2318 THE INSTITUTE OF PAPER CHEMISTRY
(Information Retrieval)
Project Reports

PROJECT REPORT FORM

PROJECT NO. 2318
COOPERATOR Institute of Paper Chemistry
REPORT NO. Six
DATE January 28, 1974
NOTE BOOK
PAGE TO
SIGNED John J. Church, Jr.

Copies to: Files
McClenahan
Weiner
Holm

AUTOMATED INFORMATION RETRIEVAL

INTRODUCTION

Machine readable copy of the Abstract Bulletin of the Institute of Paper Chemistry (ABIPC) is available beginning with Volume 40. The Keyword Supplement to the ABIPC is also available in machine readable form.

Since early 1972 the Division of Information Services has offered a combined version of the two data bases for lease, and computerized information retrieval service using the new data base. This new data base is called "Complete Text plus Keywords" for the ABIPC.

The need for the combined version of the data base came as requests for literature searches that could not be found by the keywords alone began to arrive. Along with the data base which was created, a new information retrieval programming system was developed. Since then another system has been developed which permits the division to perform the information searches in less time on the computer.

PROGRAMMING SYSTEMS

The Institute of Paper Chemistry now uses and offers for lease the following information retrieval systems, LIRES-IF and LIRES-DF. The acronym LIRES implies LIterature REtrieval System. The postscripts distinguish the form of the data base which is actually searched for information: IF for searching the inverted file, and DF for the direct file.

A direct file contains information arranged as it is seen in the printed versions of our publications. An inverted file is one in which all "words" have been separated and rearranged such that with each word, all abstract numbers in which it appeared follow. The definition of "word" is qualified according to the rules found in the File Description section of this report.

Introduction

The LIRES-DF (LIterature REtrieval System — Direct File) program has been designed to permit information searches of the Complete Text, or the new Complete Text with Keywords data base tapes of the Abstract Bulletin. Search profile terms may be words to be found in the TITLE and/or the TEXT and/or the KEYWORDS of an abstract. Each search profile may search separate and distinct portions of an abstract. Profile terms may be truncated right or left or both, and may be assigned weights. The Abstract Bulletin data base tape and one temporary data file is all that is necessary for processing with this program. The current program system was written in the basic FORTRAN IV language with nine assembler language subprograms, and designed to run in 54K of core storage or less on an IBM 360 computer.

A glossary of terms which are used in this program description manual follow:

1. LOGIC GROUP — a collection of words which have the usual "OR" connotation.

2. SEARCH PROFILE — a collection of logic groups which comprise the words of interest for an information search.

3. LOGIC STATEMENT — a statement of what combinations of logic groups constitute a "HIT" for a search profile.

4. HIT — an abstract which satisfies (or matches) a logic mask or masks for a search profile.

5. PROFILE TERMS — a word in the form that is to be used in a search against the data file, i.e., its truncated or full word form.

6. PROFILE TERM WEIGHT — a number assigned to a profile term, which if the term is found in an abstract is to be accumulated into a total for the abstract.

7. SEARCH PROFILE THRESHOLD — a minimum number which is compared against the accumulated profile term weights. If the logic mask of a profile indicates that an abstract is a hit, but the total of the weights does not equal or exceed the threshold, the abstract is not retrieved. If the logic mask is not satisfied, the abstract is not a hit regardless of the accumulated weights.

Program Description

The LIRES-DF program will accept a number of search profiles and search the Complete Text or Complete Text with Keywords data file of the Abstract Bulletin. A particular search profile may specify that all of its profile terms are to be searched against the abstract file, or against the text of the abstract, or against the keywords assigned to the abstract; or any combination of the three. The instructions for a particular search may specify that the output consist of: (1) only abstract numbers of hits, or (2) citations (title, journal reference, and keywords) for hits, or (3) complete abstracts and keywords for hits are to be printed as results. Items (2) and (3) automatically have the profile terms which caused the hit printed with each abstract.

The following restrictions apply to a SEARCH PROFILE:

1. Not more than ten (10) logic groups may be defined in one search profile.

2. Profile terms should not exceed forty (40) characters in length.

3. Profile term weights may not exceed 999 or -999.

4. Profile terms which are to be searched against the title and/or text of an abstract must be single word terms (all forms of truncation apply).

5. A profile term which is to be searched against the assigned keywords of an abstract must be an accepted keyword term (all forms of truncation apply). The "keyword" may of course consist of several individual words.

The following restrictions apply to one program run (one pass of the data file):

1. Not more than thirty-two (32) search profiles may be included.

2. The total character count for profile terms for all profiles must not exceed 4000 characters (no restrictions for a particular search).

3. Not more than 100 logic masks may be specified (no restrictions for a particular search).

4. Not more than 320 words in total may be specified (no restrictions for a particular search).

5. Not more than 100 logic groups may be specified (not more than 10 per search, but otherwise no restriction).

All of the above restrictions may be lifted or modified by some simple modifications of the source program decks and recompiling the programs.

<u>Preparing a Search Profile</u>

The following instructions apply to one search profile which will ultimately be combined with others to perform a search of the data base.

Organize a list of profile term words which describe the interest profile into logic groups, and indicate the type of truncation desired for the words by placing asterisks as follows:

WORD1 — no truncation desired, must have an exact match

*WORD2 — pre-truncated word, any data base word with this ending is to be found

WORD3 — post-truncated word, any data base word with this beginning is to be found

WORD4 — root form word, any data base word with this root is to be found also write the profile term weight which is to be assigned. Prepare the information for the search profile parameter card, which consists of the following information:

1. The type of printed output desired.

   1 — Abstract numbers only.
   2 — Citations and keywords for hits.
   3 — Full text of each abstract for hits.

2. The portion of each abstract that is to be searched for all profile terms in this search.

   1 — Search abstract title only.
   2 — Search abstract text only.
   3 — Search abstract title and text.
   4 — Search abstract keywords only.
   5 — Search abstract title and keywords.
   6 — Search abstract text and keywords.
   7 — Search abstract title, text, and keywords.

3. The search profile threshold; the accumulation of weights must be at least equal to this threshold number for an abstract to be a hit.

4. The number of logic statements which are to be entered with this profile.

5. The list of numbers indicating the number of profile terms in each logic group.

The logic statement(s) should be prepared at this time. A logic statement defines which groups are to be "AND"ed. More than one logic statement may be entered describing combinations of logic groups which if satisfied will constitute a hit. "NOT" logic is indicated in logic statements by punching a minus sign by the logic group number. A logic statement, for example to define logic groups 1, 2, and not 4 as a hit is written (and punched) as 1 2 -4. At least one logic statement must be present for a search profile. Logic masks are generated from the statements on a one-for-one basis except for a statement with "NOT" logic. An extra mask is generated for each statement containing "NOT" logic.

As an example, consider a profile concerning computers in process control; the keywords only are to be searched, and citations only are to be printed. The keywords will be searched for exact match (no truncation). The deck for this search profile would be punched:

Card	Information Punched
1	COMPUTERS IN PROCESS CONTROL – SAMPLE PROFILE
2	....2....4....0....2....1....6....6
3	COMPUTERS
4	ANALOG COMPUTERS
5	AUTOMATIC CONTROL
6	DIGITAL COMPUTERS
7	PAPER MILLS
8	PROCESS CONTROL
9	PULP MILLS
10	BATCH PROCESS
11	COMPUTER PROGRAMS
12	CONTROL SYSTEMS
13	DIGESTERS
14	PERMANGANATE NUMBER
15	TEMPERATURE
16	..1
17	..2..3

Card 1 of the deck is the title of the search; it will be printed on the output. Card 2 is the parameter card (the dots indicate blank columns) where the 2 will cause a printout of citations and keywords; the 4 will cause only the keywords to be searched; the 0 is the search profile threshold; the 2 indicates that there are two logic statements describing what combinations of logic groups will be hits; the last three numbers indicate that there are 3 logic groups of 1, 6, and 6 words, respectively. The absence of asterisks indicate no truncation. Cards 3 through 15 are the profile terms. Cards 16 and 17 are the logic statements; 1 indicates that any abstract containing the word from logic group 1 (COMPUTERS) is to be cited; and the 2 and 3 indicate that if an abstract contains any of the words from logic group 2 AND logic group 3, it is also to be cited. See the sample run for the results of this search.

The format for punching the search profile parameter card is five (5) columns for each parameter entered (maximum of 14 parameters); and three (3) columns for each logic group number indicated on a logic statement card (maximum of 10 logic group numbers). All numbers must be right-justified in their position. All other cards are free format alphanumeric information and should begin in column one of the card.

Selection of Profile Terms

The choice of profile terms is arbitrary for those searches which are to be processed against the title and text of an abstract. A concordance of Volume 41 of the Abstract Bulletin is provided to assist in the selection of profile terms. Particular attention should be paid to the standard abbreviations used by the abstractors.

The Pulp and Paper Research Institute of Canada's Thesaurus of Pulp and Paper Terms, Second Edition, should be used when choosing profile terms which are to be searched against the keywords assigned to an abstract.

Program Modifications

Installations using the LIRES-DF program may wish to ease some of the restrictions previously listed. The limitation of thirty-two searches per run cannot be altered easily. The LOR and LAND functions which manipulate binary bits are designed to operate on one word of core storage (32 bits) and modifications would be significant. Other alterations require changes in the DIMENSION statement and various input/output, and control statements. These may be tailored to the particular installation and the amount of core storage available.

Modifications of the restrictions may be effected as follows (refer to the source and cross reference listings):

1. In order to increase the number of logic groups (over 10), change the read statement (No. 24) in the main line program to (WNO(K), K=1, n) where 'n' is the number of logic groups that is to be accommodated. In subroutine RDPROF change the limit of the DO loop statement (No. 7) to 'n'. Take care to alter the FORMAT statement 9002 to accommodate the number of items to be read.

2. In order to accommodate more than 320 words in total alter the DIMENSIONs for the variables WB, WL, WTS, TAB, LOG, and WT to the desired number in the main line program. Statement 16 in RDPROF will have to be changed to IF (NE - n) 24,24, 17 also.

3. The limit of 4000 characters for profile terms can be increased by changing the DIMENSION for the variable WRD to m = (n/8) where the 'n' is the desired character storage and 'm' is the number to be put in the dimension. The statement LIMIT = 4000 (No. 17) should be changed to 'n'. All of these changes are in the main line program.

Alterations of the LIRES-DF program to process a data file with a different format, or with different types of information is possible. Subroutine RDTEXT (see statement 88) would have to be replaced with a module which would read the new data file. The important information which is returned is the text and the parameter lists TB and TE. The text is expected to be in one continuous string. The variable lists TB and TE contain the beginning (TB) and ending (TE) character locations for the type of information (title, text, keywords, etc.) that is to be searched. Statements 84 and 85 in the main line program read the abstract heading information independently. They would also have to be altered to process a different type of record.

Introduction

The LIRES-IF (LIterature REtrieval System-Inverted File) system has been developed at the Computer Center of The Institute of Paper Chemistry. The system is designed to permit searches of an inverted file for information retrieval purposes. It is written in the basic FORTRAN-IV and in basic Assembler Languages. The IBM Sort/Merge utility is used at several stages in the searching process. The system is designed to run in 66K of core storage, and requires two tape drives and sufficient disk space for temporary storage of data sets.

A glossary of terms which are used in this program description manual follows:

1. LOGIC GROUP — a collection of words which have the usual "OR" connotation.
2. PROFILE — a collection of logic groups which comprise the words of interest for an information search.
3. LOGIC STATEMENT — a statement of what combinations of logic groups constitute a "HIT" for a profile.
4. HIT — an abstract which satisfies (or matches) a logic mask or masks for a search profile.
5. PROFILE TERMS — a word in the form that is to be used in a search against the data file, i.e., its truncated or full word form.
6. PROFILE TERM WEIGHT — a number assigned to a profile term, which if the term is found in an abstract is to be accumulated into a total for the abstract.
7. PROFILE THRESHOLD — a minimum number which is compared against the accumulated profile term weights. If the logic mask of a profile indicates that an abstract is a hit, but the total of the weights does not equal or exceed the threshold, the abstract is not retrieved. If the logic mask is not satisfied, the abstract is not a hit regardless of the accumulated weights.

Program Description

The LIRES-IF program will accept up to 255 profiles to be searched against an inverted file. A direct file of the information contained in inverted form is required for the purpose of printing citations (and abstracts if desired).

A profile is comprised of identification and parameter information, search profile terms, and logic statements. The profile terms may be words up to 39 characters in length; they may be right truncated; and they may be weighted. Logic statements are used to indicate the combinations of logic groups which are to be called "hits." At least one word from each logic group identified in a logic statement must be found in order for an abstract (or document) to be retrieved.

The program searches the inverted file for the profile terms and collects the abstract (or document) numbers as the words are found. The numbers are sorted and compared to the logic statements. As a document number is found to satisfy a logic statement, the direct file is used to obtain the citation (and abstract or text) information. After gathering all citations that are to be printed, they are sorted in the following order:

1. Profile number.
2. The citation weight (from profile term weights).
3. The number of profile terms found in citation.
4. The document number.

The user has the option of specifying that the citation only or the citation and accompanying text be printed. Profile terms found in the citation are always printed.

The user may consider some profile terms more important than others for his search. He may assign unequal weights to the profile terms, and supply a threshold weight of zero. This will cause the citations with those important terms to be printed ahead of those of relatively lesser importance.

Profile Preparation

A search requires the following cards:

a. Search identification card — free form, use all 80 columns
b. Parameter card — IND, NLOG, MAX, LIM, NW1, NW2,...
c. Profile terms — follow the term immediately with an asterisk (*) for truncation
d. Logic statement(s) — LG1, LG2,... (3 columns each, right justified)

Repeat the instructions above for other searches (maximum of 255/pass). Place a card punched "END" in columns 1, 2, 3 after the last search to be included in the processing run.

Parameter Card Definitions (5 columns each, right justified)

IND = 1 to print citations, = 2 to print citation and text of abstract
NLOG = The number of logic statements for the search
MAX = The profile threshold
LIM = Limit on the number of hits to be printed
NW1-NW10 = The number of words in logic groups one through ten.

Logic statement(s) are prepared according to the following scheme: A logic statement defines which combination of groups are to be "AND"ed together. More than one logic statement may be entered. These statements describe combinations of logic groups which, if satisfied, will constitute a hit. "NOT" logic is indicated in logic statements by punching a minus sign by the logic group number. A logic statement, for example, which defines logic groups 1, 2, and not 4 as a hit is written (and punched) as 1 2 -4. At least one logic statement must be present for a search profile. Logic masks are generated from the statements on a one-for-one basis; a maximum of 500 masks are permitted for the 255 or fewer searches.

Restrictions

The following restrictions apply to a PROFILE:

1. Not more than ten (10) logic groups may be defined in one profile.
2. Profile terms should not exceed 39 characters in length.
3. Profile term weights should be in the range -999 to +999.

The following restrictions apply to one processing run (one pass of the files):

1. Not more than 255 profiles.
2. The total word count for any one or all of the profiles may not exceed 2000.
3. The total number of logic statements for any one or all profiles may not exceed 1000.

The restrictions on the number of words and the number of logic statements may be eased at the expense of more core storage for the program. Four bytes (one computer word) is required for each additional word and/or logic statement. More profiles may be handled at the expense of core storage and also disk storage. Profile terms, document numbers, and citations are stored on disk temporarily. An increase in the number of profiles would cost sixteen bytes (four computer words) for each additional profile to be handled. The system programmer should be consulted about the job control and disk space allocation for the additional profiles. The coding in the system makes it theoretically possible to handle more than a half-million profiles in one processing run.

A system programmers guide is provided with the source program to guide and assist the systems group in the installation of the system. The LIRES-IF system is being used under the Operating System (Release 19.6) and the RAX time sharing system at The Institute of Paper Chemistry.

FILE DESCRIPTION

Full Text with Keywords

The full text with keywords file of the Abstract Bulletin of The Institute of Paper Chemistry is prepared on 9 track, 800 b.p.i. computer tape. Each record contains 80 bytes (blocked 80/3600) and an abstract records arranged as follows:

Record	Bytes	Information	Mode
1	1-4	Total Length of abstract data	Binary
	5-8	Abstract number	Binary
	9-12	Byte location of Author data	Binary
	13-16	Byte location of the Title	Binary
	17-20	Byte location of Reference data	Binary
	21-24	Byte location of Text data	Binary
	25-28	Byte location of Keywords	Binary
	29-32	Byte location of Volume, Issue	Binary
	33-36	Total length of abstract data	Binary
	37-80	The beginning of the abstract (44 bytes)	EBCDIC
2-n	1-80	Abstract data	EBCDIC

There will be a varying number of records for each abstract, depending upon the length of the various parts. One may determine the number "n" by dividing the total length (obtained from either location) by 80. The file is terminated by a single 80 byte record with zeroes stored in all pointer and length locations. The EBCDIC data is in upper and lower case.

Inverted File

The inverted file of the Full Text with Keywords and the Keyword Supplement of the Abstract Bulletin of The Institute of Paper Chemistry is prepared on 9 track, 800 b.p.i. computer tape. Each record contains 80 bytes (blocked 80/3600).

In the case of the Full Text with Keywords File, all "words" have been converted to upper case EBCDIC characters. The meaning of "word" is as follows:

1. An author's full name is maintained as a word, i.e., blank spaces, and punctuation.

2. The name of the publishing journal is maintained as a word.

3. Keywords are as defined in the Thesaurus of Pulp and Paper Terms.

4. All other words are defined as the information between two blank characters after all punctuation has been removed.

All words on the inverted file have a maximum of 40 characters. Company names which appear as authors in patent abstracts that exceed the maximum have been truncated.

Abstract numbers have a source code affixed. The source codes have the following meaning:

1 - This word came from an author record.

2 - This word came from a title record.

5 - This word came from a journal reference

6 - This word came from the text of an abstract.

8 - This word came from the assigned keywords.

All numbers occupy four bytes (20 per record) in binary. The source code can be stripped from the abstract number by dividing the number by ten.

MULTIPLE CARD FORM

1		RECORD LAYOUT - INVERTED FILE
2	TAPE HEADER RECORD	← VOLUME AND ISSUE(S) IDENTIFICATION (EBCDIC) →
3	KEYWORD RECORD	INVERTED FILE "WORD" EBCDIC → | No. OF ABSTRACT NUMBERS (BINARY) | ← UNUSED → | No. OF ABSTRACT NUMBER (EBCDIC
4	ABSTRACT NUMBER RECORD	ABSTRACT NUMBERS WITH SOURCE CODES (ALL IN BINARY): #1 | #2 | #3 | #4 | #5 | #6 | #7 | #8 | #9 | #10 | #11 | #12 | #13 | #14 | #15 | #16 | #17 | #18 | #19 | #20
5		
6		

Form 902
9/9/58
CY

MULTIPLE CARD FORM

1

RECORD LAYOUT - FULL TEKT WITH KEYWORDS FILE

9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79

2 RECORD 1 OF ABSTRACT

1–4	5–8	9–12	13–16	17–20	21–24	25–28	29–32	33–36	37–79
TOTAL LENGTH OF ABSTRACT (BIN)	ABST. NUMB. (BIN)	LOC. OF AUTHOR (BIN)	LOC. OF TITLE (BIN)	LOC. OF REF. (BIN)	LOC. OF TEXT (BIN)	LOC. OF KWDS (BIN)	(BIN) LOC. OF VOL. ISSUE	(BIN) TOTAL LENGTH OF ABSTRACT	← FIRST 44 BYTES OF ABSTRACT RECORD EBCDIC →

9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79

3 REMAINING RECORDS

← 80 BYTES OF ABSTRACT RECORD (EBCDIC) →
(LAST RECORD PADDED WITH BLANKS)

9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79

4 LAST RECORD ON FILE

1–4	5–8	9–12	13–16	17–20	21–24	25–28	29–32	33–36	37–79
BINARY ZEROES - LAST RECORD									← BLANK →
Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	

9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79

5

9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79

6

9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79

Form 902
9/9/58
CY

PROJECT REPORT FORM

PROJECT NO. 2318
COOPERATOR Institute of Paper Chemistr
REPORT NO. Five
DATE January 25, 1974
NOTE BOOK
PAGE TO
SIGNED John A. Church

Copies to: Files
Church
Holm
McClenahan
Weiner

AUTOMATED POSTING OF BROAD TERMS TO THE KEYWORD SUPPLEMENT

INTRODUCTION

Beginning with Volume 42 of the Abstract Bulletin of The Institute of Paper Chemistry, indexing of abstracts has been performed using the Thesaurus of Pulp and Paper Terms as the authority. The thesaurus provides a hierarchy for all terms to be used in indexing; narrower, broader, related, used for, or use index terms if any are listed for each permissible term.

MANUAL INDEXING

Indexers at the Institute manually assign index terms to abstracts to the extent that the essential content of the abstract is indicated. These "keywords" are keypunched and verified, then are processed through the computer system providing the inverted file and direct file of the Keyword Supplement of the Abstract Bulletin in machine-readable and hard copy versions.

SEMI-AUTOMATIC INDEXING

With Volume 44, number 1 a new dimension was added to the procedure. After the manual indexing process is completed, the broad terms for keywords assigned are automatically added before producing the magnetic tapes and printed versions. Now someone searching for information, and using the Keyword Supplement, about computer applications only has to look up "COMPUTERS" which is automatically posted for articles which had "DIGITAL COMPUTERS" or "ANALOG COMPUTERS" manually assigned. We say that the Keyword Supplement is <u>upwards general</u> or <u>downwards specific</u>.

COMPUTERIZED INFORMATION RETRIEVAL

Computer systems have been developed which will search the magnetic tape version of the Keyword Supplement and report back abstracts which had specified combinations of keywords assigned to them. Unless the request is very general or very specific, two groups of index terms are usually organized and abstracts which have a word from each of the groups are retrieved. For instance, if "computer applications in information processing" was the topic of interest, the two groups of words might be

DIGITAL COMPUTERS	INFORMATION RETRIEVAL
COMPUTER PROGRAMS	INFORMATION SYSTEMS
DATA PROCESSING	DATA RETRIEVAL
AUTOMATION	

All abstracts assigned a word from column one and a word from column two would be retrieved. References which are not pertinent to the topic are sometimes retrieved along with the useful abstracts. Sometimes these "false drops" are more numerous than the "hits." The Institute offers monthly information on twenty-seven standard subject profiles. Twelve of these profiles have been tested to determine the effect of having the broad terms posted. Table I shows the results of the two searches.

TABLE I

SEARCH OF VOLUME 44, NUMBER 7

Profile Number	Broad Terms Not Posted: Hits	False Drops	Total	Broad Terms Posted: Hits	False Drops	Total	Extra Hits	Extra False Drops
1	13	2	15	16	7	23	3	5
2	5	4	9	7	4	11	2	0
3	4	0	4	4	0	4	0	0
4	15	1	16	15	1	16	0	0
5	36	2	38	45	3	48	9	1
8	12	7	19	12	7	19	0	0
9	23	5	28	25	6	31	2	1
10	11	1	12	11	1	12	0	0
11	16	1	17	16	1	17	0	0
12	11	0	11	11	0	11	0	0
13	4	0	4	4	0	4	0	0
15	19	0	19	19	0	19	0	0
	169	23	192	185	30	215	16	7

From these data we draw the following conclusions:

1. That the addition of the broad terms does not produce an "avalanche" of retrieved references. There was a modest increase of 23 in this experiment.
2. That the ratio of hits to the total does not change significantly. The ratio was 88% before posting and 86% after.
3. That better results are obtained using the file with broad terms posted. An increase of 9% in the number of hits (16/169) was obtained.

Custom profiles are usually designed for a narrower interest than the standards tested above. Improved results should be forthcoming with the new file for custom profiles too. The searcher may choose a keyword at any level in the hierarchy and will obtain all references to more specific subject matter related to that keyword.

PROJECT REPORT FORM

Copies to: Files
R. Holm
J. Weiner

PROJECT NO. 2318-02
COOPERATOR Institute of Paper Chem.
REPORT NO. 4
DATE March 16, 1972
NOTE BOOK
PAGE 10
SIGNED Robert A. Holm
Dr. Robert A. Holm

PPRIC THESAURUS

Computer Storage and Use of the Pulp and Paper Thesaurus

INTRODUCTION

The second edition of the pulp and paper thesaurus initiated by the Pulp and Paper Research Institute of Canada (PPRIC) and jointly supported and improved by PPRIC and The Institute of Paper Chemistry has recently been published. This source document is very useful in controlling the concepts and keyterms used in indexing information for efficient retrieval to serve the research and management needs of the pulp and paper industry.

In the development of lists of keywords to describe a particular area of concern, one often begins with only a few major keywords, and then proceeds to use the thesaurus to guide him in the selection of other narrower, broader, or related terms with which to improve the effectiveness of his search profile. This procedure becomes very tedious, time-consuming, and subject to error when the number of original terms is above five or so, or the concepts concerned are fairly general so the associated keyterms become numerous.

The objective of this work was threefold.

1. To store the complete pulp and paper thesaurus (PPT) on the Institute computing system library.
2. To design, develop, and test a series of modular subroutines which would allow flexible access to the information thus stored.

3. To develop and test a program which demonstrates the conceptual capabilities of the system and which has a direct practical usefulness in developing better search profiles.

HOW TO GET THREE QUARTS OF WATER INTO A ONE QUART JAR

A complete magnetic tape copy of the PPT was sent to the Institute courtesy of Peter Nobbs. In its original complete form the thesaurus occupies 3,400,000 bytes of space, roughly sixty percent of all the available space on our computer system file disk. It would have been possible to leave the PPT on tape and use it directly in that form, but then its use would be restricted to the single card-printer terminal in the main computer room. In order to use the PPT from one of the six video terminals, it was necessary to find some way of shoehorning it into a more reasonable portion of the system file disk.

In order to do this, the complete logical contents of the PPT were analyzed using specially designed computer programs, to break the complete contents of the PPT down into three separate, mutually dependent files which contained the essence of the complete PPT, and from which, should need arise, any portion of characteristic of the PPT could be regenerated.

The three files which were generated and which are used in all the subsequent programs occupy only one fifth of the original file size and yet contain all the information of the original PPT. A description of these files and their use follows.

The Keyterm File

This is the only truly alphabetic file of the three. This file contains the proper English listing of each keyterm in alphabetic order.

The Address File

This numeric file is a series of six-number groups, one group for each of the 6197 keyterms. The six numbers in each group give information on the following six items for each keyterm:

1. The address within the "number" file where the list of narrower terms begins.
2. The number of narrower terms assigned to that keyterm.
3. The address within "number" where the list of broader terms begins.
4. The number of broader terms assigned to that keyterm.
5. The address within "number" where the list of related terms begins.
6. The number of related terms assigned to that keyterm.

This file essentially collapses the logical structure of the PPT into one list of addresses and associated keyterm counts.

The Number File

This is also a numeric file, but rather than storing addresses and counts as the above file, it is a compact, complete list of the keyterm numbers for the associated keyterms actually listed in the PPT. The listing of the keyterm numbers rather than the alphabetic words themselves is the major source of savings in storage space.

GOING AROUND YOUR ELBOW TO GET TO YOUR THUMB

A typical example of how each file is dependent on the other can be seen from the series of operations which could be made to retrieve and print an elementary result, the alphabetic list of all the keyterms which are posted as "related terms" for a particular word, e.g. 'bleaching'. The computer looks up 'bleaching' in the keyterm file and finds its key-term number.

The computer then goes to the address file and finds the address and the number of related keyterms. The computer then goes to the number file at that address and gets that keyterm number and the following related numbers. Finally, the computer goes back to the keyterm file where it started and prints out the alphabetic words corresponding to each of the keyterm numbers retrieved from the number file.

This is an involved process, but the interaction of the three files in this manner can be accomplished in a direct and error-free manner with the equipment and programs on hand, and at the same time save the need for investment in tens of thousands of dollars worth of additional storage devices.

BOOTSTRAPS AND SHOESTRINGS

In order to effectively use this stored set of files, a series of Fortran subroutines was written to perform each of the sequential operations needed in using the files. By combining these subroutines with simple sorting routines (such as "SORTAL") quite a broad range of programs can be developed to select, combine, analyze, and report on the relationships of the words implicit in the PPT. The final demonstration program described below shows only one example of the use of these subroutines in a connected, coherent manner.

The subroutines which are used to get to the information are FORM, NKT, ADDR, REF and WORDS. The detailed list of parameters and description of the operation of these subroutines are included in the program listing in the 2318-02 work files, but a qualitative description of these routines is included here for your information.

Form Subroutine

This subroutine takes a full paragraph of words which have been typed into the video terminal with some arbitrary dividing mark (in the present case, a comma followed by a blank) and stretches it out so that the terms occur on evenly-blocked records (in this case 40 characters per keyterm). All of the subsequent operations use this evenly-blocked word list.

Nkt Subroutine

This subroutine takes a series of words in a list and locates the corresponding values of the keyterm identification number assigned in the PPT.

Addr Subroutine

This subroutine retrieves the addresses and counts which tell where the keyterm numbers associated with a given keyterm are located in the third file. The addresses and counts specify these values for narrower, broader, and related keyterm numbers.

Ref Subroutine

This subroutine references the third file using the addresses and count supplied and pulls out the needed keyterm numbers.

Words Subroutine

This subroutine uses a finally compiled and sorted (if desired) list of keyterm numbers and transfers the actual alphabetic words into a word list for reporting or further analysis.

AUTOPROFILE ONE

A helper program called A U T O P R O F I L E O N E was conceived, programmed and tested to demonstrate the capabilities of the individual segments described above to serve a particular need, in this case, the accumulation and reporting of generic and associated words related to a given set of original keyterms. The resulting list can then be used to suggest possible additional keyterms which might have been overlooked in the original analysis of the search question or abstract. A typical set of input data are listed in Table I. The output from this particularly extensive and complex profile analysis is given in Table II. Those related terms which occurred most frequently in the accumulation of the 286 related terms are given at the head of the list, together with a count of the frequency of occurrence of each term. Only the first hundred terms are printed, as these should be more than adequate for suggesting additional terms for practical search profiles.

THE MILLS OF THE GODS GRIND SLOW, BUT THEY GRIND EXCEEDING FINE

The concept of the program development and its operation have been proven. Unfortunately the program runs very slowly due to the inefficient method in which the present subroutines read the disk files. A simple profile takes about 3 minutes on the time-shared system, and

TABLE I

TYPICAL INPUT TO PROGRAM PPT

```
ØINCLUDE PPT
INDUSTRIAL WASTES, POLLUTION, POLLUTION CONTROL, RECYCLING,
WASTE DISPOSAL, WASTES,
BARK, ELECTROSTATIC PRECIPITATORS, FLY ASH, LANDFILL, RECLAIMED FIBERS,
REJECTS, RESIDUES, SAWMILL RESIDUES, SCRAP, SCREENINGS, SLUDGE,
SLUDGE DISPOSAL, SOLIDS, WASTE PAPERS, WOOD WASTE,



ØEND RUN
```

TABLE II
TYPICAL OUTPUT FROM PROGRAM PPT

THERE WERE 21 BASIC, 25 GENERIC, AND 286 RELATED TERMS

BASIC KEYWORDS

1	BARK
1	ELECTROSTATIC PRECIPITATORS
1	FLY ASH
1	INDUSTRIAL WASTES
1	POLLUTION
1	POLLUTION CONTROL
1	RECLAIMED FIBERS
1	RECYCLING
1	REJECTS
1	RESIDUES
1	SCRAP
1	SCREENINGS
1	SLUDGE
1	SLUDGE DISPOSAL
1	SOLIDS
1	WASTE DISPOSAL
1	WASTE PAPERS
1	WASTES
1	WOOD WASTE

NARROWER AND BROADER TERMS

2	WASTES
2	VENEER WASTE
2	SHAVINGS
2	SAW MILL RESIDUES
2	SAW DUST
2	DISPOSAL
1	WOOD WASTE
1	WATER POLLUTION
1	TAILINGS
1	SEWAGE DISPOSAL
1	SEWAGE
1	SEPARATORS
1	REFUSE
1	PLANT TISSUES
1	LAND FILL
1	INDUSTRIAL WASTES
1	ASH
1	AIR POLLUTION
1	ACTIVATED SLUDGE

RELATED TERMS

10	WASTES
7	WASTE DISPOSAL
6	INDUSTRIAL WASTES
5	SCREENINGS
5	RECOVERING
4	TAILINGS
4	SPENT LIQUORS
4	SEWAGE TREATMENT
4	POLLUTION
4	IMPURITIES
4	AIR POLLUTION

more complex ones like the one shown in Table I can take up to 15 or 20 minutes. This is still a large time saving over manual methods of comparable scope. Theoretically it should be possible to arrange more complex logic into the subroutines to permit only one search of each file for each profile, thus cutting the time to a bare minimum of approximately one or two minutes independent of the size or complexity of the profile. More efficient subroutines which have the same argument lists and produce the same results as the present brute force routines could be substituted in the future with no difficulty. Such improvements are recommended as the program's usefulness expands and the demand is demonstrated. For the present, no further work is planned. Even at the present state of the program, it can be useful for checking and expanding exceptionally complex or important profiles, such as our standard monthly keyword profiles or for quality control checking of keyword assignments to abstracts.

ACKNOWLEDGMENT

This approach to information analysis would be completely impossible without the special utility subroutines which have been written and incorporated into our RAX system by John O. Church for the direct manipulation and comparison of alphabetic strings of characters. The availability of these utility subroutines has expanded the usefulness of our machine to make the conceptual level that of much larger and more expensive equipment. Notice of this contribution is called for and herewith recorded.

PRESENT AVAILABILITY OF PROGRAM

The program has been stored on the RAX library under the name "PPT" and can be used by typing any list of keyterms into the terminal in accord with Table III.

TABLE III

TYPICAL USE OF PPT FROM VIDEO TERMINAL

```
/INCLUDE PPT
ABIES, ABIES ALBA, ABIES AMABILIS, PICEA, FORESTRY
HARVESTING, FOREST MANAGEMENT,
     (a full ten lines is required, fill in with blank
      lines or blank cards if there are not enough words.
      Words may be on separate lines, but each must be
      followed by a comma and one or more blank
      spaces.)
WOOD WASTE,
UTILIZATION,

/END RUN
```

Some difficulties have arisen for longer profiles run from the video terminals with regard to 'excessive output, job deleted.' If this occurs, try running the same program from the batch (card reader) terminal. Additional improvements to trade off the maximum word list size (now 800 total basic, generic, and related words) with the input-output buffer size may be possible. Four hundred words (before sorting) might be adequate for all but the rare exceptions.

PROJECT REPORT FORM

PROJECT NO. 2318
COOPERATOR
REPORT NO. 3
DATE May 29, 1968
NOTE BOOK
PAGE TO
SIGNED Jerry Byrne
Jerry R. Byrne

Copies to: Files
Speckhard
Bachhuber
Brown
Dickey
Holm
Nelson
Roth
Weiner
Grow

PROCEDURES FOR PRODUCING THE KEYWORD SUPPLEMENT TO THE A.B.I.P.C.

SUMMARY

This report describes the procedures for producing the Keyword Supplement to the Abstract Bulletin using the IBM 1620 digital computer and ancillary equipment. Two indexes are prepared: the monthly supplement and the semiannual index which brings together six month's keywording effort. Several keyword lists are also prepared periodically for use in thesaurus updating, analysis of keyword usage, etc.

PREPARING THE MONTHLY SUPPLEMENT

A xeroxed copy of the manuscript of the Abstract Bulletin is made available to the keyworders within a few days after the closing date of the issue. After the keywords have been assigned, the manuscript is given to the keypunch operators. The abstracts are kept in numerical order with the corresponding keywords attached. The punched cards are prepared according to the format as shown in Fig. 1. Each keyword term is separated from one another by a comma and a double space. The last word on the card and its comma cannot be further to the right than column 74 as there must be at least two blank columns before the abstract number which is punched in the last four columns of the card. In the instances where the abstract number is less than four digits in length, the number is written so that the right hand digit is

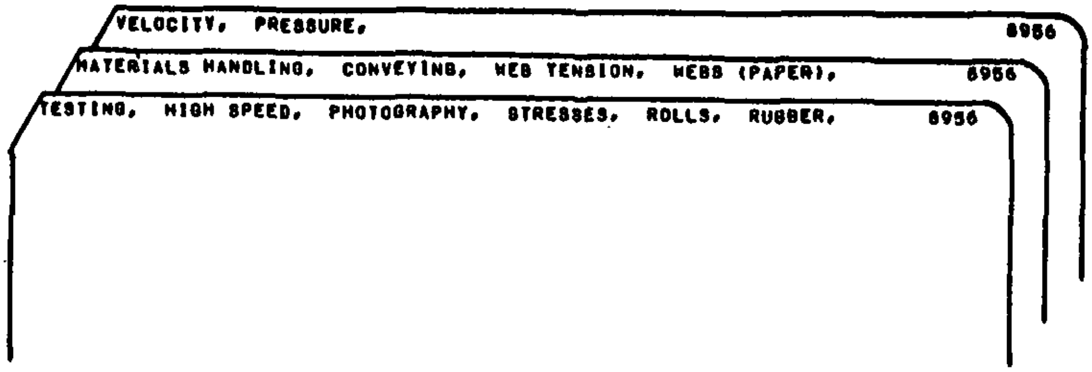

Figure 1. Punched Cards as Prepared by the Keypunch Operators

always in column 80. Each card should carry as many keywords as possible under the above limitation and as many additional cards as needed can be used. The abstract number must be repeated on every additional card used for the same abstract, however.

After all the cards are punched, the deck is listed on the printer and this list is then used to check for punching and keywording errors. The cards are corrected and repunched where necessary.

The cards are now ready to be loaded to disk in order to prepare the inverted index; i.e., keyword with corresponding abstract numbers. Both disk drives are used; the IBM 1311 disk storage drive model 3 (main drive) and the 1311 model 2 (satellite). The main drive will have the sort-merge disk pack mounted on it and the satellite will contain a scratch pack. The sort-merge disk pack contains all the programs needed for the production of the keyword supplement; therefore, all programs need only to be called up into core storage at the time they are needed rather than be loaded manually each time. The program used for loading the disk pack for inverting the index is named PLIDFI. This program reads a card and pulls off the number and the first keyword on the card. It then proceeds to make number-keyword combinations out of the rest of the keywords on the card, reads the next card, and repeats the process. This continues until all numbers and keywords are read into the computer. The number-keyword combinations are put into a large output area in the computer and when this area is filled, the records are put on the storage or scratch disk pack. The storage pack has space available for approximately 10,000 number-word combinations. The last thing this program does is put 0++ on the disk for the convenience of the sort-merge program which follows.

The sort-merge program is basically an IBM program which they call SM047 and which is stored on the sort-merge disk pack by the name of SORT. This program does the actual inverting of the index. It generates tags which are made up of up to twenty-five letters of the keyword and information giving the location of the keyword on the storage disk pack. These tags are put into the work area of the main disk pack and alphabetized by keyword in groups. The groups are then merged resulting in all the tags being in alphabetical order on the main disk. The final operation of this program is to present record number one (the first tag in the alphabetical list) to be worked on by the next program.

The next program, named PPIKI, is brought into core automatically by the sort-merge program. This program prints and punches the inverted index. The program looks at record one presented by the previous program, goes to the storage disk location indicated, stores the complete keyword found there in a new storage location, stores the number, and prints and punches the complete keyword with a - punch in column 80. It then looks at the word in the location indicated by record two. If the word is the same as the one stored previously, the abstract number is stored next to the previous number. Since the format calls for a maximum of ten numbers per card, a counter is set up to count the number of abstract numbers being stored. When the count reaches ten, the numbers are outputted along with the first twenty characters of the keyword. Figure 2 shows the so-called header card (full keyword and - punch in column 80) and its corresponding data cards. When a different keyword is encountered, the storage areas are cleared and the new record is treated as record one. In practice, the first pass of the file through the inverting procedures is only to produce a listing of the inverted index for editing purposes. Therefore, a print or print-and-punch option has been built into this program.

The corrected file of cards produced by PPIKI are then worked on by the program called annual keyword index program. This program reads the first card with a - punch in column 80 and all succeeding data cards and stops when it reaches another card with a - punch in column 80. It then arranges all the numbers into columns by last digit and prints out the final form of the inverted index. Figure 3 shows a sample of the printout from this program.

The direct index or keyword by abstract number index is prepared from the original punched cards using the program named PACKAN. This program simply prints the abstract number on the left followed by all the keywords belonging to that abstract. A by-product of this procedure is an indication of the average depth of indexing. While the program is preparing the file for printing, it is also counting the number of abstract numbers and the number of keywords. By feeding the cards into the computer in two groups, periodical section and patent section, it is possible to get the average number of keywords per abstract for each section and for the total issue.

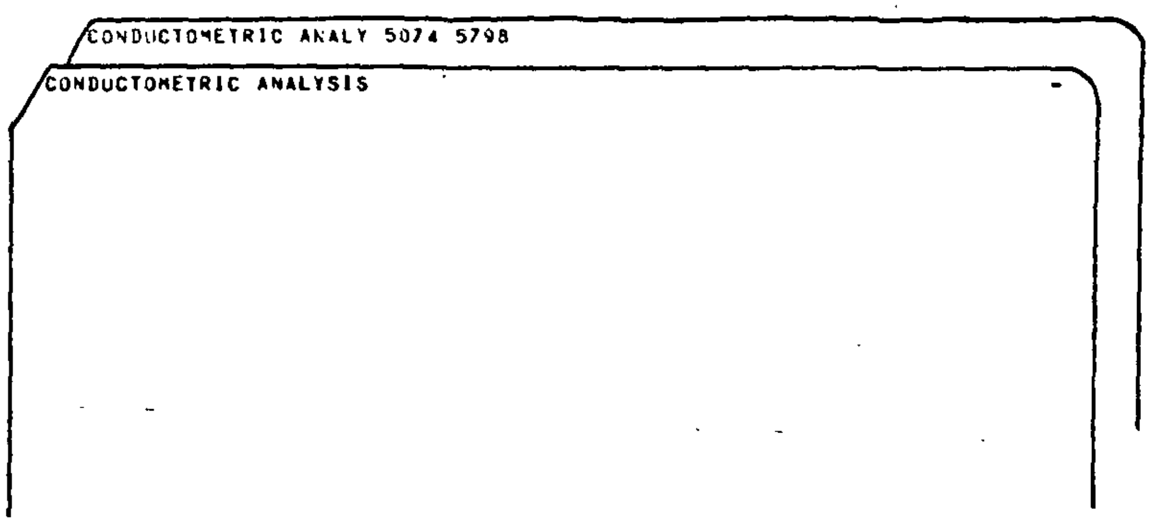

Figure 2. Header Card and Data Cards

ADDITIVES										
	7580	7301	7532	7383	7384	7395	7396	7557		7399
		7431	7562	8063	7394	7765	7576			7519
			7602				7766			8069
							8036			
							8066			
ADHESION										
					7474	7435		7397		7969
					7584					
ADHESIVE PAPERS										
			7872			7445				
ADHESIVES										
	7390	7771	7472	7773	7564	7385	7446	7537	7458	7769
	7430	7781	7732	8023	7574	7395	7796		7578	7969
	7490		7792			7575	7866			
	7580									
	7780									
ADIPIC ACID										
			7782	7783						
ADJUSTMENT										
					7824					
					8034					
AERATION										
				7723						
AFRICA										
			7672							
AGAR										
						7355				
AGGLOMERATION										
		7401								
AGING										
		7381	7612			7345	7576	7647		7319
						7445				
						7665				
AGLYCONE GROUPS										
							7346			
AIR										
				7603						

Figure 3. Sample Printout of the Inverted Index

PREPARING THE SEMIANNUAL SUPPLEMENT

Cumulative direct and inverted indexes are prepared twice a year and are distributed in place of issues 6 and 12 of the volume. The direct or keyword by number index is prepared by merely assembling the individual issue direct indexes in numerical order and putting them through the PACKAN program. The preparation of the cumulative inverted index is a little more involved.

The cards from the six monthly inverted indexes are sorted by the first letter of the keyword on an IBM 83 card sorter. The sorted cards are then loaded to the disk pack using the program named PLMIFR. This program prepares and stores records on the disk pack from the inverted file cards in the same manner that PL1DFI prepares and loads records from the direct file. Since space limitations on the disk allow for only 10,000 records to be stored at one time, the sorted inverted file must be loaded in batches. However, all the cards with keywords starting with the letter A, for example, must be loaded together. The number of records transferred to disk storage may be checked at any time during the operation by turning program switch one on at the console. After loading as many records as possible, the sort-merge program is called and the operation proceeds as in the preparation of the monthly inverted indexes.

PREPARATION OF KEYWORD LISTS

A list of all the keywords used to date along with their frequency of usage can be of use for thesaurus updating, checking consistency of class usage, determining how heavily certain keywords are posted, etc. A program has been written called CALANK to prepare a list of all keywords used in a six-month period, for example, along with the number of times the keyword has been posted during that period of time. The program takes the cards produced by PPIKI from the semiannual file, reads the first keyword and counts the number of abstract numbers associated with it. When it comes to a new keyword, it prints the first word and its frequency of usage and repeats the process. A sample of this list can be found in Fig. 4. This list is also punched on cards. The original list was prepared after 18 months of keywording. In order to incorporate the last six months and make the list reflect the usage for a full two years, it became necessary to write a program to merge the two lists. This program is called MLOKAC. The shorter word list is loaded to disk using the loading program JOHN, which simply loads the data on the cards directly with no change. MLOKAC is called and the larger list is passed by the other list by feeding it through the card reader. The program compares words and, if the words are the same, adds up the frequencies of usage and prints the word and the new total. If the words are different, the program adds the word to the list in its proper alphabetical order. The new merged list is then printed and punched on cards.

This list of all keywords used has been the starting point for a number of special lists used primarily to facilitate thesaurus updating. Programs are available for producing the following lists: keywords used only once, keywords used 100 times or more, keywords having parenthetical phrases, hyphenated keywords, multi-term keywords, keywords ending in ing, keywords ending in er/or or ers/ors, keywords ending in ate or ates. There is also a program for giving the frequency distribution of keyword posting; i.e., the number of keywords used once, twice, three times, etc. Another program, which uses the list of multi-term keywords as its source, is one which produces a permuted keyword list. A multi-term keyword is defined

16	ABATEMENT
1	ABIENOL
82	ABIES
9	ABIES ALBA
5	ABIES AMABILIS
25	ABIES BALSAMEA
6	ABIES CONCOLOR
11	ABIES GRANDIS
5	ABIES LASIOCARPA
1	ABIES MAGNIFICA
1	ABIES NORDMANNIANA
1	ABIES PINDROW
1	ABIES PINSAPO
2	ABIES SACHALINENSIS
4	ABIES SIBIRICA
1	ABIETADIENE
1	ABIETENE
19	ABIETIC ACIDS
2	ABIETINAL
1	ABIETINOL
3	ABNORMALITIES
9	ABRASION
61	ABRASION RESISTANCE
2	ABRASION RESISTANT STEELS
1	ABRASION TESTERS
11	ABRASIVE PAPERS
4	ABRASIVES
34	ABSORBENT PAPERS
5	ABSORBERS (EQUIPMENT)
8	ABSORBERS (MATERIALS)
74	ABSORPTION
26	ABSORPTION SPECTRA
21	ABSORPTIVITY
11	ACACIA
1	ACACIA ARABICA
1	ACACIA LAETA HASHAB
1	ACACIA MEARNSII
1	ACACIA NILOTICA
1	ACACIA PENNINERVIS
4	ACACIA SENEGAL
7	ACCELERATING (PROCESS)
3	ACCELERATION (MECHANICAL)
1	ACCELERATORS
1	ACCEPTABILITY
3	ACCEPTANCE
26	ACCESSIBILITY
1	ACCESSORIES
2	ACCIDENT PREVENTION
1	ACCIDENTS
6	ACCOUNTING
3	ACCUMULATION
2	ACCUMULATORS
8	ACCURACY
23	ACER
1	ACER NEGUNDO
3	ACER PLATANOIDES

Figure 4. List of Keywords Used and Frequency of Usage

as any keyword that consists of two or more words, is hyphenated, or contains a parenthetical phrase. The PICTAL/PICTAP pair of programs separates each multi-term keyword into its component words, collects all keywords that

```
ABSORPTION
     ABSORPTION SPECTRA
     ATOMIC ABSORPTION SPECTROSCOPY
     ELECTROMAGNETIC ABSORPTION
     ELECTRON ABSORPTION SPECTRA
     INK ABSORPTION
     OIL ABSORPTION
     WATER ABSORPTION

ACACIA
     ACACIA ARABICA
     ACACIA LAETA HASHAB
     ACACIA MEARNSII
     ACACIA NILOTICA
     ACACIA SENEGAL

ACCELERATING
     ACCELERATING (PROCESS)

ACCIDENT
     ACCIDENT PREVENTION

ACETAL
     ACETAL RESINS
     POLYVINYL ACETAL RESINS

ACETATE
     ACETATE PULPS
     ACETATE RAYON
     BUTYL ACETATE
     CELLULOSE ACETATE
     CELLULOSE ACETATE BUTYRATE
     CELLULOSE ACETATE PROPIONATE
     ETHYL ACETATE
     MERCURIC ACETATE
     POLYVINYL ACETATE
     VINYL ACETATE

ACETIC
     ACETIC ACID
     ACETIC ANHYDRIDE
```

Figure 5. Permuted Multi-term Keyword List

contain one of these component words, and prints them along with the component word common to all. Figure 5 is an example of this permuted keyword list.

Print-outs of all the programs mentioned in this report for producing

PLIDFI

```
*NAMEPLIDFI
     ***PRODUCE INVERTED KWD INDEX FROM KWD BY ABSTR NO INDEX***
    *****TO LOAD CARDS TO DISK FOR SM047*****
    *****PROJECT 2318-1*****
           DORG20000,,,DEFINE ORIGIN AT 20000
     N     DC  2,0,,DEFINE COUNTER FOR CARDS
     OUT   DSS 4000,,,AREA FOR 40 SECTOR OUTPUT AREA
     GM    DGM ,,,GROUP MARK FOR TRANSFER TO DISK
     START SK  DISK
           BI  *+12,00900,,TURN OFF LAST CARD INDICATOR FOR START
           CF  CL1+1,,,CLEAR FLAG
           CF  CL1+41
           CF  CL1+81
           BS  *+12,1,,SELECT BAND 1 AND GO TO NEXT STATEMENT
           BLXM*+12,OUT,10,PUT ADDRESS OF OUT IN REGISTER 1 AND GO TO
     ST1   BI  DONE,00900,,GO TO DONE IF LAST CARD
     S12   TFM K,2,10,BEGIN COLUMN COUNT AT 2
           TFM LEFT,INPUT-1
           TFM RIGHT,INPUT+1
           RACDINPUT,,,READ CARD INTO INPUT
           BLXM*+12,INPUT+1,9,PUT ADDRESS OF COLUMN IN REGISTER 2
           BLXM*+12,INPUT-1,910,PUT ADDRESS OF FIRST COLUMN IN REGISTER3
           TRNMPUTT+1,INPUT+2*75-1,,ABSTR NO TO OUTPUT
     S15   SF  (2),,,SET FLAG FOR COLUMN COMPARISON
           CM  1+(2),,10,COMPARE COLUMN TO BLANK
           BE  S10,,,IF BLANK GO TO S10
     S14   CM  1+(2),23,10,COMPARE COLUMN TO COMMA
           CF  (2),,,CLEAR FLAG FROM COLUMN
           BE  LOAD,,,IF COMMA GO TO LOAD
     S10   CF  (2),,,CLEAR FLAG FROM COLUMN
           CM  K,75,10,COMPARE COLUMN COUNT TO 75
           BE  ST1,,,IF COLUMN COUNT IS 75 GO TO ST1
     S11   AM  K,1,10,IF COLUMN COUNT NOT 75 INCREMENT COLUMN COUNT BY 1
           BXM S15,2,9,INCREMENT COLUMN ADDRESS (REGISTER 2) BY 2 AND GO
     *                TO S15
     LOAD  TR  PUTT+11,CLEAR+11,,,CLEAR OUTPUT RECORD
           TR  (2),TRAK+1,,REPLACE COMMA WITH RECORD MARK
           CF  (2),,,CLEAR FLAG FROM KEYWORD COLUMN
           TRNMPUTT+11,(3),,PUT KEYWORD IN OUTPUT AREA
           TRNM(1),PUTT-1,,TRANSFER CARD IMAGE FROM FIRST OUTPUT AREA TO
     *                SECOND OUTPUT AREA
           AM  N,1,10,INCREMENT CARD IMAGE COUNT BY 1
           CM  N,25,10,COMPARE CARD IMAGE COUNT TO 25
           BE  TRANS,,,IF 25 GO TO TRANS
           BXM EXIT,160,10,ADD 160 TO REGISTER 1 AND GO TO EXIT
     TRANS WDGNDISK
           TFM N,0,10,ZERO OUT CARD COUNT N
           BLXM*+12,OUT,10,INITIALIZE REGISTER 1 FOR OUT
           AM  DISK+5,40,10,INCREMENT SECTOR ADDRESS BY 40
           AM  TRAK,1,10,INCREMENT TRAK COUNTER BY 1
           CM  TRAK,5,10,CHECK IF CYLINDER IS FILLED
           BE  SEEK,,,IF CYLINDER IS FILLED GO TO NEXT CYLINDER
           B7  EXIT,,,IF CYLINDER IS NOT FILLED GO TO EXIT
     SEEK  SK  DISK
           TDM TRAK,0,,TRANSMIT ZERO TO TRAK COUNTER
     EXIT  NOP
     S21   AM  K,1,10,INCREMENT COLUMN COUNT BY 1
           CM  K,75,10,COMPARE COLUMN COUNT TO 75
           BE  ST1,,,IF COLUMN COUNT IS 75 GO TO ST1
```

```
S23    BXM *+12,2,9,INCREMENT COLUMN ADDRESS (REGISTER 2) BY 2 GO TO
*               NEXT STATEMENT
       SF   (2),,,SET FLAG FOR COLUMN COMPARISON
       CM   1+(2),,10,COMPARE COLUMN TO BLANK
       CF   (2),,,CLEAR FLAG FROM COLUMN
       BE   S21,,,IF BLANK GO TO S21
S24    BSX *+12,LEFT,9,PUT ADDRESS IN REGISTER 2 IN LEFT AND GO TO
*               NEXT STATEMENT
       BLX *+12,LEFT,910,PUT ADDRESS AT LEFT IN REGISTER 3 AND GO TO
*               NEXT STATEMENT
       AM   K,1,10,INCREMENT COLUMN COUNT BY 1
       BXM *+12,2,9,INCREMENT COLUMN ADDRESS (REGISTER 2) BY 1 AND
*               GO TO NEXT STATEMENT
       SF   (2),,,SET FLAG FOR COLUMN COMPARISON
       B7   S14
****TRANSMIT DIGITS AND RECORD MARKS TO MARK END OF DATA
DONE   TF   2+(1),EOF-1
       WDN DISK,,,WRITE END OF DATA MARKS ON DISK
       BS   *+12,0,,TURN OFF INDEX BAND 1 AND GO TO NEXT STATEMENT
       CALLEXIT,,,END OF PROGRAM
INPUT  DAS 1
       DAS 79
RM     DC   1,#
SEC    DS   ,20000
SECT   DS   ,040
DISK   DDA ,3,SEC,SECT,OUT
TRAK   DC   2,0
       DC   2,0#
       DC   1,#
EOF    DGM
K      DC   2,0
LEFT   DSA INPUT-1
RIGHT  DSA INPUT+1
CL1    DC   40,0
       DC   40,0
       DC   40,0
       DC   40,0
       DC   1,#
CLEAR  DS   1,CL1-38
PUTT   DAS 1
       DAS 79
       DC   1,#
       DENDSTART
```

```
      PPIKI

     **********PROGRAM TO PRODUCE INVERTED KEYWORD INDEX FROM SORTED
****************KEYWORD FILE
            DORG20000
      ST1   BS   *+12,1,,
            BD  READ,SWITCH,,IF DIGIT IN SWITCH GO TO READ
            BLXM*+12,OUTPUT+38,10,PUT ADDRESS FOR NUMBERS IN REGISTER 1
            RCTY
            WATYST3
            RCTY
            WATYST4
            RCTY
            H
            CF   ZEROS-49,,,
            CF   ZEROS+1,,,
            CF   ZEROS+41,,,
            CF   ZEROS+71,,,
            CF   ZERO-49,,,
            CF   ZERO+1,,,
            CF   ZERO+41,,,
            TRNMOUTPUT-1,ZEROS-49,,
            TDM SWITCH,1,,PUT DIGIT IN SWITCH SO SECTION 1 WILL BE SKIPPD
            SF   02686,,,SET FLAG FOR FIELD TRANSFER
            TF   ADRESS,02690,,TRANSMIT ADDRESS OF RECORD TO WORK AREA
            AM   ADRESS,00160,7,CHANGE ADDRESS TO END OF FIELD
            TD   ADRESS,RM,6,PUT RECORD MARK AT END OF RECORD
            SM   ADRESS,00160,7,CHANGE ADDRESS BACK TO BEGINNING OF FIELD
            TRNMINPUT-1,ADRESS,11,MOVE RECORD FROM SORT-MERGE TO PPIKI
            CF   02686,,,CLEAR FLAG FOR PURPOSES OF SAFETY
      TRANS SF   INPUT+1,,,
            SF   INPUT+11,,,SET FLAG FOR KEYWORD FIELD
            TF   OUTPUT+146,INPUT+158,,TRANSFER KEYWORD TO OUTPUT CARD
            TRNMOUTPUT+157,DASH-1,,TRANSFER DASH TO OUTPUT CARD
            BNC1*+24
            WACDOUTPUT,,,
            PRA OUTPUT,,,
            TF   WORD+146,INPUT+158,,TRANSFER KEYWORD TO WORD FROM INPUT
            TRNMOUTPUT-1,ZEROS-49,,ZERO OUT OUTPUT AREA
            TF   OUTPUT+38,INPUT+50,,TRANSFER 20 LETTER KEYWORD TO OUTPUT
            BXM *+12,10,10,ADD 10 TO ADDRESS FOR NUMBERS
            TF   (1),INPUT+10,,TRANSFER NUMBER TO OUTPUT
            AM   NUM,1,,INCREMENT NUMBER COUNT BY 1
            BSX 02836,0,,TURN OFF INDEX REGISTERS AND RETURN TO SORT-MERG
      READ  SF   02686,,,SET FLAG FOR FIELD TRANSFER
            TF   ADRESS,02690,,TRANSMIT ADDRESS OF RECORD TO WORK AREA
            AM   ADRESS,00160,7,CHANGE ADDRESS TO END OF FIELD
            TD   ADRESS,RM,6,PUT RECORD MARK AT END OF RECORD
            SM   ADRESS,00160,7,CHANGE ADDRESS BACK TO BEGINNING OF FIELD
            TRNMINPUT-1,ADRESS,11,MOVE RECORD FROM SORT-MERGE TO PPIKI
            CF   02686,,,CLEAR FLAG FOR PURPOSES OF SAFETY
            SF   INPUT+1,,,SET FLAG FOR NUMBER FIELD
            SF   INPUT+11,,,SET FLAG FOR KEYWORD FIELD
            C    INPUT+158,WORD+146,,COMPARE SECOND WORD TO FIRST WORD
            BE   TRANUM,,,IF SAME GO TO NUMBER TRANSFER
            CM   NUM,0,,IF NOT SAME FIND OUT IF OUTPUT HAS ANY NUMBERS
            BE   TRANS,,,IF NOT RETURN TO KEYWORD TRANSFER
      PUNCH BNC1*+24
            WACDOUTPUT
            PRA OUTPUT,,,
            TRNMOUTPUT-1,ZEROS-49,,ZERO OUT OUTPUT AREA
```

```
      TFM NUM,0,,INITIALIZE NUMBER COUNTER TO 0
      BLXM*+12,OUTPUT+38,10,PUT ADDRESS FOR NUMBERS IN REGISTER 1
      B   TRANS,,,RETURN TO KEYWORD TRANSFER
TRANUMBXM *+12,10,10,ADD 10 TO ADDRESS FOR NUMBERS
      TF  (1),INPUT+10,,TRANSFER NUMBER TO OUTPUT
      AM  NUM,1,,INCREMENT NUMBER COUNT BY 1
      CM  NUM,10,,COMPARE NUMBER COUNT TO 10
      BNE ST2
      BNC1*+24
      WACDOUTPUT,,,IF 10 PUNCH OUTPUT CARD
      PRA OUTPUT,,,
      TFM NUM,0,,INITIALIZE NUMBER COUNTER TO 0
      BLXM*+12,OUTPUT+38,10,PUT ADDRESS FOR NUMBERS IN REGISTER 1
      TRNMOUTPUT+39,ZERO-49,,BLANK OUT NUMBER AREA OF OUTPUT
ST2   BSX 02836,0,,TURN OFF INDEX REGISTERS AND RETURN TO SORT-MERG
FINISHCM  NUM,0,,SEE IF OUTPUT HAS ANY NUMBERS
      BE  STOP,,,IF NOT GO TO HALT
      BNC1*+24
      WACDOUTPUT,,,
      PRA OUTPUT,,,
STOP  BS  *+12,0,,TURN OFF INDEX REGISTERS
      H
      CALLEXIT
INPUT DAS 80,,,DEFINE INPUT AREA FOR CARD IMAGE
      DC  1,@,,
OUTPUTDAS 80,,,DEFINE AREA FOR OUTPUT CARD IMAGE
      DAC 1,@,,
DASH  DAC 1,-,,DEFINE A DASH FOR THE FIRST OUTPUT CARD
      DC  1,@,,
ZEROS DC  50,0,,DEFINE ZEROS TO BLANK OUT AREAS
      DC  40,0,,
      DC  30,0,,
      DC  40,0,,
      DC  1,@,,
WORD  DAS 74,,,DEFINE AREA FOR KEYWORD STORAGE
      DC  1,@
NUM   DC  5,0,,
      DC  1,@,,
ZERO  DC  50,0,,DEFINE ZEROS TO BLANK OUT OUTPUT AREA
      DC  40,0,,
      DC  30,0,,
      DC  1,@,,
ADRESSDS  5,,,
RM    DC  1,@,,
SWITCHDC  1,0,,ESTABLISH NON-DIGIT INDICATOR IN SWITCH
ST3   DAC 26,SWITCH 1 ON TO PUNCH ALSO@,,
ST4   DAC 12,PRESS START@,,
      DENDST1
```

ANN. KWD. INDEX PROGRAM

```
      D=CDS(6.)
      D=RCD(6.)
      D=PAS(6,8061030)
C     THIS CLEARS THE AREA FOR THE PRINTER
      DIMENSION A(10,170),J(10)
      LCNT=0
      DO 7 I=1,10
    7 J(I)=0
      PET=.05
    3 D=RCD(1.)
      IF(ZON(1.80))1,2,2
    1 D=PAS(2.8018005)
      D=PAS(2.8058080)
      D=PRT(5.)
      D=PAS(1.8058080)
      D=PRT(5.)
      LCNT=LCNT+2
      D=PAS(1.8038080)
      IF(LCNT-60)3,11,12
   11 LCNT=1
      D=PAS(3.8058080)
      D=PRT(5.)
      GO TO 3
   12 LCNT=LCNT-60
      GO TO 3
    2 IF(CMP(1.2032020))4,5,4
    4 PAUSE
      GO TO 3
    5 CAT=1.2550
      DO 6 I=1,10
      R=GET(CAT)
      IF(R)4,6,66
   66 K=R*1000.
      II=K/1000+1
      J(II)=J(II)+1
      K=J(II)
      A(II,K)=R
    6 CAT=CAT+PET
      D=RCD(1.)
      IF(ZON(1.80))100,2,2
  100 DO 40 I=1,10
      ID=J(I)
      IF(ID)4,40,101
  101 M=ID
  200 M=M/2
      IF(M)300,40,300
  300 K=ID-M
      J1=1
   41 I1=J1
   49 L=I1+M
      IF(A(I,I1)-A(I,L))60,60,50
   50 B=A(I,I1)
      A(I,I1)=A(I,L)
      A(I,L)=B
      I1=I1-M
      IF(I1)60,60,49
   60 J1=J1+1
      IF(J1-K)41,41,200
   40 CONTINUE
```

```
206 D=PAS(2.8048080)
    M=0
    DOG=4.07501
    DO 210 I=1,10
    K=J(I)
    IF(K)4,210,201
201 J(I)=J(I)-1
    M=1
    R=A(I,1)
    D=PUT(DOG)
    KM1=K-1
    DO 202 L=1,KM1
    LP1=L+1
202 A(I,L)=A(I,LP1)
210 DOG=DOG+.07
    IF(M)4,1,205
205 IF(LCNT-60)203,204,4
203 D=PAS(4.8058080)
    D=PRT(5.)
    LCNT=LCNT+1
    GO TO 206
204 LCNT=2
    D=PAS(3.8058080)
    D=PRT(5.)
    D=PAS(4.8058080)
    D=PRT(5.)
    GO TO 206
    END
```

PACKAN

```
*NAMEPACKAN
     START BS   *+12,1,,TURN ON INDEX REGISTERS BAND 1
           BLXM*+12,CARD1-1,10,LOAD REGISTER 1 WITH ADDRESS CARD1-1
           TF   COMKNT,ZERO,,SET COMMA COUNTER TO ZERO
           TFM COLKNT,0,,SET COLUMN COUNTER TO ZERO
           TFM ABKNT,0,,SET ABSTRACT NUMBER TO ZERO
           CF   ZEROS+1,,,CLEAR FLAGS FOR FIELD TRANSFER
           CF   ZEROS+41
           CF   ZEROS+81
           BI   *+12,00900,,TURN OFF LAST CARD INDICATOR
           TF   CARD1+158,ZEROS+120,,BLANK OUT CARD AREA
           RACDCARD1,,,READ FIRST CARD
           SF   CARD1+149,,,SET FLAG FOR ABSTRACT NUMBER
     ST1   TF   NUMAB,CARD1+158,,MOVE ABSTRACT NUMBER TO STORAGE
           AM   ABKNT,1,,INCREMENT ABSTRACT COUNTER BY 1
           TF   CARD2+8,CARD1+158,,MOVE ABSTRACT NUMBER TO OUTPUT CARD
     ST2   SF   (1),,,SET FLAG FOR CHARACTER COMPARE
           TFM CARD2+10,0,10,
           CF   CARD2+9
           TF   CARD2+158,CARD1+146,,MOVE KEYWORDS TO OUTPUT CARD
           PRA CARD2,,,PRINT OUTPUT
           BCOV*+24
           B    *+24
           SKIP,7,,
     ST25  CM   1+(1),23,,COMPARE CHARACTER TO COMMA
           BNE *+24,,,IF NOT A COMMA NOT INCREMENT COUNTER
           AM   COMKNT,1,,INCREMENT COMMA COUNTER BY 1
           AM   COLKNT,1,,INCREMENT COLUMN COUNTER BY 1
           CM   COLKNT,75,,COMPARE COLUMN COUNTER TO 75
           BE   ST3,,,IF EQUAL GO TO NEXT CARD
           BXM *+12,2,10,INCREMENT INDEX REGISTER BY 2
           SF   (1)
           B    ST25,,,COMPARE THE NEXT CHARACYER
     ST3   BLXM*+12,CARD1-1,10,INITIALIZE INDEX REGISTER 1
           TFM COLKNT,0,,SET COLUMN COUNTER TO ZERO
           BI   ST4,00900,,IF NO MORE CARDS GO TO OUTPUT
           TF   CARD1+158,ZEROS+120,,BLANK OUT CARD AREA
           RACDCARD1,,,READ NEXT CARD
           SF   CARD1+149,,,SET FLAG FOR ABSTRACT NUMBER
           C    NUMAB,CARD1+158,,COMPARE NEW ABSTRACT NUMBER WITH FORMER
           BNE ST1,,,IF UNEQUAL INCREMENT ABSTRACT COUNTER
           TF   CARD2+10,ZEROES,,MOVE BLANKS TO OUTPUT AREA
           B    ST2,,,IF EQUAL BEGIN CHARACTER COMPARISON
     ST4   RCTY
           RCTY
           CM   KNTRL,0,,IS THIS PERIODICALS OR PATENTS
           BNE ST5,,,IF NOT EQUAL GO TO PATENTS
           WATYST6
           TF   NUM1,COMKNT,,
           WNTYNUM1-9
           RCTY
           RCTY
           WATYST7
           TF   NUM2,ABKNT
           WNTYNUM2-4
           RCTY
           RCTY
           LD   00094,COMKNT,,LOAD COMMA COUNTER AS DIVIDEND
           D    00085,ABKNT,,DIVIDE COMMA COUNTER BY ABSTRACT COUNTER
```

```
       WATYST8
       TDM OUTPUT-1,7,,
       TD  OUTPUT,00088,,
       TDM OUTPUT+1,7,,
       TD  OUTPUT+2,00089,,
       WATYOUTPUT
       TR  00095,NUM1+1,,
       WNTY00090,,,WRITE DECIMAL QUOTIENT
       TFM KNTRL,1,,SET KNTRL TO DO PATENT SECTION OUTPUT
       B   START
ST5    RCTY
       WATYST9
       WNTYCOMKNT-9,,,WRITE NUMBER OF KEYWORDS
       RCTY
       RCTY
       WATYST10
       WNTYABKNT-4,,,WRITE NUMBER OF ABSTRACTS
       RCTY
       RCTY
       LD  00094,COMKNT,,LOAD NUMBER OF KEYWORDS AS DIVIDEND
       D   00085,ABKNT,,DIVIDE NUMBER OF KEYWORDS BY NUMBER OF ABSTR
       WATYST11
       TDM OUTPUT-1,7,,
       TD  OUTPUT,00088,,
       TDM OUTPUT+1,7,,
       TD  OUTPUT+2,00089,,
       WATYOUTPUT,,,WRITE FIRST PART OF QUOTIENT
       TR  00095,NUM1+1,,SET RECORD MARK AT END OF QUOTIENT
       WNTY00090,,,WRITE DECIMAL QUOTIENT
       A   NUM1,COMKNT,,ADD PERIODICAL KEYWORDS TO PATENT KEYWORDS
       A   NUM2,ABKNT,,ADD PATENT ABSTRACTS TO PERIODICAL ABSTRACTS
       RCTY
       RCTY
       RCTY
       WATYST12
       WNTYNUM1-9,,,WRITE TOTAL NUMBER OF KEYWORDS
       RCTY
       RCTY
       WATYST13
       WNTYNUM2-4,,,WRITE TOTAL NUMBER OF ABSTRACTS
       RCTY
       RCTY
       LD  00094,NUM1,,LOAD NUMBER OF KEYWORDS AS DIVIDEND
       D   00085,NUM2,,DIVIDE BY NUMBER OF ABSTRACTS
       WATYST14
       TDM OUTPUT-1,7,,
       TD  OUTPUT,00088,,
       TDM OUTPUT+1,7,,
       TD  OUTPUT+2,00089,,
       WATYOUTPUT,,,WRITE FIRST PART OF QUOTIENT
       TR  00095,NUM1+1,,MOVE RECORD MARK TO END OF QUOTIENT
       WNTY00090,,,WRITE DECIMAL PART OF QUOTIENT
       H
       TFM KNTRL,0,,RESET KNTRL FOR PERIODOCAL
       B   START
CARD1  DAS 80,,,CARD INPUT
CARD2  DAS 80,,,CARD OUTPUT
       DAC 1,@,,
NUMAB  DS  10,,,ABSTRACT NUMBER
ABKNT  DS  5,,,NUMBER OF ABSTRACTS
       DC  1,@,,
```

```
COMKNTDS  10,,,NUMBER OF COMMAS
      DC  1,*,,
ZERO  DC  10,0,,ZERO FIELD FOR INITIALIZATION
COLKNTDS  5,,,NUMBER OF COLUMNS
NUM1  DS  10
      DC  1,*,,
NUM2  DS  5
      DC  1,*,,
ZEROS DC  40,0,,BLANK FOR INITIALIZATION
      DC  40,0,,
      DC  40,0,,
      DC  40,0,,
OUTPUTDAS 2
      DAC 1,.,,
      DAC 1,*,,
ZEROESDC  12,0,,
KNTRL DC  5,0,,
ST6   DAC 33,NUMBER OF PERIODICAL KEYWORDS = *
ST7   DAC 34,NUMBER OF PERIODICAL ABSTRACTS = *
ST8   DAC 50,AVERAGE NUMBER OF PERIODICAL KEYWORDS PER ABSTRACT
      DAC 4, = *
ST9   DAC 29,NUMBER OF PATENT KEYWORDS = *
ST10  DAC 30,NUMBER OF PATENT ABSTRACTS = *
ST11  DAC 50,AVERAGE NUMBER OF PATENT KEYWORDS PER ABSTRACT = *
ST12  DAC 28,TOTAL NUMBER OF KEYWORDS = *
ST13  DAC 29,TOTAL NUMBER OF ABSTRACTS = *
ST14  DAC 43,AVERAGE NUMBER OF KEYWORDS PER ABSTRACT = *
      DENDSTART
```

PLMIFR

```
*NAMEPLMIFR
*ID NUMBER 0217
            DORG20000
      ST1   CF   ZEROS-39,,,
            CF   ZEROS+1,,,
            CF   ZEROS+41,,,
            CF   ZEROS+81,,,
            TFM KOUNTR,0,,
            SK   DISK,,,
            BI   *+12,00900,,TURN OFF LAST CARD INDICATOR
            BS   *+12,1,,TURN ON BAND 1 INDEX REGISTERS
            BLXM*+12,STORE,10,LOAD REGISTER 1 WITH ADDRESS OF STORE
            BLXM*+12,CARD2+39,9,LOAD INDEX REGISTER 2 WITH ADDRESS OF NOS
      READ1 RACDINPUT,,,READ FIRST CARD
            SF   INPUT+157,,,SET FLAG FOR COLUMN 80 COMPARE
            CM   INPUT+158,20,10,COMPARE COLUMN 80 TO -
            CF   INPUT+157,,,
            BE   CAT,,,IF EQUAL SKIP ERROR ROUTINE
            WATYERMES1,,,TYPE ERROR MESSAGE 1
            H
            B    READ1
      CAT   TRNMCARD1-1,INPUT-17,MOVE FIRST CARD TO FIRST INPUT STORAGE
            SF   CARD1-1,,,SET FLAG FOR WORD COMPARE
            RACDINPUT,,,READ SECOND CARD
            SF   INPUT+157,,,SET FLAG FOR COLUMN 80 COMPARE
            CM   INPUT+158,20,10,COMPARE COLUMN 80 TO -
            CF   INPUT+157,,,
            BNE DOG,,,IF NOT EQUAL SKIP ERROR ROUTINE
      ERR2  WATYERMES2,,,TYPE ERROR MESSAGE 2
            H
            B    READ1
      DOG   TRNMCARD2-1,INPUT-1,,MOVE SECOND CARD TO SECOND INPUT STORAGE
            SF   CARD2-1,,,SET FLAG FOR WORD COMPARE
            C    CARD2+38,CARD1+38,,COMPARE KEYWORDS
            BNE ERR2
      ELK   SF   (2),,,SET FLAG FOR ABSTRACT NUMBER
            C    9+(2),BLANK,,COMPARE ABSTRACT NUMBER TO BLANK
            BE   FOX,,,IF BLANK GO TO NEXT COLUMN
            TF   OUTPUT+10,9+(2),,MOVE ABSTRACT NUMBER TO OUTPUT
            SF   CARD1-1,,,SET FLAG FOR KEYWORD TRANSFER
            TF   OUTPUT+158,CARD1+146,,MOVE KEYWORD TO OUTPUT
            CF   OUTPUT+1
            CF   OUTPUT+11
            TRNM(1),OUTPUT-1,,MOVE OUTPUT RECORD TO STORAGE
            AM   KOUNTR,1,,
            BNC1BYRNE
            RCTY
            RCTY
            WATYST2
            WNTYKOUNTR-4
            RCTY
      BYRNE BXM *+12,00160,10,INCREMENT INDEX REGISTER 1 BY 160
            TRNMOUTPUT-1,ZEROS-39,,BLANK OUTPUT AREA
            AM   NUMREC,1,10,INCREMENT RECORD COUNTER BY 1
            CM   NUMREC,25,10,COMPARE RECORD COUNTER TO 25
            BNE FOX,,,IF STORAGE NOT FULL SKIP TRANSMIT ROUTINE
      TRANS TD   GM,EOF,,
            WDGNDISK
            BLXM*+12,STORE,10,INITIALIZE INDEX REGISTER 1
```

```
        TFM NUMREC,0,10,ZERO OUT RECORD COUNTER
        AM  DISK+5,40,10,INCREMENT SECTOR COUNTER BY 5
        AM  TRAK,1,10,INCREMENT TRAK COUNTER BY 1
        CM  TRAK,5,10,COMPARE TRAK COUNTER TO FULL
        BNE FOX
SEEK    SK  DISK,,,
        TDM TRAK,0,,INITIALIZE TRAK COUNTER
FOX     BXM *+12,10,9,INCREMENT REGISTER 2 BY 10
        AM  NUMNUM,1,10,INCREMENT NUMBER COUNTER BY 1
        CM  NUMNUM,10,10,COMPARE NUMBER COUNTER TO 10
        BNE ELK,,,IF NOT EQUAL GO TO NEXT ABSTRACT NUMBER
GOAT    BI  FINISH,00900,,IF NOT MORE CARDS GO TO FINISH
        BLXM*+12,CARD2+39,9,INITIALIZE NUMBER ADDRESS IN REGISTER 2
        TFM NUMNUM,0,10,INITIALIZLE NUMBER COUNTER
        RACDINPUT,,,READ ANOTHER CARD
        SF  INPUT+157,,,SET FLAG FOR COLUMN 80 COMPARE
        CM  INPUT+158,20,10,COMPARE COLUMN 80 TO -
        CF  INPUT+157
        BE  CAT,,,IF EQUAL MOVE CARD TO FIRST INPUT STORAGE
        B   DOG,,,IF NOT EQUAL MOVE CARD TO SECOND INPUT STORAGE
FINISHSF  TRAK+1
        TF  2+(1),TRAK+3,,TRANSFER RECORD MARKS TO OUTPUT AREA
        WDGNDISK,,,WRITE OUTPUT AND RECORD MARKS ON DISK
        BS  *+12,0,,TURN OFF INDEX REGISTERS FOR JERRY BYRNE
        RCTY
        RCTY
        WATYST2
        WNTYKOUNTR-4
        RCTY
        CALLEXIT
INPUT   DAS 80,,,
        DAC 1,@,,
ERMES1DAC 50,FIRST CARD HAS NO - IN COLUMN 80   REARRANGE AND P
        DAC 10,RESS START
        DAC 1,@,,
CARD1   DAS 80,,,
        DAC 1,@,,
ERMES2DAC 50,CARDS OUT OF ORDER   REARRANGE STARTING WITH LAST
        DAC 27,HEADER CARD AND PRESS START,,
        DAC 1,@,,
CARD2   DAS 80,,,
        DAC 1,@,,
OUTPUTDAS 80,,,
        DAC 1,@,,
BLANK   DC  10,0,,
ZEROS   DC  40,0,,
        DC  40,0,,
        DC  40,0,,
        DC  40,0,,
        DC  1,@,,
KOUNTRDS  5,,,
        DC  1,@,,
NUMRECDC  2,0,,
NUMNUMDC  2,0,,
        DAC 1,@,,
STORE DSS 4000,,,
GM      DGM
SEC     DS  ,20000
SECT    DS  ,040
DISK    DDA ,3,SEC,SECT,STORE
TRAK    DC  2,0,,
```

CALANK

```
*NAMECALANK
     ST1    BS   *+12,1,,TURN ON INDEX REGISTERS BAND 1
            CF   BLANK+1
            CF   BLANK+41
            CF   BLANK+81
            TF   INPUT+158,BLANK+120,,
            TF   OUTPUT+158,BLANK+120,,
            TFM NUMKNT,0,,INITIALIZE NUMBER COUNTER
            SPTY
            WATYST6
            RACDOUTPUT
            PRA OUTPUT
            TF   OUTPUT+158,BLANK+120,,
            PRA OUTPUT
            PRA OUTPUT
            PRA OUTPUT
     ST2    RACDINPUT,,,READ FIRST INPUT CARD
            SF   INPUT+157,,,SET FLAG FOR - COMPARE
            CM   INPUT+158,20,10,CHECK COLUMN 80 FOR -
            CF   INPUT+157
            BNE ST25,,,IF NO MINUS PUNCH COUNT NUMBERS
     ST23   SF   INPUT-1,,,SET FLAG FOR KEYWORD TRANSFER
            TF   OUTPUT+158,INPUT+124,,TRANSFER KEYWORD TO OUTPUT
            B    ST2,,,GO TO NEXT INPUT CARD
     ST25   BLXM*+12,INPUT+39,10,LOAD INDEX REGISTER 1 WITH ADDRESS
            BLXM*+12,INPUT+48,9,LOAD INDEX REGISTER 2 WITH ADDRESS
            TFM COLKNT,0,,SET COLUMN COUNTER TO 0
     ST3    SF   (1),,,SET FLAG FOR NUMBER COMPARE
            C    (2),ZEROS,,COMPARE FIELD TO BLANK
            CF   (1)
            BE   ST35,,,IF BLANK DO NOT INCREMENT NUMBER COUNTER
            AM   NUMKNT,1,,INCREMENT NUMBER COUNTER BY 1
     ST35   AM   COLKNT,1,,INCREMENT COLUMN COUNTER BY 1
            CM   COLKNT,10,,COMPARE COLUMN COUNTER TO 10
            BE   ST4,,,IF EQUAL GO TO NEXT CARD
            BXM *+12,10,10,IF NOT EQUAL INCREMENT ADDRESS BY 10
            BXM *+12,10,9,INCREMENT ADDRESS FOR FIELD COMPARE BY 10
            B    ST3,,,COMPARE NEXT FIELD
     ST4    RACDINPUT,,,READ NEXT CARD
            SF   INPUT+157,,,SET FLAG FOR - COMPARE
            CM   INPUT+158,20,10,COMPARE COLUMN 80 TO -
            CF   INPUT+157,,,
            BNE ST25,,,IF NO - COUNT NUMBERS
            TNF OUTPUT+28,NUMKNT,,TRANSFER COUNTER TO ALPHAMERIC OUTPUT
            BD   ST5,OUTPUT+20,,
            TDM OUTPUT+19,0,,
            BD   ST5,OUTPUT+22,,
            TDM OUTPUT+21,0,,
            BD   ST5,OUTPUT+24,,
            TDM OUTPUT+23,0,,
            BD   ST5,OUTPUT+26,,
            TDM OUTPUT+25,0,,
     ST5    TFM OUTPUT+30,0,10,
            BNC1*+24
            WACDOUTPUT
            PRA OUTPUT
            BCOV*+24,,,
            B    *+24
            SKIP,7,,
```

```
        SF   BLANK-39
        TF   OUTPUT+158,BLANK+120,,
        TFM NUMKNT,0,,INITIALIZE NUMBER COUNTER
        B    ST23
ZEROS DC   10,0,,
NUMKNTDS   5,,,NUMBER COUNTER
COLKNTDS   5,,,NUMBER FIELD COUNTER
INPUT DAS 80,
OUTPUTDAS 80,
        DAC 1,@,,
BLANK DC   40,0,,
        DC   40,0,,
        DC   40,0,,
        DC   40,0,,
ST6     DAC 40,TURN SWITCH 1 ON FOR PUNCHED OUTPUT ALSO,,
        DAC 1,@,,
        DENDST1
```

MLOKAC

```
*NAMEMLOKAC
     ST1    BS  *+12,1,,TURN ON INDEX REGITERS BAND 1
            BI  *+12,00900,,TURN OFF LAST CARD INDICATOR
            SPTY
            WATYST10
            RACDINPUTC,,,READ HEADER CARD
            PRA INPUTC,,,READ HEADER CARD
            SPIM,3,,
            SK  DISK
            CF  BLANK+1
            CF  BLANK+41
            CF  BLANK+81
     ST12   RDN DISK,,,BRING RECORDS FROM DISK
            AM  DISK+5,40,10,INCREMENT SECTOR COUNTER
            AM  TRAK,1,10,
            CM  TRAK,5,10,
            BNE *+36,,,IF NOT EQUAL READ FIRST RECORD
            SK  DISK
            TDM TRAK,0,,INITIALIZE TRAK COUNTER
            BLXM*+12,INPUT,10,PUT ADDRESS OF INPUT IN REGISTER 1
            TFM NUMREC,0,10,INITIALIZE RECORD COUN
     ST17   CM  NUMREC,25,10,CHECK IF ALL RECORDS USED
            BE  ST12,,,IF SO BRING IN NEW SET OF RECORDS FROM DISK
            SF  (1),,,SET FLAG FOR RECORDS TRANSFER
            TF  INPUTD+158,BLANK+120,,BLANK OUT INPUT AREA
            TF  INPUTD+158,159+(1),,TRANSFER RECORDSFOR COMPARISON
            BXM *+12,160,10,INCREMENT INDEX REGISTER BY 160
            AM  NUMREC,1,10,INCREMENT RECORD COUNTER BY 1
            SF  INPUTD+31,,,SET FLAG FOR KEYWORD COMPARE
     ST19   NOP ST3,,,CAN BE CHANGED TO BRANCH TO SKIP CARD READ
     ST2    TF  INPUTC+158,BLANK+120,,BLANK OUT INUT AREA
            RACDINPUTC,,,READ CARD RECORD
            SF  INPUTC+31,,,SET FLAG FOR KEYWORD COMPARE
     ST3    BNR *+24,INPUTD+1,,CHECK TO SEE IF DISK RECORD IS LAST ONE
            B   ST7,,,IF SO GO TO END ROUTINE
            C   INPUTC+158,INPUTD+158,,COMPARE KEYWORDS
            BE  ST4
            BP  ST6
            BNC1*+24,,,IF NEGATIVE CHECK SWITCH U FOR PUNCHING
            WACDINPUTC,,,IF SWITCH 1 ON PUNCH CARD
            PRA INPUTC,,,PRINT CARDP
            BCOV*+24
            B   *+24
            SKIP,7,,
            BI  ST8,00900,,
            B   ST2
     ST4    SF  INPUTC+19,,,SET FLAG FOR NUMBER TRANSFER
            SF  NUMC-4
            TNS INPUTC+28,NUMC,,MOVE NUMBER TO NUMBERIC FIELD FOR ADD
            SF  INPUTD+19,,,SET FLAG FOR NUMBER TRANSFER
            SF  NUMD-4
            TNS INPUTD+28,NUMD,,MOVE NUMBER TO NUMERIC FIELD FOR ADDITION
            A   NUMC,NUMD,,ADD TWO NUMBERS
            SF  NUMC-4
            TNF INPUTC+28,NUMC,,RETURN NUMBER TOTAL TO OUTPUT ALPHAMERIC
            BD  ST5,INPUTC+20,,
            TDM INPUTC+19,0,,
            BD  ST5,INPUTC+22,,
            TDM INPUTC+21,0,,
```

```
       BD  ST5,INPUTC+24,,
       TDM INPUTC+23,0,,
       BD  ST5,INPUTC+26,,
       TDM INPUTC+25,0,,
ST5    TFM INPUTC+30,0,10,PUT BALNK IN OUPUT AREA
       BNC1*+24
       WACDINPUTC
       PRA INPUTC
       BCOV*+24
       B   *+24
       SKIP,7,,
       BI  ST81,00900,,IF NO MORE CARDS GO TO FINISH ROUTINE
       TDM ST19+1,1,,CHANGE ST19TO READ NEW CARD
       B   ST17
ST6    BNC1*+24
       WACDINPUTD
       PRA INPUTD
       BCOV*+24
       B   *+24
       SKIP,7,,
       TDM ST19+1,9,,CHANGE ST19 TO BRANCH AROUND CARD READ
       B   ST17
ST7    BNC1*+24
       WACDINPUTC
       PRA INPUTC
       BCOV*+24
       B   *+24
       SKIP,7,,
       BI  ST9,00900,,
       TF  INPUTC+158,BLANK+120,,BLANK OUT INUT AREA
       RACDINPUTC,,,READ CARD RECORD
       B   ST7
ST8    BNC1*+24
       WACDINPUTD
       PRA INPUTD
       BCOV*+24
       B   *+24
       SKIP,7,,
       B   ST81
ST80   RDN DISK,,,BRING RECORDS FROM DISK
       AM  DISK+5,40,10,
       AM  TRAK,1,10,
       CM  TRAK,5,10,
       BNE *+36
       SK  DISK
       TDM TRAK,0,,
       BLXM*+12,INPUT,10,
       TFM NUMREC,0,10,
ST81   CM  NUMREC,25,10,
       BE  ST80
       SF  (1)
       TF  INPUTD+158,BLANK+120,,BLANK OUT INPUT AREA
       TF  INPUTD+158,159+(1),,
       BXM *+12,160,10,
       AM  NUMREC,1,10,
       BNR ST8,INPUTD+1,,
       B   ST9
ST9    CALLEXIT
INPUTCDAS 80,,,INPUT AREA FOR CARDS
       DAC 1,@,,
INPUTDDAS 80,,,INPUT AREA FOR DISK RECORDS
```

```
       DAC 1,*,,
NUMC   DS  5,,,NUMBER FROM CARD INPUT
NUMD   DS  5,,,NUMBER FROM DISK INPUT
ST10   DAC 40;TURN SWITCH 1 ON FOR PUNCHED OUTPUT ALSO,,
       DAC 1,*,,
NUMRECDC   2,0,,COINTER FOR NUMBER OF DISK RECORDS READ
SEC    DS  ,20000
SECT   DS  ,040
DISK   DDA ,3,SEC,SECT,INPUT
TRAK   DC  2,0,,
INPUT  DSS 4000
BLANK  DC  40,0,,DEFINE ZEROSFOR ALPHAMERIC BLANK
       DC  40,0,,DEFINE ZEROSFOR ALPHAMERIC BLANK
       DC  40,0,,DEFINE ZEROSFOR ALPHAMERIC BLANK
       DC  40,0,,DEFINE ZEROSFOR ALPHAMERIC BLANK
       DENDST1
```

```
C     KEYWORD FREQUENCY DISTRIBUTION
      DIMENSION L(3000)
      DO 2 I=1,3000
      L(I)=0
    2 CONTINUE
   25 READ 3,N
    3 FORMAT (11X,I4)
      L(N)=L(N)+1
      IF(SENSE SWITCH 9)4,25
    4 PRINT 41
   41 FORMAT(1H ,5X,9HFREQUENCY,3X,9HFREQUENCY)
      PRINT 42
   42 FORMAT(1H ,6X,8HOF USAGE,3X,12HDISTRIBUTION)
      PRINT 43
   43 FORMAT(1H ,//)
      DO 5 I=1,3000
      IF(L(I))5,5,44
   44 PRINT 45,I,L(I)
   45 FORMAT(10X,I4,3X,I4)
    5 CONTINUE
      END
```

DO-ALL

```
*NAMEDO-ALL
    START CF  BLANK+1
          CF  BLANK+51
          CF  BLANK+101
          CF  INPUT+159
          BS  *+12,1,,
          BLXM*+12,OUTPUT,10,
          BLXM*+12,LIST,9,
          BLC *+12
          RCTY
          WATYMESS8
          RCTY
          WATYMESS1
          RCTY
          H
          SKIP,7,,
          PRA HD100
          SPIM,3,,
    ST1   BLC ST3
          TF  INPUT+158,BLANK+110,,
          RACDINPUT
          SF  INPUT-1
          TF  159+(1),INPUT+158,,
          BXM *+12,160,10,
          AM  OUTCNT,1,,
          CM  OUTCNT,125,,
          BNE ST2
          SK  DSKOUT
          WDGNDSKOUT
          TFM OUTCNT,0,,
          AM  DSKOUT+5,200,,
          BLXM*+12,OUTPUT,10,
    ST2   SF  NUM-3
          TNS INPUT+28,NUM,,
          CF  NUM-2
          CF  NUM-1
          CF  NUM
          CM  NUM,100,8,
          BN  ST1
          PRA INPUT
          SPIM,1,,
          BCOV*+24
          B   *+24
          SKIP,7,,
          BNC2*+24
          WACDINPUT
          BNC1ST1
          TR  (2),INPUT-1,,
          TR  162+(2),INPUT+161,,
          BXM *+12,200,9,
          AM  LSTCNT,1,,
          CM  LSTCNT,50,,
          BNE ST1
          SK  DSKLST
          WDGNDSKLST
          TFM LSTCNT,0,,
          AM  DSKLST+5,100,,
          BLXMST1,LIST,9,
    ST3   TD  (1),RM,,
```

```
        SK   DSKOUT
        WDGNDSKOUT
        BNC1ST9
        TD   (2),RM,,
        SK   DSKLST
        WDGNDSKLST
        BLXM*+12,ST9,910,
ST5     SKIP,7,,
        PRA HD100
        SPIM,3,,
        TFM DSKLST+5,20000,,
ST6     TFM LSTCNT,0,,
        SK   DSKLST
        RDGNDSKLST
        BLXM*+12,LIST,9,
ST7     BNR *+24,(2),,
        B    ST8
        PRA 1+(2)
        SPIM,1,,
        BCOV*+24
        B    *+24
        SKIP,7,,
        BNC2*+24
        WACD1+(2)
        BXM *+12,200,9,
        AM   LSTCNT,1,,
        CM   LSTCNT,50,,
        BNE ST7
        AM   DSKLST+5,100,,
        B    ST6
ST8     WATYMESS2
        RCTY
        H
        BC1 ST5
        B    (3)
ST9     WATYMESS3
        RCTY
        H
        SKIP,7,,
        PRA HD1
        SPIM,3,,
        TFM OUTCNT,0,,
        TFM LSTCNT,0,,
        TFM DSKOUT+5,30000,,
        TFM DSKLST+5,20000,,
        BLXM*+12,OUTPUT,10,
        BLXM*+12,LIST,9,
        SK   DSKOUT
        RDGNDSKOUT
ST11    BNR *+24,(1),,
        B    ST14
        SF   (1)
        TF   INPUT+158,159+(1),,
        SF   NUM-3
        TNS INPUT+28,NUM,,
        CF   NUM-2
        CF   NUM-1
        CF   NUM
        CM   NUM,1,8,
        BNE ST12
        PRA INPUT
```

```
        SPIM,1,,
        BCOV*+24
        B    *+24
        SKIP,7,,
        BNC2*+24
        WACDINPUT
        BNC1ST12
        TR   (2),INPUT-1,,
        TR   162+(2),INPUT+161,,
        BXM *+12,200,9,
        AM   LSTCNT,1,,
        CM   LSTCNT,50,,
        BNE ST12
        SK   DSKLST
        WDGNDSKLST
        TFM LSTCNT,0,,
        AM   DSKLST+5,100,,
        BLXM*+12,LIST,9,
ST12    AM   OUTCNT,1,,
        CM   OUTCNT,125,,
        BNE ST13
        AM   DSKOUT+5,200,,
        SK   DSKOUT
        RDGNDSKOUT
        TFM OUTCNT,0,,
        BLXMST11,OUTPUT,10,
ST13    BXM ST11,160,10,
ST14    BNC1ST15
        TD   (2),RM,,
        SK   DSKLST
        WDGNDSKLST
        BNC1ST15
        TFM ST5+18,HD1,,
        BLXMST5,ST15,910,
ST15    WATYMESS4
        RCTY
        H
        SKIP,7,,
        PRA HDPARP
        SPIM,3,,
        TFM OUTCNT,0,,
        TFM LSTCNT,0,,
        TFM DSKOUT+5,30000,,
        TFM DSKLST+5,20000,,
        BLXM*+12,OUTPUT,10,
        BLXM*+12,LIST,9,
        SK   DSKOUT
        RDGNDSKOUT
ST16    BNR *+24,(1),,PARENTHESIS SECTION
        B    ST21
        SF   (1)
        TF   INPUT+158,159+(1),,
        BLXM*+12,INPUT+33,8,
        TFM COLCNT,0,,
ST17    SF   (4)
        CM   1+(4),24,10,
        CF   (4)
        BE   ST18
        AM   COLCNT,1,,
        CM   COLCNT,63,,
        BE   ST19
```

```
      BXM ST17,2,8,
ST18  PRA INPUT
      SPIM,1,,
      BCOV*+24
      B   *+24
      SKIP,7,,
      BNC2*+24
      WACDINPUT
      BNC1ST19
      TR  (2),INPUT-1,,
      TR  162+(2),INPUT+161,,
      BXM *+12,200,9,
      AM  LSTCNT,1,,
      CM  LSTCNT,50,,
      BNE ST19
      SK  DSKLST
      WDGNDSKLST
      TFM LSTCNT,0,,
      AM  DSKLST+5,100,
      BLXM*+12,LIST,9,
ST19  AM  OUTCNT,1,,
      CM  OUTCNT,125,,
      BE  ST20
      BXM ST16,160,10,
ST20  AM  DSKOUT+5,200,,
      SK  DSKOUT
      RDGNDSKOUT
      TFM OUTCNT,0,,
      BLXMST16,OUTPUT,10,
ST21  BNC1ST22
      TD  (2),RM,,
      SK  DSKLST
      WDGNDSKLST
      TFM ST5+18,HDPARP
      BLXMST5,ST22,910,
ST22  WATYMESS5,,,HYPHENATED SECTION
      RCTY
      H
      SKIP,7,,
      PRA HDHYPH
      SPIM,3,,
      TFM OUTCNT,0,,
      TFM LSTCNT,0,,
      TFM DSKOUT+5,30000,,
      TFM DSKLST+5,20000,,
      BLXM*+12,OUTPUT,10,
      BLXM*+12,LIST,9,
      SK  DSKOUT
      RDGNDSKOUT
ST23  BNR *+24,(1),,
      B   ST28
      SF  (1)
      TF  INPUT+158,159+(1),,
      BLXM*+12,INPUT+33,8,,
      TFM COLCNT,0,,
ST24  SF  (4)-
      CM  1+(4),20,10,
      CF  (4)
      BE  ST25
      AM  COLCNT,1,,
      CM  COLCNT,63,,
```

```
      BE   ST26
      BXM ST24,2,8,
ST25  PRA INPUT
      SPIM,1,,
      BCOV*+24
      B    *+24
      SKIP,7,,
      BNC2*+24
      WACDINPUT
      BNC1ST26
      TR   (2),INPUT-1,,
      TR   162+(2),INPUT+161,,
      BXM *+12,200,9,
      AM   LSTCNT,1,,
      CM   LSTCNT,50,,
      BNE ST26
      SK   DSKLST
      WDGNDSKLST
      TFM LSTCNT,0,,
      AM   DSKLST+5,100,,
      BLXM*+12,LIST,9,
ST26  AM   OUTCNT,1,,
      CM   OUTCNT,125,,
      BE   ST27
      BXM ST23,160,10,
ST27  AM   DSKOUT+5,200,,
      SK   DSKOUT
      RDGNDSKOUT
      TFM OUTCNT,0,,
      BLXMST23,OUTPUT,10,
ST28  BNC1ST29
      TD   (2),RM,,
      SK   DSKLST
      WDGNDSKLST
      TFM ST5+18,HDHYPH,,
      BLXMST5,ST29,910,
ST29  WATYMESS6,,,COMPUND TERMS SECTION
      RCTY
      H
      SKIP,7,,
      PRA HDCOMP
      SPIM,3,,
      TFM OUTCNT,0,,
      TFM LSTCNT,0,,
      TFM DSKOUT+5,30000,,
      TFM DSKLST+5,20000,,
      BLXM*+12,OUTPUT,10,
      BLXM*+12,LIST,9,
      SK   DSKOUT
      RDGNDSKOUT
ST30  BNR *+24,(1),,
      B    ST36
      SF   (1)
      TF   INPUT+158,159+(1),,
      BLXM*+12,INPUT+33,8,
      TFM COLCNT,0,,
ST31  SF   (4)
      CM   1+(4),20,10,
      CF   (4)
      BE   ST33
      SF   (4)
```

```
        CM   1+(4),0,10,
        CF   (4)
        BNE  ST32
        SF   2+(4)
        CM   3+(4),0,10,
        CF   2+(4)
        BNE  ST33
        B    ST34
ST32    AM   COLCNT,1,,
        CM   COLCNT,63,,
        BE   ST34
        BXM  ST31,2,8,
ST33    PRA  INPUT
        SPIM,1,,
        BCOV*+24
        B    *+24
        SKIP,7,,
        BNC2*+24
        WACDINPUT
        BNC1ST34
        TR   (2),INPUT-1,,
        TR   162+(2),INPUT+161,,
        BXM  *+12,200,9,
        AM   LSTCNT,1,,
        CM   LSTCNT,50,,
        BNE  ST34
        SK   DSKLST
        WDGNDSKLST
        TFM  LSTCNT,0,,
        AM   DSKLST+5,100,,
        BLXM*+12,LIST,9,
ST34    AM   OUTCNT,1,,
        CM   OUTCNT,125,,
        BE   ST35
        BXM  ST30,160,10,
ST35    AM   DSKOUT+5,200,,
        SK   DSKOUT
        RDGNDSKOUT
        TFM  OUTCNT,0,,
        BLXMST30,OUTPUT,10,
ST36    BNC1ST37
        TD   (2),RM,,
        SK   DSKLST
        WDGNDSKLST
        TFM  ST5+18,HDCOMP,,
        BLXMST5,ST37,910,
ST37    SKIP,7,,
        PRA  HDALL1
        PRA  HDALL2
        SPIM,1,,
        PRA  HDALL3
        SPIM,3,,
        TFM  DSKOUT+5,30000,,
ST38    SK   DSKOUT
        RDGNDSKOUT
        TFM  OUTCNT,0,,
        BLXM*+12,OUTPUT,10,
ST39    BNR  *+24,(1),,
        B    ST41
        TD   159+(1),RM,,
        PRA  1+(1)
```

```
      SPIM,1,,
      BCOV*+24
      B    *+24
      SKIP,7,,
      BNC2*+24
      WACD1+(1)
      AM   OUTCNT,1,,
      CM   OUTCNT,125,,
      BE   ST40
      BXM  ST39,160,10,
ST40  AM   DSKOUT+5,200,,
      B    ST38
ST41  WATYMESS7
      RCTY
      H
      BC1 ST37
      CALL EXIT
BLANK DC   50,0,,
      DC   50,0,,
      DC   50,0,,
      DC   12,0,,
INPUT DAS  80
      DAC  1,@,,
      DC   37,0,,
      DC   1,@,,
RM    DAC  1,@,,
OUTPUTDSS  20000
      DGM
      DAC  1,@,,
LIST  DSS  10000
      DGM
OUTCNTDC   5,0,,
LSTCNTDC   5,0,,
DSKOUTDDA  ,3,30000,200,OUTPUT
DSKLSTDDA  ,3,20000,100,LIST
NUM   DS   4
MESS1 DAC  48,SWITCH 1 ON FOR COPIES OF LIST 100, PRESS START@,,
MESS2 DAC  50,SWITCH 1 ON FOR MORE COPIES, OFF FOR NEW LIST, PRE,,
      DAC  9,SS START@,,
MESS3 DAC  46,SWITCH 1 ON FOR COPIES OF LIST 1, PRESS START@,,
HD100 DAC  47,       LIST OF KEYWORDS USED 100 OR MORE TIMES@,,
HD1   DAC  40,             LIST OF KEYWORDS USED ONCE@,,
COLCNTDC   5,0,,
MESS4 DAC  47,SWITCH 1 ON FOR COPIES OF () LIST, PRESS START@,,
MESS5 DAC  46,SWITCH 1 ON FOR COPIES OF - LIST, PRESS START@,,
MESS6 DAC  49,SWITCH 1 ON FOR COPIES OF CMPD LIST, PRESS START@,,
MESS7 DAC  41,SWITCH 1 ON FOR MORE COPIES, PRESS START@,,
HDPARPDAC  44,             LIST OF PARENTHETICAL KEYWORDS@,,
HDHYPHDAC  41,             LIST OF HYPHENATED KEYWORDS@,,
HDCOMPDAC  39,             LIST OF COMPOUND KEYWORDS@,,
HDALL1DAC  44,                 KEYWORD AND FREQUENCY LIST@,,
HDALL2DAC  39,                     VOLUMES 37 AND 38@,,
HDALL3DAC  37,                       6022 KEYWORDS@,,
MESS8 DAC  21,SWITCH 2 ON TO PUNCH@,,
      DENDSTART
```

ING

```
*NAMEING
      *PROGRAM TO PRODUCE LIST OF WORDS ENDING IN ING
      START SF   ING-1
            CF   ZERO-49
            CF   ZERO+1
            CF   ZERO+51
            CF   ZERO+101
            BS   *+12,1,,
            WATYMESS1
            RCTY
            H
            BNC1*+24
            BLXM*+12,OUTPUT+200,9,
            SKIP,7,,
            RACDINPUT
            PRA  INPUT
            BNC1*+48
            TR   OUTPUT,INPUT-1,,
            TR   OUTPUT+162,INPUT+161,,
            TFM  CNTR,1,,
            SPIM,3,,
            BLC  *+12
      ST1   BLC  ST6
            TRNMINPUT-1,ZERO-49,,
            RACDINPUT
            BLXM*+12,INPUT+157,10,
      ST2   SF   (1)
            CM   1+(1),0,10,
            CF   (1)
            BNE  ST3
            BXM  ST2,-2,10,
      ST3   SF   (1)
            CM   1+(1),4,10,
            CF   (1)
            BNE  ST5
      ST4   BXM  *+12,-2,10,
            SF   (1)
            CM   1+(1),24,10,
            CF   (1)
            BNE  ST4
            BXM  *+12,-4,10,
      ST5   BXM  *+12,-4,10,
            SF   (1)
            C    5+(1),ING+4,,
            CF   (1)
            BNE  ST1
            PRA  INPUT
            SPIM,1,,
            BCOV*+24
            B    *+24
            SKIP,7,,
            BNC1ST1
            TR   (2),INPUT-1,,
            TR   162+(2),INPUT+161,,
            BXM  *+12,200,9,
            AM   CNTR,1,,
            CM   CNTR,100,,
            BNE  ST1
            SK   DISK
```

```
       WDGNDISK
       AM   DISK+5,200,
       TFM CNTR,0,,
       BLXMST1,OUTPUT,9,
ST6    TD   (2),RM,
       SK   DISK
       WDGNDISK
ST65   TFM CNTR,0,,
       SKIP,7,,
       SK   DISKR
       RDGNDISKR
       BLXM*+12,OUTPUT,9,
       PRA  1+(2)
       SPIM,3,,
       AM   CNTR,1,,
ST7    BXM  *+12,200,9,
ST8    BNR  *+24,(2)
       B    ST9
       PRA  1+(2)
       SPIM,1,,
       BCOV*+24
       B    *+24
       SKIP,7,,
       AM   CNTR,1,,
       CM   CNTR,100,,
       BNE  ST7
       AM   DISKR+5,200,,
       SK   DISKR
       RDGNDISKR
       TFM CNTR,0,,
       BLXMST8,OUTPUT,9,
ST9    WATYMESS2
       RCTY
       H
       TFM  DISKR+5,20000,,
       B    ST65
INPUT  DAS  80
       DAC  1,@,,
       DC   37,0,,
       DC   1,@,,
       DGM
ING    DAC  3,ING,,
ZERO   DC   50,0,,
       DC   50,0,,
       DC   50,0,,
       DC   10,0,,
       DC   1,@,,
DISK   DDA  ,3,20000,200,OUTPUT
CNTR   DC   5,0,,
MESS1  DAC  50,TURN SWITCH 1 ON FOR MULTIPLE COPIES, PRESS START@,,
DISKR  DDA  ,3,20000,200,OUTPUT
RM     DAC  1,@,,
OUTPUTDSS  20000
       DGM
MESS2  DAC  29,PRESS START FOR ANOTHER COPY@,,
       DENDSTART
```

EORATE

```
*NAMEEORATE
      START SKIP,7,,
            BS   *+12,1,,
            PRA HEAD1
            SPIM,3,,
            BLC *+12
      ST0   TDM OUTPUT,0,,
            AM   CNT,1,,
            CM   CNT,20000,,
            BE   *+36
            AM   ST0+6,1,,
            B    ST0
            TFM CNT,0,,
      ST1   BLXM*+12,OUTPUT,9,
            TFM CNT,0,,
      READ  BLC ST6
            RACDINPUT
            BLXM*+12,INPUT+157,10,
      ST2   SF   (1)
            CM   1+(1),0,10,
            CF   (1)
            BNE ST3
            BXM ST2,2,1011,
      ST3   SF   (1)
            CM   1+(1),4,10,
            CF   (1)
            BNE ST5
      ST4.  BXM *+12,2,1011,
            SF   (1)
            CM   1+(1),24,10,
            CF   (1)
            BNE ST4
            BXM *+12,4,1011,
      ST5   BXM *+12,2,1011,
            SF   (1)
            C    3+(1),ER+2,,
            CF   (1)
            BE   ST53
            SF   (1)
            C    3+(1),OR+2,,
            CF   (1)
            BE   WRITE1
            BXM *+12,2,1011,
            SF   (1)
            C    5+(1),ERS+4,,
            CF   (1)
            BE   ST55
            SF   (1)
            C    5+(1),ORS+4,,
            CF   (1)
            BE   WRITE1
            SF   (1)
            C    5+(1),ATE+4,,
            CF   (1)
            BE   WRITE2
            BXM *+12,2,1011,
            SF   (1)
            C    7+(1),ATES+6,,
            CF   (1)
```

```
        BE   WRITE2
        B    READ
ST53    BXM  *+12,6,1011,
        SF   (1)
        C    9+(1),PAPER+8,,
        CF   (1)
        BNE  WRITE1
        B    READ
ST55    BXM  *+12,6,1011,
        SF   (1)
        C    11+(1),PAPERS+10,,
        CF   (1)
        BNE  WRITE1
        B    READ
WRITE1PRA INPUT
        SPIM,1,,
        BCOV*+24
        B    *+24
        SKIP,7,,
        WACDINPUT
        B    READ
WRITE2TR   INPUT+158,RM,,
        TR   (2),INPUT-1,,
        AM   CNT,1,,
        CM   CNT,125,,
        BE   STORE
        BXM  READ,160,9,
STORE   SK   DISK
        WDGNDISK
        AM   DISK+5,200,,
        B    ST1
ST6     TR   (2),RM,,
        SK   DISK
        WDGNDISK
ST65    SKIP,7,,
        PRA  HEAD2
        SPIM,3,,
        TFM  DISK+5,20000,,
ST9     SK   DISK
        RDGNDISK
        AM   DISK+5,200,,
        TFM  CNT,0,,
        BLXM*+12,OUTPUT,9,
ST7     BNR  *+24,(2),,
        B    ST10
        PRA  1+(2)
        SPIM,1,,
        BCOV*+24
        B    *+24
        SKIP,7,,
ST8     WACD1+(2)
        AM   CNT,1,,
        CM   CNT,125,,
        BE   ST9
        BXM  ST7,160,9,
ST10    TDM  ST8,4,,
        TDM  ST8+1,1,,
        AM   CNTR,1,,
        CM   CNTR,5,,
        BNE  ST65
        CALLEXIT
```

```
HEAD1 DAC 33,                   THE ER/OR COMPLEX*,,
HEAD2 DAC 36,                   THE ATE/ATES PERPLEX*,,
RM    DAC 1,*,,
OUTPUTDSS 20000
      DGM
CNT   DC  5,0,,
INPUT DAS 80
      DAC 1,*,,
ER    DAC 2,ER,,
OR    DAC 2,OR,,
ERS   DAC 3,ERS,,
ORS   DAC 3,ORS,,
ATE   DAC 3,ATE,,
ATES  DAC 4,ATES,,
PAPER DAC 5,PAPER,,
PAPERSDAC 6,PAPERS,,
DISK  DDA -,3,20000,200,OUTPUT,
CNTR  DC  5,0,,
      DENDSTART
```

PICTAL

```
*NAMEPICTAL
     START BS   *+12,1,,
           BLC *+12
           SK   DISK
           CF   CNTRL-49
           CF   CNTRL+1
           CF   CNTRL+51
           CF   CNTRL+101
           CF   CNTRL+151
           CF   CNTRL+201
           CF   CNTRL+251
           CF   CNTRL+301
           CF   BLANK-49
           CF   BLANK+1
           CF   BLANK+51
           CF   BLANK+101
           TRNMINPUT-1,BLANK-49,,
           TRNMZERO,BLANK-39,2,
           AM   *-6,150,,
           CM   *-18,ZERO+1950,,
           BNE *-36
           TRNMZERO+1950,BLANK+61,,
           TRNMSTORE,ZERO,,
           BLXM*+12,STORE,9,
     ST2   BLC ST6
           RACDINPUT
           BLXM*+12,INPUT+33,10,
     ST22  TF   319,309,,
     ST23  SF   (1)
           CM   1+(1),0,10,
           CF   (1)
           BE   ST3
           SF   (1)
           CM   1+(1),20,10,
           CF   (1)
           BNE *+36
           BLXMST8,*+12,810,
           B    ST4
           BXM ST23,2,10,
     ST3   TD   (1),RM,,
           TRNM(2),(3),,
           TDM (1),0,,
     ST35  TRNM124+(2),INPUT+33,,
           AM   KNTR,1,,
           CM   KNTR,8,,
           BNE *+36
           BLXMWRITE,*+12,8,
           B    ST4
           BXM *+12,250,9,
     ST4   BXM *+12,2,10,
           SF   (1)
           CM   1+(1),0,10,
           CF   (1)
           BE   ST2
           SF   (1)
           CM   1+(1),24,10,
           CF   (1)
           BNE ST22
     ST5   BXM *+12,2,10,
```

```
      TF   319,309,,
ST51  SF   (1)
      CM   1+(1),4,10,
      CF   (1)
      BE   ST54
      SF   (1)
      CM   1+(1),0,10,
      CF   (1)
      BE   ST55
      SF   (1)
      CM   1+(1),20,10,
      CF   (1)
      BNE  *+36
      BLXMST8,*+12,810,
      B    ST5
      BXM  ST51,2,10,
ST54  TD   (1),RM,,
      TRNM(2),(3),,
      TDM  (1),0,,
      BXM  ST35,2,10,
ST55  TD   (1),RM,,
      TRNM(2),(3),,
      TDM  (1),0,,
      TRNM124+(2),INPUT+33,,
      AM   KNTR,1,,
      CM   KNTR,8,,
      BNE  *+36
      BLXMWRITE,*+12,8,
      B    ST5
      BXM  ST5,250,9,
ST8   TD   (1),RM,,
      TRNM(2),(3),,
      TDM  (1),2,,
      TRNM124+(2),INPUT+33,,
      AM   KNTR,1,,
      CM   KNTR,8,,
      BNE  *+36
      BLXMWRITE,*+12,8,
      B    (5)
      BXM  (5),250,9,
WRITE WDGNDISK
      TRNMSTORE,ZERO,,
      TFM  KNTR,0,,
      AM   DISK+5,20,,
      BLXM*+12,STORE,9,
      AM   CNT,1,,
      CM   CNT,10,,
      BNE  (4)
      SK   DISK
      TFM  CNT,0,,
      B    (4)
ST6   TDM  (2),0,,
      TD   1+(2),RM,,
      TD   2+(2),RM,,
      WDGNDISK
ST7   BS   *+12,0,,
      SK   DSKSRT
      WDGNDSKSRT
      CALLLINK,SORT,
RM    DAC  1,@,,
STORE DSS  2000
```

```
       DGM
INPUT  DAS 80
       DC   1,●,,
KNTR   DC   5,0,,
DISK   DDA  ,3,20000,20,STORE
CNT    DC   5,0,,
DSKSRTDDA  ,1,00000,004,CNTRL-49
       DAC  1,●,,
CNTRL  DC   50,01000250050000000001200000000024700000000000000000,,
       DC   50,00000000000000000000000000001000011000000000000000,,
       DC   50,00000000000000000000000000000000000000000000000000,,
       DC   50,00100000002073727070707070707070707070707372707070,
       DC   50,70707373757070707070707170717070727075767470707070,
       DC   50,70707070707070707070707070707070707070707070707070,
       DC   50,70707070707070707070730000000000000000000000000000,
       DC   50,00000000000000000000000000000000000000000000000000,,
       DGM
BLANK  DC   50,0,,
       DC   50,0,,
       DC   50,0,,
       DC   10,0,,
       DC   1,●,,
ZERO   DSS 2000
       DC   1,●,,
       DENDSTART
```

PICTAP

```
*NAMEPICTAP
*ID NUMBER 0247
            DORG20000
     START  BD   ST2,DIGIT,,
            TDM  DIGIT,1,,
            CF   ZERO-49
            CF   ZERO+1
            CF   ZERO+51
            CF   ZERO+101
            CF   ZERO+151
            CF   ZERO+201
            CF   ZERO+251
            CF   ZERO+301
            SKIP,7,,
            SF   02690,,6,
            AM   02690,123,,
     ST1    TF   TERM+122,02690,11,
            AM   02690,1,,
            SF   02690,,6,
            AM   02690,125,,
            TF   PHRASE+124,02690,11,
            PRA  TERM
            BCOV*+24
            B    *+36
            SKIP,7,,
     ST5    PRA  TERM
     ST4    PRA  PHRASE-10
            BCOV*+24
            B    02836
            SKIP,7,,
            TDM  KEY,1,,
            B    02836
     ST2    SF   02690,,6,
            AM   02690,123,,
            C    TERM+122,02690,11,
            BE   ST3
            TDM  KEY,0,,
            SPIM,1,,
            BCOV*+24
            B    ST1
            SKIP,7,,
            B    ST1
     ST3    AM   02690,1,,
            SF   02690,,6,
            AM   02690,125,,
            TF   PHRASE+124,02690,11,
            BD   *+24,KEY,,
            B    ST4
            TDM  KEY,0,,
            B    ST5
     FINAL  SK   DISK
            WDGNDISK
            CALLEXIT
     TERM   DAS  62
            DAC  1,@,,
            DAC  5,       ,,
     PHRASEDAS 63
            DAC  1,@,,
     DIGIT  DC   2,0,,
```

```
KEY    DC  2,0,,
       DAC 1,*,,
ZERO   DC  50,0,,
       DC  50,0,,
       DC  50,0,,
       DC  50,0,,
       DC  50,0,,
       DC  50,0,,
       DC  50,0,,
       DC  50,0,,
       DGM
DISK   DDA ,1,00000,004,ZERO-49,
       DENDSTART
```

PROJECT REPORT FORM

PROJECT NO. 2318
COOPERATOR
REPORT NO. 2
DATE November 20, 1967
NOTE BOOK
PAGE TO
SIGNED Richard W. Nelson

Copies to: Files
Nelson
Bachhuber
Brown
Roth
Byrne
Weiner
Holm
Dickey

EXPERIMENTAL SEARCH SYSTEM

SUMMARY

This report describes an experimental search system, with which one may explore the possibilities of document retrieval by way of an inverted keyword file. The data, which are extensive enough to permit meaningful tests to be made, consist of the record of keywords for the documents in Volume 37 of the Abstract Bulletin. These have been stored, in encoded and compressed form, on (a portion of) a disk pack. A group of computer programs provide for the storage and extraction of records, and for processing the logic of search definitions involving as many as twenty keywords.

The construction of this system, and the execution of a number of searches with it, have given substance to several hypotheses: (1) it is feasible to carry out mechanized searches with a comparatively small computer system; (2) efficient design, even when a large computer is available, will require the elimination of unnecessary features, such as English text output in large quantities; (3) an effective information system will require a mixture of storage media and search procedures, including printed books (which are by no means uneconomical for housing large masses of tabular material); and (4) effective use will demand adequate mutual comprehension of policy and practice among the persons in search of information, the specialists who prepare abstracts, and the designers of programs.

THE INVERTED FILE AND DISK PACKING PROCESS

It has been found possible to store a one-year accumulation of keywords and document identifications on approximately one-fourth of one disk pack, which is mounted in one of the two 1311 drives of an IBM 1620 II computer. The information in question corresponds to Volume 37 of the Abstract Bulletin. The inverted file of keywords was punched on cards when prepared (for production of the Keyword Supplement), and this series had been merged (by hand) to give a single file for the complete volume. The file consists of title cards, containing keywords as these appear in the PPRIC Thesaurus and its extensions, each title card followed by a list of document numbers pertaining to it. The document numbers are the serial numbers of entries in the Abstract Bulletin.

The machine program (172A3) which constructs the disk record reads this file and as it does so performs the following functions: (1) as each new keyword is encountered, a code (six decimal digits) is assigned to it, beginning with $\bar{0}00001$, which happens to be the keyword ABATEMENT; (2) the disk address of the disk area which will next be loaded is also recorded, together with the keyword itself, in a "directory" which is printed for later reference as the processing proceeds (in the present system the first address is 20000); (3) the keyword code is entered in a strip of 500 core positions, preceded by a record mark, and following without a break whatever information had already been placed there; (4) the document numbers following the given keyword are read, edited (7001 becomes $\bar{3}707001$, where the first two digits indicate the volume of the Abstract Bulletin), and the resulting seven-digit groups are loaded in the strip following the keyword code; (5) when all the document numbers have been read and processed, the next keyword is read, i.e., the cycle begins again at (1); (6) whenever the 500 core position strip is full, the contents are loaded to the disk, and the current disk address (to be used in the next loading) is increased by 5 units.

Figure 1 contains a portion of the "directory", Fig. 2 examples of coded and packed information as stored on the disk, and Fig. 3 a listing of the machine program. The preparation of the disk file proceeds essentially at card-reading speed and requires about 3 hours. The keyword accumulations for successive years can only be merged by reconstituting the file, since any differences in keyword vocabularies will require a new encoding for the combination.

In explanation of the choice of a six-digit field for the keyword code, even though no one presently contemplates the application of 10^6 keywords, it may be remarked that the extra digit or two will permit experimentation with keywords which do not belong to the present scheme (as examples, one might consider descriptions of the nature of the document, the language in which it is written, its security status, etc.), and that the packed file is not thereby lengthened excessively.

RECORD TRANSFER FROM DISK FILE

A subprogram (175A7), constructed as an SPS subroutine which can be called by Fortran mainline programs, extracts information from the disk in 5 sector blocks, and stores it in a strip of 500 core positions. Transfer begins at a sector address which must be furnished to the subprogram by the calling program. A "common" area is defined for communication between programs.

000001	20000	ABATEMENT
000002	20000	ABIENOL
000003	20000	ABIES
000004	20000	ABIES ALBA
000005	20000	ABIES AMABILIS
000006	20000	ABIES BALSAMEA
000007	20000	ABIES CONCOLOR
000008	20000	ABIES GRANDIS
000009	20000	ABIES LASIOCARPA
000010	20000	ABIES MAGNIFICA
000011	20000	ABIES NORDMANNIANA
000012	20000	ABIES PINDROW
000013	20005	ABIES PINSAPO
000014	20005	ABIES SIBIRICA
000015	20005	ABIETADIENE
000016	20005	ABIETIC ACIDS
000017	20005	ABIETINAL
000018	20005	ABIETINOL
000019	20005	ABNORMALITIES
000020	20005	ABRASION
000021	20005	ABRASION RESISTANCE
000022	20005	ABRASION RESISTANT STEELS
000023	20005	ABRASION TESTERS
000024	20005	ABRASIVE PAPERS
000025	20005	ABRASIVES
000026	20005	ABSORBENT PAPERS
000027	20005	ABSORBENTS
000028	20005	ABSORBERS (EQUIPMENT)
000029	20010	ABSORBERS (MATERIALS)
000030	20010	ABSORPTION
000031	20010	ABSORPTION SPECTRA
000032	20010	ABSORPTIVITY
000033	20010	ACACIA
000034	20010	ACACIA NILOTICA
000035	20010	ACACIA SENEGAL
000036	20010	ACCELERATING (PROCESS)
000037	20010	ACCELERATION (MECHANICAL)
000038	20010	ACCEPTANCE
000039	20010	ACCESSIBILITY
000040	20010	ACCESSORIES
000041	20010	ACCIDENT PREVENTION
000042	20010	ACCOUNTING
000043	20015	ACCUMULATION
000044	20015	ACCUMULATORS
000045	20015	ACCURACY
000046	20015	ACER
000047	20015	ACER PLATANOIDES
000048	20015	ACER PSEUDOPLATANUS
000049	20015	ACER RUBRUM
000050	20015	ACER SACCHARINUM
000051	20015	ACER SACCHARUM
000052	20015	ACETALDEHYDE
000053	20015	ACETAL RESINS
000054	20015	ACETALS
000055	20015	ACETATE PULPS
000056	20015	ACETATE RAYON
000057	20015	ACETATES
000058	20025	ACETIC ACID
000059	20025	ACETIC ANHYDRIDE
000060	20025	ACETOLYSIS

Figure 1

02̄001500539000	**Control Field**
073133̄7071943̄7073153̄7073163̄707308‡0̄000563̄707191‡0̄0	**Contents**
00573̄7073223̄7071633̄707313‡0̄000583̄7073083̄707318‡0̄00	
0593̄707128‡0̄000603̄7071473̄7071483̄707149‡0̄000613̄7075	
003̄7075203̄7075403̄7074443̄7074743̄7072943̄7075463̄70750	
73̄7073173̄7073483̄7072183̄7073083̄7074793̄707299‡0̄00062	
3̄7074403̄7074503̄7078403̄7072503̄7074723̄7074333̄707473̄	
7074043̄7074453̄7074663̄707369‡0̄000633̄707197‡0̄00064 3̄7	
07178‡0̄000653̄707198‡0̄000663̄7073333̄7073363̄707816‡0̄0	
00673̄7073333̄707336‡0̄000683̄707179‡0̄000693̄7072043̄707	
307‡0̄000703̄707307‡0̄000713̄707564‡0̄000723̄707216‡0̄000	
‡0000000000000000000000000000000000000000000000000	

02̄002000539000	**Control Field**
733̄707664‡0̄000743̄7072163̄707878‡0̄000753̄7071813̄70786	**Contents**
8‡0̄000763̄7078723̄7078633̄7071453̄7075373̄707159‡0̄00077	
3̄7071453̄707537‡0̄000783̄707162‡0̄000793̄707159‡0̄000803̄	
7073213̄707503‡0̄000813̄707209‡0̄000823̄7071643̄707155 3̄7	
071563̄707258‡0̄000833̄707494‡0̄000843̄7071203̄7074003̄70	
7460 3̄7071123̄7073023̄7071533̄7071633̄7071933̄7073833̄707	
1143̄7071543̄7072843̄7071053̄7071153̄7071453̄7071553̄7071	
063̄7071963̄7071973̄707119‡0̄000853̄7071363̄7071383̄70712	
9‡0̄000863̄7075933̄707863‡0̄000873̄707911‡0̄000883̄707141	
3̄707136‡0̄000893̄707162‡0̄000903̄7075313̄707375‡0̄000913̄	
‡0000000000000000000000000000000000000000000000000	

Two Consecutive Portions of Encoded File, as Stored on Disk (each portion is 500 digits long; the final row of zeros is not present on the disk and is to be disregarded)

Figure 2

```
C      DISK LOADING PROGRAM  172A3  REMOVE COMMENT CARDS BEFORE LOADING

C      PART L  LOADER
3620000000500322000000000015200220000‡3120004̄200104900000

C      PART A  PRELIMINARIES
08500      323615800000
08512      15351080000‡
08524      1635155L9000
08536      15350560000‡
08548      1635005K0000
08560      1635055-0001
08572      1635051000-0
08584      733598135055
08596      733599535005
08608      153616000000
08620      15361610000‡
08632      163500800-05
08644      1635013L9000
08656      153500000000
08668      343500000701
08680      323600000000
08692      153804000000
08704      15380410000‡
08716      490900000000

C      PART B  READ-IN
09000      373600100500
09012      1436159000K0
09024      470950001200
09036      490910800000
09048      153515N0000‡
09060      1135155000-1
09072      313515N35050
09084      1135155-0006
09096      490800000000
09108      263803936039
09120      1609499-9000
09132      490904800000
```

Figure 3

```
C        PART C  DOCUMENT CODE EDITING
09500       490968000000
09512       323510300000
09524       723520N35107
09536       1435107-0000
09548       460963201200
09560       333510300000
09572       1635102000L7
09584       313515N35101
09596       1135155-0007
09608       1609499-9632
09620       491000000000
09632       1135205000J0
09644       1435205L6139
09656       470951201100
09668       490900000000
09680       243603938039
09692       460976401200
09704       340000000102
09716       390978900100
09728       393799700100
09740       480000000000
09752       490900000000
09764       1635205L6049
09776       490951200000
09788       430ǂ

C        PART D  PACKING
10000       263525535155
10012       1235255L9500
10024       471020401100
10036       313975039500
10048       15395000000ǂ
10060       15101790000ǂ
10072       490750000000
10096       1635155L9000
10108       450949R10179
10120       313900039750
10132       213515535255
10144       45101683515N
10156       490949R00000
10168       480000000000
10180       490949R00000
10204       470949R01200
10216       15395000000ǂ
10228       151017900000
10240       4907500
```

Figure 3 (Continued)

```
C       PART E  PRINT DIRECTORY
08000       393594700900
08012       460808402500
08024       470804803400
08036       340000000971
08048       490813200000
08060       733598135055
08072       491000000000
08084       480000000000
08096       490800000000
08132       733615935055
08144       393600100400
08156       323615800000
08168       1135055000-1
08180       490806000000

C       PART F  DISK TRANSFER
07500       383500000700
07512       460774003600
07524       460774003800
07536       460774000700
07548       460774001600
07560       460774001700
07572       363500000701
07584       460774003600
07596       460774003700
07608       460774003800
07620       460774000600
07632       460774001600
07644       460774001700
07656       1135005000-5
07668       343500000701
07680       733599535005
07692       491009600000
07740       480000000000
07752       490750000000

C       PART G  STARTER
4908500
```

Figure 3 (Continued)

The subprogram now scans the strip to find a keyword code which agrees with a keyword code previously established by the calling program. The document codes which follow the selected keyword code are edited and transmitted to an output strip, in which they are accessible to the mainline program. The output strip provides for 100 floating-point variables, and the document code editing will, for example, change $\overline{3}707001$ to $\overline{3}707001\overline{0}02$ before transfer to the output strip.

If the output strip becomes full before the last document number has been processed, a signal is set in the "common" area and control is transferred to the mainline program. The latter then processes the output strip and returns control to the subprogram which then assembles more output.

On occasion, part of the desired keyword code will be filed at the beginning of the next group of five sectors. The subprogram then saves the beginning of the keyword code, reads in the next group of five sectors, and proceeds as above. If no part of the specified keyword code is to be found in the 500 core position strip, an error indicator is set and control returns to the main program.

The user must find the keyword code and sector address in the "directory" and supply these to the mainline program. While this part of the retrieval process could also be mechanized, it is considered that this feature (which brings with it a substantial storage space problem) belongs to a later stage of development.

A listing of the subprogram (which has the name ACCESS) appears in Fig. 4.

```
*ASSEMBLE RELOCATABLE
*NAME ACCESS
*STORE RELOADABLE
*LIST PRINTER
    S      DS   ,*+101
           DC   6,987898,5-S
           DAC  6,ACCESS,7-S
           DVLC22-S,5,LENGTH,2,8,2,4,5,ENTRY-6,5,0,30,0
           DSC  17,0,0
           DORGS-100
           DAS  10
    BAND   DAS  251
           DGM  BAND+499
    MARK   DS   5
    LOC    DS   5
    LOW    DC   5,20000
    HIGH   DC   5,39999
    CYL    DC   3,0
    DISK   DAS  7
           DDA  DISK-1,0,20000,5,BAND-1
    MSG2   DAC  1,F
           DAC  1,@
           DC   5,0
    ENTRY  AM   ENTRY-1,1,10
           CM   38979,0,8
           BI   INIT,01100
           TFM  MARK,39006,7,              RESET LOADING ADDRESS
           TFM  38983,0,8,                 RESET OUTPUT COUNTER
           TFM  LOC,BAND+499,7,            RESET LOCATION COUNTER
           CF   38996
           SF   38994,,,                   ASSEMBLE TITLE CODE
           CF   38988
           SF   38987,,,                   ASSEMBLE SECTOR ADDRESS
           TF   DISK+4,38991,,             TRANSMIT TO CONTROL FIELD
           TFM  FILL-1,START,7,            SET RETURN
           C    DISK+4,LOW
           BNI  ERROR1,01300
           C    DISK+4,HIGH
           BI   ERROR1,01100
           LD   00099,DISK+4
           DM   00097,200,9
           C    00096,CYL
           BI   FILL+12,01200
           TF   CYL,00096
           SK   DISK-1,00701,,             SEEK
           B    FILL+12,,,                 GO TO TRANSFER FROM DISK
```

Figure 4

```
INIT   TFM MARK,39006,7,          RESET LOADING ADDRESS
       TFM 38983,0,8,             RESET OUTPUT COUNTER
       B   AGAIN
START  TFM LOC,BAND-1,7,          RESET LOCATION COUNTER
NEXT   BNR STEP,LOC,11
       B   CYCLE
STEP   AM  LOC,1,10
       CM  LOC,BAND+498,7
       BNI NEXT,01100
       B   ERROR1
CYCLE  AM  LOC,6,10
       CM  LOC,BAND+498,7
       TFM FILL-1,R1,7
       BI  FILL,01100
R1     C   LOC,38999,6
       BNI STEP,01200
AGAIN  AM  LOC,1,10
       CM  LOC,BAND+498,7
       TFM FILL-1,R2,7
       BI  FILL,01100
R2     BNR SKIP,LOC,11
       TFM 38979,0,8,             SET CONTROL COUNTER FOR EXIT
       B   ENTRY-1,,6,            RETURN
SKIP   AM  LOC,6,10
       CM  LOC,BAND+498,7
       TFM FILL-1,R3,7
       BI  FILL,01100
R3     TF  MARK,LOC,611
       AM  MARK,1,10
       TDM MARK,0,6
       AM  MARK,2,10
       TFM MARK,2,610
       AM  MARK,7,10
       AM  38983,1,10
       CM  MARK,39996,7
       BNI AGAIN,01100
       TFM 38979,1,8,             SET CONTROL COUNTER FOR REFILL
       B   ENTRY-1,,6,            RETURN
ERROR1TFM 38979,1,811,            SET CONTROL COUNTER FOR MISS
       B   ENTRY-1,,6,            RETURN
       DORG*+6
FILL   TF  BAND-2,BAND+498
       SM  LOC,500,8
READ   RDGNDISK-1,00700
       BI  ERROR2,03600
       BI  ERROR2,03700
       BI  ERROR2,03800
       BI  ERROR2,00600
       BI  ERROR2,01600
       BI  ERROR2,01700
```

Figure 4 (Continued)

```
          AM   DISK+4,5,10
          LD   00099,DISK+4
          DM   00097,200,9
          C    00096,CYL
          BI   FILL-1,01200,6
          TF   CYL,00096
          SK   DISK-1,00701
          B    FILL-1,,6
ERROR2K        ,00102
          WATYMSG2
          H
          B    READ
LENGTHDC       1,@
          DEND
```

Figure 4 (Continued)

THE SEARCH DEFINITION

The search program, discussed in the next section, processes a search definition which may contain a maximum of twenty keywords. Each keyword is presented to the program in the form of a keyword code and a sector address.

A typical search definition, intelligible to the program, is of the form

$$(A \cup B \cup C) \cap (D \cup E) \cap (F \cup G \cup H \cup I) \ . \ . \ .$$

in which a letter represents a set of documents filed under a keyword. Provided the maximum number of keywords is not exceeded, any arrangement of parentheses is accepted.

The symbols present in the search definition may also be of the form $\overline{Q}$, where Q is the set of documents filed under a keyword and $\overline{Q}$ is the complement of that set. This makes it possible to exclude documents having certain qualities, and the program will process such definitions correctly provided that there is at least one parenthesis which contains no complements of sets.

The exceptional cases are not of practical importance. For example, the search definition $(\overline{A})$ would call for all documents not belonging to the set A; the correct response would be an utterly useless list of nearly 10,000 documents. The computer program, in such a case, would indicate that no documents had the required property. In any event, the program can apply the search definition only to those documents which are listed, in the file, under one of the keywords which appear in the search definition.

SEARCH PROGRAM

The work of this program proceeds in two phases. In the first, a search definition is read and processed. As it lists the keyword codes and sector addresses and records the formal structure of the definition, the program extracts from the keyword file the document numbers listed under each of the keywords. These are stored, with duplications deleted, in a 500 position stack. There is provided an overflow area of the same size for use when more than 500 document numbers are returned. As each document is entered in the stack, an entry is made in the corresponding row of a 500 by 20 core position matrix (previously cleared), in the column corresponding to the keyword being processed. If the search definition refers to the set belonging to a certain keyword, the entry made is 1 when the document is listed under the keyword and 0 otherwise; but if the definition refers to the complement of the set, the entries are respectively 0 and 1. When the document had previously been entered in the stack, the entry is made in the appropriate row and the same column of the matrix.

In the second phase, the program scans the matrix, row by row, to detect entries which conform to the search definition. The program tests agreement with each parenthesis in turn, starting from the left, and the item is rejected whenever a lack of agreement is observed. If the item is acceptable, the location (row) and the corresponding document number are printed. When the scan is complete, and at the option of the user (indicated by program switch settings), the program prints the contents of the document stack and the matrix.

If document numbers have been filed in the overflow area, the program provides for a second reading of the search definition, and these document numbers are then processed as before, with similar output. When more than 1000 document numbers (without duplications) are returned from the disk file, the excess is discarded and the search proceeds with the items which have been retained; a warning message is printed. This limitation can be circumvented, in many cases, by changing the order in which the keywords appear in the search definition; this changes the order in which document numbers appear in the stack, and the items discarded may now be those which were retained in a previous trial.

The size of the matrix and of the stack and overflow area appear to be best for the available core capacity (40K). Temporary storage on disk as a means of increasing the effective dimensions of the work area is a possibility, but will require increased processing time, which is already appreciable for problems which do not exceed present capacities. Some processing time could be saved by more efficient coding (in SPS) of critical portions of the program.

A listing of the mainline program (176A10, SEARCH) and the other of its two subprograms (187A7, TABLES) is given in Fig. 5.

PRELIMINARY TRIALS

In the examples of output which follow this section, the program switches were set for complete listings. Example 1 shows a search which was intended to retrieve documents relating to the abatement of pollution in streams. The search definition is of the form

$$(A \cup B \cup C \cup D) \cap (E \cup F) \cap (G \cup H)$$

in the first parenthesis, A is the set of document numbers listed under the keyword

```
*FANDK0804
*LDISKSEARCH
*LIST PRINTER
C     UTILITY PROGRAM  176A10       25 APR 1967
C     PRELIMINARY VERSION OF DOCUMENT RETRIEVAL SYSTEM
C          PART A, PRELIMINARIES
C               COMMUNICATION AREA FOR  ACCESS  SUBPROGRAM
      DIMENSION A(100)
      COMMON A, NKEY1, NKEY2, NSEC1, NSEC2, NR, KT
C               COMMUNICATION AREA FOR  TABLES  SUBPROGRAM
      DIMENSION JUN(20), JSGN(20)
      COMMON JUN, JSGN, JTO, JGR, JKWD, JDCM, JTAB
C               STORAGE ARRANGEMENTS FOR THIS PROGRAM
      DIMENSION D(1000), ITEXT(21)
      PRINT 308
  101 LOAD = 1
      IRECYC = 0
      DO 103  I = 1, 1000
  103 D(I) = 0.0
  102 JKWD = 0
      JTAB = 1
      CALL TABLES
      JTAB = 2
      KT = 0
C          PART B, INPUT
      READ 301, J
      IF (J)  407, 407, 408
  408 DO 405  I = 1, J
      READ 307
  405 PRINT 307
      PRINT 302
  407 READ 301, JTO, JGR
      PRINT 301, JTO, JGR
      PRINT 302
C               LOOP FOR INPUT OF GROUPS
      DO 107  I = 1, JGR
      READ 303, JUN(I)
      PRINT 303, JUN(I)
C               LOOP FOR INPUT OF KEYWORDS
      K = JUN(I)
      DO 105  J = 1, K
      JKWD = JKWD + 1
      READ 304, NKEY2, NKEY1, NSEC2, NSEC1, JSGN(JKWD), ITEXT
      TEMP1 = NKEY1
      TEMP2 = NKEY2
      P = 10000.0 * TEMP2 + TEMP1
      TEMP1 = NSEC1
      TEMP2 = NSEC2
      Q = 10000.0 * TEMP2 + TEMP1
```

Figure 5

```
      PRINT 305, P, Q, JSGN(JKWD), ITEXT
      CALL ACCESS
C                LOOP FOR CONSTRUCTION OF DOCUMENT CODE LIST
      IF (KT)  105, 570, 570
  570 IF (IRECYC - 1)  526, 526, 726
  526 DO 515  L = 1, NR
      LOADM1 = LOAD - 1
      DO 521  M = 1, LOADM1
      IF (A(L) - D(M))  521, 535, 521
  535 IF (M - 500)  551, 551, 515
  551 JDCM = M
      CALL TABLES
      GO TO 515
  521 CONTINUE
      IF (LOAD - 1000)  554, 554, 555
  555 IRECYC = 1
      GO TO 515
  554 D(LOAD) = A(L)
      IF (LOAD - 500)  552, 552, 553
  552 JDCM = LOAD
      CALL TABLES
  553 LOAD = LOAD + 1
  515 CONTINUE
      IF (KT)  105, 105, 525
  525 CALL ACCESS
      GO TO 526
  105 CONTINUE
  107 PRINT 302
      IF (IRECYC - 1)  556, 557, 575
  557 PRINT 311
      GO TO 561
  556 IF (LOAD - 501)  575, 575, 576
  576 IRECYC = 1
  561 TYPE 310
C           PART C, OUTPUT
  575 J = 0
      DO 601  I = 1, JGR
  601 J = J + JUN(I)
      J = J - JTO
      IF (J)  602, 603, 602
  602 PRINT 309
  603 JTAB = 3
      IF (IRECYC - 1)  650, 650, 850
  650 LOADM1 = LOAD - 1
      JDCM = LOADM1
      IF (JDCM - 500)  610, 610, 611
  611 JDCM = 500
  610 CALL TABLES
      IF (JTAB - 4)  604, 604, 605
```

Figure 5 (Continued)

```
  604 PRINT 306, JDCM, D(JDCM)
      GO TO 610
  605 PRINT 302
      IF (SENSE SWITCH 1)  652, 653
  652 DO 651  I = 1, LOADM1
  651 PRINT 306, I, D(I)
      PRINT 302
  653 IF (SENSE SWITCH 2)  654, 655
  654 CALL TABLES
      PRINT 302
  655 IF (IRECYC - 1)  101, 670, 670
  670 IRECYC = 2
      GO TO 102
C           PART D, RECYCLE PROCEDURE
  726 DO 715  L = 1, NR
      DO 721  M = 1, LOADM1
      IF (A(L) - D(M))  721, 735, 721
  735 JDCM = M - 500
      IF (JDCM)  715, 715, 736
  736 CALL TABLES
      GO TO 715
  721 CONTINUE
  715 CONTINUE
      IF (KT)  105, 105, 725
  725 CALL ACCESS
      GO TO 726
  850 JDCM = LOADM1 - 500
  810 CALL TABLES
      IF (JTAB - 4)  804, 804, 805
  804 I = JDCM + 500
      PRINT 306, I, D(I)
      GO TO 810
  805 PRINT 302
      IF (SENSE SWITCH 2)  854, 855
  854 CALL TABLES
      PRINT 302
  855 GO TO 101
C           PART E, I/O ARRANGEMENTS
  301 FORMAT (1H , 4X, I5, 5X, I5)
  302 FORMAT (1H0)
  303 FORMAT (1H , 4X, I5)
  304 FORMAT (2X, 2I4, 2X, 2I4, 5X, I5, 21A2)
  305 FORMAT (1H , 20X, F10.0, 10X, F10.0, 10X, I5, 21A2)
  306 FORMAT (1H , 15X, I5, 5X, F10.5)
  307 FORMAT (1H , 4X, 67H
     1                                        )
  308 FORMAT (1H , 4X, 31HUTILITY PROGRAM  176A10  SEARCH/)
  309 FORMAT (1H , 4X, 26HKEYWORD COUNT INCONSISTENT/)
  310 FORMAT (23HRELOAD CURRENT DATA SET)
  311 FORMAT (1H , 4X, 22HLIST CAPACITY EXCEEDED/)
      END
```

Figure 5 (Continued)

```
*LDISKTABLES
*LISTPRINTER
C     UTILITY PROGRAM  176A7       25 APR 1967
      SUBROUTINE TABLES
      DIMENSION BLANKA(100)
      COMMON BLANKA, KLANKB, KLANKC, KLANKD, KLANKE, KLANKF, KLANKG
      DIMENSION JUN(20), JSGN(20)
      COMMON JUN, JSGN, JTO, JGR, JKWD, JDCM, JTAB
      DIMENSION JET(500,5)
      GO TO (401, 402, 403, 404, 405), JTAB
  401 DO 415  I = 1, 500
      DO 415  J = 1, 5
  415 JET(I,J) = 0.0
      RETURN
  402 I = JDCM
      J = (JKWD + 3) / 4
      K = JKWD - ((JKWD - 1) / 4) * 4
      GO TO (501, 502, 503, 504), K
  501 JET(I,J) = JET(I,J) + 1000
      RETURN
  502 JET(I,J) = JET(I,J) + 100
      RETURN
  503 JET(I,J) = JET(I,J) + 10
      RETURN
  504 JET(I,J) = JET(I,J) + 1
      RETURN
  403 JTAB = 4
      JEND = JDCM
      JDCM = 1
  711 NGR = 1
      INIT = 1
  716 IFIN = INIT + JUN(NGR) - 1
      DO 708  JKWD = INIT, IFIN
      I = JDCM
      J = (JKWD + 3) / 4
      K = JKWD - ((JKWD - 1) / 4) * 4
      GO TO (701, 702, 703, 704), K
  701 L = JET(I,J) / 1000
      GO TO 705
  702 M = JET(I,J)
      N = (JET(I,J) / 1000) * 1000
      L = (M - N) / 100
      GO TO 705
  703 M = JET(I,J)
      N = (JET(I,J) / 100) * 100
      L = (M - N) / 10
      GO TO 705
```

Figure 5 (Continued)

```
704 M = JET(I,J)
    N = (JET(I,J) / 10) * 10
    L = M - N
705 IF (JSGN(JKWD))  751, 752, 752
751 IF (L)  709, 709, 708
752 IF (L)  708, 708, 709
708 CONTINUE
404 JDCM = JDCM + 1
    IF (JDCM - JEND)  711, 711, 725
725 JTAB = 5
    RETURN
709 NGR = NGR + 1
    INIT = IFIN + 1
    IF (NGR - JGR)  716, 716, 717
717 RETURN
405 DO 410  I = 1, JEND
410 PRINT 301, I, JET(I,1), JET(I,2), JET(I,3), JET(I,4), JET(I,5)
    RETURN
301 FORMAT (1H , 15X, I5, 5X, 5I4)
    END
```

Figure 5 (Continued)

ABATEMENT, B the set listed under the keyword ELIMINATION, etc. Thus, the first parenthesis refers to abatement and three synonyms, the second parenthesis to pollution and a synonym (actually a narrower term), and the third to streams and a synonym.

One item was retrieved (at row 5 in the stack and matrix, and the reference is to abstract 6654 of Volume 37 of the Abstract Bulletin). The same item would have been found if the search definition had been $(A) \cap (E) \cap (H)$. The synonyms, in this example, were superfluous. But Example 2 shows that it is well to include them (when space limitations permit). Here the last line of the matrix indicates that a document was listed under STREAM POLLUTION but not under POLLUTION. Reinforcement of the original cross-referencing by the user ordinarily costs little and may help to promote familiarity with the Thesaurus and more flexible design of search definitions.

Other qualitative conclusions which follow from preliminary experiments are that the user should be prepared to test a number of search definitions of varying structure and degrees of sharpness, and that the use of some heavily posted keywords can be expensive in terms of storage space and processing time.

The output for Examples 1 and 2 is shown in Fig. 6.

FURTHER DEVELOPMENTS

It has been considered unnecessary, in an experimental system, to provide such conveniences as output in the form of title, author(s), and publication reference. The storage of such information requires a magnetic-tape system, but apart from programming effort and other expense there should be no prohibitive difficulty.

```
SEARCH 07800 15234 LOADED
TABLES 23034 12866 LOADED
ACCESS 35900 01584 LOADED
03     37484 00474 LOADED
```

```
UTILITY PROGRAM  176A10  SEARCH

EXAMPLE 1

8          3

4
              1.        20000.       0      ABATEMENT
           1363.        21330.       0      ELIMINATION
           3074.        23000.       0      PREVENTION
           3240.        23225.       0      REDUCTION

2
           2953.        22835.       0      POLLUTION
           3768.        23645.       0      STREAM POLLUTION

2
           3325.        23295.       0      RIVERS
           3769.        23645.       0      STREAMS

        5      37.06654

        1      37.09070
        2      37.09071
        3      37.09066
        4      37.09069
        5      37.06654
        6      37.00395
        7      37.04011
        8      37.04161
        9      37.04165
       10      37.09051
       11      37.06390
       12      37.06504
       13      37.04682
       14      37.00896
       15      37.02877
       16      37.03423
```

Figure 6

17	37.07144
18	37.07219
19	37.06985
20	37.08148
21	37.08930
22	37.08876
23	37.08929
24	37.09009
25	37.08291
26	37.08296
27	37.08297
28	37.07274
29	37.07275
30	37.07309
31	37.07039
32	37.05170
33	37.05528
34	37.05963
35	37.05638
36	37.04219
37	37.01930
38	37.01886
39	37.02017
40	37.02615
41	37.03276
42	37.03509
43	37.07150
44	37.07241
45	37.07513
46	37.07504
47	37.07495
48	37.07518
49	37.09082
50	37.08869
51	37.06537
52	37.06658
53	37.05163
54	37.05065
55	37.05165
56	37.05921
57	37.05923
58	37.05934
59	37.05924
60	37.04321
61	37.04015
62	37.04305
63	37.00396
64	37.00027
65	37.00399
66	37.00037
67	37.02071
68	37.02066
69	37.03631
70	37.03632
71	37.03323
72	37.08280
73	37.08298
74	37.09078
75	37.05942

Figure 6 (Continued)

76	37.03333
77	37.00410
78	37.00403
79	37.07521
80	37.07514
81	37.07509

1	10011000	0	0	0
2	1000 0	0	0	0
3	10011000	0	0	0
4	1000 0	0	0	0
5	10001001	0	0	0
6	1000 0	0	0	0
7	10001000	0	0	0
8	100 0	0	0	0
9	100 0	0	0	0
10	100 0	0	0	0
11	100 0	0	0	0
12	100 0	0	0	0
13	10 0	0	0	0
14	10 0	0	0	0
15	10 0	0	0	0
16	10 0	0	0	0
17	10 0	0	0	0
18	10 0	0	0	0
19	10 0	0	0	0
20	10 0	0	0	0
21	1 0	0	0	0
22	1 0	0	0	0
23	1 0	0	0	0
24	1 0	0	0	0
25	1 0	0	0	0
26	1 0	0	0	0
27	1 0	0	0	0
28	1 0	0	0	0
29	1 0	0	0	0
30	1 0	0	0	0
31	1 0	0	0	0
32	1 0	0	0	0
33	1 0	0	0	0
34	1 0	0	0	0
35	1 0	0	0	0
36	1 0	0	0	0
37	1 0	0	0	0
38	1 0	0	0	0
39	1 0	0	0	0
40	1 0	0	0	0
41	1 0	0	0	0
42	1 0	0	0	0
43	01000	0	0	0
44	01000	0	0	0
45	01000	0	0	0
46	01000	0	0	0
47	01000	0	0	0
48	01001	0	0	0
49	01000	0	0	0
50	01000	0	0	0
51	01000	0	0	0

Figure 6 (Continued)

52	01000	0	0	0
53	01000	0	0	0
54	01100	0	0	0
55	01000	0	0	0
56	01000	0	0	0
57	01000	0	0	0
58	01101	0	0	0
59	01000	0	0	0
60	01000	0	0	0
61	01000	0	0	0
62	01000	0	0	0
63	01000	0	0	0
64	01000	0	0	0
65	01000	0	0	0
66	01000	0	0	0
67	01100	0	0	0
68	01100	0	0	0
69	01000	0	0	0
70	01100	0	0	0
71	01100	0	0	0
72	0 100	0	0	0
73	0 10	0	0	0
74	0 10	0	0	0
75	0 10	0	0	0
76	0 10	0	0	0
77	0 1	0	0	0
78	0 1	0	0	0
79	0 1	0	0	0
80	0 1	0	0	0
81	0 1	0	0	0

Figure 6 (Continued)

EXAMPLE 2

2	2				
1		2953.	22835.	0	POLLUTION
1		3768.	23645.	0	STREAM POLLUTION

15	37.05065
19	37.05934
29	37.02071
30	37.02066
32	37.03632
33	37.03323

1	37.07150
2	37.07241
3	37.07513
4	37.07504
5	37.07495
6	37.07518
7	37.09070
8	37.09082
9	37.09066
10	37.08869
11	37.06654
12	37.06537
13	37.06658
14	37.05163
15	37.05065
16	37.05165
17	37.05921
18	37.05923
19	37.05934
20	37.05924
21	37.04011
22	37.04321
23	37.04015
24	37.04305
25	37.00396
26	37.00027
27	37.00399
28	37.00037
29	37.02071
30	37.02066
31	37.03631
32	37.03632
33	37.03323
34	37.08280

1	1000	0	0	0	0

Figure 6 (Continued)

2	1000	0	0	0	0
3	1000	0	0	0	0
4	1000	0	0	0	0
5	1000	0	0	0	0
6	1000	0	0	0	0
7	1000	0	0	0	0
8	1000	0	0	0	0
9	1000	0	0	0	0
10	1000	0	0	0	0
11	1000	0	0	0	0
12	1000	0	0	0	0
13	1000	0	0	0	0
14	1000	0	0	0	0
15	1100	0	0	0	0
16	1000	0	0	0	0
17	1000	0	0	0	0
18	1000	0	0	0	0
19	1100	0	0	0	0
20	1000	0	0	0	0
21	1000	0	0	0	0
22	1000	0	0	0	0
23	1000	0	0	0	0
24	1000	0	0	0	0
25	1000	0	0	0	0
26	1000	0	0	0	0
27	1000	0	0	0	0
28	1000	0	0	0	0
29	1100	0	0	0	0
30	1100	0	0	0	0
31	1000	0	0	0	0
32	1100	0	0	0	0
33	1100	0	0	0	0
34	100	0	0	0	0

Figure 6 (Continued)

There is no reason why the existing inverted file, stored on disk, cannot be reinverted to form a direct file. A direct file, also stored on disk, would make it possible to try direct search systems, and would extend the usefulness of the inverted file system. Thus, after a search of the inverted file, as described in this report, one could analyze the regularities (if any) in the keywords belonging to the documents which had been selected in a search. Additional search questions may be suggested by any patterns which develop.

PROJECT REPORT FORM

PROJECT NO. 2318
COOPERATOR Institute of Paper Chemistry
REPORT NO. 1
DATE October 5, 1962
NOTE BOOK
PAGE
SIGNED Edward F. Thode
Edward F. Thode

Copies to: Files, C. L. Brown, E. E. Dickey, A. E. Grummer, R. W. Nelson, L. E. Roth, M. L. Scribner, J. G. Strange, E. F. Thode, Jack Weiner, R. P. Whitney, Reading Copy

DEVELOPMENT OF MECHANIZED METHODS FOR TECHNICAL INFORMATION RETRIEVAL

INTRODUCTION

Institute activity in the development of retrieval systems for technical information has had two main goals:

1. The development of a generalized system of retrieval applicable to collections of various sizes in the paper industry with full compatibility of coding and searching procedures among the various collections (sub-systems).

2. The development of a highly mechanized and highly efficient retrieval system -- for our large, centralized collection -- which speaks to the needs of our staff, students and member companies.

The elements of experimentation on this project include six major steps; namely,

1. Preparation of a pulp and paper supplement to the Chemical Engineering Thesaurus.

2. Coding a large sample of documents from A.B.I.P.C.

3. Checking this coding for different sources of variation.

4. Experimenting with search configurations.

5. Providing means for inclusion of various externally and internally generated documents in the file.

6. Developing effective techniques for processing questions from and answers to users of the system.

Also involved in this effort is considerable education on the part of the study group on the techniques tried and found useful (or not) by others - furthermore, there has been established close co-operation with the parallel program at the Pulp and Paper Research Institute of Canada.

PROBLEMS OF AN INFORMATION RETRIEVAL SYSTEM

Before embarking on as complicated a venture as this, one must have the goal clearly in mind. Even more generalized than the goals just cited is that of the Engineers Joint Council program, of which our activity may properly be considered a sub-set. Slightly paraphrased, their statement of objectives is: "To improve the EFFICIENCY of ENGINEERS and TECHNOLOGISTS in the function of OBTAINING PERTINENT INFORMATION from the CURRENT TECHNICAL LITERATURE." The problems faced in setting up an information retrieval system must be viewed with keywords of this statement in mind.

We must consider, for one thing, "Who is to be served?" The answer is scientists, engineers and technologists in the pulp and paper industry.

Next, "What do they want?" The answer, pertinent technical information from the current (this term needs some defining) literature.

"Why do they want it?" - To be more effective and efficient on the job.

In other words, we must focus on the information needs of a very special audience and must serve these needs in such a way as to save the individuals time and effort in their vocational activities.

In dealing with communication between any author and any reader (or, searcher for information) certain language problems always intervene. The four major classes of language problems which crop up in technical communications may be categorized as follows:

VIEWPOINT
MEANING
WORD ORDER
FAMILY RELATIONSHIPS

The problems of viewpoint and meaning are interrelated. A given term may have several different meanings regardless of viewpoint and requires use of a modifier for specificity. The term, "track", for example, may mean anything from a pair of iron rails attached to cross-ties to a sequence of animal footprints in snow, soft earth, etc. Furthermore, a term may have various meanings depending on the viewpoint of the observer or writer. To a heating engineer, writing about oil burners, "oil" is a mixture of paraffinic and naphthenic hydrocarbons of a restricted range of density and viscosity obtained by the distillation of crude petroleum. To the wildcatter, oil is the crude itself; to the French chef, oil is something entirely different. Imagine the difficulties terms such as "chest","headbox", "furnish" present to the person unaquainted with the language and viewpoint of the papermaker. However, in a specialized information retrieval scheme, the viewpoint of the author and reader may be very close, so that it is possible to restrict greatly the possible range of meanings for a given term.

Now, the way in which a writer arranges the words in a sentence or thoughts in a paragraph establishes a certain meaning by context. Contrast two titles, "The Organization of a Republican Form of Government" and "The Form and Government of a Republican Organization". An individual concerned with Republican Party affairs would have no interest in the former article, yet he would surely retrieve it, among others, if it were indexed solely by keywords with no indication of context. To avoid excessive "false drops" of non-pertinent information in a mechanized system, some provision must be made for preserving the contextual relationships of terms defined by the author.

Finally, there is the problem of family relationships of words. Different words in the English language may be used to describe exactly the same concept. "Methanol" and "methyl alcohol" are synonyms; imagine the confusion which would arise if half the documents in a system relating to this concept were coded or indexed under "methanol" and the other half under "methyl alcohol". Synonyms must be eliminated from a retrieval system; but when this is done, there arises the more complex problem of <u>closely related</u> terms describing <u>slightly different</u> concepts. Consider "crushing", "milling" and "pulverizing"; these are quite properly distinct concepts and an author may quite properly use one or another of these terms to precisely designate both mode and degree of size reduction, according to the context of his paper. The <u>seeker</u> of information, however, may be uncertain about the <u>precise</u> mode and degree of size reduction about which he needs information; if so, he must be provided with a guide for exploring concepts closely related to that which originally comes to mind. Examination of related concepts, furthermore, must be both vertical and horizontal. The seeker of information on milling may wish merely to explore this concept plus those of crushing and pulverizing; on the other hand, he may decide that <u>all</u> related concepts need coverage, in which case the search should be directed to the class concept, "size reduction". Thus, a guide to vertical relationships is also needed.

CONCEPT CO-ORDINATION

An appreciation and analysis of the above problems has led various literature specialists to the development of the technique of <u>Concept Co-ordination Indexing</u> as a means of handling large or small collections of information in which complex logical relationships of thought may be found.

In the communication process, we must begin with the author, who in preparing his manuscript, first evolves a mental image (concept) of an action or a thing and then codes this concept in a language natural to him, committing this code (word) to paper. The thought processes of the author follow some logical pattern, which he attempts to communicate via the language by linking up the words representing his concepts in an orderly, co-ordinated fashion. The entire document may be represented by one (or more) logical structures describing the relationships of the principal concepts. The words used to describe these principal concepts are usually referred to as "keywords".

Although he seldom thinks of it in explicit terms, the searcher for information usually has in mind certain main concepts and a certain (incomplete) logical structure relating these concepts. In searching for information he is usually interested in finding out the answers to one or more of the following:

1. Has anyone linked up these particular concepts in the logical structure I have used?

2. If so, what results or observations derived from this co-ordination?

3. Assuming my structure is partial, what extensions of the logical structure have previous workers made?

4. Of what larger set of ideas is my argument a sub-set?

5. If my argument is highly generalized, what sub-sets exist?

The function of the indexer now becomes clear: He must so code the document for entry into the storage and retrieval system that both the concepts and the logical processes of the author may be operated upon in response to the thought processes of the searcher. It is apparent that this coding must largely circumvent the language problems mentioned earlier.

Absolutely basic to a successful concept co-ordination system, then, is some means of defining the words, or terms, used to describe concepts and of establishing the relationships among terms. This is referred to as control of language and vocabulary; the task is eased greatly if the environment is limited in some manner. In our case, we limit the environment to the pulp and paper industry, and then define the terms by specifying that the language shall be English, as written by engineers and scientists in North America. The final step of vocabulary control is accomplished by compilation of a thesaurus, or collection of words, in which elimination of synonymous terms is accomplished in a prescribed fashion and the relationships between terms are rigidly defined. By reference to this thesaurus, the indexer may code the thoughts of the author for entry to the information system using a consistent set of words, not necessarily the words the author originally employed in describing his concepts. The author, for example, may have used the term "methyl alcohol" for one of his principal concepts. Reference to the thesaurus shows that "methyl alcohol" is not an accepted term but that its synonym, "methanol", is. The indexer will therefore see that the concept is represented by the term, "methanol" and, in addition, by the generic term, "alcohol".

To meet the various needs just described, a thesaurus is essential; it should contain the following elements, when needed, for each acceptable term:

1. Scope notes, where the dictionary applies the word to two or more different concepts. For example, in a technical thesaurus it is necessary to enter the term referring to the most ancient physical science as MECHANICS / NOT PERSONNEL/!

2. List of synonyms - not accepted for entry in the system.

3. List of closely (horizontally) related terms - accepted for entry in system.

4. List of important subordinate concept terms.

5. Name of term designating the class of concepts to which subject term belongs.

In addition, terms not accepted for entry in the system must be interfiled in the alphabetic listing of terms, with cross reference to the appropriate accepted term.

Documents may be entered into a concept co-ordination indexing system in either of two standard ways - a sequential file in which the records are filed by accession number of the document and in which each record contains all the keyword codes (and modifiers, if any) pertaining to the document, or, an inverted file in which the records are filed by keyword and each record contains the accession numbers of all documents described by that keyword. The inverted file has the great advantage that only a small portion of the file need be searched for any given inquiry and that the logic of the search may be accomplished by appropriate manipulation of keyword combinations.

The form of the records in the active system (library cards, edgenotched cards, Termatrex cards, IBM cards, paper tape, magnetic tape, photographic image, etc.) and the techniques of entering and retrieving the data are of relatively minor importance, and will vary depending on the number of terms and documents in the system, as well as the nature and frequency of inquiries.

Experience has shown that, except for the very small collection, it is of critical importance that the establishment and maintenance of the system be the function of experienced technical workers and that these technically trained individuals do the actual programming of searches conducted in response to inquiries. By working closely with the originator of the inquiry, the technical information specialist is able to extract the maximum pertinent information with maximum efficiency.

PROGRESS ON IPC PROJECT

THESAURUS

Prior to the establishment of Project 2318, the Institute Information Retrieval Study Committee had decided to adopt the concept co-ordination system advanced by the American Institute of Chemical Engineers and to use the Chemical Engineering Thesaurus as a basic word-book. Since the experience of others clearly indicate that even experimental coding of documents should not be done without a thesaurus, the first item of business was the preparation of a pulp and paper supplement to the basic Chemical Engineering Thesaurus.

Drawing on word-lists from various sources, a first, rather abbreviated draft was prepared at IPC in January and forwarded to PPRIC. This draft was reconciled with a longer PPRIC word list in conference in February, after which PPRIC prepared an extensive second draft. After this second draft had been studied and criticized, the PPRIC staff prepared a third draft of what they call the Pulp Technology Thesaurus in July, followed by a Forestry Thesaurus in late August. Sufficient well-related terms are now on hand in these three collections to permit the beginning of coding (i.e. the work of attaching index terms.) However, the various Thesauri are nowhere near complete and will require modification and extension as the work proceeds.

CODING ABSTRACTS FOR RETRIEVAL EXPERIMENTS

Since September 1, both IPC and PPRIC have been active in the task of assigning keywords to abstracts from the current volume of the Abstract Bulletin (IPC).

It was suggested that in our particular environment, many inquiries would be concerned with publications of but a restricted viewpoint, say, mill experience, on one hand, or theoretical research, on another. It was, accordingly, decided to attach to the list of keywords describing a document, a type-of-publication keyword which would enable the inquirer to eliminate information from sources not pertinent to his viewpoint. These categories are:

RESEARCH, THEORY
RESEARCH, LABORATORY EXPERIMENT
PRODUCT DEVELOPMENT, LABORATORY
PRODUCT DEVELOPMENT, MILL SCALE
PROCESS DEVELOPMENT, LABORATORY AND PILOT PLANT
PROCESS DEVELOPMENT, MILL SCALE
PRODUCTION EXPERIENCE
DATA COMPILATION
REVIEW PAPER, THEORETICAL AND TECHNICAL
GENERAL TECHNICAL ARTICLE
NON-TECHNICAL REVIEW
PATENT, PRODUCT
PATENT, PROCESS
PATENT, N.E.C.

Reaction to this classification technique has been mixed; it is not certain how far it will be carried along with the experiment.

So far, there has been lack of agreement between the two institutions concerning the use of "links" and "roles" in applying keywords to documents, as recommended by the A.I.Ch.E. The attachment of role indicators to keywords is one way of preserving and indicating the context in which the corresponding word was used in a source document. Our committee believes that the problem with false drops will become severe with a large collection unless a system such as this is used. At the moment, the abstracts being coded at IPC all have role indicators attached.

It will eventually be necessary to translate this English-language keyword code to a more economical "language" for entry on punched cards. With this in mind a dictionary (or, more properly for now, a list) of all possible combinations of four alphabetic characters (from the Latin set) has been prepared on the 1620 computer at IPC.

OTHER EXPERIMENTS

In the absence of a significantly large set of properly coded documents, no extensive experimentation with search procedures has been carried out, although some programming of machine methods has been carried to the test-run stage at both institutions. One point seems quite clear from such early thought and activity with search configurations. With the large system we expect to have, it will be essential to provide for all the common types of logical operations used in information retrieval. It will not suffice to use the simple intersection of sets to retrieve related concepts; but search methods providing both for union of possibly desired sets and exclusion of undesired material will be necessary.

FUTURE WORK

Work is proceeding at both institutions on the task of applying keywords to abstracts. A meeting has been set up for October 17 in Montreal to reconcile possible differences in approach in this phase of the experiment. Following this meeting, completion of the keyword assignments will depend on the time available of the technical people involved.

After the keyword assignment has been completed, a list of the keywords actually employed will be prepared and a set of mnemonic four-letter codes will be arbitrarily assigned to these words. The information on the abstract and keyword sheets will then be transferred to punched cards in this manner:

For each keyword on each abstract a card containing the document number and the single keyword will be punched. The cards will be sorted according to keyword, producing an inverted file, and then the information concerning documents to which a given keyword applied compressed into the minimum number of unit records (cards for IPC, tape for PPRIC).

At this point, the analysis of search configurations can begin in earnest. A tentative time schedule for completion of the various above phases will be considered at the Montreal meeting.

PERSONNEL

All of the members of the Literature Retrieval Committee have contributed to the thinking and the progress of the project; these are Curtis Brown, Edgar Dickey, Richard Nelson and Jack Weiner. Lillian Roth has done much of the indexing to date and has contributed valuable comments. Others doing active indexing have been E. Dickey and the writer. John Bachhuber has prepared the dictionary of four-letter words and performed other computer-related chores.

DEVELOPMENT OF MECHANIZED METHODS FOR TECHNICAL INFORMATION RETRIEVAL

INTRODUCTION

Institute activity in the development of retrieval systems for technical information has had two main goals:

1. The development of a generalized system of retrieval applicable to collections of various sizes in the paper industry with full compatibility of coding and searching procedures between the various collections (sub-systems).

2. The development of a highly mechanized and highly efficient retrieval system for our large, centralized collection which speaks to the needs of our staff, students and member companies.

The elements of experimentation on this project include six major steps; namely,

1. Preparation of a pulp and paper supplement to the Chemical Engineering Thesaurus.

2. Coding a large sample of documents from A.B.I.P.C.

3. Checking this coding for different sources of variation.

4. Experimenting with search configurations.

5. Providing means for inclusion of various externally and internally generated documents in the file.

6. Developing effective techniques for processing questions from and answers to users of the system.

Also involved in this effort is considerable education on the part of the study group on the techniques tried and found useful (or not) by others - furthermore, there has been established close co-operation with the parallel program at the Pulp and Paper Research Institute of Canada.

PROBLEMS OF AN INFORMATION RETRIEVAL SYSTEM

Before embarking on as complicated a venture as this, one must have the goal clearly in mind. Even more generalized than the goals just cited is that of the Engineers Joint Council program, of which our activity may properly be considered a sub-set. Slightly paraphrased, their statement of objectives is: "To improve the EFFICIENCY of ENGINEERS and TECHNOLOGISTS in the function of OBTAINING PERTINENT INFORMATION from the CURRENT TECHNICAL LITERATURE." The problems faced in setting up an information retrieval system must be viewed with keywords of this statement in mind.

We must consider, for one thing, "Who is to be served?" The answer is scientists, engineers and technologists in the pulp and paper industry.

Next, "What do they want?" The answer, pertinent technical information from the current (this term needs some defining) literature.

"Why do they want it?" - To be more effective and efficient on the job.

In other words, we must focus on the information needs of a very special audience and must serve these needs in such a way as to save the individuals time and effort in their vocational activities.

In dealing with communication between any author and any reader (or, searcher for information) certain language problems always intervene. The four major classes of language problems which crop up in technical communications may be categorized as follows:

VIEWPOINT
MEANING
WORD ORDER
FAMILY RELATIONSHIPS

The problems of viewpoint and meaning are interrelated. A given term may have several different meanings regardless of viewpoint and requires use of a modifier for specificity. The term, "track", for example, may mean anything from pair of iron rails attached to cross-ties to a sequence of animal footprints in snow, soft earth, etc. Furthermore, a term may have various meanings depending on the viewpoint of the observer or writer. To a heating engineer, writing about oil burners, "oil" is a mixture of paraffinic and naphthenic hydrocarbons of a restricted range of density and viscosity obtained by the distillation of crude petroleum. To the wildcatter, oil is the crude itself; to the French chef, oil is something entirely different. Imagine the difficulties terms such as "chest", headbox", "furnish" present to the person unaquainted with the language and viewpoint of the papermaker. However, in a specialized information retrieval scheme, the viewpoint of the author and reader may be very close, so that it is possible to restrict greatly the possible range of meanings for a given term.

Now, the way in which a writer arranges the words in a sentence or thoughts in a paragraph establishes a certain meaning by context. Contrast two titles, "The Organization of a Republican Form of Government" and "The Form and Government of a Republican Organization". An individual concerned with Republican Party affairs would have no interest in the former article, yet he would surely retrieve it, among others, if it were indexed solely by keywords with no indication of context. To avoid excessive "false drops" of non-pertinent information in a mechanized system, some provision must be made for preserving the contextual relationships of terms defined by the author.

Finally, there is the problem of family relationships of words. Different words in the English language may be used to describe exactly the same concept. "Methanol" and "methyl alcohol" are synonyms; imagine the confusion which would arise if half the documents in a system relating to this concept were coded or indexed under "methanol" and the other half under "methyl alcohol". Synonyms must be eliminated from a retrieval system; but when this is done, there arises the more complex problem of <u>closely related</u> terms describing <u>slightly different</u> concepts. Consider "crushing", "milling" and "pulverizing"; these are quite properly distinct concepts and an author may quite properly use one or another of these terms to precisely designate both mode and degree of size reduction, according to the context of his paper. The <u>seeker</u> of information, however, may be uncertain about the <u>precise</u> mode and degree of size reduction about which he needs information; if so, he must be provided with a guide for exploring concepts closely related to that which originally comes to mind. Examination of related concepts, furthermore, must be both vertical and horizontal. The seeker of information on milling may wish merely to explore this concept plus those of crushing and pulverizing; on the other hand, he may decide that <u>all</u> related concepts need coverage, in which case the search should be directed to the class concept, "size reduction". Thus, a guide to vertical relationships is also needed.

CONCEPT CO-ORDINATION

An appreciation and analysis of the above problems has led various literature specialists to the development of the technique of <u>Concept Co-ordination Indexing</u> as a means of handling large or small collections of information in which complex logical relationships of thought may be found.

In the communication process, we must begin with the <u>author</u>, who, in preparing his manuscript, first evolves a mental image (<u>concept</u>) of an action or a thing and then codes this concept in a language natural to him, committing this code (<u>word</u>) to paper. The thought processes of the author follow some logical pattern, which he attempts to communicate via the language by linking up the words representing his concepts in an orderly, co-ordinated fashion. The entire document may be represented by one (or more) logical structures describing the relationships of the principal concepts. The words used to describe these principal concepts are usually referred to as "keywords".

Although he seldom thinks of it in explicit terms, the <u>searcher</u> for information usually has in mind certain main concepts and a certain (incomplete) logical structure relating these concepts. In searching for information he is usually interested in finding out the answers to one or more of the following:

1. Has <u>anyone</u> linked up these particular concepts in the logical structure I have used?

2. If so, what results or observations derived from this co-ordination?

3. Assuming my structure is partial, what extensions of the logical structure have previous workers made?

4. Of what larger set of ideas is my argument a sub-set?

5. If my argument is highly generalized, what sub-sets exist?

The function of the <u>indexer</u> now becomes clear: He must so code the document for entry into the storage and retrieval system that both the <u>concepts</u> and the <u>logical processes</u> of the author may be operated upon in response to the thought processes of the searcher. It is apparent that this coding must largely circumvent the language problems mentioned earlier

Absolutely basic to a successful concept co-ordination system, then, is some means of defining the words, or terms, used to describe concepts and of establishing the relationships among terms. This is referred to as control of language and vocabulary; the task is eased greatly if the environment is limited in some manner. In our case, we limit the environment to the pulp and paper industry, and then define the terms by specifying that the language shall be English, as written by engineers and scientists in North America. The final step of vocabulary control is accomplished by compilation of a thesaurus, or collection of words, in which elimination of synonymous terms is accomplished in a prescribed fashion and the relationships between terms are rigidly defined. By reference to this thesaurus, the indexer may code the thoughts of the author for entry to the information system using a consistent set of words, not necessarily the words the author originally employed in describing his concepts. The author, for example, may have used the term "methyl alcohol" for one of his principal concepts. Reference to the thesaurus shows that "methyl alcohol" is not an accepted term but that its synonym, "methanol" is. The indexer will therefore see that the concept is represented by the term, "methanol" and, in addition, by the generic term, "alcohol".

To meet the various needs just described, a thesaurus is essential: it should contain the following elements, when needed, for each acceptable term:

1. Scope notes, where the dictionary applies the word to two or more different concepts. For example, in a technical thesaurus it is necessary to enter the term referring to the most ancient physical science as MECHANICS / NOT PERSONNEL/.

2. List of synonyms - not accepted for entry in the system.

3. List of closely (horizontally) related terms - accepted for entry in system.

4. List of important sub [illegible] concept terms.

5. Name of term designating the class of concepts to which subject term belongs.

In addition, terms not accepted for entry in the system must be interfiled in the alphabetical listing of terms with cross reference to the appropriate accepted term.

Documents may be [illegible] coordination indexing system in either of two standard [illegible] in which the [illegible] are filed by accession number of the document and in which each record contains all the keyword codes (and modifiers, if any) pertaining to the document, or, an inverted file in which the records are filed by keyword and each record contains the accession numbers of all documents described by that keyword. The inverted file has the great advantage that only a small portion of the file need be searched for any given inquiry and that the logic of the search may be accomplished by appropriate manipulation of keyword combinations.

The form of the records in the active system (library cards, edge-notched cards, Termatrex cards, IBM cards, paper tape, magnetic tape, photographic image, etc.) and the techniques of entering and retrieving the data are of relatively minor importance, and will vary depending on the number of terms and documents in the system, as well as the nature and frequency of inquiries.

Experience has shown that, except for the very small collection, it is of critical importance that the establishment and maintenance of the system be the function of experienced technical workers and that these technically trained individuals do the actual programming of searches conducted in response to inquiries. By working closely with the originator of the inquiry, the technical information specialist is able to extract the maximum pertinent information with maximum efficiency.

PROGRESS ON IPC PROJECT

THESAURUS

Prior to the establishment of Project 2318, the Institute Information Retrieval Study Committee had decided to adopt the concept co-ordination system advanced by the American Institute of Chemical Engineers and to use the Chemical Engineering Thesaurus as a basic word-book. Since the experience of others clearly indicated that even experimental coding of documents should not be done without a Thesaurus, the first item of business was the preparation of a pulp and paper supplement to the basic Chemical Engineering Thesaurus.

Drawing on word-lists from various sources, a first, rather abbreviated draft was prepared at IPC in January and forwarded to PPRIC. This draft was reconciled with a longer PPRIC word list in conference in February, after which PPRIC prepared an extensive second draft. After this second draft had been studied and criticized, the PPRIC staff prepared a third draft of what they call the Pulp Technology Thesaurus in July, followed by a Forestry Thesaurus in late August. Sufficient well-related terms are now on hand in these three collections to permit the beginning of coding (i.e. the work of attaching index terms.) However, the various Thesauri are nowhere near complete and will require modification and extension as the work proceeds.

CODING ABSTRACTS FOR RETRIEVAL EXPERIMENTS

Since September 1, both IPC and PPRIC have been active in the task of assigning keywords to abstracts from the current volume of the Abstract Bulletin (IPC).

It was suggested by PPRIC personnel that in our particular environment, many inquiries would be concerned with publications of but a restricted viewpoint, say, mill experience, on one hand, or theoretical research, on another. It was, accordingly, decided to attach to the list of keywords describing a document, a type-of-publication keyword which would enable the inquirer to eliminate information from sources not pertinent to his viewpoint. These categories are:

RESEARCH, THEORY
RESEARCH, LABORATORY EXPERIMENT

PRODUCT DEVELOPMENT, LABORATORY
PRODUCT DEVELOPMENT, MILL SCALE
PROCESS DEVELOPMENT, LABORATORY AND PILOT PLANT
PROCESS DEVELOPMENT, MILL SCALE
PRODUCTION EXPERIENCE
DATA COMPILATION
REVIEW PAPER, THEORETICAL AND TECHNICAL
GENERAL TECHNICAL ARTICLE
NON-TECHNICAL REVIEW
PATENT, PRODUCT
PATENT, PROCESS
PATENT, N.E.C.

So far, there has been lack of agreement between the two institutions concerning the use of "links" and "roles" in applying keywords to documents, as recommended by the A.I.Ch.E. The attachment of role indicators to keywords is one way of preserving and indicating the context in which the corresponding word was used in a source document. Our committee believes that the problem with false drops will become severe with a large collection unless a system such as this is used. At the moment, the abstracts being coded at IPC all have role indicators attached.

It will eventually be necessary to translate this English-language keyword code to a more economical "language" for entry on punched cards. With this in mind a dictionary (or, more properly for now, a list) of all possible combinations of four alphabetic characters (from the Latin set) has been prepared on the 1620 computer at IPC.

OTHER EXPERIMENTS

In the absence of a significantly large set of properly coded documents, no extensive experimentation with search procedures has been carried out, although some programming of machine methods has been carried to the test-run stage at both institutions. One point seems quite clear from such early thought and activity with search configurations . With the large system we expect to have, it will be essential to provide for all the common types of logical operations used in information retrieval. It will not suffice to use the simple intersection of sets to retrieve related concepts; but search methods providing both for union of possibly desired sets and exclusion of undesired material will be necessary.

FUTURE WORK

Work is proceeding at both institutions on the task of applying keywords to abstracts. A meeting has been set up for October 17 in Montreal to reconcile possible differences in approach in this phase of the experiment. Following this meeting, completion of the keyword assignments will depend on the time available of the technical people involved.

After the keyword assignment has been completed, a list of the keywords actually employed will be prepared and a set of mnemonic four-letter codes will be arbitrarily assigned to these words. The information on the abstract and keyword sheets will then be transferred to punched cards in this manner: For each keyword on each abstract a card containing the document number and the single keyword will be punched. The cards will be sorted according to keyword, producing an inverted file, and then the information concerning documents to which a given keyword applied compressed into the minimum number of unit records (cards for IPC, tape for PPRIC).

At this point, the analysis of search configurations can begin in earnest. A tentative time schedule for completion of the various above phases will be considered at the Montreal meeting.

END