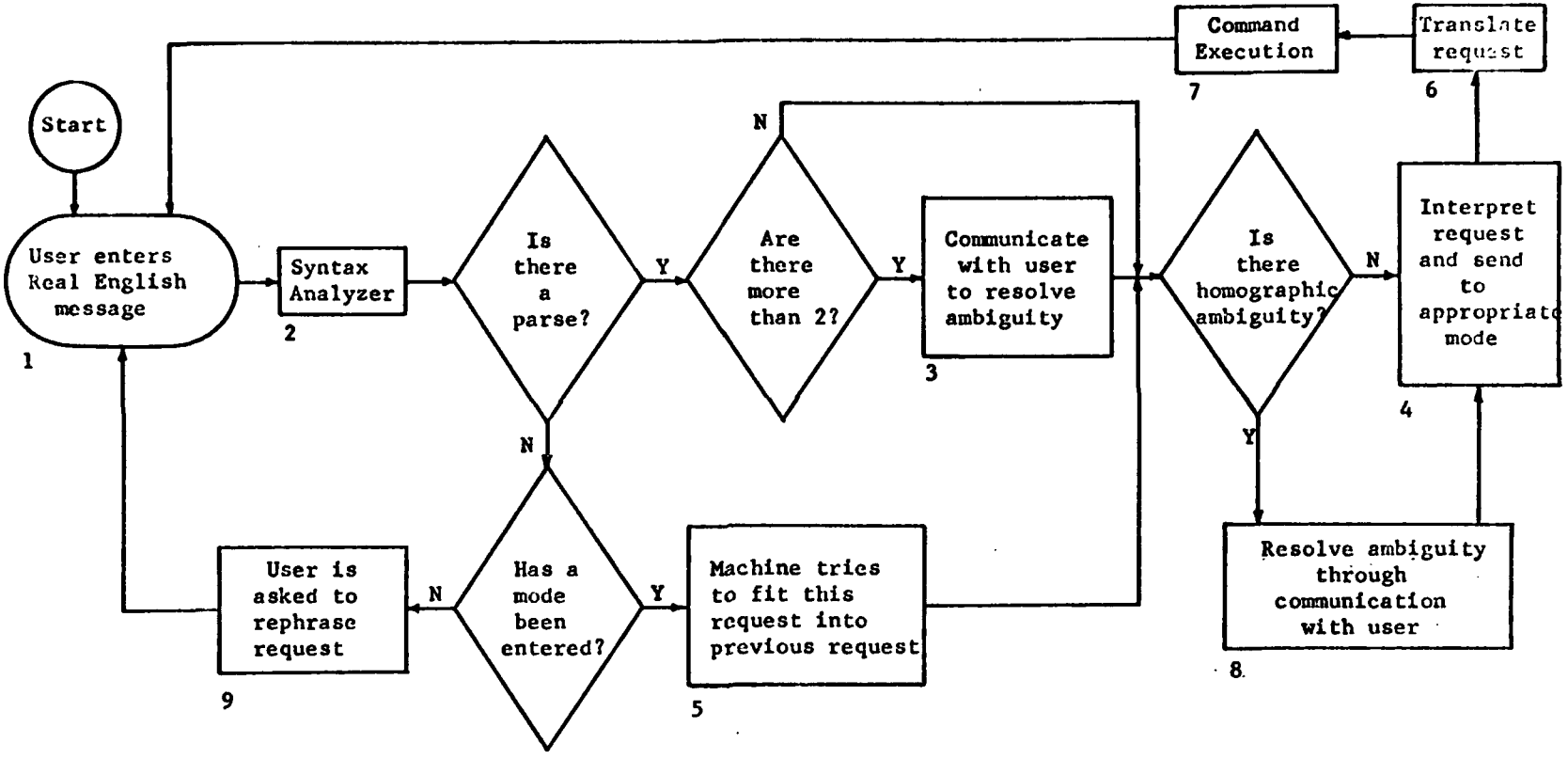


Figure 1

Real English Flowchart

In addition to the problems of homographic ambiguity, multiple parses, and incomplete sentences discussed above, the Real English system must anticipate the problem of contextual ambiguity and the problem of determining the meaning of complete sentences that refer to previously entered sentences.

Contextual ambiguity is caused by the English language's high degree of context sensitivity. The meaning of a clause is not independent of its context in the sentence. Consider the following sentence which might be a typical command to the Real English system:

I would like all the documents by Carr.

Here the user is making a request for the retrieval of all the documents in the system data base that were written by Carr. By adding the word "purged" at the end of this sentence, we obtain:

I would like all the documents by Carr purged.

Although the first eight words of the above sentence are identical to those of the first sentence, the addition of the word "purged" has completely changed the meaning of those first eight words. Now the user desires that the system delete from its data base all documents that were written by Carr.

The problem of determining the meaning of complete sentences that refer to previously entered sentences is most easily explained with an example. Consider the following dialogue:

Do you have any documents co-authored by Carr and Gorn?

YES

Let me have them.

"Them" in the above sentence refers to "documents co-authored by Carr and Gorn". The Real English system will have to be able to recognize that the meaning of the object "them" is to be found in a previous inquiry. Another problem involving complete sentences can also be shown with the above dialogue. Simply change the second user command to:

Let me have all the documents co-authored by Carr and Gorn.

At this point the system has already retrieved all the documents requested and it should not duplicate its retrieval a second time.

To implement the Real English system, several sentences for each anticipated mode of operation were written. From these, sentence commands are to be devised which would perform the functions implied by the sentences. The sum total of these commands will institute a symbolic command language which should be able to fit in, as closely as possible, with the present retrieval system. The various modes with their commands follow:

SEARCH Mode

1. RETRIEVE - Retrieves documents satisfying a criterion made up of bibliographic data and subject matter. The user may use any logical combination of such terms with the "and", "or", and "and not" logical operators

FORMAT - RETRIEVE a_1 b_1 c_1 a_2 b_2 c_2 a_3 b_3

where a_i - is a section code designating any one of the above categories plus abstracts

b_i - is a term from any category

c_i - a logical operator

Parentheses may be used to express more complex logical constructions. The output of such a command is a list of accession numbers which satisfy the logical expression of the request.

SENTENCES - Let me see the information on graph theory written by Allen.

What do you have about graph theory?

I would like something on graph theory.

Could I see the material on graph theory?

Do you have anything listed under graph theory, Allen and trees?

2. COMBINE - Given a set a = n ($n \leq 8$) descriptors, COMBINE will determine how many documents have been indexed under exactly n, n - 1,...,1 of the descriptors.

COMBINE $a_1/a_2/.../a_n$

a - a descriptor consisting of a section code and an index term. If a section is omitted from an index term, the last previous section code is associated with this term.

SENTENCES - Let me have anything indexed by any of the following: A,B,C, or D.

What do you have on A,B, or C?

I would like anything on A or B or C.

3. CLUSTER - Presents a list of accession numbers which satisfy a comparative request (greater than 8, less than or equal to 3) of index terms.

CLUSTER - n, M, $d_1/d_2/.../d_j$ $\quad 1 \leq j \leq 8$

n - a number from 1 to 8

M - a submode G - greater than

GE - greater than or equal to

E - equal

LE - less than or equal to

L - less than

d_j - a descriptor consisting of a section code and an index term. If a section code is omitted from an index term, the last previous section code is attached to this term. No connectives (i.e., and, or, but not) are permitted with this command. The output is a list of accession numbers representing the documents which are indexed by M,n index terms. For example, if M = G and n = 2, the list would correspond to all documents indexed by greater than 2 of the listed index terms.

SENTENCES - I would like all material indexed by more than those of the following: A,B,C or D.

Give me documents characterized by any two of the following terms: A,B,C,D.

Do you have anything listed under any one of the following: A,B,C,D and E?

4. INDEP - Search in depth.

A list of accession numbers satisfying the RETRIEVE, CLUSTER, or DISPLAY request is first obtained. The information corresponding to the section codes specified in the INDEP command is then given to the searcher.

INDEP (n_1, n_2,..., n_i) (command)

where n - any section code

command - either a complete RETRIEVE, CLUSTER or DISPLAY request.

SENTENCES - Give me the title of all references written by Wilson on graph theory.

I would like title, author and date of all works on graph theory appearing in the CACM.

List the author of all papers on citation indexing.

DICTIONARY Mode

DEFINE (a_1, a_2, a_3) (t_1/ t_2 .../ t_i)

where: a_i - refers to the level of expansion, i.e., L = 1 may be a one-line definition, L = 2 may be a paragraph description, L = 3 may be an illustrative example.

t_i are terms making a definition.

Operation: The system will extract the a_i's of the dictionary record of t_1, t_2,..., t_i. The system will ask if the user wants a further explanation. If yes, the next higher level is retrieved for the user.

Structure: Each dictionary word has its own record in the file.

SENTENCES: What does radar mean?

Do altitude and attitude mean the same thing?

Can A and B be used as synonyms?

Give me an example of an interpretative program.

What is radar?

For each word in a thesaurus, generic terms, specific terms and the word used for indexing, if the given word is not, are given. Basically, a thesaurus consists of a group of classification tables along with the indexing term. For this reason, all queries pertaining to any classification table will be considered in the TABLES mode.

TABLES Mode

ELEMENT command will retrieve all left or right elements of a term in a given relation.

ELEMENT (a_1, a_2, a_i) (r_1, r_2, r_j) $(t_1, t_2, \ldots, t_k)$

where: a_i - is a number designating left elements, right elements or indexing terms (i.e., authority list entry used for the term)

r_j - designators for the various relations

t_k - terms being investigated

Structure - Every term in each relation gets a record in the file. Within each record, entries for the various designators are placed.

SENTENCES - Are A and B related at all?

What is A generic to?

What is A a synonym of?

What should I use for radar? (e.g., of pragmatic ambiguity)

STATISTICAL Mode

Data Structure - The data will be organized in a linked list structure. There will be a sublist corresponding to each index term. The first element of this sublist will contain a count of the number of times the term was used. The remaining elements will contain information on each document that is retrieved using that index term. Specifically,

these elements contain a count and a pointer to the next index term used with that document. There are also nodes for every document. They contain counts of the number of times the documents were retrieved and a pointer to the most frequently used index term.

1. TIMES (n_i, x_1, x_2,...)

   where: n_i = 1 means number of time each of the documents x_1, x_2,... has been retrieved.

   n_i = 2 means number of times the index terms were used.

   n_i = 3 means number of times the system was used.

   x_j = Either document number or index term depending on value of n_i.

   Operation: Depending on n_i, the system will go to the node corresponding to the document, index term, or system and extract the count of the number of times it was used.

   SENTENCES - How often was document 113 retrieved?

   Approximately how many requests for information do you get per day?

   Give me a count of the number of times each of these was used _____, _____, and _____?

   How many times was "Computer Logic" by Jones requested?

   Has the FORTRAN IV Language Specifications Manual been retrieved more than 2 times?

2. TERMS (d_1, d_2, d_3,...)

   where: d_i are document numbers.

   Operation: For each d_i the system will search through the linked list and return the index terms used to retrieve d_i and the number of times the specific index term was used.

SENTENCES - Were any of these terms used in the retrieval of

Programming by Smith: _____, _____, and _____?

What terms were used to retrieve _____?

Has _____ been used as an index term to retrieve document 100?

Was machine language ever used as an index term used to retrieve?

3. DOCS (IT_1, IT_2, IT_3,...)

where: IT_i are index terms.

Operation: The system will return all the documents that have been retrieved using index term IT_i and a frequency count of the number of times IT_i was used for each document.

SENTENCES - What documents have been retrieved using "FORTRAN"?

Has document 201 been retrieved with "FORTRAN" used as an index term?

How many documents has "ALGOL" been associated with?

Was reentrant ever used as an index term to retrieve the IBM S/360 Assembly Language Manual?

In order to get some feel for the syntax analyzer output for our sentence set, so as to develop a semantic analyzer, the sentences were parsed by hand and the various strings of the grammar constituting the parse were noted. This analysis showed that the set of sentences can be analyzed by about 40 strings many of which are combinations of others in the set. There were about 10 object strings. The set of sentences produced about 128 distinct words excluding all index terms. It is proposed that by properly classifying verbs with their objects, a semantic interpretor can be developed.

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*

University of Pennsylvania
Moore School of Electrical Engineering
Philadelphia, Pennsylvania 19104

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

REAL ENGLISH PROJECT REPORT

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

scientific; interim

5. AUTHOR(S) *(First name, middle initial, last name)*

Harvey Cantin and Edward Regan

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
November, 1967	12	

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
AF 49(638)-1421	
b. PROJECT NO. 9769-01	
c. 61102F	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d. 681304	AFOSR 68-1838

10. DISTRIBUTION STATEMENT

1. This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
TECH, OTHER	Air Force Office of Scientific Research Directorate of Information Sciences Arlington, Virginia 22209

13. ABSTRACT

Requirements are discussed for an information retrieval language that enables users to employ natural language sentences in interaction with computer-stored files. Anticipated modes of operation of the system are outlined. These are: the search mode, the dictionary mode, the tables mode, and the statistical mode. Analysis of sample sentences parsed manually indicate that sentences can be machine analyzed by about 40 strings, many of which are combinations of others. The sentences produced about 190 distinct words excluding all index terms. It is suggested that, by properly classifying verbs with their objects, a semantic interpreter can be developed.

DD FORM 1 NOV 65 1473

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Information retrieval Machine language Linguistics Semantics Natural language Man-machine communication						