User's Manual for the SMART Information Retrieval System

Joel Zumoff

March 1971
(Revised July 1971)

Technical Report
No. 71-95
March 1971

Department of Computer Science
Upson Hall
Cornell University
Ithaca, New York 14850

Acknowledgements

The author wishes to acknowledge the work of the following people who have contributed to the SMART project.

Mrs. Donna Williamson prepared the preliminary version of the SMART user's manual. This technical report is an extensive rewrite and update of that manual. Mr. Robert Williamson designed and programmed a large part of the Cornell implementation of the SMART system. Mrs. Marcia Kerchner provided many suggestions for modifications and improvements to the system. Mr. Steven Worona and Mr. Robert Crawford helped in debugging the system and adding modifications to it. Mrs. Barbara Galaska read the manuscript and suggested improvements in the wording. Mrs. Eileen Gudat and Mrs. Rosalind Pasquali did an excellent job typing the manuscript.

Finally, the author wishes to thank the SMART project director, Professor Gerard Salton, for his support of this work.

Table of Contents

Figures

User's Manual
for the
SMART
Information Retrieval
System

I. Introduction

The Cornell SMART information retrieval system operates as a private program stored on disk. When the SMART program is invoked it interrogates control cards in the input stream to ascertain which SMART routines are desired. Control is transferred to those routines in the requested sequence.

Each of the SMART routines is controlled by a set of cards containing the name of the routine and a set of parameters to control the processing of the routine. Most of the parameters have default values and do not need to be explicitly given on the control cards unless non-standard values are required for the particular application under consideration.

The parameters required for each SMART routine are described in detail in the descriptions given in Section IV. Sample detailed card deck setups are shown with each description.

Two important characteristics of the card deck setup for a SMART run should be noted. First, after invoking SMART, control is transferred to the SMART system programs, and the only valid input cards are SMART control cards specified in the routine descriptions. Second, no fixed calling sequence exists for the SMART routines other than a required initial call to RESTART (to open the desired collections data set) and a final call to STOP (to end the run).

II. SMART Collections

All collections used in SMART runs must be stored on the collections data set being used. There are two different collections data sets. Data set 1 is a private SMART disk and data set 2 is an on-line disk.

Figure 1 gives the names of the collections, the number of items in each collection, and the type of each collection, on the two data sets. Each name consists of two eight character fields. Blank spaces in the name fields are an intrinsic part of the name. There are four text types. Abstracts refer to collections formed from the abstracts of documents. Queries are collections of questions. Centroids are collections of centroid trees. Inverted are collections inverted on concepts. There are three types of dictionaries. Thesaurus dictionaries group several terms per concept class, word form dictionaries ignore plural endings (-s), and word stem (null) dictionaries ignore endings from a given suffix list.

III. Job Control Language and SMART Control Cards

Special job control language (JCL) is needed to run the SMART system on the IBM System/360 Model 65 at Cornell. The JCL needed is filed in several Card Data Sets (CDS's) under account MNQ and should be inserted into the job input stream as indicated below. The files that may be used and their functions are:

MNQ.SMART - contains JCL needed to use the rest of the SMART JCL and must be included in every job as the first job step. It defines the data sets used by the other SMART JCL files.

MNQ.READCRDS - reads SMART control information from cards or CDS's.

MNQ.PNCHCRDS - punches output that has been prepared for punching as a result of a SMART run (routines COLCDS and CONCDS).

MNQ.SMARTRUN - contains the JCL needed to execute SMART. It defines the input and output data sets. It must be immediately preceded by the EXEC statement which invokes SMART.
Data cards for the run must have been previously read in using READCRDS.

MNQ.SMARTGO - similar to SMARTRUN except data cards for the SMART run

Sequence Number	Collection Name		Number of Items	Text Type	Dictionary Type
1	CRN4S	QUESTS	155	queries	word form
2	CRN4S	DOCS	424	abstracts	word form
3	CRN4S	QUESTS30	30	queries	word form
4	ADIABTH	QUESTS	35	queries	thesaurus
5	ADIABTH	DOCS	82	abstracts	thesaurus
6	CRN2TH	DOCS	200	abstracts	thesaurus
7	CRN2TH	QUESTS	42	queries	thesaurus
8	REW-CRN4	S TREE	78	centroids	word form
9	CRN14NUL	DOCS	1400	abstracts	word stem
10	CRN14NUL	QUESTS	225	queries	word stem
11	CRN2NUL	QUESTS	42	queries	word stem
12	CRN2NUL	DOCS	200	abstracts	word stem
13	CRN14S	DOCS	1400	abstracts	word form
14	CRN14S	QUESTS	225	queries	word form
15	CRN4NUL	DOCS	424	abstracts	word stem
16	CRN4NUL	QUESTS	155	queries	word stem
17	CRN2S	QUESTS	42	queries	word form
18	CRN2S	DOCS	200	abstracts	word form
19	CRN4S	TREE1	82	centroids	word form
20	CRN14NUL	INVERTED	4953	inverted	word stem
21	CRN14TH	DOCS	1400	abstracts	thesaurus
22	CRN4NUL	INVERTED	2654	inverted	word stem

SMART Disk (Data Set 1)

Sequence Number	Collection Name		Number of Items	Text Type	Dictionary Type
1	ADIABTH	QUESTS	35	queries	thesaurus
2	ADIABTH	DOCS	82	abstracts	thesaurus
3	ADIABTH	TREE1	13	centroids	thesaurus
4	ADINUL	DOCS	82	abstracts	word stem
5	ADINUL	QUESTS	35	queries	word stem

On-line Disk (Data Set 2)

<u>The SMART Collections</u>

Figure 1

are not read in by READCRDS; they are simply included in the input stream. This procedure requires more input and output operations but needs a smaller region size and should be used with small amounts of input data.

MNQ.SMARTER - defines the data sets used by SMART.

MNQ.FT01 - defines the SMART disk.

MNQ.FT90 - defines an auxiliary work space.

Figure 2 shows the order of inclusion in the card deck of the cards to call out the SMART JCL files. Examples are shown to illustrate the use of the two "GO" files: SMARTRUN (for large amounts of input data) and SMARTGO (for small amounts of input data). Data refers to both SMART control cards and collection cards.

The region size on the '/*MNQ.SMART' and '/*MNQ.READCRDS' cards <u>must</u> be the same as the region size on the 'EXEC' card. The <u>first card</u> of the <SMART run cards> <u>must</u> be a RESTART card to designate the collections data set to be used. This card has the word 'RESTART' in columns 1-7 and a '1' in column 10 if the SMART disk is desired or a '2' in column 10 if the on-line disk is wanted. The last of the <SMART run cards> must be a STOP card with 'STOP' in columns 1-4.

If the SMART disk is desired (a '1' on the RESTART card) the private SMART disk must be mounted by inserting the card

/*SETUP UNIT=2314,ID=D00006,TEXT='SMART'

immediately after the '/*MAIN' card. In addition, the card '/*MNQ.FT01' must be inserted immediately after the '/*MNQ.SMARTER' card. It is usually better to use the on-line disk rather than the SMART disk if at all possible since there is an additional charge for mounting the SMART disk.

The user may wish to use collections other than those listed in Figure 1, or he may wish to modify some of these collections. If this is the case, then the

Use of SMARTRUN

```
//JOB                                                CLASS=K,REGION=XXXK
/*MAIN LINES=(W),CARDS=(Z)
/*SETUP UNIT=2314,ID=D00006,TEXT='SMART'   (include to set up the SMART disk)
/*MNQ.SMART REGION=XXXK
/*MNQ.READCRDS REGION=XXXK
  <SMART run cards>
/*
//SMART EXEC PGM=SMART,REGION=XXXK,TIME=YY
/*MNQ.SMARTRUN
/*MNQ.SMARTER
/*MNQ.FT01    (include if SMART disk is set up)
/*MNQ.FT90    (may not be needed)
/*MNQ.PNCHCRDS      (may not be needed)
/*CREATE cdsname FROM FT07F001  (include to file punched output)
/*
```

Use of SMARTGO

```
//JOB                                                CLASS=K;REGION=XXXK
/*MAIN LINES=(W),CARDS=(Z)
/*SETUP UNIT=2314,ID=D00006,TEXT='SMART'  (include to set up the SMART disk)
/*MNQ.SMART REGION=XXXK
//SMART EXEC PGM=SMART,REGION=XXXK,TIME=YY
/*MNQ.SMARTGO
/*MNQ.SMARTER
/*MNQ.FT01    (include if SMART disk is set up)
/*MNQ.FT90    (may not be needed)
  <SMART run cards>
/*MNQ.PNCHCRDS  (may not be needed)
/*CREATE cdsname FROM FT07F001   (include to file punched output)
/*
```

Use of SMART JCL

Figure 2

new collections must be placed temporarily on one of the collections data sets by using either routine CRDCOL or card decks previously punched by routines COLCDS or CONCDS (see the descriptions of these routines in Section IV for details). Collections placed on a collections data set by these routines can only be accessed for the duration of the run. They are deleted at the end of the job. Thus they must be restored every time a run is made.

If collections are large the standard auxiliary files used by SMART may not be large enough. In this case, an alternate work space must be defined by inserting the card '/*MNQ.FT90' after the '/*MNQ.SMARTER' card and by setting the variable "AUXSET" to 90 on the appropriate SMART control card (see Section IV for routines CDSCOL, COLAUX, CRDCEN, DCLSTR, INVERT, RECODE, and SUBCOL).

The card deck structures as given will produce punched card output as a result of punching operations by routines COLCDS and CONCDS. If there is no punched output from these routines, do not include the '/*MNQ.PNCHCRDS' card. If you would prefer to have the COLCDS and CONCDS output stored on a CDS instead of being punched on cards, include the card '/*CREATE cdsname FROM FT07F001' after the '/*MNQ.PNCHCRDS' card . This procedure is recommended to avoid getting invalid cards from an incorrect run. The CDS may be listed to check the validity of the cards, which may then be punched directly from the CDS. When CDS's are used in later runs, they should always be copied 'NOSEQ'.

Other important considerations in setting up the runs are the specifications of CLASS, REGION, TIME, LINES, and CARDS. These parameters should be set reasonably accurately. Too small a specification will result in either job cancellation or incomplete runs. Too large a limit on the region causes heavy extra charges (these charges do not go up linearly). Therefore it is recommended that the region be estimated as accurately as possible, especially on long runs, and TIME, LINES, and CARDS should be set to expected amounts plus sizable safety margins. The figures given below are merely estimates that vary with the amount of work space and collection sizes in each run.

Estimates for LINES are as follows:

full SEARCH runs - 2 pages per query
cluster SEARCH runs - 3 pages per query per iteration
other routines - the system defaults are usually sufficient.

Estimates for CARDS are as follows:

CRDCOL, COLCDS, CONCDS item collection - 10 cards per item
COLCDS result collection - 5 cards per query per iteration

Estimates for TIME are as follows:

SEARCH, CLUSTER, and DCLSTR are usually the only runs needing more that the standard default. The time depends on the collection size. For SEARCH runs approximately 0.007 seconds are required for each correlation. Therefore, the time for 2 iterations on the ADI collection is (0.007 * 35 queries * 82 documents * 2 iterations) or about 40 seconds. The Cranfield 200 collection takes about 1 minute per iteration, and the Cranfield 424 collection takes about 7½ minutes per iteration. The Cranfield 1400 collection takes about 33 minutes per iteration.

Estimates for REGION are as follows:

SEARCH, AVERAGE, DCLSTR, CLUSTER, and CRDCEN - 280K
all other runs - 240K
Check the region actually used when a run is made and adjust the region size accordingly for subsequent similar runs.

The final consideration is the CLASS on the job card. Only classes J, K, and L should be used to take advantage of the facilities of the SMART JCL. Most jobs will be class K (balanced I/O and computation). Long runs may be class L (compute bound).

If runs are submitted on a "CRBE" slow speed terminal, the card '/*FORMAT TP,DDNAME=FT02F001,DEST=destination' should be inserted before the '/*MAIN' card to avoid punching of auxiliary cards generated by the SMART JCL.

IV. SMART Routines

The following pages give descriptions and control card formats for the routines of the SMART system. The routines are presented in alphabetical order. All field specifications must be given exactly as indicated. Integers must always be right justified in the fields. Literals must be given exactly as indicated. Blank spaces are important in literal fields. If it is stipulated that a list must end with a blank (or zero) field and the last actual element of the list is in the last field of a card, then a blank card must follow the cards of the list to supply the blank or zero field.

In the following descriptions, names of SMART routines are given in all capital letters (e.g., SEARCH). Names of variables for which values are substituted are enclosed in double quotes (e.g., "AUXSET"). Values punched on cards are given in single quotes (e.g., 'RELEVANT').

Routine AVERAGE

Routine AVERAGE prints and plots average recall level and document level values for selected members of up to eight result collections from a collections data set (collections produced by SEARCH or RESCOL). In addition, the selected members of each collection are written on auxiliary data sets for use by VERIFY (if desired). The following input cards are necessary to use the routine:

1. The AVERAGE call out card containing the following information:

Columns	Description
1-7	'AVERAGE'
11-18	If this field contains 'COLIDS ' there is a list of collection names defining the collections to be used following this card. Only these collections will be used. If this field contains 'NOCOLIDS' the routine will use all the collection names of the result collections created by the last previous call (in this run) to SEARCH or RESCOL. A blank field is equivalent to 'NOCOLIDS'.
21-28	This field controls the selection of members in each collection. Three possibilities exist: (1) ' ' - All members in each collection will be used. (2) 'SAMPAIRS' - One set of cards specifying the members follows the list of collection names. The same members are used from all collections. (3) 'DIFPAIRS' - A set of cards specifying the members for each collection follows the entire list of collection names.
39-40	This field specifies the number of printed output copies desired. If blank, one is assumed.

Routine [illegible] 50,52 — The first symbol to be used for plotting recall level and document level graphs respectively. Successive symbols are taken sequentially from the following table of symbols: '0' - '9', 'A' - 'Z'. If the symbol presented is blank or a symbol not in the table, '0' is assumed.

61-67 — This field specifies how to assign the unrecovered relevant documents.
'UNIFORM' distributes them uniformly throughout the remainder of the unrecovered documents.
'WORST ' assigns them to the lowest possible ranks.
The default is 'WORST '.

71-78 — This field specifies at what rank the precision is held constant.
'=LASTREL' holds it constant after the last recovered relevant document.
'=LASTREC' holds it constant after the last recovered document if all relevant have already been recovered.
The default is '=LASTREL'.

2. If 'COLIDS ' is specified, cards must be added here specifying the names of the collections to be used. Up to eight collections can be used. Five names may appear on a card in columns 1-16, 17-32, 33-48, 49-64, 65-80. A blank field <u>must</u> terminate the list of collection names.

3. If 'SAMPAIRS' or 'DIFPAIRS' is specified, the next cards are read with the format (A8, 2X, 14I5) and can contain up to 7 valid pairs of number Each pair of numbers consists of a "from" key and a "to" key which indicate an inclusive range of sequence keys for the items to be averaged. A particular pair can be any place in the list (ordering is not necess and can overlap, or be overlapped by, any other pair(s). A pair indicates only one item key when the "to" key is either the same as the "from" key, zero, or blank. Columns 1-8 of the first card (of each set if 'DIFPAIRS' is specified) must contain either '   DO' if the keys included by each pair <u>are</u> to be averaged, or 'DON'T DO' if the

inclusive keys are not to be averaged. Successive cards must contain blanks in columns 1-8. Each set is terminated by a blank "from" key.

4. If routine VERIFY is not to be called after routine AVERAGE, a blank card must be inserted at this point. Otherwise insert the cards to call out the VERIFY routine. Routine VERIFY should not be called if only one collection is processed by AVERAGE.

Example of Use of AVERAGE

Assume there are two result collections, "RESULT FULL " and "RESULT FLUID " filed in CDS "RESULT". (Methods of producing this file are given in the descriptions of SEARCH and COLCDS). We want to produce separate averages of each collection (collections are never averaged together). For "RESULT FULL " we want to use the results of queries 1-10, 15, and 23-32 while for "RESULT FLUID " we want to use all queries except 11-19. Verification is not wanted. The deck setup is as follows:

```
//JOB                                                CLASS=K,REGION=260K
/*MNQ.SMART REGION=260K
/*MNQ.READCRDS REGION=260K
RESTART  2
/*COPY RESULT NOSEQ
AVERAGE    COLIDS     DIFPAIRS
RESULT  FULL     RESULT  FLUID
      DO      1   10   15   15   23   32
DON'T DO     11   19
<blank card>
STOP
/*
//SMART EXEC PGM=SMART,REGION=260K
/*MNQ.SMARTRUN
/*MNQ.SMARTER
/*
```

Routine CDSCOL

Routine CDSCOL places a collection punched by COLCDS on an auxiliary data set and on the collections data set being used for use by other routines. The CDSCOL cards are punched in hexadecimal (non-printable) format by COLCDS and normally do not require altering. The first card of the deck punched by COLCDS is the CDSCOL callout card and has a format as follows:

Columns	Description
1-6	'CDSCOL'
13-14	"AUXSET" - left blank to default to data set 50 in the CDSCOL run.
21-24	'ITEM' if you want one line printed for each item processed, blank if this printing is to be suppressed. COLCDS leaves this field blank.
78-80	'100' - the sequence number of the first card.

If this collection is large, it may not fit on the data set 50 when it is used. In this case, set "AUXSET" to 90 by punching '90' in columns 13-14 of the CDSCOL callout card and include the card '/*MNQ.FT90' after the '/*MNQ.SMARTER' card.

If you want the number of bytes in each item, punch 'ITEM' in columns 21-24. In any case, the sequence number of the last item processed is given.

Example of Use of CDSCOL

Assume a collection was punched by COLCDS and stored in CDS "ADIDOCS". To place the collection on auxiliary data set 90, the deck setup is as follows:

```
//JOB                                              CLASS=K,REGION=220K
/*MNQ.SMART REGION=220K
/*MNQ.READCRDS REGION=220K
RESTART  2
CDSCOL      90
/*COPY ADIDOCS NOSEQ 2-END
<SMART run cards to use the collection>
STOP
/*
//SMART EXEC PGM=SMART,REGION=220K
/*MNQ.SMARTRUN
/*MNQ.SMARTER
/*MNQ.FT90
/*
```

Routine CLUSTER

Routine CLUSTER generates a centroid collection of single-level clusters using a modified version of Rocchio's clustering algorithm. The input cards are as follows:

1. The CLUSTER callout card containing the following:

Columns	Description
1-7	'CLUSTER'
11-26	The collection name of the document collection to be clustered.
31-46	A collection name for the centroid collection that will be produced from the clustering process.

2. A card containing parameter information:

Columns	Description
16-20	'BLEND' - if blending is to be done. If blank, no blending is done, in which case loose documents are completely lost. Therefore blending is recommended.
26-30	An integer indicating the first document to be checked for use as a root. If blank, the first document in the collection is used.
31-35	The number of documents to be batched for checking as possible cluster roots. If blank, an optimum value is used.
36-40	An auxiliary data set for the unblended clusters. If blank, 46 is used.
41-45	An auxiliary data set for the blended clusters. If blank, 47 is used. If the clusters will be large, use data sets 88 and 89 and include the DD cards as in the example for INVERT on page 46.

3. An 80-column title for the centroid collection being defined.

4. A card containing CLUSTER control parameters.

Columns	Description
1-10	The number of documents that must correlate above "RHOMAX" with any given document for that given document to be a root.
11-20	"RHOMAX" — the larger correlation cutoff value. This number must be punched with a decimal point.
21-30	The number of documents that must correlate above "RHOMIN" with a document for that document to be a root.
31-40	"RHOMIN" — The smaller correlation cutoff value. This number must be punched with a decimal point.
41-50	The minimum number of documents per cluster.
51-60	The maximum number of documents per cluster.

5. A card containing normalization parameters.

Columns	Description
1-8	Let this field be "XXXXYYYY". If "XXXX" is 'NEG.' negative weights are permitted; otherwise only positive weights will be kept after definition. "YYYY" can be either ' ', 'ABS.', or 'AVE.'. If "YYYY" is blank, only concepts with weight zero are deleted from the new query. (This obviously does not change correlations.) If "YYYY" is 'ABS.' then all concepts with absolute weight less than "PERDRP" are deleted. If "YYYY" is 'AVE.' then all concepts with absolute weight less than ("PERDRP" * the sum of absolute weights)/(100 * the number of unique concepts) are deleted. The former method is used to delete weights less than a specific value. The latter method permits dropping all weights less than a certain percentage of the average weight.

11-20 "PERDRP" — this number is a percentage (like 25.0) or weight and must include a decimal point.

21-26 If this field contains the word 'COSINE', all weights in a given vector are normalized according to the cosine correlation prior to being added to the composite for the new query. This makes all documents used have the same weight regardless of length. This is accomplished by multiplying each weight by "MULT" and dividing by the square root of the sum of squared weights of the vector being added. To prevent weights from disappearing (due to integer arithmetic), make "MULT" large when using this feature. If this field contains the word 'LINEAR', normalization is accomplished by dividing by the sum of absolute values of all weights in the vector being added. If blank, no normalization will be done.

31-40 The minimum correlation a document must have in order to be kept when correlating documents and centroids. If punched, the field must have a decimal point. A blank field is equivalent to 10^{-75}.

51-58 The type of correlation to be used. If blank, the 'COSINE ' correlation is used.

61-64 "MULT" — the multiplier for every document. If blank, 1 is used as the multiplier.

Example of Use of CLUSTER

Assume we wish to cluster the "CRN2S   DOCS   " collection (from the SMART disk) using Rocchio's algorithm. We will punch out the clustered collection. A sample set of parameters is given by the following deck setup.

```
//JOB                                                         CLASS=L,REGION=280K
/*MAIN CARDS=(20)
/*SETUP UNIT=2314,ID=D00006,TEXT='SMART'
/*MNQ.SMART REGION=280K
//SMART EXEC PGM=SMART,REGION=280K,TIME=10
/*MNQ.SMARTGO
/*MNQ.SMARTER
/*MNQ.FT01
RESTART  1
CLUSTER   CRN2S   DOCS          CRN2SDOC CLUSTER
              BLEND                  5
  CLUSTER COLLECTION FOR THE CRN2S   DOCS      COLLECTION
         2       0.50         10       0.10          5         50
    AVE.         50.5                  0.10
COLCDS    CRN2SDOC CLUSTER
STOP
/*MNQ.PNCHCRDS
/*
```

Routine COLAUX

Routine COLAUX copies all of the items of a specified collection from the collections data set being used onto a temporary auxiliary data set. The collection can only be used during the run, as it is lost at the end of the run. This routine may be used to set up a collection for modification (see CRDCOL). To copy a collection that is not permanently on the collections data set onto an auxiliary data set, the collection should be temporarily placed on the collections data set by CRDCOL, and then copied using COLAUX. Only one card is needed to call COLAUX.

Columns	Description
1-6	'COLAUX'
11-26	The collection name of the collection to be copied. This name must be exactly the same as the name on the data set from which it is to be copied.
31-32	"AUXSET" — the logical unit number of the data set which is to received the copied collection. If this field is blank, "AUXSET" is set to 50. If the collection being copied has more than 200 items, or has vectors of longer than 500 concepts, "AUXSET" must be set to '90'. In addition a '/*MNQ.FT90' card must be inserted after the '/*MNQ.SMARTER' card.

Example of Use of COLAUX

Assume we wish to copy the "CRN4NUL DOCS " collection from the SMART disk onto an auxiliary data set so that the collection may be modified by a call to CRDCOL. Since the collection contains 424 items, we must use data set 90. The deck setup is as follows:

```
//JOB                                                  CLASS=L,REGION=220K
/*MAIN CARDS=(10)
/*SETUP UNIT=2314,ID=D00006,TEXT='SMART'
/*MNQ.SMART REGION=220K
//SMART EXEC PGM=SMART,REGION=220K
/*MNQ.SMARTGO
/*MNQ.SMARTER
/*MNQ.FT01
/*MNQ.FT90
RESTART  1
COLAUX    CRN4NUL DOCS        90
 <cards to perform the merging operation; see CRDCOL>
STOP
/*
```

Routine COLCDS

Routine COLCDS punches out the contents of a document, query, centroid, inverted, or search result collection, not including the description cards, in a form suitable for punching or storing on a Card Data Set (CDS). These cards are punched in hexadecimal format and are not printable. The last eight columns however contain a printable sequence number (in increments of 100). COLCDS is the fastest way of saving and restoring a collection. If the output of COLCDS resides on CDS "XYZ", the card '/*COPY XYZ NOSEQ' will place the collection onto the collections data set being used. Essentially, COLCDS punches out a deck which comprises a CDSCOL run. Only one card is needed to call COLCDS.

Columns	Description
1-6	'COLCDS'
11-26	The collection name of the collection to be punched from the collections data set. This is either a permanent collection or one placed on the data set by the present SMART run.
33-34	'NO' if you do not want a listing of the card numbers on which each item is located, blank if you do want this information.

To file the punched output on CDS "XYZ", the last card in the deck must be '/*CREATE XYZ FROM FT07F001'. Care must be taken not to create too large a CDS (more cards than allotted to the account). If the output is possibly too large to store, the deck should be punched instead of stored. In either case, the card '/*MNQ.PNCHCRDS' must follow the SMART "GO" step. A '/*MAIN' card may be needed to punch or file the cards. If the collection being punched has more than 200 items, or has vectors of longer than 500 concepts, auxiliary data set 90 must be specified when the punched results are used. This is done by inserting a '/*MNQ.FT90' card after the '/*MNQ.SMARTER' card and by punching the number 90 in columns 13-14 of the CDSCOL callout card (the first card punched by COLCDS).

Example of Use of COLCDS

Assume we wish to punch out the collection "ADIABTH DOCS ". Instead of cards, we want to store the card images in CDS "ADIDOCS". The deck setup is as follows:

```
//JOB                                                  CLASS=K,REGION=220K
/*MAIN CARDS=(10)
/*MNQ.SMART REGION=220K
//SMART EXEC PGM=SMART,REGION=220K
/*MNQ.SMARTGO
/*MNQ.SMARTER
RESTART  2
COLCDS     ADIABTH DOCS            NO
 STOP
/*MNQ.PNCHCRDS
/*CREATE ADIDOCS FROM FT07F001
/*
```

Routine COMORN

Routine COMORN determines whether or not the concepts in a document collection are common concepts depending on a statistical test associated with the distribution of the concepts in the relevant documents of the associated query collection. This routine uses inverted collections of the document and query collections. The document collection should be inverted on concepts, and the query collection should be inverted on relevancy decisions. (See routine INVERT to set up the inverted collections.)

The test determines the probability that each concept comes from a uniform distribution. The user determines the cutoff value which stipulates the amount of uniformity desired. If the concept has a probability of being uniform which exceeds the cutoff value, it is considered to be a common concept.

This routine can punch out a deck of cards which comprises the transfer mapping table for routine RECODE. All those concepts which are judged common (uniform) by COMORN would be mapped into zero and hence dropped from the collection produced by RECODE.

The control cards are as follows:

1. COMORN callout card:

Columns	Description
1-6	'COMORN'
11-26	The collection name of the inverted (on concepts) document collection.
31-46	The collection name of the inverted (on relevancy decisions) query collection.
51-55	The probability cutoff value. This field should be punched with a decimal point and should be a number between 0.0 and 1.0. As this value is increased, the number of concepts judged common (i.e., having

a uniform distribution) decreases.

56-60 The logical unit number on which cards for RECODE will be punched. If no cards are wanted, leave the field blank. To place the cards on the standard punch file, code '   7'. If a different file is wanted for these cards, a DD card must be supplied after the '/*MNQ.SMARTER' card (see the example where file 20 is used). Do not include the '/*MNQ.PNCHCRDS' card for <u>this</u> output <u>unless</u> '   7' is used as the punch file. If any file except 7 is used, the cards <u>must</u> be filed in a CDS. They will not be directly punched.

61-65 The number of documents in the original document collection from which the inverted collection was produced

66-70 "PRINT" - results are given every "PRINT" queries. If "PRINT" is greater than the number of queries in the original query collection, then only one line per concept is given (at the conclusion of the testing) giving summary information for each concept. If "PRINT" is less than the negative of the number of queries, then one line per uniform (common) concept is given (at the conclusion of the testing).

71-75 The number of queries in the original query collection from which the inverted collection was produced.

2. An 80 column title card to be printed at the top of every page of the COMORN output.

3. Cards giving the ranges of concept numbers to test for uniformity. These numbers are given in pairs (first and last number in each range) in format (5(2I5)), 5 pairs per card. If the last number in each pair is zero, blank or the same as the first number, only that concept is tested. If the first concept of a pair is blank or zero, the rest of that card is ignored.

COMORN terminates processing when the first number on a card is zero or blank.

<u>Example of use of COMORN</u>

Assume we wish to test the concepts in the "ADIABTH DOCS " collection. We will place the RECODE cards on data set 20 and save them in a CDS rather than punch them on cards. The deck setup is as follows:

```
//JOB                                               CLASS=K, REGION=260K
/*MAIN CARDS=(10)
/*MSQ.SMART REGION=260K
//SMART EXEC PGM=SMART,REGION=260K
/*MSQ.SMARTGO
/*MSQ.SMARTER
//FT20F001   DD  SYSOUT=B
RESTART  2
INVERT     ADIABTH DOCS          CONCEPTS  DOCS     CON                          NOPRINT
INVERT     ADIABTH QUESTS        RELEVANT  QUESTS   REL                          NOPRINT
COMORN     DOCS     CON          QUESTS  REL          0.10    20    82    36    35
TEST FOR COMMON CONCEPTS IN THE ADIABTH COLLECTION
    1  560
<blank card>
STOP
/*CREATE RECODE FROM FT 20F001
/*
```

Routine CONCDS

Routine CONCDS will punch out the contents of a document, query, or centroid collection in a form that is printable and suitable for storing a Card Data Set (CDS). The collection can then be placed on the collections data set being used (and hence be accessible as a collection) without any additional cards. If the output of CONCDS resides on CDS "XYZ", the card '/*COPY XYZ NOSEQ' will place the collection onto the active SMART collections data set. CONCDS places sequence numbers (in increments of 100) in the last eight columns of each card. Essentially, CONCDS punches out a deck which comprises a CRDCOL run.

It should be noted that routine COLCDS punches the same information as CONCDS, but in a hexadecimal form, which is much faster to read as input. Therefore, unless the card input is to be modified or verified, COLCDS should be used instead of CONCDS.

Only one card is needed to call CONCDS.

Columns	Description
1-6	'CONCDS'
11-26	the collection name of the collection to be punched from the collections data set. This is either a permanent collection or one placed on the data set by the present SMART run.
31-34	' NO' if you do not want a listing of the CRDCOL deck produced, 'ONLY' if you want only the first line of each item listed, blank if you want all of each item listed.

To file the punched output on CDS "XYZ", the last card in the deck must be '/*CREATE XYZ FROM FT07F001'. Regardless of whether cards are to be punched or filed in a CDS, the card '/*MNQ.PNCHCRDS' must follow the

SMART "GO" step. A '/*MAIN' card may be needed to punch or file the cards.

Example of use of CONCDS

Assume we wish to punch the collection "CRN2S QUESTS " from the SMART disk and that we want the cards. The deck setup is as follows:

```
//JOB                                                  CLASS=L,REGION=260K
/*MAIN CARDS=(50)
/*SETUP UNIT=2314,ID=D00006,TEXT='SMART'
/*MNQ.SMART  REGION=260K
//SMART EXEC PGM=SMART,REGION=260K
/*MNQ.SMARTGO
/*MNQ.SMARTER
/*MNQ.FT01
RESTART  1
CONCDS    CRN2S   QUESTS      ONLY
STOP
/*MNQ.PNCHCRDS
/*
```

Routine CRDCEN

This routine creates a centroid collection defined by document numbers from a SMART document collection and cluster numbers from a SMART centroid collection. The following input cards are necessary:

1. The CRDCEN call out card containing the following:

Columns	Description
1-6	'CRDCEN'
11-26	The collection name of the centroid collection that will be formed.
31-46	The collection name of the document collection from which the centroid collection is to be formed.
51-66	The collection name of the previously prepared centroid collection that will be used. Either a document collection or a centroid collection or both must be used. Both fields (columns 31-46, 51-66) cannot be blank.

2. A card containing normalization information.

Columns	Description
3-4	"AUXSET" — the data set number of the data set which can be used as a scratch disk. If this field is blank, "AUXSET" is set to 46. If the collections are large set "AUXSET" to '90' and include the card '/*MNQ.FT90' after the '/*MNQ.SMARTER' card.
11-15	"MULT" — the multiplier to use for documents or centroids which have an indicated multiplier of zero (see description of card 4). If this field is blank or zero it is set to 1.
31-36	If this field contains the word 'COSINE', all

weights in a given vector are normalized according to the cosine correlation prior to being added to the composite for the new query. This makes all documents used have the same weight regardless of length. This is accomplished by multiplying each weight by the suitable multiplier and dividing by the square root of the sum of squared weights of the vector being added. To prevent weights from disappearing (due to integer arithmetic), make the multipliers large when using this feature. If this field contains the word 'LINEAR', normalization is accomplished by dividing by the sum of absolute values of all weights in the vector being added. If blank, no normalization will be done.

41-48 Let this field be "XXXXYYYY". If "XXXX" is 'NEG.' negative weights are permitted. Otherwise only positive weights will be kept after definition. "YYYY" can be either '   ', 'ABS.', or 'AVE.'. If "YYYY" is blank, only concepts with weight zero are deleted from the new query. (This obviously does not change correlations.) If "YYYY" is 'ABS.' then all concepts with absolute weight less than "PERDRP" are deleted. If "YYYY" is 'AVE.' then all concepts with absolute weight less than("PERDRP"* the sum of absolute weights)/(100* the number of unique concepts) are deleted. The former method is used to delete weights less than a specific value. The latter method permits dropping all weights less than a certain percentage of the average weight.

51-60 This number is a percentage (like 25.0) or weight and must include a decimal point.

3. An 80-column title for the centroid collection to be defined.

4. Sets of cards for defining the clusters. Each set consists of either document numbers from the document collection being used, or centroid numbers (indicated by a minus sign preceding each centroid number) from the centroid collection being used. The first card for each set contains the cluster number for the cluster being defined, and 5 pairs of alternating item numbers and multipliers for the weights in that item, in the format (I10,10X,5(2I5)). The cluster number may be left blank for sequential numbering of clusters. (If a previously prepared cluster collection is used in defining the new cluster collection, then the new cluster numbers on the cards should start with a number one larger than the number of the root cluster from the old collection.) The remaining cards for that cluster contain only 5 pairs of item numbers and multipliers in the format (20X,5(2I5)). Blank or zero multipliers for the weights are changed to "MULT". Centroid numbers (indicated by a minus sign) must follow the entire set of document numbers for each cluster. Clusters of loose documents are defined by only a cluster number and no defining documents or centroids. A blank item number must terminate each set of item numbers and multipliers, and the set of numbers for the next cluster must start on a new card. At present, cluster numbers must be consecutive.

5. A card with a '-1' in columns 9-10 to indicate the end of cards defining clusters.

6. Sets of cards indicating the members of the clusters previously defined — one set for each cluster previously defined. The cards for each set contain 5 items in the format (20X,5(I5,5X)). Centroid numbers (indicated by a minus sign) must follow the entire set of document numbers for each cluster. A blank item number must terminate each set of item numbers and the set of numbers for the next cluster must start on a new card.

The sets of cards containing members of clusters must be in the same order as the sets of cards defining clusters.

7. The last card (cards) must give the roots (the top level of centroids) of the cluster tree being defined. The centroid numbers (with their minus signs) are given in the format (20X,10I5). A blank centroid number must terminate the set of roots. Note that the highest level of each centroid group must be given, otherwise clusters at lower levels can never be reached.

If a tree with more than one level is to be defined, the centroids at the lowest level (with only document sons) must be defined first. The centroids should then be defined working up the tree to the root.

Example of use of CRDCEN

Assume we wish to cluster the "ADIABTH DOCS " collection. The following deck setup will produce a typical cluster collection and file it in CDS "TREE".

```
//JOB                                                  CLASS=K,REGION=280K
/*MAIN CARDS=(10)
/*MSQ.SMART REGION=280K
//SMART EXEC PGM=SMART,REGION=280K,TIME=20
/*MSQ.SMARTGO
/*MSQ.SMARTER
/*MSQ.FT90
RESTART  2
CRDCEN    ADIABTH TREECOLL    ADIABTH DOCS
  90          1
CLUSTER TREE COLLECTION FOR ADIABTH DOCS
        1               1    2    3    2    5    2    7    2    9    2
                       11    2   13    2   15    2   17    2   19    2
                       21    2

        2               2         4         6         8        10
                       12        14        16        18        20
                       22        24        26        28

        3              35    1   40    2   45    3   50    4   55    5
                       60    6   65    7   70   10

        4              71    1   72    1   73    1   74    1   75    1

        <blank card>

        5

       -1

                        1         3         5         7         9
                       11        13        15        17        19
                       21
                        2         4         6         8        10
                       12        14        16        18        20
                       22        24        26        28
                       35        40        45        50        55
                       60        65        70
                       71        72        73        74        75

       <blank card>
                       21        23        25        27        29
                       76        77        78        79        80
                       81        82
                       -1   -2   -3   -4   -5
COLCDS    ADIABTH TREECOLL
STOP
/*MSQ.PNCHCRDS
/*CREATE TREE FROM FT07F001
/*
```

Routine CRDCOL

Routine CRDCOL can add a collection to the collections data set being used from cards alone, or it can merge items (e.g. documents, queries, or centroids) from an auxiliary data set (previously set up) with new items from cards to create a new collection. (The auxiliary data set may be set up by a call to COLAUX.) The finished collection is placed on the collections data set being used. The collection can be accessed as a regular SMART collection only for the duration of the run. To save the collection for further runs, the collection must be punched out by either COLCDS or CONCDS. Item numbers of collection members need not be given in sequential order. If they are not given in sequential order however, they will be renumbered (not reordered) and an appropriate message printed. If CRDCOL is used for merging, cards need to be supplied only for those items that will be modified.

CRDCOL is called by one general parameter card, followed by a format card (if non-standard format bands are used) and at least one title card. Then comes the collection, followed by an "end of collection" card.

The first card to CRDCOL is the general parameter card described below.

Columns	Description
1-6	'CRDCOL'
11-26	A collection name for the new collection. It is suggested that this 16 character field be used as two 8 character fields, and the collection be named in the manner of the permanent collections.
28-35	The printing option. 'SUPPRESS' will suppress all printing. ' DELETE' will print only a one line summary of each item. 'PRINTNEW' prints the

concepts, weights, and relevancy decisions of items being added or modified. 'PRINTALL' prints the concepts, weights, and relevancy decisions of all items. If the print option is left blank, 'PRINTNEW' is assumed.

36-42 'NOT ALL' is specified if you are merging and wish to put only new or modified items on the data set. '   ALL' will put both the new or modified items as well as unmodified items from the auxiliary data set on the new data set. If the merge option is left blank, '   ALL' is assumed.

43-50 '   SUM' is specified if you are merging and wish to add a new weight (which may be positive or negative) to the old weight when the concept numbers match. 'DISPLACE' causes the new weight to displace the old weight on matched concept numbers. If these columns are blank, 'DISPLACE' is assumed.

51-58 '   NOREL' is specified if there are <u>no cards</u> for relevancy decisions. The word 'RELEVANT' indicates that there are <u>cards</u> for relevancy decisions. If '   NOREL' is specified, and CRDCOL is being used to merge or modify, the relevancy decisions will be taken from the auxiliary data set for each item on the data set. If these columns are left blank, 'RELEVANT' is assumed.

61-65 "NUMAUX" — this field must specify the highest numbered item from the old collection (on the auxiliary data set) that is to be placed on the new data set if merging is to be done. If "NUMAUX" is set larger than the number of items in the old collection, then "NUMAUX" is made equal to that number. Note carefully: if this routine is not being used to perform a merge, "NUMAUX" <u>must</u> be left blank.

67-72	'FORMAT' is specified if non-standard format bands are desired to read the item header card, the concepts and weights, or the relevancy decisions. The standard format bands are (I5, 3X, 8A8, 2I4) for the item header card, (8(I5, 2X, I3)) for the concepts and weights, and (15I5) for the relevancy decisions. If the format option is left blank, the standard format bands will be used.
73-80	'OVERRIDE' must appear in these columns if the new collection name matches identically the name of a previous collection on the collections data set. This should never be done without expressed authority.

If the standard format bands are used, the second and later cards are title cards, and are discussed after the format card. If non-standard format bands will be used (the format option has been set), the second card for CRDCOL must specify the new format bands. Only those format bands that will be non-standard should be given.

Columns	Description
1-24	the format band for the item header card. The format is punched left-justified within the field, and is enclosed by parentheses. Formats must be given to read in an integer item number, 8 fields of alphamerics for a description, an integer for "HANCON", and an integer for "ANREL" in that order.
25-32	the number of concept-weight pairs per card. The default is 8.
33-56	the format band for the concept-weight pairs. The format is punched left-justified within the field, and is enclosed by parentheses.
57-64	the number of relevancy decisions per card. The default is 15.

65-80 the format band for the relevancy decisions. The format is punched left-justified within the field, and enclosed by parentheses.

The next input card must contain an 80 column title for the collection. This title will be printed at the top of every page of the output whenever the items in this collection are referred to. Following this card come as many collection description cards as desired. These must be ended by a card which contains 'NO MORE' in columns 1-7. This card must appear even if there are no collection description cards.

Next come the items (e.g., documents or queries) in the collection. The first card of each item contains that item's sequence number in columns 1-5. Columns 9-72 should contain a hollerith description for that item. Columns 73-76 and 77-80 may contain the number of concept-weight pairs, "HANCON", and the number of relevant, "ANREL", for that item. If nothing is punched for "HANCON", then the concept-weight pairs are read until a concept of zero or a blank concept field is read. Concepts must be punched in ascending sequence.

If "ANREL" is blank or zero, and columns 51-58 of the CRDCOL call out card are blank or contain the word 'RELEVANT', relevant keys are read until a blank or zero relevant field is read. The keys of the relevant items must begin a new card. At least one card will be read. If columns 51-58 of the CRDCOL card contain the word '   NOREL', and "NUMAUX" is blank (a collection is being added from cards), it is assumed that there are no relevancy decisions (i.e., no cards for relevancy decisions). If columns 51-58 of the CRDCOL card contain the word '   NOREL', and "NUMAUX" is not blank (merging or modifying a collection), then there are no relevancy decisions on cards and the relevancy decisions will be taken from the auxiliary data set for those items on the data set. If an item is present on cards but not on the auxiliary dat set, it is assumed that there are no relevancy decision for that item.

The two methods of determining "HANCON" and "ANREL" can be independently selected, and different methods may be used for different items in the same set.

An "end of collection" card must follow the item cards. This is either a blank card or a card with 'END OF COLLECTION' in the first seventeen positions of the description field (usually card columns 9-25).

Example of use of CRDCOL

Assume we wish to modify collection "CRN14S QUESTS " by changing queries 5 and 10. We wish to delete some concepts from these queries. The following deck will put the modified collection in CDS "NEWCOL".

```
//JOB                                                      CLASS=L,REGION=260K
/*MAIN CARDS=(25)
/*SETUP UNIT=2314,ID=D00006,TEXT='SMART'
/*MSQ.SMART REGION=260K
//SMART EXEC PGM=SMART,REGION=260K
/*MSQ.SMARTGO
/*MSQ.SMARTER
/*MSQ.FT01
/*MSQ.FT90
RESTART  1
COLAUX     CRN14S  QUESTS       90
CRDCOL     CRN14S  MODQUEST                                  NOREL     225 FORMAT
(I2,8A1,2I5)                   14(14(I3,I2))
MODIFIED QUERIES FOR CRN14S  QUESTS
NO MORE
 5            4
  1 0  9 0 27 0 90 0
10
  1 0  6 0  8 0 17 0 25 0 37 0 90 0
<blank card>
COLCDS     CRN14S  MODQUEST
STOP
/*MSQ.PNCHCRDS
/*CREATE NEWCOL FROM FT07F001
/*
```

Routine DCLSTR

Routine DCLSTR automatically generates centroid clusters from a collection of items (normally documents). Any number of levels of clusters may be generated. In addition, pseudo-levels are also permitted in order to speed up the clustering time.

The time needed to run DCLSTR is:

$T = K \cdot N \cdot p \cdot \log_p m$ where

K = constant,

N = total number of documents in the collection,

p = the number of clusters produced from each previous cluster (assume constant for all levels),

m = total number of clusters on final level (assume $p^x = m$, for some positive integer x).

For any N and any m, T is minimized for $p=2$, and T is monotone increasing with p. Therefore, the algorithm runs fastest when many levels are produced with a small increase in the number of clusters from one level to the next.

Frequently, a specific number of levels is desired, so p cannot be chosen to optimize the time T. However, "pseudo-levels" can be generated, where a pseudo-level is a level of clusters used in the algorithm but not output as a level in the centroid tree.

For example, suppose the "CRN14TH DOCS " collection is to be clustered into one level of 100 centroids. Then,

$$T = K \cdot 1400 \cdot 100 \cdot \log_{100} 100 = 1.4 \times 10^5 K \quad .$$

However, suppose the 100 centroids are produced by using two levels of the algorithm. The first level is a pseudo-level of 10 centroids, and the second level consists of the 100 centroids that are output. Then,

$$T = K\cdot 1400\cdot 10\cdot \log_{10}100 = K\cdot 1400\cdot 10\cdot 2 = 2.8\text{x}10^{4}K \quad .$$

Thus the algorithm runs five times faster using an extra pseudo-level.

DCLSTR Data Cards are as follows:

Card 1

Columns	Description
1-6	'DCLSTR'
11-26	The collection name of the collection to be clustered. The collection must be on the collections data set being used.

Card 2

Columns	Description
4-5 9-10 14-15 19-20 24-25	These variables specify the unit numbers for scratch disk space. They must all be different, and can be chosen from any integer between 46-53. However, if the 424 or 1400 collection is being clustered, place '90' in 4-5, '92' in 9-10, '93' in 14-15, and '91' in 24-25. In addition, DD cards must be supplied for these data sets and placed after the '/*MNQ.SMARTER' card (see example).
31-35	A five digit integer that is used as a seed for the random number generator used to form the initial clusters.

Card 3

Columns	Description
11-26	The collection name for the centroid tree that is to be generated.

Remaining Cards (one for each level or pseudo-level)

Columns	Description
1-5	The approximate number of clusters desired at the end of this level.
9-10	The final percentage of loose documents. This is the percent of documents left unclustered at the end of the algorithm. However, if 'C' is punched, some documents might still remain loose, so blending should be specified if no loose documents are desired.
15	'0' if loose documents are <u>not</u> to be blended, '1' if loose documents <u>are</u> to be blended into the nearest cluster.
19-20	The percent of overlap between clusters (should not be greater than 15).
23-25	The percent of the remaining loose documents to be clustered on the next iteration.
28-30	The percent of the concepts in the clusters to be used in defining the centroids. This number is an approximation, and normally any number greater than 75 results in all of the concepts being used.
34-35	The maximum number of cycles allowed per iteration. If left blank, 10 is used.
40	A '1' indicates a pseudo-level of clusters.

A blank card <u>must</u> terminate the DCLSTR data cards.

<u>Suggested Parameters</u>:

As shown in the description of the data cards, eight parameters are specified for each level or pseudo-level of the algorithm. These parameters can be adjusted to satisfy different experimental conditions.

However, if some of the parameters are not involved in the experiment, then the following values are suggested:

Columns	Description
1-5	This parameter depends on the particular level or pseudo-level being generated. If the user does not care how many levels or clusters are generated, then output only one level of centroids, and let this value be the integer closest to $\sqrt{N}$, where N is the total number of documents in the collection. However, to speed up the algorithm one pseudo-level can also be used with $\sqrt[4]{N}$ used.
9-10	10
15	1; loose documents are blended
19-20	2
23-25	60
28-30	50
34-35	5
40	1 if the current level is not to be output.

<u>Example of use of DCLSTR</u>

Assume we want to cluster the "CRN4NUL DOCS " collection. The following deck setup will give two levels.

```
//JOB                                                  CLASS=K,REGION=280K
/*MAIN CARDS=(100)
/*SETUP UNIT=2314,ID=D00006,TEXT='SMART'
/*MNQ.SMART REGION=280K
//SMART EXEC PGM=SMART,REGION=280K,TIME=120
/*MNQ.SMARTGO
/*MNQ.SMARTER
/*MNQ.FT01
/*MNQ.FT90
//FT91F001 DD DCB=*.A,SPACE=(TRK,100,,CONTIG),UNIT=SYSSC
//FT92F001 DD DCB=*.A,SPACE=(TRK,100,,CONTIG),UNIT=SYSSC
//FT93F001 DD DCB=*.A,SPACE=(TRK,100,,CONTIG),UNIT=SYSSC
RESTART  1
DCLSTR     CRN4NUL DOCS
   90   92   93   50   91        100
           CRN4NUL TREE
    5   10    1    2   60   50    5
   20   10    1    2   60   50    5
<blank card>
COLCDS     CRN4NUL TREE
STOP
/*MNQ.PNCHCRDS
/*
```

Routine INVERT

Routine INVERT produces an inverted collection of items (documents, queries, or centroids) versus either relevant documents or concepts. For example, an inverted file of documents versus concepts would list each concept with a list of all documents in the collection having that given concept. Similarly an inverted file of queries versus relevant documents lists each relevant document with a list of all queries to which the given document is relevant. The inverted collection may be saved by using routine COLCDS.

Only one control card is needed.

Columns	Description
1-6	'INVERT'
11-26	The collection name of the collection to be inverted.
31-38	'RELEVANT' if the inversion is to be done versus relevant documents, 'CONCEPTS' if the inversion is to be done versus concept numbers. If these columns are blank, 'RELEVANT' will be assumed.
41-56	The collection name to be given to the inverted collection. If no name is given the inverted collection will be printed but not put on the collections data set.
61-65	If this field is blank, scratch data sets 46 and 47 will be used. If the collections are large, then this field should contain 'LARGE' and DD cards should be included after the '/MNQ.SMARTER' card for data sets 88 and 89 (see example).
71-77	'NOPRINT' if printing is to be suppressed. Blank if the inverted collection is to be printed.

Example of use of INVERT

We wish to invert the "CRN4S QUESTS " collection on relevancy decisions and on concepts. The following deck will do this and file both inverted collections in CDS "INVERT".

```
//JOB                                             CLASS=K,REGION=260K
/*MAIN CARDS=(20)
/*MNQ.SMART REGION=260K
//SMART EXEC PGM=SMART,REGION=260K
/*MNQ.SMARTGO
/*MNQ.SMARTER
/*MNQ.FT01
//FT88F001 DD DCB=*.A,SPACE=(TRK,200,,CONTIG),UNIT=SYSSC
//FT89F001 DD DCB=*.A,SPACE=(TRK,200,,CONTIG),UNIT=SYSSC
RESTART  1
INVERT    CRN4S   QUESTS               CRN4S  INVREL      LARGE
INVERT    CRN4S   QUESTS     CONCEPTS  CRN4S  INVCON      LARGE
COLCDS    CRN4S   INVREL
COLDCS    CRN4S   INVCON
STOP
/*MNQ.PNCHCRDS
/*CREATE INVERT FROM FT07F001
/*
```

Routine LSTCOL

Routine LSTCOL prints the contents of one document, query, centroid or inverted collection from the collections data set being used.

Only one input card is needed.

Columns	Description
1-6	'LSTCOL'
11-26	the collection name of the document, query, centroid or inverted collection to be printed.

Example of use of LSTCOL

```
//JOB                                                    CLASS=K,REGION=260K
/*MAIQ.SMART REGION=260K
//SMART EXEC PGM=SMART,REGION=[illegible]
/*[illegible].SMARTGO
[illegible].SMARTER
[illegible]START   2
LSTCOL     ADIABTH QUESTS
LSTCOL     ADIABTH DOCS
STOP
/*
```

Routine PROBTY

Routine PROBTY calculates the probability of a document being relevant to each query (and cumulatively to groups of queries) based on the number of matching concepts between the documents and the queries. Statistics may be gathered for individual queries as well as any cumulative number of successive queries. The concept weight range to be considered is supplied by the user, so certain concepts may be eliminated. The percentage of nonrelevant documents used in the test is also an input parameter. Those nonrelevant used are chosen at random for each query.

Two sets of statistics are produced. The first set gives the probability that a document is relevant to a query if it contains X number of matching concepts with the query (or queries) where X=0,1,2,...,20,20+. These statistics are computed by counting the number of documents with the indicated number of matching concepts with the queries.

The second set of statistics is called Document Concept Frequency. There are two parts to this set. A Document Concept Frequency of zero is associated with concepts in the documents that do not match the concepts in the queries. The numbers in this category signify the number of concepts in the relevant and nonrelevant documents that do not match any concepts in the query (or queries). The probability that a non-matching concept is in a relevant document is then computed. The second part of this statistic is for values of the Document Concept Frequency of 1,2,...,20,20+. This statistic is based on the weights of matching concepts. Each number multiplied by twelve gives the upper limit of the weight in that category (the lower limit is one more than twelve times the next lower number). The counts give the number of concepts with a weight in a document in the given range that match a concept in the query (or queries) regardless of the concept weight in the query. The probabilities that a concept within the given weight ranges is in a relevant document are then computed.

Two cards are needed to use PROBTY.

1. The PROBTY parameter card:

Columns	Description
1-6	'PROBTY'
11-26	The collection name of the document collection.
31-46	The collection name of the query collection.
51-55	The minimum concept weight to be used.
56-60	The maximum concept weight to be used.
61-65	"PQ" — individual query results will be given every "PQ" queries. (Always given after the first and last query.)
66-70	"PC" — cumulative results will be given every "PC" queries. (Always given after the first and last query.)
71-75	Disregard this number of random numbers in selecting the nonrelevant documents to be used.
76-80	"PCNTNR" — Use "PCNTNR"% of the nonrelevant documents in the test with each query. This number may be punched with a decimal point, but should be given as a percentage rather than as a decimal fraction.

2. An 80 column title to be printed on each page of the PROBTY output.

Example of use of PROBTY

Assume we wish to use the "ADIABTH" collections. We will include all concepts in the test and use 50% of the nonrelevant documents with each query. Statistics will be printed for every query and cumulatively every 5 queries. The card deck setup is as follows:

```
//JOB                                                    CLASS=K,REGION=260K
/*MNQ.SMART REGION=260K
//SMART EXEC PGM SMART,REGION=260K
/*MNQ.SMARTGO
/*MNQ.SMARTER
RESTART  2
PROBTY    ADIABTH DOCS        ADIABTH QUESTS        1  560    1    5    1 50.
PROBABILITY STATISTICS FOR THE ADIABTH COLLECTION
STOP
/*
```

Routine RECODE

Routine RECODE allows a new collection to be formed from a collection existing on a collections data set by modifying the concept numbers of the existing collection. New concept numbers may be assigned to take the place of old concept numbers without regard to numerical order or uniqueness. For example, concept 1 may become concept 10 while concept 3 becomes concept 6, and concepts 20 and 25 may both become concept 50, etc. The weights are carried along with the concepts and become the weights of the new concepts. If more than one old concept number is mapped into a single new number, the weights of the old concepts are added to form the weight of the new concept. Concepts may be dropped by mapping the old concept numbers into zero. New concept numbers must be given for every old concept number, even if the old number remains the same. Concept numbers not mapped explicitly into new numbers are automatically mapped into zero and dropped.

The control cards are as follows:

1. A card giving collection and printing information:

Columns	Description
1-6	'RECODE'
11-26	The collection name of the existing collection.
31-46	A collection name for the new collection.
51-60	If the old collection is to be printed, punch 'LIST '. If the old collection is not to be printed, punch 'DON'T LIST' or leave the field blank.
61-70	If the new collection is not to be printed, punch 'DON'T LIST'. If the new collection is to be printed, punch 'LIST ' or leave the field blank.

73-74	The number of an auxiliary data set to be used. If the field is blank, data set 50 will be used. For large collections, use data set '90' and includ the '/*MNQ.FT90' card after the '/*MNQ.SMARTER' ca

2. A card giving an 80 column description for the collection.
3. Sets of cards giving the ranges of old concept numbers and the new concept numbers with which they are to be replaced. Ranges should be given in increasing order.

The first card of each set contains:

Columns	Description
1-5	The minimum old concept number to be changed in this set.
6-10	The maximum old concept number to be changed in this set.

The second and succeeding cards of each set contain the new concept numbers to replace the (maximum - minimum + 1) concepts in the set. The new concept numbers are punched ten numbers per card in columns 1-5, 6-10, 11-15, 16-20, ... 45-50. Fields left blank are taken to be zero.

If the maximum old concept number in the original collection is included in the last range set, the sets must be followed by a blank card. If the maximum concept number is not included in the last range set, the sets must be terminated by a card which has the maximum old concept number in columns 6-10 with columns 1-5 blank or '   0'.

Example of use of RECODE:

Assume we want to modify the "ADIABTH QUESTS " collection. We will call the new collection "ADI NEW QUESTS ". It will be formed by dropping some concepts and grouping others. The card deck is as follows:

```
//JOB                                             CLASS=K,REGION=260K
/*MAIN CARDS=(20)
/*MNQ.SMART REGION=260K
//SMART EXEC PGM=SMART,REGION=260K
/*MNQ.SMARTGO
/*MNQ.SMARTER
RESTART  2
RECODE    ADIABTH QUESTS      ADI NEW QUESTS      LIST
    1   20
    5    0    5    0    0    1    0    0    5    0
    0    1    0    0    0    0    0    2    5    0
   61   70
    0    0    0    0   24    0    5    0    0    0
  277  277
    5
    0  533
COLCDS    ADI NEW QUESTS
STOP
/*MNQ.PNCHCRDS
/*
```

Routine RESCOL

Routine RESCOL allows result collections to be defined using hand-ranked documents. The collections are assumed to be results of iterations of searches on the same document collection. Each iteration of results will comprise a separate collection. The collections can be used by the standard evaluation procedures during the run, or punched for storage on a CDS by routine COLCDS. T! following input cards are necessary:

1. The RESCOL call out card contining the following

Columns	Description
1-6	'RESCOL'

2. A complete set of the following cards are needed for each iteration. The first card for each iteration contains the following information:

Columns	Description
1-16	Collection name for the result collection to be generated. Columns 1-8 must be the same for each iteration, while columns 9-16 must be different for each iteration. Columns 1-8 must not be blank.
17-26	"TOTDOC" — the total number of documents in the searched document collection for this iteration.
28-33	If this field contains 'DELETE' no message will be printed for query numbers that are not consecutive, but the query will be renumbered and included in the collection.

The second card for each iteration contains an 80-column description of the iteration.

The third and successive cards for each iteration are sets of cards, one set for each query. Query numbers should be consecutive, but will be renumbered if they are incorrect. The first card for each query is as follows [Columns 16-70 (containing statistics for the runs) may be left blank if the information is not needed for significance tests]:

Columns	Description
1-5	The query sequence number.
6-10	The total number of relevant documents for this query. This field cannot be blank or zero.
16-20	The total number of document-query correlations needed to generate this ordering.
21-25	The total number of centroid-query correlations needed to generate this ordering.
26-30	The number of document-query correlations less than "MINCOR". (See page 59.)
31-35	The number of items correlated with for ranking.
36-40	The number of documents previously seen by the user.
41-45	The number of relevant documents previously seen by the user.
46-50	The number of new items shown to the user on this iteration.
51-55	The number of items in the definition of the query with a positive multiplier.
56-60	The number of items in the definition of the query with a negative multiplier.
61-65	The correlation of this query with the original query.
66-70	The recall ceiling seen by the user.

The second and successive cards for each query contain the paired document numbers and ranks of the relevant documents in the format (7(2I5)), 7 documents per card, with the document number and then the corresponding rank. The document numbers must be given in ascending rank order. Not all relevant documents need be given, and a blank pair must terminate the input pairs.

The last card for each iteration must be blank to indicate the end of the queries for that iteration.

After all iterations have been read in, a blank input card indicates the end of input cards. This means that two blank cards are necessary at the end, one to terminate the last iteration and one to indicate the end of the input cards.

Example of use of RESCOL

Assume we wish to define a small result collection for three iterations of searches on the "ADIABTH DOCS " collection. Only three queries are in the query collection. The deck setup would be as follows:

```
//JOB                                                    CLASS=K,REGION=260K
/*MAIN CARDS=(10)
/*MSQ.SMART REGION=260K
//SMART EXEC PGM=SMART,REGION=260K
/*MSQ.SMARTGO
/*MSQ.SMARTER
RESTART  2
RESCOL
RESULT  RUN1              82
FULL SEARCH RUN
    1    5
    5    1    9    4    6   10   32   11   20   20
    2    2
    3    4   10    6
    3    1
   30    2
<blank card>
RESULT  RUN2              82
FROZEN FEEDBACK RUN
    1    5
    5    1    9    4   32   10   20   12    6   35
    2    2
    3    4   10    6
    3    1
   30    2
<blank card>
RESULT  RUN3              82
FLUID FEEDBACK RUN
    1    5
    5    1   32    3    9    6    6   10   20   11
    2    2
    3    1   10    2
    3    1
   30    1
<blank card>
<blank card>
COLCDS     RESULT  RUN1
COLCDS     RESULT  RUN2
COLCDS     RESULT  RUN3
STOP
/*MSQ.PNCHCRDS
/*CREATE RESULT FROM FT07F001
/*
```

Routine SEARCH

Routine SEARCH runs an initial search and up to 3 iterations of feedback on either clustered or unclustered documents. If desired, the queries used for any of the iterations can be produced by SEARCH in CRDCOL format (see the discussion following the description of the set of iteration control cards). The type of feedback done, and the type of cluster searching done, depend on user-specified parameters given on input cards. The collections to be used in the search process must be on one of the collections data sets. They may be permanent collections, temporary collections put on by routine CRDCOL, or collections put on by punched decks from routines COLCDS or CONCDS.

The results of SEARCH runs are usually averaged and verified with significance tests by calls to routines AVERAGE and VERIFY respectively. If the results are to be saved for later AVERAGE and VERIFY runs they should be punched using routine COLCDS, and stored either on a CDS or on cards.

The following describes the control cards for SEARCH. The first card contains the following information:

Columns	Description
1-6	'SEARCH'
11-26	the collection name of the query collection to be used.
31-46	the collection name of the document collection to be searched.
51-55	'MULTI' if multi-level (cluster) searching is to be done, blank if no cluster searching is to be done. If cluster searching is to be done, the cluster parameter cards must be included in the proper place as defined later in this description.

61-68	the first 8 characters of every result collection's name. The last 8 characters of each iteration's result collection name are taken from the first 8 columns of the appropriate field of "SCHTIT" (see description of the third card).
69-72	the minimum number of documents to be listed in direct order of recovery. In any case, the correlation and rank of each relevant recovered document is always listed unless it has a correlation less than "MINCOR" (see below). This number should be set to at least the total of all the documents to be used for feedback for the iterations. If it is set too low SMART will reset it to that total.
76-80	The maximum number of queries in a batch. If left blank, 99999 is used.

The second input card is printed at the top of every page as a title for the run.

The third card is divided into 4 fields of 20 columns each as search titles ("SCHTIT") for the four iterations to be done for each query. The first eight columns of each of the four fields are used as the last 8 characters of the respective iteration's result collection name.

Next come up to four sets (of two or three) iteration control cards to specify the query definition for successive iterations. The first card of the set of iteration control cards contains:

Columns	Description
1-5	Multiplier of original query
6-10	Multiplier of previous iteration's query
11-15	"MINCOR" — any document with a correlation less than "MINCOR" is considered not worth showing the user and is deleted from the recovered list prior to

sorting into correlation order. The higher this value the faster the system can answer a query. If punched, the field must include a decimal point. A blank field is equivalent to 10^{-75} (to delete documents with zero correlations)

17-22 The type of correlation to be used. If blank, the correlation of the previous iteration is used. If blank for the zeroth iteration, the 'COSINE' correlation is used. 'COSINE' is the only available correlation at present.

25-30 If this field contains the word 'NORMAL', at each definition "RMULT" (see below) is divided by the number of relevant used in that definition. "NMULT" (see below) is likewise divided by the number of nonrelevant used in feedback. If blank, no division is done.

33-35 This field contains 'YES' if specific documents a: multipliers are given for each and every query for this iteration. If blank, no specific documents are expected.

41-43 This field contains 'YES' if a specific vector of concepts and weights is supplied for each and every query for this iteration. If blank, no specific vectors are expected.

49-54 If this field contains 'FREEZE', the documents seen by the user defining the query are frozen in the order seen. Otherwise, all rank positions are available and correlations are calculated for all documents. That is a "FLUID" search is performed. 'FLUID' or blanks produce "FLUID" feedback.

57-62 If this field contains the word 'COSINE' all weig in a given vector are normalized according to the cosine correlation prior to being added to the

composite for the new query. This makes all documents used have the same weight regardless of length. This is accomplished by multiplying each weight by the suitable multiplier ("RMULT", "NMULT", or the user supplied document multiplier) and dividing by the square root of the sum of squared weights of the vector being added.
To prevent weights from disappearing (due to integer arithmetic) make "RMULT" and "NMULT", and user supplied document multipliers large when using this feature. If this field contains the word 'LINEAR', normalization is accomplished by dividing by the sum of absolute values of all weights in the vector being added. If blank, no normalization will be done.

65-72 Let this field be "XXXXYYYY". If "XXXX" is 'NEG.' negative weights are permitted for concepts in the query. Otherwise only positive weights will be kept. "YYYY" can be either '   ', 'ABS.', or 'AVE.'. If "YYYY" is blank, only concepts with weight zero are deleted from the new query. (This obviously does not change correlations.) If "YYY" is 'ABS.' then all concepts with absolute weight less than "PERDRP" are deleted. If "YYYY" is 'AVE.' then all concepts with absolute weight less than

$$\frac{\text{"PERDRP"} * \text{sum of absolute weights}}{100 \quad * \text{number of unique concepts}}$$

are deleted. The former method is used to delete weights less than a specific value. The latter method permits dropping all weights less than a certain percentage of the average weight.

73-80 "PERDRP". This is a percentage (like 25.0) or weight and must be punched with a decimal point.

The second card of each set of iteration control cards gives specific constants for selecting specified documents whose concept vectors are to be used in defining the queries for the iteration. It also includes the keys determining whether the queries used for the iteration are to be punched and/or printed.

Columns	Description
1-5	"RMULT" — all weights of relevant documents used in feedback are multiplied by this number. If blank, no relevant feedback will be done.
6-10	"ALLOF" — all relevant documents with iteration ra above "ALLOF" according to the previous iteration' ordering are used in defining the new query. If blank, no relevant documents are selected due to this parameter.
11-15	"IFOVER" — all relevant documents with a correlati above this value are also used. (This value must include a decimal point.) If "IFOVER" is zero or blank, no relevant documents are selected due to this parameter.
16-20	"ATLEST" — at least "ATLEST" relevant documents will be fed back (if they exist). If the "ATLEST' relevant documents are found above rank "ALLOF", then <u>at most</u> "ALLOF" relevant documents are used. If the "ATLEST" relevant documents are not found above rank "ALLOF", then <u>at most</u> "ATLEST" relevan' documents are used. If blank, no relevant docume are selected due to this parameter.
21-25	"NOMOR" — However, no more than "NOMOR" documents will be searched to provide the documents for feedback. "NOMOR" is set to the largest of "ALLOF", "ATLEST", and "NOMOR". The total number documents used for feedback is the sum of documen selected due to the "ALLOF" parameter, plus docu-

ments selected due to the "IFOVER" parameter, plus documents selected due to the "ATLEST" parameter.

26-30 "NMULT" — all weights of nonrelevant documents used in feedback are multiplied by this number. It should be given with a minus sign. To signify that negative feedback is not desired "NMULT" is blank or zero.

31-35 "$ALLOF" — all nonrelevant documents with iteration ranks above "$ALLOF" according to the previous iteration's ordering are used in defining the new query. If blank, no nonrelevant feedback will be done due to this parameter.

36-40 "$IFOVR" — all nonrelevant documents with a correlation above this value are also used.

41-45 "$ATLST" — at least "$ATLEST" nonrelevant documents will be fed back.

46-50 "$NOMOR" — However, not more than "$NOMOR" nonrelevant documents will be used. Note that only documents scanned to satisfy the desire for relevant documents for positive feedback are used in attempting to find nonrelevant documents for negative feedback. If "$NOMOR" is blank, it is set to the larger of "$ALLOF" and "$ATLEST".

51-55 "UNLESS". Negative feedback is done except when "UNLESS" relevant documents are found. If "UNLESS" are found, no negative feedback at all will be done. If "UNLESS" is blank or zero, and "NMULT" is non-zero, then negative feedback is attempted regardless of the number of relevant documents actually used in positive feedback.

56-58 'YES' — if the user wishes to stop considering documents for feedback once all the relevant documents have been found. If set to 'NO ' documents will be considered until the specifications of the other feedback parameters have been completed. The default is 'NO '.

60-64	"PRECVT" — If the precision after "ALLOF" document is over "PRECVT", and if the precision after more relevance judgments drops below "PRECVT", the judging of documents ceases. If blank, "PREC\ is zero.
65-72	'SILENT '— if search queries are not to be printed. 'STANDARD' if search queries are to be printed. If blank, 'STANDARD' is the default.
77-79	This field contains the query filing key ("PKEY") and may be: ' ' if the queries used for this iteration are <u>not</u> to be filed in a CDS. 'CON' if only the concepts (no relevancy decisions) of this iteration's queries are to be included. 'REL' if concepts <u>and</u> relevancy decisions of this iteration's queries are to be included.

If the queries for this iteration are to be filed in a CDS (they cannc be directly punched) ("PKEY" is 'CON' or 'REL') a third card must be given witł the collection name for the query collection. This name should be given as twc eight character fields similar to the existing collections. The name is punche in columns 11-18 and 19-26. If the queries are not to be punched ("PKEY" is ' '), <u>no</u> card should be included here (the set will consist of two rather than three cards).

A blank card <u>must</u> come after the last set of iteration control cards tc indicate the end of this group of cards.

A description of the remaining cards needed for SEARCH is given after the following discussion of the "PKEY" option. If "PKEY" is 'CON' or 'REL' foı an iteration, the queries used for that iteration's search are set up in CRDCOI format. These card images are similar to output from CONCDS except that they do not have sequence numbers. If 'CON' is coded, the actual relevancy decisions for the queries are not included, but a card with '0' in column 5 is included instead. The queries used for each iteration are placed on separate data sets. They must be filed in CDS's at the end of the run. The data sets used are 10, 11, 12 and 13 for iterations 0, 1, 2, and 3 respectively. These data sets are not punched out on cards. Do not include the '/*MNQ.PNCHCRDS' card to get this output (it should be included however if COLCDS or CONCDS is invoked in the run). To have the cards from iteration X (X = 0, 1, 2, 3) saved in a CDS, include the card '/*CREATE cdsname FROM FT1XF001' after the SMART "GO" step.

The cards for multi-level (cluster) searching follow if cluster searching is to be done. If cluster searching is not done, the following cluster control cards are not included, and the cards for user specified information described after the cluster cards are included at this point. If no user information is to be included, no additional cards are added.

When multi-level (cluster) searching is being done, cards to specify cluster search parameters are placed after all the iteration query definition control cards. A set of five cards must be given for each iteration. The first card of each set controls the number of documents with which the queries will be correlated.

Columns	Description
11-26	The name of the collection containing the centroid tree being used. If blank, the name from the previous iteration is used. If blank for the zeroth iteration a full search is done.
31-36	The type of correlation to be used for query-centroid correlations. If blank, the correlation of the previous iteration is used. If blank for the zeroth iteration, 'COSINE' is substituted. Cosine is the only available correlation at present.

46-50	"CORDOC" — At least this many documents will be correlated with on this iteration.
51-55	"TIMALL" — This number times "ALLOF" (for this iteration) documents are additionally correlated with on this iteration.
56-60	"TIMREL" — this number times the number of relevant documents not yet retrieved are addition to be correlated with on this iteration.
61-65	"TIMNMR" — this number times "NOMOR" (for this iteration) documents are to be additionally correlated with on this iteration.

The total number of documents to be correlated with on a given iteration is "WANTED", where "WANTED" is defined by:

"WANTED" = "CORDOC" + "TIMALL" * "ALLOF"
+ "TIMREL" * (the number of relevant documents not yet retrie
+ "TIMNMR" * "NOMOR"

"ALLOF" and "NOMOR" are user-supplied constants indicating how many documents are used in relevance feedback. Therefore the second and fourth terms of the parameter "WANTED" are constants, like "CORDOC", for a given iteration. These constants are expressed by three parameters (rather than being lumped into one for user convenience. Most users will set "TIMALL" and "TIMNMR" to zero. The parameter "TIMREL" allows the number of documents searched to be related to the number of relevant documents previously not found.

The second card of each set controls the selection of the centroid nodes to be used in a given iteration.

Columns	Description
11-15	"MINNOD" — at least this many nodes are expanded for this iteration. If blank or zero, 1 is used.

16-20	"MAXNOD" — no more than this number of nodes are expanded for this iteration. If blank or zero, 1 is used.
26-30	If there exist two nodes between "MINNOD" and "MAXNOD" which have a correlation difference greater than this number, all nodes above the node with the smaller correlation are expanded. If this field is blank or zero, it is automatically changed to 32767.0. If punched, the field should contain a decimal point.
31-35	Any nodes within this value of the latest node selected for expansion are also to be expanded.
46-50	"MNGOOD" — Any node with a "GOOD" (see description of the third card of the set) of less than this value is not to be retained for expansion.
51-55	"MNCORR" — Any node with a correlation less than this value with the query is not retained for expansion.
56-60	Only those nodes whose combined "CROWN" is greater than this percentage of the size of the collection being searched need be retained for expansion.
61-65	Only nodes whose combined "CROWN" is greater than this number times the number of documents to be correlated with are retained for expansion.

The third and fourth cards of each set control the ranking of the expanded nodes. These nodes are ranked on a parameter "GOOD" which is defined as follows:

$$\text{"GOOD"} = \text{"COEF"} + \text{"MCN"} \times \text{"CROWN"}^{\text{"PCN"}} + \text{"MLV"} \times \text{"LEVEL"}^{\text{"PLV"}}$$

$$+ \text{"MCFCN"} \times \text{"COEF"}^{\text{"PFCFCN"}} \times \text{"CROWN"}^{\text{"PNCFCN"}}$$

$$+ \text{"MCNLV"} \times \text{"CROWN"}^{\text{"PNCNLV"}} \times \text{"LEVEL"}^{\text{"PVCNLV"}}$$
$$+ \text{"MCFLV"} \times \text{"COEF"}^{\text{"PFCFLV"}} \times \text{"LEVEL"}^{\text{"PVCFLV"}}$$
$$+ \text{"MALL"} \times \text{"COEF"}^{\text{"PALLCF"}} \times \text{"CROWN"}^{\text{"PALLCN"}} \times \text{"LEVEL"}^{\text{"PALLLV"}}$$

where "COEF" is the correlation between the node and the query,
"CROWN" is the number of nodes that are the sons of the given node,
"LEVEL" is the level of the given node.

It should be noted that the formula for "GOOD" contains many combinations of "CROWN" and "LEVEL", making the formula extremely flexible for experimental purposes. It is expected that most users will use only two or three terms, most parameters in "GOOD" being usually set to zero.

Third Card

Columns	Description
11-15	MCN
16-20	PCN
21-30	MLV
31-35	PLV
36-45	MALL
46-50	PALLCF
51-55	PALLCN
56-60	PALLLV

Fourth Card

Columns	Description
6-15	MCFCN
16-20	PFCFCN
21-25	PNCFCN
26-35	MCNLV
36-40	PNCNLV
41-45	PVCNLV
46-55	MCFLV
56-60	PFCFLV
61-65	PVCFLV

The fifth card gives the following:

Columns	Description
1-3	'YES' — prints all the sons of an expanded centroid. 'NO ' prints no sons of expanded centroid. If blank, no sons are printed.
11-18	'YES ' — all correlations of centroids with queries will be printed. 'DROPONLY' — only correlations for centroids dropped due to

parameters "MNCORR", or "MNGOOD" will be printed. 'NO ' — correlation printing will be done. The default is 'DROPONLY'

21-23 'YES' — prints all the centroids to be expanded or retained. 'NO ' — no printing of centroids to be expanded or retained. The default is 'YES'.

If user specified information is provided for each query, the order is: all information for the first query, all information for the second query, etc. If user specified information is given, there must be user specified information for <u>every</u> query in the collection. Within one query, the cards pertaining to the zero-th iteration appear first, then the cards for the first iteration, second iteration, and last of all the cards for the third iteration. Within each iteration, the cards specifying specific documents and multipliers preceed the cards specifying specific concepts and weights.

To specify specific documents and multipliers, the user first punches the number of documents to be used in columns 1-5. The multiplier of the first document is placed in columns 11-15 and the key of the first document is placed in column 16-20. A total of five multipliers and documents can be specified per card (using columns 11-15, 21-25, 31-35, 41-45, and 51-55 for the multipliers and 16-20, 26-30, 36-40, 46-50, and 56-60 for the documents). If necessary, additional cards may be punched in columns 11-60 as appropriate.

To specify specific concepts and weights, the user first punches the number of concepts to be entered in columns 1-5. The first concept number and weight are placed in columns 11-15 and 16-20, respectively. Additional concepts and weights are placed as with multipliers and documents.

<u>Example of use of SEARCH:</u>

Assume we wish to perform a full search, a frozen feedback search, and a fluid feedback search of the "ADI" collection. We will save the result collections in CDS "RESULT". The frozen feedback queries will be saved in CDS "FROZEN". The card deck is as follows:

```
//JOB                                                    CLASS=L,REGION=280K
/*MAIN CARDS=(20),LINES=(40)
/*MNQ.SMART REGION=280K
//SMART EXEC PGM=SMART,REGION=280K,TIME=5
/*MNQ.SMARTCO
/*MNQ.SMARTER
RESTART  2
SEARCH    ADIABTH QUESTS      ADIABTH DOCS                    RESULT     30
ADI SEARCHES
FULL                FREEZE              FLUID
   1
<blank card>
     1                                       FREEZE
   5   10  0.5    2   20    -1    5  0.8    1    5             SILENT      REL
      ADIABTH FREEZE
   1    0
   5   10  0.5    2   20    -1    5  0.8    1    5
<blank card>
COLCDS    RESULT  FULL
COLCDS    RESULT  FREEZE
COLCDS    RESULT  FLUID
STOP
/*MNQ.PNCHCRDS
/*CREATE RESULT FROM FT07F001
/*CREATE FROZEN FROM FT11F001
/*
```

Routine SUBCOL

Routine SUBCOL allows a subset of items from a collection on the collections data set to be copied from that collection and formed into a new collection. The old collection remains intact on the collections data set. The new collection is written onto an auxiliary data set and then copied onto the collections data set. If in the same run the subcollection is used by CPDCOL in the merging process, then COLAUX need not be called to place the subcollection on an auxiliary data set since it is already there.

SUBCOL renumbers (but does not sort) the items copied from the original collection so that the new subcollection has items in proper order. Thus, SUBCOL can be used to change the order of the items in a collection. The sequence number of each item from the original collection is stored in the corresponding subcollection item's description field.

The cards needed for routine SUBCOL are as follows.

The first card contains control information:

Columns	Description
1-6	'SUBCOL'
11-26	The name of the collection on the collections data set from which the subcollection will be drawn.
31-46	The name to be given to the subcollection.
54-55	The number of the auxiliary data set to be used. If this field is left blank, 46 is used. If the subcollection is large, set this field to '90' and include the '/*MNQ.FT90' card after the '/*MNQ.SMARTER' card.

The second card contains an 80 column title for the subcollection.

The third and succeeding cards give the numbers of the members to be copied from the old collection to form the subcollection. There are six pairs of numbers per card in columns (11-15, 16-20), (21-25, 26-30), ... , (61-65, 66-70). The numbers in each pair give an (inclusive) range of items to be copied from the old collection. The second number of each pair cannot be less than the first number or only the first number will be used. If a range is to include only one item, the second number of the pair should be either the same as the first number, zero ('0'), or blank (' '). The list ends when the first number of a pair is zero or blank.

Example of use of SUBCOL:

Assume we wish to form a subcollection of the "CRN4S DOCS " collection and punch out the subcollection. The card deck is as follows:

```
//JOB                                                   CLASS=K,REGION=240K
/*MAIN CARDS=(40),LINES=(10)
/*SETUP UNIT=2314,ID=D00006,TEXT='SMART'
/*MNQ.SMART REGION=240K
//SMART EXEC PGM=SMART,REGION=240K
/*MNQ.SMARTGO
/*MNQ.SMARTER
/*MNQ.FT01
/*MNQ.FT90
RESTART  1
SUBCOL    CRN4S   DOCS        CRN4SSUBDOCS          90
  SUBCOLLECTION OF THE CRANFIELD 424 COLLECTION
           10   25   30   50   75  150  200  250  275       330
          350  400  410  410  420  420    0
COLCDS    CRN4SSUBDOCS
STOP
/*MNQ.PNCHCRDS
/*
```

Routine VERIFY

Routine VERIFY runs three significance tests on results from routine AVERAGE (which must be called immediately preceding a call to routine VERIFY). VERIFY should not be called unless at least two collections are used by AVERAGE. The sign test, the T-test, and the Wilcoxon signed rank test are calculated for each pair of search result collections. The main control card for routine VERIFY is as follows:

Columns	Description
1-6	'VERIFY'
9-14	When performing paired tests the user may specify a number, "DELTA" for each item to be used in the tests. If the two searches being tested disagree by a quantity less than this delta, they will be treated as if they were tied. If the user wishes to give his own "DELTA"s he should punch 'DELTAS' in these columns and include a set of delta cards following the VERIFY call out card. If "DELTA" is left blank, the standard deltas (all zero) will be used.
17-24	'NOWILCOX' if no Wilcox test is desired. If blan a Wilcox test will be made. In any case, sign and T-tests are run.
27-29	Indicates which items will be compared. 'R ' — recall level averages. 'D ' — document level recall and precision avera 'W ' — items from the cost statistics (see subroutine RESCOL for a complete list of these statistics). 'RW ' — both options "R" and "W" 'RD ' — both options "R" and "D" 'DW ' — both options "D" and "W" 'RDW' — options "R", "D" and "W". ' ' — if blank, option "R" will be run as a def

34,36 38,40 42,44 . . .	The numbers of the iterations to be compared (as given to routine AVERAGE. The first column of the pair (such as column 34, 38, 42, ...) contains one iteration number; the second column (such as column 36, 40, 44, ...) contains the second iteration number of the pair. In the one-sided tests the second iteration number is tested for performance better than the first. Up to 12 pairs may be given. If no pairs are given, the default is 0-1, 0-2, ..., . 0-"NITER", where "NITER" is one less than the number of result collections given to routine AVERAGE.

If "DELTA" cards are given, they follow the VERIFY call out card. There must be 12 cards with 8 numbers on each card. The numbers are punched in E10.5 format.

Example of use of VERIFY

Assume we have performed a normal search run and have saved the result collections in CDS "RESULT". The collection names are "RESULT FULL " and "RESULT FEEDBACK". We wish to average and verify these collections. The cards are as follows:

```
//JOB                                                    CLASS=K,REGION=260K
/*MNQ.SMART REGION=280K
/*MNQ.READCRDS REGION=280K
RESTART  2
/*COPY RESULT NOSEQ
AVERAGE   COLIDS
RESULT  FULL    RESULT  FEEDBACK
VERIFY
STOP
/*
//SMART EXEC PGM=SMART,REGION=280K
/*MNQ.SMARTRUN
/*MNQ.SMARTER
/*
```