AN INTERACTIVE DOCUMENT RETRIEVAL SYSTEM

D.E. CARROLL, R.T. CHIEN, K.C. KELLY, F.P. PREPARATA
P. REYNOLDS, S.R. RAY & F.A. STAHL

UNIVERSITY OF ILLINOIS – URBANA, ILLINOIS
This work was supported in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under Contract DAAB 07-67-C-0199 and in part by OE C-1-7-071213-4557.

Reproduction in whole or in part is permitted for any purpose of the United States Government.

Distribution of this report is unlimited.
AN INTERACTIVE DOCUMENT RETRIEVAL SYSTEM

D. E. Carroll    P. Reynolds
R. T. Chien      S. R. Ray
K. C. Kelley     F. A. Stahl
F. P. Preparata

Coordinated Science Laboratory
University of Illinois, Urbana, Illinois

ABSTRACT

In this paper the REQUEST document retrieval system is described. Included is a description of the data base, the interactive mode, the modular implementation in ISL—a string manipulating language, and the multiple-level Boolean hierarchy query format.

*This work was supported in part by the Joint Services Electronics Program (U. S. Army, U. S. Navy, U. S. Air Force) under Contract No. DAAB-07-67-C-0199 and in part by OE C-1-7-071213-4557.
1. Introduction

The document retrieval system at Coordinated Science Laboratory is implemented on a Control Data 1604 computer system. It is interactive in nature and possesses several unique features.

Recorded information from each source article includes 1) a rather complete list of conventional bibliographic data such as authorship, author affiliation, title, year of publication, etc., 2) detailed bibliographic data from each of the cited references, and 3) citations in context, i.e., the sentence (or sentences) which contain the citations. The source articles are selected from the IEEE Transactions on Computers, Journal of the Association for Computing Machinery, and Documentation.

The searches are conducted in interactive manner through a REQUEST system, which is a man-machine interface language. Via the simple formats of REQUEST the user provides information concerning the kinds of documents he wishes to obtain. The format employed in REQUEST is a multiple-level Boolean hierarchy which provides maximum flexibility. Very little knowledge about the system is required of the user.

The searches are conducted in another language called ISL for Information Search Language. ISL is a string-possessing language similar to SNOBOL. It is capable of searching variable-length information strings containing don't care symbols. Through imbedding of the ILLAR Assembly Language, ISL also has acquired all necessary bookkeeping functions as well as some power for doing arithmetic.

The searches are conducted serially at the moment due to the lack of any mass storage component. When a document is found, its image is displayed on the screen of the display unit. Several options of retaining the
information are now open to the user, such as printing, photography or storage on another tape. He may indicate his commands with a light pen. The system is quite flexible and can carry out almost any search strategy the user wishes to employ. The essential limitation in speed is now imposed by the serial organization as implemented on tapes. Some of the more advanced search strategies will be applied when a disc-file is available. [4]

In Section 2 of this report a brief description of the data base is given, insofar as to make this document somewhat self-contained. A more detailed account of this work is given elsewhere. [1]

Section 3 describes the REQUEST system from a systems point of view; Section 4 describes the Information Search Language.

Reports that describe the detailed implementation of both ISL and REQUEST are available as separate documents. [2],[5]
2. The Data Base

The data base for the REQUEST system consists of a rather complete list of information items for each document. Each document file contains one group of source items and a number of groups of citation items. Each item is a variable-length string that is prefixed by an item-type marker. The source items contain all the bibliographic information for the source document. Table 2.1 lists the permissible source items and their associated item-type markers.

There is a group of citation items associated with each entry that the source document cites. Each group of citation items is prefixed by an item-type marker (CR or CF) and the number of the cited work. The permissible citation items are the same as the source items with the addition of the citation-context string (item-type marker ST). The citation-context string is the textual material that the source author has associated with the cited document.

Item-type markers are followed with a number if there is multiple occurrence of that marker within that item group. So, for example, if a source article was co-authored, then the item-type markers associated with the authors would be AP1 and AP2.

The Appendix contains a typical information item for a document.
<table>
<thead>
<tr>
<th>ITEM-TYPE MARKER</th>
<th>TYPE OF INFORMATION ITEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>author (corporate class)</td>
</tr>
<tr>
<td>AE</td>
<td>author (editor class)</td>
</tr>
<tr>
<td>AP</td>
<td>author (personal class)</td>
</tr>
<tr>
<td>AT</td>
<td>author (translator class)</td>
</tr>
<tr>
<td>CH</td>
<td>chapter number</td>
</tr>
<tr>
<td>DS</td>
<td>descriptor (assigned or derived class)</td>
</tr>
<tr>
<td>ED</td>
<td>edition number</td>
</tr>
<tr>
<td>FA</td>
<td>affiliation of the author (institutional association)</td>
</tr>
<tr>
<td>IT</td>
<td>identification number</td>
</tr>
<tr>
<td>JL</td>
<td>journal (or other work containing the source article)</td>
</tr>
<tr>
<td>KP</td>
<td>acknowledged person or group</td>
</tr>
<tr>
<td>LA</td>
<td>language of title</td>
</tr>
<tr>
<td>LF</td>
<td>literary form (book, translation, patent, etc.)</td>
</tr>
<tr>
<td>LT</td>
<td>language of text (if different from that of title)</td>
</tr>
<tr>
<td>MO</td>
<td>month (Jan., Feb., etc.) or season (fall, winter, etc.)</td>
</tr>
<tr>
<td>NR</td>
<td>number of issue</td>
</tr>
<tr>
<td>PA</td>
<td>page numbering (beginning number and ending number)</td>
</tr>
<tr>
<td>PL</td>
<td>place of publication</td>
</tr>
<tr>
<td>PT</td>
<td>part number</td>
</tr>
<tr>
<td>PU</td>
<td>publisher</td>
</tr>
<tr>
<td>SP</td>
<td>sponsor of work</td>
</tr>
<tr>
<td>TI</td>
<td>title</td>
</tr>
<tr>
<td>VO</td>
<td>volume</td>
</tr>
<tr>
<td>YR</td>
<td>year of publication or issue</td>
</tr>
</tbody>
</table>

Table 2.1
3. Description of the REQUEST System

The REQUEST system is a series of interdependent interactive programs written in the ISL system. It receives, displays, and translates a user's query into a format that can interrogate the bibliographic collection accessible through the computer's bulk storage and then disseminate the results of the interrogation for display, printing, or storage on magnetic tape for later use.

In order to best illustrate the use of this system, we give a number of annotated examples.

3.1. Verbal description: Find all documents in the collection that cite articles by Borko.

<table>
<thead>
<tr>
<th>Typewriter</th>
<th>Display</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>go, request.</td>
<td></td>
<td>User initiates system</td>
</tr>
<tr>
<td>STATE YOUR REQUEST</td>
<td></td>
<td>REQUEST responds and asks the user for a query. The user states that the desired information is located in the citation part of the document.</td>
</tr>
<tr>
<td>citation.</td>
<td>YOUR REQUEST IS: CITATION</td>
<td></td>
</tr>
<tr>
<td>and</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>citation=author.</td>
<td>YOUR REQUEST IS: (CITATION) CITATION=(AUTHOR)</td>
<td>REQUEST asks what to look for regarding the citation part. The user replies, &quot;look for the author.&quot;</td>
</tr>
<tr>
<td>and</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>author =borko.</td>
<td>YOUR REQUEST IS: (CITATION) CITATION=(AUTHOR)</td>
<td>REQUEST asks what to look for regarding the author. The user replies the author's name.</td>
</tr>
<tr>
<td>and</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(3)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. The form and content of this collection is described in Section 2.

2. In the examples that follow, the underlined portions are the responses of the REQUEST system, the "." represents a carriage return typed by user, and the "." is typed by the user to terminate a subdivisional response.
At this point REQUEST recognizes that the query has been completely stated and awaits a command from the user regarding the next phase of operation. The user may now (a) reformulate the query, (b) get a printed copy of the query as it appears on the scope, (c) get a Polaroid photo of the query as it appears on the scope, or (d) initiate the interrogation. The user may choose (b) or (c) as many times as desired by typing PRINT or PHOTO for each copy, and then initiate either (a) or (d) by typing ERROR or SEARCH.

The command SEARCH initiates the interrogation of the bibliographic collection. Upon finding a document which satisfies the query, the REQUEST system displays the document on the scope as Fig. 3.1:
The text represents the bibliographic material regarding the document that satisfied the query. Many times the entire text cannot be displayed on the scope, so the first two arrows are used to "roll" the text in scroll fashion in front of the user. The P takes a photo of only the textual material appearing on the scope. After sufficient examination of the bibliographic material the user may press the light pen against the right-most arrow to display a list of further options.

- RESTART
- CONTINUE
- EXIT
- HOLD
- PRINT
- TAPE7

These options are:

(a) **RESTART** - tells REQUEST that the user wants to formulate another query

(b) **CONTINUE** - tells REQUEST to look for another document that satisfies the current query

(c) **EXIT** - tells REQUEST to return the ILLAR monitor

(d) **HOLD** - tells REQUEST to restore current document as illustrated in Fig. 3.1

(e) **PRINT** - tells REQUEST to print the entire current document

(f) **TAPE7** - tells REQUEST to store the current document on magnetic tape 7

Thus, the user may build up a collection of desirable bibliographic references using the above-mentioned techniques. The user may choose to build up his collection on photos, printed copy, or magnetic tape. If he chooses the magnetic tape collection scheme, he may, by use of other available system routines, display, print, photograph, or duplicate onto another magnetic
tape any part of the contents of his collection.

3.2. Verbal description: Find all articles in the collection that cite either Borko or Jacobson.

The transmission of this query to REQUEST proceeds in the same manner as in 3.1, except for the part labeled (3) which, in this query, appears as follows:

<table>
<thead>
<tr>
<th>Typewriter</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3) <code>author =borko+jacobson. and ^</code></td>
<td><code>YOUR REQUEST IS</code></td>
</tr>
<tr>
<td></td>
<td><code>(CITATION)</code></td>
</tr>
<tr>
<td></td>
<td><code>CITATION=(AUTHOR)</code></td>
</tr>
<tr>
<td></td>
<td><code>AUTHOR =(BORKO+JACOBSON)</code></td>
</tr>
</tbody>
</table>

Here the "+" indicates that the descriptor `AUTHOR` can be satisfied by either of the authors. In the same manner the user may at any point use the "+" feature. Some examples are:

<table>
<thead>
<tr>
<th>Typewriter</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>source+citation</code></td>
<td><code>(SOURCE+CITATION)</code></td>
</tr>
<tr>
<td><code>and ^</code></td>
<td></td>
</tr>
<tr>
<td><code>source =journal+publish+year</code> and ^</td>
<td><code>SOURCE =(JOURNAL+PUBLISH+YEAR)</code></td>
</tr>
<tr>
<td><code>citation=title+author</code> and ^</td>
<td><code>CITATION=(TITLE+AUTHOR)</code></td>
</tr>
</tbody>
</table>

3.3. Verbal description: Find all documents in the collection that cite articles by Hardwick and Jacobson.

Again we proceed as in the previous examples. In this case the part labeled (3) appears as:

<table>
<thead>
<tr>
<th>Typewriter</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3) <code>author =borko. and jacobson. and ^</code></td>
<td><code>YOUR REQUEST IS</code></td>
</tr>
<tr>
<td></td>
<td><code>(CITATION)</code></td>
</tr>
<tr>
<td></td>
<td><code>CITATION=(AUTHOR)</code></td>
</tr>
<tr>
<td></td>
<td><code>AUTHOR =(BORKO)*(JACOBSON)</code></td>
</tr>
</tbody>
</table>
Here, the "*" indicates that the descriptor AUTHOR must be satisfied by the occurrence of both authors' names. Again, the "*" feature may be used at any point. For example:

<table>
<thead>
<tr>
<th>Typewriter</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>source *</td>
<td>(SOURCE)*(CITATION)</td>
</tr>
<tr>
<td>and citation *</td>
<td></td>
</tr>
<tr>
<td>and *</td>
<td></td>
</tr>
<tr>
<td>source = journal *</td>
<td>SOURCE = (JOURNAL)<em>(MONTH)</em>(YEAR)</td>
</tr>
<tr>
<td>and month *</td>
<td></td>
</tr>
<tr>
<td>and year *</td>
<td></td>
</tr>
<tr>
<td>and *</td>
<td></td>
</tr>
<tr>
<td>citation = title *</td>
<td>CITATION = (TITLE)*(AUTHOR)</td>
</tr>
<tr>
<td>and author *</td>
<td></td>
</tr>
<tr>
<td>and *</td>
<td></td>
</tr>
<tr>
<td>title = information *</td>
<td>TITLE = (INFORMATION)*(RETRIEVAL)</td>
</tr>
<tr>
<td>and retrieval *</td>
<td></td>
</tr>
<tr>
<td>and *</td>
<td></td>
</tr>
</tbody>
</table>

Of course, we may combine the use of the "+" and "*" features, as for example:

<table>
<thead>
<tr>
<th>Typewriter</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>source = journal + publish</td>
<td>SOURCE = (JOURNAL+PUBLISH)*(YEAR)</td>
</tr>
<tr>
<td>and year *</td>
<td></td>
</tr>
<tr>
<td>and *</td>
<td></td>
</tr>
</tbody>
</table>

and, in general, we may express any "product of sums" of terms by this technique.

3.4. Verbal description: Find all documents in the collection that are either written by or reference BORKO.
Notice that both SOURCE and CITATION are satisfied by a variable called AUTHOR which has the value BORKO in both cases. Also, note that the REQUEST system only asks for the value of AUTHOR once. REQUEST assumes that if multiple occurrences of a variable term\(^1\) appear, then this variable term has only one interpretation.

### 3.5

In contrast, suppose that the user wants to find all documents in the collection that are written by BORKO or reference JACOBSON:

<table>
<thead>
<tr>
<th>Typewriter</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>STATE YOUR REQUEST</strong></td>
<td><strong>YOUR REQUEST IS:</strong></td>
</tr>
<tr>
<td>source+citation</td>
<td>(SOURCE+CITATION)</td>
</tr>
<tr>
<td>and</td>
<td></td>
</tr>
<tr>
<td>source =author</td>
<td>SOURCE =(AUTHOR1)</td>
</tr>
<tr>
<td>and</td>
<td>CITATION=(AUTHOR2)</td>
</tr>
<tr>
<td>citation=author2</td>
<td>AUTHOR1 =(BORKO)</td>
</tr>
<tr>
<td>and</td>
<td></td>
</tr>
<tr>
<td>author =borko</td>
<td>AUTHOR2 =(JACOBSON)</td>
</tr>
<tr>
<td>and</td>
<td></td>
</tr>
</tbody>
</table>

Notice that if the user replies:

AUTHOR2 = BORKO

the resultant query is equivalent to that in 3.4.

The rules that govern the choosing of the variable term names, e.g., AUTHOR, AUTHOR1, AUTHOR2, etc., are as follows:

---

\(^1\)A variable term is any term that can appear to the left of an equal sign, e.g., SOURCE, CITATION, AUTHOR in example 4.
1) No variable term may exceed eight characters.

2) The first character of the variable term must be chosen as specified in 3). All subsequent characters are chosen at the user's discretion, except for the use of the blank and + characters.

3) The first character of the variable term must be chosen in accordance with the following table:

<table>
<thead>
<tr>
<th>First letter of variable</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>author</td>
</tr>
<tr>
<td>C</td>
<td>citation</td>
</tr>
<tr>
<td>D</td>
<td>descriptor</td>
</tr>
<tr>
<td>E</td>
<td>edition</td>
</tr>
<tr>
<td>F</td>
<td>affiliation</td>
</tr>
<tr>
<td>G</td>
<td>page</td>
</tr>
<tr>
<td>I</td>
<td>item</td>
</tr>
<tr>
<td>J</td>
<td>journal</td>
</tr>
<tr>
<td>K</td>
<td>acknowledged person</td>
</tr>
<tr>
<td>L</td>
<td>language or literary form</td>
</tr>
<tr>
<td>M</td>
<td>month</td>
</tr>
<tr>
<td>N</td>
<td>number</td>
</tr>
<tr>
<td>P</td>
<td>place, publisher, part</td>
</tr>
<tr>
<td>R</td>
<td>referring string</td>
</tr>
<tr>
<td>S</td>
<td>source</td>
</tr>
<tr>
<td>T</td>
<td>title</td>
</tr>
<tr>
<td>U</td>
<td>universal (no restriction)</td>
</tr>
<tr>
<td>W</td>
<td>sponsor</td>
</tr>
<tr>
<td>X</td>
<td>chapter</td>
</tr>
<tr>
<td>Y</td>
<td>year</td>
</tr>
</tbody>
</table>

An atom term is any term that is not a variable term, e.g., BORKO, JACOBSON, INFORMATION RETRIEVAL, and, in general, any string of characters with the following restrictions:

1) No atom term may contain a "+". The + is reserved for the "+" feature that "or's" two or more terms.

2) The last atom term used with the "+" feature (including the vacuous case) may not terminate in a blank character. If terminal blanks are desired, the user must indicate so by use of a "%" immediately following the last blank. Otherwise, a terminal blank character causes the entire working line to be erased. Thus, if the user wants to type "INFORMATION
RETRIEVAL + AUTOMATIC INDEXING" and types "INFOR..." by mistake, he may type a blank and a carriage return causing the line to be erased and permitting the line to be typed again.

3) The atom term may contain the "don't care" character as described in the SEARCH instruction.

3.6. Verbal description: Journal papers written since 1965 dealing with information retrieval that reference journal or technical papers written by BORKO on information retrieval.

As a final example we demonstrate a more sophisticated query using most of the features available.

<table>
<thead>
<tr>
<th>Typewriter</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>STATE YOUR REQUEST</strong></td>
<td><strong>YOUR REQUEST IS:</strong></td>
</tr>
<tr>
<td>source and citation and .</td>
<td>(SOURCE)*(CITATION)</td>
</tr>
<tr>
<td>source =litform1 and year and descript and .</td>
<td>SOURCE =(LITFORM1)<em>(YEAR)</em>(DESCRIPT)</td>
</tr>
<tr>
<td>citation=author and descript and litform1+litform2 and .</td>
<td>CITATION=(AUTHOR)*(DESCRIPT)*LITFORM1+LITFORM2)</td>
</tr>
<tr>
<td>litform1=JL paper and .</td>
<td>LITFORM1=(JL PAPER)</td>
</tr>
<tr>
<td>descript=information retrieval+automatic indexing and .</td>
<td>DESCRIPT=(INFORMATION RETRIEVAL+AUTOMATIC INDEXING)</td>
</tr>
<tr>
<td>author =borko and .</td>
<td>AUTHOR =(BORKO)</td>
</tr>
<tr>
<td>litform2=tech rept and .</td>
<td>LITFORM2=(TECH REPT)</td>
</tr>
</tbody>
</table>

A sample retrieval in this query appears on Fig. 3.1. The entire bibliographic entry of this document appears in the appendix.

---

1See section 4 for the description of the search instruction. The typewriter character that represents the "don't care" character is the "$\$."

---
4. The ISL System

The Information Search Language (ISL) is a problem-oriented language designed to facilitate the manipulation of textual material. A preliminary form of the language has been described in an earlier report. [3] The present version is an extension and evolution of the previous form, and may be more accurately termed a "system" consisting of a basic language (ISL), a sophisticated assembler (ILLAR), a large and expanding set of sub-programs to handle special functions of the system, and an interactive program that allows programmers to initiate commands from the display console.

The ISL language has been designed in such a fashion as to allow sophisticated programming (in the sense of the implementation of the REQUEST System in ISL) and yet be very easily used by persons who have had no previous programming experience.

The user need only become familiar with a very modest number of easy-to-use instructions to input, manipulate, and output the data. In addition, if the manipulation is such that it requires decisions on the part of the user, then the interactive mode instructions may be used to alter the program.

The basic ISL language consists of only twenty-two instructions:

- word-oriented: LOAD, STORE, PLUS, MINUS, CONVERT
- character-oriented: STRING, MOVE, SEARCH, VSEARCH, SEEK
- transfer: IF, GOTO, TJUMP
- input output: TSTRING, READ, WRITE, PRINT, ISLTV, TVOFF/STOPTV, STRTTV
- entry and exit: BEGIN, RETURN

The first group of basic instructions are concerned with arithmetic
operation on the ISL "accumulator." This location is designated by the name ISLACC. In addition to the standard word-oriented instructions, the convert instruction converts the value of the ISLACC to a BCD character string representing this value's decimal integer equivalent and designates the resultant string by its beginning and ending addresses in core.

The character-oriented instructions identify strings of characters by their beginning and ending address. These are 18-bit addresses (i.e., a byte address). The STRING instruction is used to define strings of characters. Its analogy in a usual assembly language would be a BCD pseudo-op. The MOVE instruction allows the moving of a string from one area to another within the character-organized portion of core. The move instruction is provided with the specification of what characters are to be moved and where the beginning of the new area is to be. The move instruction then returns the value of the end of the string.

The SEARCH and VSEARCH instructions specify what string of characters is to be searched for, and what area of core is to be scanned. These instructions return a success-fail flag. Upon success they return the beginning and ending addresses of the "matched" string of characters. Included in the search specification string may be any number of "don't care" characters. The "don't care" character is used in the string of characters to be searched for, to indicate that we "don't care" what characters come between the previous string and the following string.

The SEEK instruction is used to look for the occurrence of a single given character in a given area of core. Although its function could be performed by the SEARCH instruction, the SEEK is much faster and has the
added feature that it will seek in either direction on a character string.

There are three transfer instructions. The IF instruction is a conditioned transfer and the GOTO instruction is an unconditional transfer. The TJUMP instruction is a multiple-branch transfer, based on typewriter control.

The input and output instructions allow extremely easy use of the peripheral equipment. In interactive programs the execution of the TSTRING instruction allows the user to enter a character string from the console typewriter.

The PRINT instruction allows the programmer to specify the area to be printed, the column number in which to start, and how many lines to skip.

There are two instructions used to display character strings. The first of these is ISLTV which simply displays an area of core with no checking of number of characters on a line, total number of characters, or total number of lines. The other routine STRTIV checks all of these items and ensures that what is put out to the scope does not wrap around the end of a line or the bottom of the screen. In the event that the characters will not fit on the screen, this routine provides light-pen pointers which allow a scroll-like roll of lines of characters up and down on the screen. The routine also has provision for taking photographs of the material displayed or printing what is displayed on the screen.

Because of the nature of ISL, the tape READ and WRITE routines are also oriented toward strings of characters rather than "card-images." ISL tape records are in variable lengths binary format. Each record is tagged with its length and the terminal byte position. This bookkeeping is
done exclusively by the READ and WRITE routines. By convention, records are always read into core starting at a predefined location called MID and the READ instruction returns the end address of the string read. The WRITE instruction always writes the string of characters beginning at a predefined location called ANT and ending at a location specified by the programmer.

The BEGIN and RETURN statements in ISL take care of entry to and exit from ISL programs or subprograms. These instructions communicate the necessary arguments and facilitate the modularization of the system.

In addition to the basic ISL instructions, there are several ILLAR instructions which are needed to complete the basic language, that is, to make it usable. Most of these are self-explanatory or are explained more fully in [2]. They are:

- **EQU** - a standard equate statement
- **DEFINE** - to define variables (like BSS)
- **PAUSE** - to pause and proceed
- **EFMARK** - to write and end of file on magnetic tape
- **REWIND** - to rewind a magnetic tape
- **BACKSPAC** - to backspace a magnetic tape
- **IDENT** - (first instruction of a program) to identify program name and arguments
- **LIB** - (LIB ISL is the second instruction of an ISL program) to identify an ISL program
- **END** - to signify end of a program
- **FINIS** - to signify the end of a series of programs to be assembled together.

Since the ISL language is imbedded in the ILLAR system, any of the
features of ILLAR are available to the ISL programmer who wishes to take advantage of them. Of course, all of the machine language operations basic to the 1604 computer are available for use. The programmer who has experience in ILLAR or some similar assembly language normally will write a machine-language-like program, using the ISL instructions only as the need arises. Thus index registers are available for loops and counts, threshold and equality searches can be used to compare counters or characters, floating point arithmetic operations can be performed and logical operations are immediately available.

However, the ILLAR system offers the programmer much more. Recursive macros may be used, literals may be used in lieu of separately defining every number to be used, and fortran-like compile arithmetic statements are available. ILLAR also provides very good tools for communication of arguments to subroutines on a call-by-name basis. All of the ILLAR system routines are also available to the ISL programmer. Among these are routines to do normal (i.e., non-ISL) input and output, number conversion, light-pen functions, scope display of character or graphic data, and output of typed messages. The edit routines used for ILLAR are, of course, used to edit ISL programs as well.

There exist on the library tape a large number of subroutines written in ISL that accomplish the more complex operations that ISL was designed to handle and also handle the interactive system described in section 3. Among these are PUSH and POP to retrieve and store a single character of a string, LJUMP which is just a light-pen-controlled equivalent
of TJUMP and various routines to set up strings of characters for display. There also exist typewriter driven routines to list and copy ISL tapes, to translate standard "card-images" on tape to ISL records, and to display from magnetic tape to the scope. Routines used in handling the data collection include a packing routine and a syntax-checking routine, and various routines to arrive at a key-word in context sorting of titles in the data collection.
5. **Summary**

The ISL-REQUEST system in the form presented in sections 3 and 4 incorporates the following features:

1. It is a real-time retrieval system.
2. It permits an easy and flexible means of stating queries.
3. It allows the dissemination of information on various output devices.
4. It allows a wide variety of information to be used for the retrieval base.
5. The system is modular in design, thereby facilitating future modifications.
6. It allows relatively inexperienced persons to interrogate the data base.
7. It implements a very natural type interrogation scheme with the feedback feature.
APPENDIX

IT 45
TI AUTOMATIC SUBJECT RECOGNITION IN SCIENTIFIC PAPERS..AN EMPIRICAL STUDY
AP 0=CONNOR, JOHN
FA INSTITUTE FOR SCIENTIFIC INFORMATION, PHILADELPHIA, PENN
SP1 INFORMATION SYSTEMS BRANCH OFFICE OF NAVAL RESEARCH
SP2 INFORMATION RESEARCH DIVISION, AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
JL ACMJ
VO 12
NR 4
MO OCT
YR 1965
PA 490 TO 515
LA ENGLISH
LF JL PAPER
CR1
IT AUTOMATIC ABSTRACTING
AC COMPUTER DIVISION, RAMO WOOLDRIDGE
PL CANOGA PARK, CAL
MO FEB
YR 1963
NR C 107 301
LA ENGLISH
LF TECH REPORT
ST SEVERAL AUTHORS HAVE SUGGESTED THAT OCCURRENCE IN FIRST AND LAST SENTENCES
OF PARAGRAPHS OR FIRST AND LAST PARAGRAPHS OF SECTIONS INDICATES EMPHASIS.
CR2
TI MACHINE MADE INDEX FOR TECHNICAL LITERATURE, AN EXPERIMENT
AP BAXENDALE, P.
JL IBMJRD
VO 2
NR 4
MO OCT
YR 1958
PA 354
LA ENGLISH
LF JL PAPER
ST SEVERAL AUTHORS HAVE SUGGESTED THAT OCCURRENCE IN FIRST AND LAST SENTENCES
OF PARAGRAPHS OR FIRST AND LAST PARAGRAPHS OF SECTIONS INDICATES EMPHASIS.
CR3
TI ACCURACY OF TITLES IN DESCRIBING CONTENT OF BIOLOGICAL SCIENCES ARTICLES
AP1 BERNARD, J.
AP2 SHILLING, C.
PU AMERICAN INSTITUTE OF BIOLOGICAL SCIENCES
MO MAY
YR 1963
There have been previous studies using biomedical titles but not full texts. The inadequacy of titles alone for some purposes is indicated by some of these tests.

Some of these methods have been tested empirically. About 60 percent of the papers were longer than 1000 words (the average length 2000) rather than letters to the editor or abstracts. The subject recognition rates in six of the previous studies ranged from 30 to 70 percent with corresponding overassigning less than seven percent. Some of the studies used judgments agreed on by two or three subject specialists.

Towards the establishment of a computer based classification system for scientific documentation, report number TM 1763.

Some of these methods have been tested empirically. About 60 percent of the papers were longer than 1000 words (the average length 2000) rather than letters to the editor or abstracts. The subject recognition rates in six of the previous studies ranged from 30 to 70 percent with corresponding overassigning less than seven percent. Some of the studies used judgments agreed on by two or three subject specialists.

Automatic syntax analysis in machine indexing and abstracting.

It has been suggested that syntactic centrality in a sentence or clause measures emphasis or importance in a message. The basic idea of ... can be roughly summarized as follows. How many modifying phrases and words must be added to a core clause (or core word) in a title or heading before the nested expression is reached, preserving grammaticality during the additions.

Some clumping experiments for associative document retrieval

Some of these methods have been tested empirically. About 60 percent of the papers were longer than 1000 words (the average length 2000) rather than letters to the editor or abstracts.

The results in the other four tests were 100 percent recognition and five percent overassigning, 94 percent recognition and 40 percent overassigning, 80 percent recognition and 10 percent overassigning, 80 percent recognition and 13 percent overassigning, and 60 percent recognition and 10 percent overassigning.

Some of the studies used judgments agreed on by two or three subject specialists.

In other studies the average overassignment ratios were between 0.3 and 1.5 except for a 94 percent recognition costing 8.5 and an 80 percent recognition costing 2.0.

The only other studies reporting recognition rates over 80 percent were Salton's, still in the early phase of which the corpus searched is that used to build the thesaurus, and Dale's in which the recognition rate cost 40 percent overassigning.

The dictionary used was ....
Thus syntactic distance might be better than word distance for measuring word relations. Giuliano and Jones have also suggested adding sentence co-occurrence or syntactic information to the word list representing a paper.

It has been suggested that automatic subject recognition can be improved by augmenting papers with words closely associated statistically with the words in the papers.

In a paper the frequency of clues for a subject specification might indicate the emphasis on that subject in the paper and thus be useful in reducing familiar fact overassigning.

Some of these methods have been tested empirically.
About 60 percent of the papers were longer than 1000 words (the average length 2000) rather than letters to the editor or abstracts. The subject recognition rates in six of the previous studies ranged from 30 to 70 percent with corresponding overassigning less than seven percent.

Machinelike indexing by people

Method for using computers in information classification

It has been suggested that automatic subject recognition can be improved by augmenting papers with words closely associated statistically with the words in the papers.

Searching titles by man, machine, and chance

There have been previous studies using biomedical titles but not full texts.

The inadequacy of titles alone for some purposes is indicated by some of these tests.

Nonconventional technical information systems in current use
LA ENGLISH
LF TECH REPT
ST THE SYSTEM DESCRIBED IN DETAIL IN ....
CR17
TI CORRELATION OF INDEXING HEADINGS AND TITLE WORDS IN THREE MEDICAL
INDEXING SYSTEMS
AP 0=CONNOR,J.
JL AD
VO 15
NR 2
MO APR
YR 1964
PA 96 TO 104
ST1 THERE HAVE BEEN PREVIOUS STUDIES USING BIOMEDICAL TITLES BUT NOT FULL
TEXTS
ST2 THE INADEQUACY OF TITLES ALONE FOR SOME PURPOSES IS INDICATED BY SOME
OF THESE TESTS
CR18
TI MECHANIZED INDEXING METHODS AND THEIR TESTING
AP 0=CONNOR,J.
JL ACMJ
VO 11
NR 4
MO OCT
YR 1964
PA 437 TO 449
LA ENGLISH
LF JL PAPER
ST1 MANY METHODS HAVE BEEN PROPOSED IN RECENT YEARS FOR SUBJECT IDENTIFI-
ICATION OF SCIENTIFIC PAPERS BY COMPUTER. FOR A REVIEW SEE ...
ST2 FOR A BRIEF SUMMARY, SEE....
CR19
TI MECHANIZED INDEXING STUDIES OF MSD TOXICITY, PART 1 (APPENDIX D). AD436 523
AP 0=CONNOR,J.
PU DEFENSE DOCUMENTATION CENTER, CAMERON STATION, ALEXANDRIA, VA.
LA ENGLISH
LF TECH REPT
ST FULL TEXTS OF SEVERAL PAPERS WITHOUT TOXICITY KEYWORDS ARE FOUND IN...
CR20
TI MECHANIZED INDEXING STUDIES OF MSD TOXICITY, PART 2 (APPENDIX B). AD 437
868
AP 0=CONNOR,S.
PU DEFENSE DOCUMENTATION CENTER, CAMERON STATION, ALEXANDRIA, VA.
LA ENGLISH
LF TECH REPT
ST MANY OTHER EXAMPLES OF SUCH EXPRESSIONS ARE FOUND IN ....
CR21
TI INFORMATION STORAGE AND RETRIEVAL
AP1 STALTON,G.
AP2 ET AL
ST1 SOME OF THESE METHODS HAVE BEEN TESTED EMPIRICALLY
ST2 ABOUT 60 PERCENT OF THE PAPERS WERE LONGER THAN 1000 WORDS (THE AVERAGE LENGTH 2000) RATHER THAN LETTERS TO THE EDITOR OR ABSTRACTS
ST3 THE RESULTS IN THE OTHER FOUR TESTS WERE 100 PERCENT RECOGNITION AND FIVE PERCENT OVERASSIGNING, 94 PERCENT RECOGNITION AND 40 PERCENT OVERASSIGNING, 80 PERCENT RECOGNITION AND 10 PERCENT OVERASSIGNING, 80 PERCENT RECOGNITION AND 13 PERCENT OVERASSIGNING, AND 60 PERCENT RECOGNITION AND 10 PERCENT OVERASSIGNING
ST4 TWO OF THE 17 SUBJECTS IN THE SALTON TESTS OCCURRED IN 0.5 OF THE COLLECTION AND TWO MORE IN 0.8 OF THE COLLECTION
ST5 THE LAST RULE SOMEWHAT RESEMBLES SALTON=S STATISTICAL PHRASES, OF THE S LIST IS VERY LONG AND A WORD NEARNESS CLOSER THAN SENTENCE CO-OCCURRENCE IS USED.
ST6 THE ONLY OTHER STUDIES REPORTING RECOGNITION RATES OVER 80 PERCENT WERE SALTON=S, STILL IN THE EARLY PHASE IN WHICH THE CORPUS SEARCHED IS THAT USED TO BUILD THE THESAURUS, AND DALE=S IN WHICH THE RECOGNITION RATE COST 40 PERCENT OVERASSIGNING.

CR22
TI PUNCHED CARDS, 2 ED
AP0 SCHULTZ, C.K.
AE1 PERRY, J.
AE2 CASEY, R.
ED 2
LF BOOK
PU REINHOLD
YR 1958
LA ENGLISH
ST THE SYSTEM DESCRIBED IN DETAIL IN ....

CR23
TI STEEDMAN=S MEDICAL DICTIONARY 20 ED
LF BOOK
PU WILLIAMS AND WILKINS
YR 1961
ED 20
LA ENGLISH
ST THE DICTIONARY REFERRED TO HERE IS ....

CR24
TI TRAINING A COMPUTER TO ASSIGN DESCRIPTORS TO DOCUMENTS. EXPERIMENT IS IN AUTOMATIC ABSTRACTING
AP1 STEVENS, M.
AP2 URBAN, C.H.
JL PMJCC
PA 563 TO 575
ST1 Some of these methods have been tested empirically.
ST2 About 60 percent of the papers were longer than 1000 words (the average length 2000) rather than letters to the editor or abstracts.
ST3 The subject recognition rates in six of the previous studies ranged from 30 to 70 percent with corresponding overassigning less than seven percent.
ST4 The subject specifications in previous studies, except ..., would not usually be used conjunctively in searching, while a Merck Sharp and Dohme search is usually specified by a conjunction of index terms.

CR25
TI Association factor in information retrieval.
AP Stiles, H.E.
JL ACMJ
VO 8
NR 2
MO APR
YR 1961
PA 271 TO 279
LA ENGLISH
LF JL PAPER
ST It has been suggested that automatic subject recognition can be improved by augmenting papers with words closely associated statistically with the words in the papers.

CR26
TI Searching natural language text by computer.
AP Swanson, D.R.
JL 8
VO 132
NR 34
MO OCT
YR 1960
PA 1099
LF JL PAPER
LA ENGLISH
ST Some of these methods have been tested empirically.
ST2 About 60 percent of the papers were longer than 1000 words (the average length 2000) rather than letters to the editor or abstracts.
ST3 The subject recognition rates in six of the previous studies ranged from 30 to 70 percent with corresponding overassigning less than seven percent.
ST4 Some of the studies used judgments agreed on by two or three subject specialists.

CR27
TI Interrogating a computer in natural language.
AP SWANSON, D.R.
JL IFIPSCP
LF CONF PROC
PL MUNICH
YR 1962
LA ENGLISH
ST1 SOME OF THESE METHODS HAVE BEEN TESTED EMPIRICALLY
ST2 ABOUT 60 PERCENT OF THE PAPERS WERE LONGER THAN 1000 WORDS (THE AVERAGE LENGTH 2000) RATHER THAN LETTERS TO THE EDITOR OR ABSTRACTS
ST3 SOME OF THE STUDIES USED JUDGMENTS AGREED ON BY TWO OR THREE SUBJECT SPECIALISTS
ST4 THE RESULTS IN THE OTHER FOUR TESTS WERE 100 PERCENT RECOGNITION AND FIVE PERCENT OVERASSIGNING, 94 PERCENT RECOGNITION AND 40 PERCENT OVERASSIGNING, 80 PERCENT RECOGNITION AND 10 PERCENT OVERASSIGNING, 80 PERCENT RECOGNITION AND 13 PERCENT OVERASSIGNING, AND 60 PERCENT RECOGNITION AND 10 PERCENT OVERASSIGNING
ST5 IN OTHER STUDIES THE AVERAGE OVERASSIGNMENT RATIOS WERE BETWEEN 0.3 AND 1.5 EXCEPT FOR A 94 PERCENT RECOGNITION COSTING 8.5 AND AN 80 PERCENT RECOGNITION COSTING 2.0
CR28
TI RESEARCH PROCEDURES FOR AUTOMATIC INDEXING
AP SWANSON, D.R.
JL MACHINE INDEXING
LF BOOK
PA 281 TO 304
PU AMERICAN U.
YR 1962
LA ENGLISH
ST THE FREQUENCY RESULT CONTRASTS WITH SWANSON=S THAT FREQUENCY DID NOT HELP.
CR29
TI AUTOMATIC LANGUAGE ANALYSIS, AD 297 381
AP THORNE, J.P.
PU DEFENSE DOCUMENTATION CENTER, CAMERON STATION, ALEXANDRIA, VA.
YR 1962
LA ENGLISH
LF TECH REPT
ST1 IT HAS BEEN SUGGESTED THAT SYNTACTIC CENTRALITY IN A SENTENCE OR CLAUSE MEASURES EMPHASIS OR IMPORTANCE IN A MESSAGE
ST2 THE BASIC IDEA OF ... CAN BE ROUGHLY SUMMARIZED AS FOLLOWS. HOW MANY MODIFYING PHRASES AND WORDS MUST BE ADDED TO A CORE CLAUSE (OR CORE WORD) IN A TITLE OR HEADING BEFORE THE NESTED EXPRESSION IS REACHED, PRESERVING GRAMMATICALITY DURING THE ADDITIONS.
CR30
TI DISCRIMINANT METHOD FOR AUTOMATICALLY CLASSIFYING DOCUMENTS
AP WILLIAMS, J.H., JR.
JL JCCFP
PA 161 TO 166
PU SPARTAN
YR 1963
Some of these methods have been tested empirically. About 60 percent of the papers were longer than 1000 words (the average length 2000) rather than letters to the editor or abstracts.

The subject recognition rates in six of the previous studies ranged from 30 to 70 percent with corresponding overassigning less than seven percent.

Results of classifying documents with multiple discriminant functions.

AP Williams, J.H. Jr.
JL SSAMMDP
PU IBM FEDERAL SYS DIV, BETHESDA, MD
MO MAR
YR 1964

Some of these methods have been tested empirically. About 60 percent of the papers were longer than 1000 words (the average length 2000) rather than letters to the editor or abstracts.

The results in the other four tests were 100 percent recognition and five percent overassigning, 94 percent recognition and 40 percent overassigning, 80 percent recognition and 10 percent overassigning, 80 percent recognition and 10 percent overassigning, and 60 percent recognition and 10 percent overassigning.
REFERENCES


Distribution List as of November 1, 1968

1 Dr. A.K. Dougaid  
   Ass't Director (Research)  
   Dept of Defense Res & Eng  
   Washington, D.C. 20301

1 Office of Deputy Director  
   (Research and Technology)  
   Washington, D.C. 20301

1 Director Advanced Research Projects Agency  
   Washington, D.C. 20301

1 Director for Information Sciences  
   Advanced Research Projects Agency  
   Department of Defense  
   Washington, D.C. 20301

1 Director for Materials Sciences  
   Advanced Research Projects Agency  
   Department of Defense  
   Washington, D.C. 20301

1 Headquarters  
   Defense Communications Agency (333)  
   The Pentagon  
   Washington, D.C. 20305

20 Defense Documentation Center  
   Attn: JERA  
   Cameron Station, Bldg 5  
   Alexandria, Virginia 22314

1 Director  
   National Security Agency  
   Fort George G. Meade, Maryland 20755

1 Weapons System Evaluation Group  
   Attn: Col Daniel W. McIntosh  
   Department of Defense  
   Washington, D.C. 20305

1 National Security Agency  
   Attn: Mr. S.T. Jones  
   Office of Research  
   Fort George G. Meade, Maryland 20755

1 Central Intelligence Agency  
   Attn: OCR-23D Publications  
   Washington, D.C. 20305

1 Colonel Kee  
   AFPRC  
   Rap, USAF  
   Room 10-045, The Pentagon  
   Washington, D.C. 20339

1 Aerospace Medical Division  
   AND (AM)  
   Brooks Air Force Base, Texas 78235

1 AUSL-9663  
   Newell AFB, Alabama 36112

1 AFPC (FYPF-2)  
   Technical Library  
   Edwards AFB, Calif. 93513

1 By SANDU (ENTER/ Lt Nelson)  
   AF 511 Post Office  
   Los Angeles, California 90065

1 Lt Col Charles M. Neary  
   USAF (ARTF-3)  
   The Pentagon  
   Washington, D.C. 20301

1 SSD (SINE/AF Starback)  
   AFPSD  
   Los Angeles, California 90065

1 DoD (GOM, LOOR)  
   Air Force Post Office  
   Los Angeles, California 90065

1 ARL (ART)  
   Wright-Patterson AFB, Ohio 45433

1 Dr. H.T. Noble  
   Air Force Aviation Laboratory  
   Wright-Patterson AFB, Ohio 45433

1 Mr. Peter Murray  
   Air Force Aviation Laboratory  
   Wright-Patterson AFB, Ohio 45433

1 AFAL (ATMN/ R.D. Lerman)  
   Wright-Patterson AFB, Ohio 45433

2 Commanding General  
   Attn: TMPE-MG-2T  
   White Sands Missile Range  
   New Mexico, 88001

1 RAIC (UKAL101)  
   Griffin AFB, New York 13442  
   Attn: Documents Library

1 Mr. H.E. Webb (ERSA)  
   Rome Air Development Center  
   Griffiss AFB, New York 13442

1 Academy Library (OFFS)  
   U.S. Air Force Academy  
   Colorado Springs, Colorado 80912

1 Mr. Morton F. Fossett, Chief  
   AFSC Scientific and Liaison Office  
   26 Federal Plaza  
   New York, N.Y. 10007

1 Lt Col Bernard S. Morgan  
   AFETR (ETTG-3)  
   Wright-Patterson AFB, Ohio 45433

1 Technical Library, AFETR  
   Attn: Col Daniel W. McIntosh  
   Fort George G. Meade, Maryland 20755

1 Lt Col Robert B. Lalisch  
   AFSC (SCTR)  
   Patrick AFB, Florida 85315  
   Attn: STEWS-WS-VT

1 AFSC (SCBB)  
   Andrews AFB, Maryland 20331  
   Attn: Library/Documents

1 AFSC (SCT)  
   Andrews AFB, Maryland 20331  
   Attn: Library/ Documents

1 Commanding General  
   U.S. Army Strategic Command  
   Fort Bragg, North Carolina 28210

1 Commanding General  
   U.S. Army Material Command  
   Attn: AMXDO-TI  
   Fort Belvoir, Virginia 22060

1 Commanding General  
   U.S. Army Strategic Command  
   Fort Bragg, North Carolina 28212

1 Commanding Officer  
   Army Materials & Tech. Center  
   Watertown Arsenal  
   Watertown, Mass. 02172

1 Commanding Officer  
   U.S. Army Ballistics Research Laboratory  
   Attn: AMXDO-DMX  
   Aberdeen Proving Ground  
   Aberdeen, Maryland 21005

1 Commandant  
   U.S. Army Armament Command  
   Attn: Technical Library  
   E Benjamin Arsenal, Alabama 35009

1 U.S. Army Ordnance Command  
   Attn: Technical Information Command  
   Picatinny Arsenal, New Jersey 07881

1 Commanding Officer  
   U.S. Army Ordnance Command  
   Attn: Technical Information Command  
   Picatinny Arsenal, New Jersey 07881

1 U.S. Army Ordnance Command  
   Attn: Technical Information Command  
   Picatinny Arsenal, New Jersey 07881

1 Commanding Officer  
   U.S. Army Ordnance Command  
   Attn: Technical Information Command  
   Picatinny Arsenal, New Jersey 07881

1 Commanding Officer  
   U.S. Army Ordnance Command  
   Attn: Technical Information Command  
   Picatinny Arsenal, New Jersey 07881

1 Commanding Officer  
   U.S. Army Research Office  
   Attn: DOD-AAC-P (Richard O. Hedges)  
   Fort Leavenworth, Kansas 66103

1 Commanding Officer  
   U.S. Army Research Office  
   Attn: DOD-AAC-P (Richard O. Hedges)  
   Fort Leavenworth, Kansas 66103

1 Commanding Officer  
   U.S. Army Research Office  
   Attn: DOD-AAC-P (Richard O. Hedges)  
   Fort Leavenworth, Kansas 66103

1 Commanding Officer  
   U.S. Army Research Office  
   Attn: DOD-AAC-P (Richard O. Hedges)  
   Fort Leavenworth, Kansas 66103
In this paper the REQUEST document retrieval system is described. Included is a description of the data base, the interactive mode, the modular implementation in ISL—a string manipulating language, and the multiple-level Boolean hierarchy query format.
Document Retrieval
Information Retrieval
String Manipulation
Interactive Programming
Boolean Query