II. The Cornell Implementation of the SMART System

D. Williamson

Abstract

This section covers the systems organization of the SMART programs prepared for operation in a batch processing mode on the IBM 360/65. Covered in particular are the basic input and text analysis routines, the document clustering programs, the search routines and the feedback operations.

1. Introduction

The present report contains a brief description of the SMART programs implemented on the Cornell 360/65. Two major criteria were used in the design of the Cornell implementation of the SMART system [1]. The primary need of such an experimental system is flexibility. The requirement for mixing different processing methods, such as clustering, relevance feedback, and searching, implies that the programming system should be written in terms of many small blocks, in such a way that any one process would be synthesized by using several such blocks put togehter. In this manner, not only can a process be carried out using many different combinations of methods, but a change in any part of the system does not require major alterations of the other parts of the system.

The second major design requirement is operating speed. The large size of the collections being used in the system make it necessary to plan on fast operations for any given process. Each process may then be carried out at a reasonable cost. Processing speed is gained in the SMART operations by making it possible to process several queries in parallel. The number of queries that can be processed simultaneously depends on the number of documents in the collection (only a given amount of storage space is available). As an example, a collection of 1500 document abstracts could be used with a parallel process for about 10 queries.

The SMART system is also designed to store the results obtained from any run in order to make it possible to generate comparisons between runs at a later time. This feature is of use especially for the more complex runs, for which averages and statistics are calculated that combine, or compare, a variety of evaluation parameters.

2. Basic Cornell System Organization

The SMART system is designed for the exploration, testing, and measurement of proposed algorithms for document retrieval. The system can be run by a person not knowing Fortran or Assembly Language since all routines are entirely data deck controlled. However, to permit the implementation of new procedures, which might necessitate adding new routines or modifying old routines, the system is written as a set of logically distinct subroutines with clear, explicit interfaces. This permits changes to be made to certain sections without destroying the integrity or usability of the other routines. As an example, to test a new correlation coefficient, it is necessary only to add an appropriate section of code to the present inner product routine (INNER). The entire body of feedback, centroid, and evaluation routines may still be used unchanged.

Basically, the SMART information retrieval process can be divided into four sections. The first involves the reading of text (e.g. abstracts, queries) and the conversion of given text into numeric concept vectors with weights. One possible conversion process may involve the use of suitable dictionaries, thesauruses, etc. At present, all routines for this purpose are processed at Harvard. Routines for conversion of text are to be implemented at Cornell in the fall of 1968. Fig. 1 and Table 1 contain specifications for the proposed text analysis routines. These routines are independent of the rest of the SMART system, in that the vectors produced from the input text are the only items exchanged between routines — no control parameters are passed. In a possible feedback process that uses actual query modification (as opposed to feedback using only relevant document numbers), the text analysis routines will produce additional proper text vector on call.

The second section involves pre-grouping (clustering) the documents of a given collection prior to the search process. The simplest form of grouping consists of considering the whole collection as one group — thus producing the situation which obtains a full search. For actual nontrivial clustering, the method credited to Rocchio [2] is now a part of the SMART system; other clustering methods are to be added later to reduce the amount of run time required for clustering. Multi-level clustering will also be added to the basic system. The systems chart of Fig. 2 shows that the clustering section of the system is completely independent of the

Name	Description
TEXT	Controls the text handling with control card directions. This is the only routine reading control cards in the text analysis package.
UNITIZ	Controls the conversion of a stream of text into a concept vector. Since it reads no cards, it can be called by "MODIFY" when updating or initializing a query to analyze the information supplied at the console.
SETUP	Obtains a dictionary and various tables for use of later routines.
STATUS	Prints statistical summary information saved from the dictionaries.
TOCOL	Places all result vectors onto the SMART SCDS if storage is desired.
STREAM	Supplies the complete text associated with one document or query
PARSE	Controls word generation of input stream
SPLIT	Splits input stream into words.
PEEL	Removes punctuation marks from words.
LOOKUP	Controls dictionary lookup routines.
LOCATE	Locates the concept number of a word if the word is already known.
HYPHEN	Handles hyphenated words.
SUFFIX	Finds the stem of the word being looked up.
INSERT	Inserts the proper stem of the word into the dictionary.

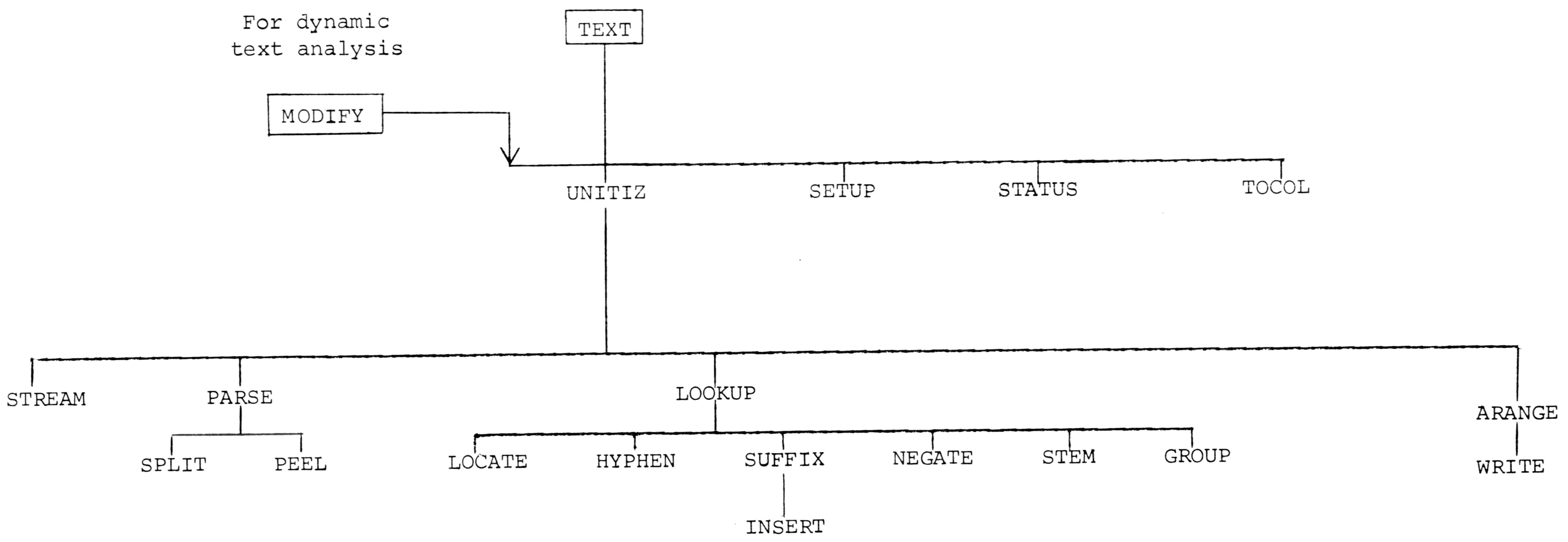
SMART Text Analysis Routines

Table I

Name	Description
NEGATE	Determines if the weight of the word should be negative. All words, including common words are passed through this routine.
STEM	Obtains the new concept number for the stem found in SUFFIX.
GROUP	Locates the concept numbers of thesaurus groups containing the word, if this is requested.
ARANGE	Sorts the list of concept numbers for a document (or query), and sums the weights of repeated concept numbers.
WRITE	Stores the resulting concept vector in auxiliary storage.

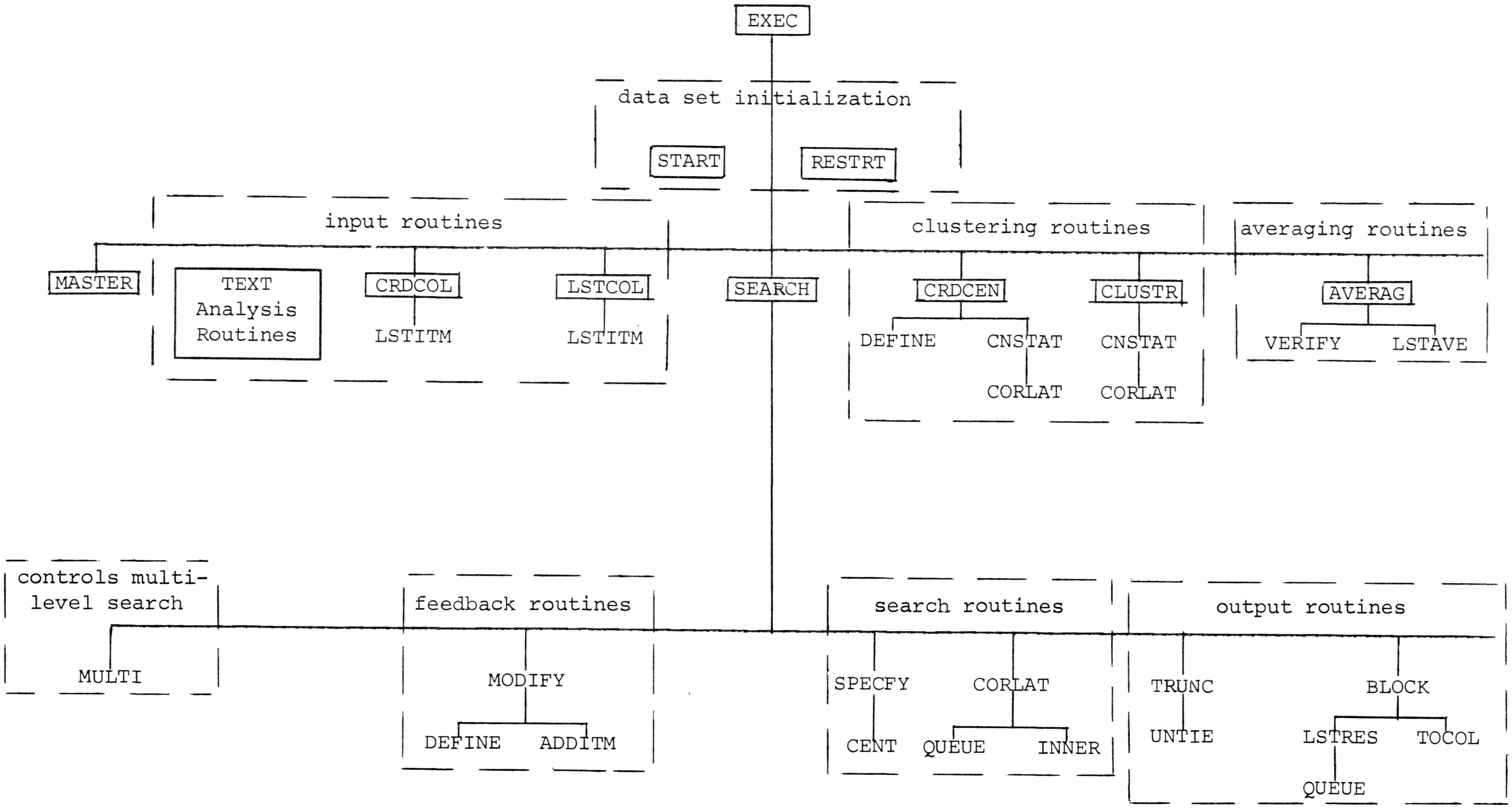
SMART Text Analysis Routines

Table 1 (contd)



Text Analysis Routines

Fig. 1



Boxed routines are control routines
Unboxed routines are inner system routines

Main SMART Systems Chart (exclusive of document and query analysis)

Fig. 2

other sections, such as searching. Both clustering and search sections are controlled, via parameters, by the system user, and any type of search can be used for any type of clustering process.

The search section selects the document groups to be examined in the retrieval process. Parameters are used to specify which clusters are to be searched, or which documents are to be searched, if a full search is desired. The number of documents that are to be searched determines the number to be correlated with each of several queries. The number of documents considered to be retrieved — i.e., the number either shown to the user or used in evaluation — may be specified by the user through a separate parameter.

Following the selection of the document groups to be correlated, the documents are compared with the queries, and ordered, for each query, according to their similarity with respect to that query (in decreasing order of the correlation coefficient). Several different inner-product search strategies and feedback processes can be performed by specifying appropriate parameters on control cards. The ordering obtained for the retrieved documents is evaluated by using relevance judgments previously made for each query.

A special section of the programming system computes evaluation averages and statistics for large groups of queries. Many different retrieval strategies, using the statistics from several retrieval runs, can be compared.

The clustering, search, and evaluation sections are implemented now at Cornell and may be requested on control cards. The basic routines involved are CLUSTR, SEARCH, and AVERAG, respectively, for the three last sections. The SMART system exists at Cornell as a private library system, located on a disk, and is accessible by reading in sets of control cards. When the SMART programs are loaded, a routine called EXEC receives control. This routine interrogates control cards in the data stream to ascertain which routines are desired and transfers control to those routines in the sequence requested.

A typical deck setup for the system is reproduced as follows:

```
                           /JOB . . . . . . (parameters). . . . . .
                           /EDIT
 initiates                 /SETUP  D00006
 SMART                   {  /COPY 86M.RUN
 routines                  /COPY 86M.SMT
                           /COPY 86M.AUX
                          //SYSIN DD*

 reopens SMART
 collection data         [  RESTRT . . . . . (parameters). . . . . .
 file

 sets up                    CLUSTR . . . . . (parameters). . . . . .
 document                    . . . . . . . . (parameters). . . . . .
                         {   .
 groups for a                .
 collection already          .
 on file

 performs retrieval         SEARCH . . . . . (parameters). . . . . .
 runs using methods          . . . . . . . . (parameters). . . . . .
 called for by the       {   .
 parameter cards             .

 performs statistical       AVERAG . . . . . (parameters). . . . . .
 averages for the        {   . . . . . . . . (parameters). . . . . .
 previous search             .

                         [  STOP

 signals end of job         /*
                         {  /OS
                            /ENDJOB
```

The parameters indicated for each routine are described in detail on the computer listing of the SMART routines.

It should be noted that no rigid calling sequence exists for these routines (other than the first call to RESTRT to open the collection data file). The call to CLUSTR is used to group documents for the next call to SEARCH, or alternatively, the document groups could be stored for later runs. Similarly, the call to SEARCH might have its parameters set so that SEARCH would use the previous CLUSTR document groups to perform a feedback run, or a different collection previously stored for the retrieval run might be used. A call to AVERAG could use data from retrieval runs made immediately prior to the call, or stored data from previous runs might be used in computing the statistical data. The following calling sequences are shown to illustrate the possible intermixing of routines:

Calling Sequence I

RESTRT

CLUSTR (parameters set to cluster collection *A* which is already in storage)

SEARCH (parameters set to do a full search on collection *A*)

SEARCH (parameters set to use the clusters as defined by the preceding call to CLUSTR)

AVERAG (parameters set to calculate the averages for both preceding calls to SEARCH)

STOP

Calling Sequence II

RESTRT

CRDCOL (parameters set to modify collection *A* which is already in storage)

SEARCH (parameters set to do a full search on the modified version of collection *A*)

SEARCH (parameters set to do a full search on the original version of collection *A*)

AVERAG (parameters set to calculate the averages for both preceding calls to SEARCH)

STOP

Calling Sequence III

RESTRT

SEARCH (parameters set to do a full search on collection *B* which is already in storage)

AVERAG (parameters set to calculate the average for the preceding call to SEARCH)

CLUSTR (parameters set to cluster collection *B*)

STOP

Calling Sequence IV

RESTRT

SEARCH (parameters set to use clusters previously stored for collection *B*)

AVERAG (parameters set to calculate the averages for both the preceding call to SEARCH, and results stored from a previous run)

STOP

3. The SMART System Routines

The SMART routines fall into two categories: The routines that can be called with control cards, and the routines that can only be called by other routines. The latter set is interconnected by means of complex internal vectors, designed to make the most efficient use of in-core storage. A list of the main routines is included in Table 2.

A) Control Routines

Eleven routines can be called by control cards — three major ones, and eight minor ones. The three major ones are: CLUSTR, SEARCH, and AVERAG.

CLUSTR-CLUSTR is the general clustering subprogram that calls one of several clustering methods. The major method, and the only one presently programmed, is Rocchio's clustering algorithm [2]. CLUSTR is used to group a given set of documents; instructions must be furnished to specify the clustering parameters. The generated clusters are stored for immediate or future use.

SEARCH-SEARCH is the major retrieval controlling routine in the system. All the necessary parameters are read in by SEARCH, and numerous minor routines are called to execute the desired document-query matching. Up to four iterations of feedback can be called, using any combination of the many feedback methods available. Different types of search patterns can be used, and any combination of document and query collections can be used. The final results are produced in terms of ranked lists of the retrieved documents for each query, along with recall and precision figures calculated for each retrieved document. Averages for groups of queries are calculated using the AVERAG routine.

AVERAG-AVERAG calculates the four standard global retrieval measures (rank recall, log precision, normalized recall and normalized precision) averaged over the desired collection. Recall-level and document-level recall-precision graphs are also constructed for the collection averages. AVERAG can call on another routine, named VERIFY, to generate several statistical tests for the comparison of the collections used by AVERAG.

Name	Description
EXEC	Controls program flow.
START	Starts a new SMART collections data set (SCDS).
RESTRT	Restarts a previously initialized SCDS.
MASTER	Entry point reachable from EXEC so a user can easily gain master control of the system.
CRDCOL	Adds a collection to the SCDS or modifies an existing collection.
LSTITM	Lists an item (such as a document or query) given the vector of the item.
LSTCOL	Lists a collection in SCDS.
SEARCH	Controls the search portion of the system, searching one query collection over one document collection with provisions for multi-level search and relevance feedback.
CRDCEN	Defines a set of centroids given the documents to be included in each cluster.
DEFINE	Defines the composite (such as the centroid in clustering or the updated query in relevance feedback) from a group of document vectors.
CNSTAT	Generates some centroid statistics for printing purposes.
CORLAT	Correlates a batch of queries against specified documents.
CLUSTR	Clusters documents according to various given algorithms.
AVERAG	Computes the average of up to four retrieval runs stored as collections in SCDS.

SMART System Routines

Table II

Name	Description
VERIFY	Runs significance tests on results using a sign test and a t-test.
LSTAVE	Lists and plots the graphs for the averages calculated by AVERAG.
MULTI	Initializes data sets needed to provide multi-level search capacity.
MODIFY	Modifies a query to permit relevance feedback, using a given feedback algorithm.
ADDITM	Adds the concepts of an item to a composite to be constructed by DEFINE.
SPECFY	Specifies which documents are to be correlated with on this search.
CENT	Runs one level of a multi-level centroid search to ascertain which centroids are to be used.
QUEUE	Maintains a queue of location pointers of items ranked by value.
INNER	Forms the inner product of two vectors.
TRUNC	Sorts the correlations of documents that have been correlated with a query and assigns ranks to these documents.
UNTIE	Assigns positions to relevant documents with identical correlations.
BLOCK	Sets up the results of up to 4 runs for printing.
LSTRES	Prints results of up to 4 runs for one query.
TOCOL	Places results of runs on SCDS as collections.

SMART System Routines

Table 2 (contd)

Other routines called by control cards are

START-START initializes a new SMART collection Data Set (SCDS). If temporary use of a non-standard collection is needed for a single run, START can be used to set up a temporary data set; this operation will not require the mounting of the SMART disk.

RESTRT-RESTRT opens the SCDS for use, allowing either a new collection to be received in storage, or providing for use an older standard collection. It is necessary to call either START or RESTRT before any further use is made of the SMART systems routines.

CARDCOL-CARDCOL reads a new collection (documents, queries, or centroids) from a set of cards, or from an auxiliary data set, and puts the new collection on the SCDS. CARDCOL can also make modifications in existing collections such as adding new concepts to existing documents, modifying old concepts or weights, and deleting concepts, or entire documents, from the collection. The modified collection is placed on the SCDS.

LSTCOL-LSTCOL lists any given collection on the SCDS.

COLAUX-COLAUX redefines a collection of documents or queries existing in the SCDS as a collection on an auxiliary data set.

COLCEN-COLCEN redefines a collection of centroids in the SCDS as a collection of centroids on an auxiliary data set.

CRDCEN-CRDCEN defines a set of centroids from parameters specifying the documents to be included in the centroid.

MASTER-MASTER is a dummy subroutine, reachable from EXEC, used to link user programs to the system routines. In this manner, an experienced user can control the system routines not accessible through cards, or perform functions not built into the system without requiring the use of difficult system linkage steps.

B) Inner System Routines

The remaining SMART system routines cannot be called by control cards. To make the most efficient use of storage, complex transfer vectors have been set up between the inner system routines and the major routines previously described; these vectors must be properly filled if the inner system routines are to be used separately (through MASTER). The routines are listed here to illustrate the various parts of the system, and to provide the experienced user with a convenient list of the subroutines available.

LSTITM-LSTITM lists the concept vector contained in COMMON/AUX/.

DEFINE-DEFINE computes the composite vector from a group of individual document vectors. The composite vector can be a centroid for a cluster defined by a document group, or it can be an updated query vector used for relevance feedback, the updated query being defined by a group of documents.

ADDITM-ADDITM adds the concepts of an item to the composite to be constructed by DEFINE.

CNSTAT-CNSTAT generates centroid statistics, including a list of documents used to generate the centroid, the average number of concepts in the documents, and other similar statistics. These statistics are calculated mainly for output printing purposes, although some are used by other routines.

CORLAT-CORLAT correlates a batch of queries (as many as can be processed in parallel) against the documents specified by SPECFY.

SPECFY-SPECFY determines which documents are to be used for query correlation for a given iteration and a given batch of queries. For a full search, SPECFY would identify all the documents in a collection; otherwise, parameters passed through SEARCH instruct SPECFY how to determine the appropriate list of documents to be used.

INNER-INNER forms the inner product of the two vectors previously specified by CORLAT. The only correlation coefficient presently implemented is the cosine correlation.

CENT-CENT determines which clusters are to be used in a given search. Parameters transferred through SPECFY identify the cluster search method to be used by CENT.

QUEUE-QUEUE maintains a queue of location pointers for the transfer vectors. The queue is ordered by the value of the item stored in queue, and is used as a major servicing routine by the system.

TRUNC-TRUNC sorts the correlations of documents calculated by CORLAT, and assigns ranks to all documents. The correlations that are not needed (those for nonrelevant documents with ranks below a given rank) are deleted from the results at this point.

UNTIE-UNTIE assigns ranks to relevant documents in the case of a tie in correlation coefficients.

BLOCK-BLOCK sets up the results of up to four runs for printing, and calls TOCOL to add these results to the SCDS.

TOCOL-TOCOL places the results of a run on the SCDS.

LSTRES-LSTRES prints the retrieval results for one query (for up to four iterations of that query).

MODIFY-MODIFY performs the necessary modifications in a query for relevance feedback.

MULTI-MULTI initializes data sets needed to provide multi-level search capability (called once per call to SEARCH).

LSTAVE-LSTAVE lists and plots the values for recall level and document level averages, using information stored in the transfer vectors. LSTAVE is the major routine used by AVERAG.

The remaining SMART routines are small service routines, used by other routines to perform single tasks. The following is a list of the principal routines of this type:

Special SMART In-Out Routines:

NEWCOL	Opens a new collection in the SCDS to LOCITM.
LOCITM	Locates an item in the collection opened by NEWCOL.
FUTITM	Obtains an item for future use.

REDITM	Reads items from SCDS.

General Purpose I-0 Routines:

HREAD 1 UREAD 1 HREAD UREAD	Read records stored temporarily on disk.
STPIN	Stops input.
BUFIN	Device dependent routine to buffer in.
CHKIN	Device dependent routine to check input.
HWRITE UWRITE	Write records to be stored temporarily on disk.
STPOUT	Stops output.
BUFOUT	Device dependent routine to buffer out.
CHKOUT	Device dependent routine to check output.
BACKSP	Device dependent routine to "backspace" a data set.
REWIND	Device dependent routine to "rewind" a data set.

Sorting Routines:

SORTUP	Sorts a vector into ascending sequence.
SORTDN	Sorts a vector into descending sequence.

Copying Routines:

HMOVE UMOVE	Move vectors quickly in core storage.

Labeling Routines:

TALLY	Counts printed lines to eject and label pages.
TIME	Operates ten elapsed time clocks.

Conversion Routines:

HOLINT Converts Holerith integers to binary.

INTHOL Converts binary integers to Holerith.

Calculation Routines:

PRMFAC Calculates the logarithmic values of the permutations of a given number cf things taken a given number at a time.

References

[1] E. Ide, R. Williamson, and D. Williamson, The Cornell Programs for Cluster Searching and Relevance Feedback, Information Storage and Retrieval, Report ISR-12 to the National Science Foundation, Section IV, Department of Computer Science, Cornell University, June 1967.

[2] J. J. Rocchio, Jr., Document Retrieval Systems — Optimization and Evaluation, Harvard Doctoral Thesis, Information Storage and Retrieval, Report No. ISR-10 to the National Science Foundation, Harvard Computation Laboratory, Cambridge, March 1966.