I. The Cornell Implementation of the SMART System

D. Williamson, R. Williamson, M. Lesk

Abstract

The systems organization of the SMART programs is discussed as implemented 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. Sample computer output is shown in each case to illustrate the operations.

1. Introduction

The SMART system is designed for the exploration, testing and measurement of proposed algorithms for document retrieval. The system takes documents and search requests in English, performs a fully-automatic content analysis of the texts, matches analyzed documents with analyzed search requests, and retrieves those stored items believed to be most similar to the queries. The request authors (users) can submit information to improve their queries (relevance feedback), and this information is used by several experimental procedures to improve search results. The time required to match large collections of documents to requests can be reduced by grouping these documents (clustering) and matching requests against a representative of the entire group. Finally, exhaustive evaluation procedures can be used to ascertain the effectiveness of various methods used in searching.

Several important criteria are incorporated in the implementation of the SMART system. [1] 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 assembling several blocks into one unit. 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 fast processing speed necessary to process large collections is gained by making it possible to process several queries in parallel.

2. Basic System Organization

The SMART information retrieval process can be divided into five basic sections: the input of printed text, the grouping of documents for searching purposes (clustering), the selection of a group of documents to be searched, the searching of the document group, and the evaluation of the search.

The printed text specifying the queries and documents must be converted into a form more easily handled by a computer. For this purpose various automatic language analysis devices can be used which reduce each query and document to "concept" vector form.

To produce fast searching algorithms, documents can be grouped into classes of similar documents. The grouping (clustering) is done by placing documents containing similar concepts together, into the same group; a representative central item is then constructed for each group.

The search of a document group (cluster) is done by first matching requests against clusters. Certain clusters are picked as most likely to contain documents of interest. These documents are then searched in the normal manner, one item at a time. After seeing some retrieved documents, the requestor can modify his request, either by physically changing it, or using the requestor's relevance assessments to automatically modify the query.

Several measures of retrieval performance are computed to evaluate each search. The sign test, T test, and Wilcoxon Rank Sum test are also used to determine the significance of the evaluation measurements.

A) Input of Printed Text

The first section involves the reading of text (e.g., abstracts, queries) and the conversion of a given text into numeric concept vectors with weights. The conversion process may involve the use of suitable dictionaries, thesaurus, and other language normalization aids. At present, a relatively simple PL/1 program is used to implement this section. A more flexible Fortran IV program is planned for later implementation as described in report ISR-14 [2] and included in the system flowcharts, part 4 of the report.

The presently available text-handling program, LOOKUP, is a procedure which performs dictionary lookups on a large IBM 360-series computer. It accepts a dictionary, suffix list, and texts and produces "concept vectors" for the texts. Words missing from the dictionary are also processed. The algorithm is essentially that of Sussenguth [3] although the tree structure storage format is not used. LOOKUP is designed primarily for ease of programming, and is coded entirely in PL/1.

The overall operation of LOOKUP is divided into three parts. First, the dictionary and suffix list are read into memory, sorted alphabetically and necessary initialization is performed. Secondly, text is read in, divided into words, and the words looked up in the dictionary and suffix lists. Third, the concept numbers derived from the words in each document are sorted and condensed into a properly weighted vector. The vector can be printed and/or stored in machine-readable form. The lookup program finds a match between an input word and a dictionary entry under the following conditions:

1) the word exactly matches a dictionary entry; or
2) it matches a dictionary entry with a final "e" dropped and a suffix beginning with a vowel added; or
3) it matches a dictionary entry plus a suffix; or
4) it matches a dictionary entry with a final "y" changed to "i" and a suffix added; or
5) it matches a dictionary entry, with a final consonant doubled and a suffix added.

When several possible matches are found, the match involving the longest stem is preferred; within stems of the same length, preference is in numerical order as above. Thus, if "cop", "cope", and "copy" are all stems in the dictionary, and all normal English suffixes are included in the suffix list, "cops" is found from "cop" under rule 3; "copes" or "coping" is found from "cope" under rule 2; "copying" from "copy" under rule 3; "copies" from "copy" under rule 4; and "copper" from "cop" under rule 5. Other morphological features of English are not recognized; such word pairs as "mouse" and "mice", "sing" and "sung", "fight" and "fought", or "court-martial" and "courts-martial" must be entered explicitly in the dictionary if both members are to be recognized. Special rules exist which specify that all stems must be at least three letters long (to avoid, for example, finding "wing" from "we" under rule 2 or "inning" from "in" under rule 5); furthermore, all words are truncated at 24 characters.

The program can distinguish titles from the body of the text, if asked; and it may either split the weight of an ambiguous word among its concept numbers, or weight all concept occurrences equally. The suffix list may be omitted from the lookup, in which case only words that exactly match a dictionary entry can be found; and the programmer may choose whether hyphenated words are to be considered as a unit or as separate words. As in the previous SMART implementations, concept numbers of zero or concept numbers of 32000 or more are considered to be nonsignificant and are dropped from the vector.

Fig. 1 shows a typical output of LOOKUP. First the title is given, and then the text of the document (or query in this case). The resulting numeric concept vector is next printed, consisting of pairs of concept numbers followed by the respective concept weights (for example, concept 927 with weight 12, 2574 with weight 12, etc.). Concepts are listed in the vector in increasing numeric order.

B) Document Clustering for Search Purposes

At present two clustering algorithms are in operation at Cornell — CLUSTR, which uses Rocchio's clustering algorithm [4], and DCLSTR, a variation of Doyle's clustering algorithm. [5]

Rocchio's clustering algorithm is based on the following methodology: an unclustered document is selected as a possible cluster center. Then, all of the other unclustered documents are correlated with it, and the document is subjected to a density test to see if a cluster should be formed around it. The density test specifies that at least N_1 documents should have corre-

```
LISTING OF INPUT TEXT, MISSING WORDS, AND VECTORS   PAGE103

 *FIND QA7PAPERS
DESCRIBE PRESENTLY WORKING AND PLANNED SYSTEMS FOR PUBLISHING
AND PRINTING ORIGINAL PAPERS BY COMPUTER, AND THEN SAVING THE
BYPRODUCT, ARTICLES CODED IN DATA-PROCESSING FORM, FOR FURTHER
USE IN RETRIEVAL .

VECTOR:    927/ 12/, 2574/ 12/, 3509/ 12/, 4087/ 12/, 4989/ 12/, 4999/ 12/,
          5068/ 12/, 5253/ 12/, 5432/ 12/, 5440/ 12/, 5469/ 12/, 5515/ 12/,
          5543/ 12/, 5554/ 12/, 5569/ 12/, 5576/ 12/, 5602/ 12/, 5605/ 12/,
             0/  0/,

 1007
*FIND QA8INDEXING
DESCRIBE INFORMATION RETRIEVAL AND INDEXING IN OTHER LANGUAGES . WHAT
BEARING DOES IT HAVE ON THE SCIENCE IN GENERAL .QUE

VECTOR:   3931/ 12/, 4369/ 12/, 4762/ 12/, 4989/ 12/, 4999/ 12/, 5372/ 12/,
          5489/ 12/, 5598/ 12/, 5606/ 12/,    0/  0/,

 1008
*FIND QA9ANALYSIS
WHAT POSSIBILITIES ARE THERE FOR AUTOMATIC GRAMMATICAL AND
CONTEXTUAL ANALYSIS OF ARTICLES FOR INCLUSION IN AN INFORMATION
RETRIEVAL SYSTEM .QUE

VECTOR:   3338/ 12/, 4821/ 12/, 4916/ 12/, 4999/ 12/, 5469/ 12/, 5474/ 12/,
          5511/ 12/, 5605/ 12/, 5606/ 12/,    0/  0/,

 1009
*FIND QA10GROUP
THE USE OF ABSTRACT MATHEMATICS IN INFORMATION RETRIEVAL,
E.G. GROUP THEORY .

VECTOR:   2883/ 12/, 4999/ 12/, 5491/ 12/, 5555/ 12/, 5602/ 12/, 5606/ 12/,
```

lations higher than a specified parameter p_1 with the document in question, and that at least N_2 documents should have correlations higher than p_2 (p_2 is generally larger than p_1). This test ensures that documents on the edge of large groups do not become cluster centers. If the document passes the density test, thus becoming a cluster center, a cutoff correlation, p_{min} , is determined from the cluster size limits and the distribution of correlation values. The cutoff correlation becomes p_1 if fewer documents than the minimum cluster size (M_1) have correlations above p_1 . If more such documents exist, the cutoff correlation is chosen at the greatest correlation difference between M_2 adjacent documents, where M_2 is the maximum cluster size.

A classification vector is then formed by taking the centroid of all the document vectors having correlations above p_{min} . This centroid vector is matched against the entire collection, and the cutoff parameters for cluster size are recalculated to create an altered cluster.

As a result of this process, some documents may appear in more than one cluster; and some which were in a cluster when the centroid was originally formed may not remain in any cluster. These documents, as well as those which failed the density test, are termed "loose", and those within the cluster are termed "clustered".

This entire procedure is repeated with all unclustered documents, the first pass terminating when all items are either clustered or loose. Figs. 2, 3, and 4 illustrate the formation of a cluster. Document 2 is first correlated with all previously unclustered documents in the collection (9 documents of the 82 documents in the collection had previously been clustered around document 1). The correlations are ranked, and the ranks, docu-

CLUSTERING ABOUT DOCUMENT 2

RANK	DOC	CORR	RANK	DOC	CORR	RANK	DOC	CORR
1	2	1.0000	2	64	0.4002	3	27	0.3631
6	68	0.2512	7	61	0.2475	8	18	0.2367
11	12	0.1990	12	55	0.1867	13	14	0.1861
16	34	0.1697	17	33	0.1689	18	22	0.1634
21	50	0.1445	22	82	0.1420	23	48	0.1400
26	19	0.1239	27	6	0.1235	28	30	0.1006
31	77	0.0934	32	81	0.0934	33	32	0.0921
36	53	0.0854	37	78	0.0748	38	25	0.0729
41	26	0.0583	42	58	0.0578	43	38	0.0539
46	42	0.0460	47	15	0.0454	48	49	0.0437
51	21	0.0394	52	35	0.0385	53	54	0.0374
56	43	0.0337	57	36	0.0335	58	44	0.0283
61	0	0.0000	62	0	0.0000	63	0	0.0000
66	0	0.0000	67	0	0.0000	68	0	0.0000
71	0	0.0000	72	0	0.0000	73	0	0.0000

RANK	DOC	CORR	RANK	DOC	CORR
4	39	0.3466	5	41	0.2628
9	29	0.2258	10	71	0.2174
14	66	0.1749	15	73	0.1715
19	16	0.1515	20	69	0.1511
24	9	0.1257	25	23	0.1257
29	8	0.0934	30	65	0.0934
34	67	0.0891	35	17	0.0880
39	75	0.0691	40	24	0.0665
44	7	0.0511	45	10	0.0467
49	80	0.0432	50	79	0.0417
54	63	0.0353	55	59	0.0347
59	0	0.0000	60	0	0.0000
64	0	0.0000	65	0	0.0000
69	0	0.0000	70	0	0.0000
74	0	0.0000	75	0	0.0000

DOCUMENT 2 HAS PASSED THE DENSITY TEST.
CUTOFF WILL BE CHECKED.

The Testing of Document 2
as a Possible Cluster Center

Fig. 2

CCRRELATICNS FUR CENTROID 2

RANK	DCC	CORR	RANK	DOC	CORR	RANK	DCC	CORR
1	2	C.7221	2	64	C.5501	3	27	0.5252
6	68	0.4657	7	18	C.4358	8	29	0.4343
11	3C	C.3859	12	55	C.2971	13	66	0.2784
16	33	0.2591	17	22	0.2467	18	23	0.2458
21	17	C.2326	22	9	0.2258	23	69	0.2253
26	34	C.2090	27	14	0.2054	28	58	0.1937
31	28	0.1862	32	8	C.1857	33	78	0.1595
36	77	C.1485	37	53	0.1480	38	80	0.1473
41	63	0.1384	42	79	0.1281	43	82	0.1237
46	44	0.1190	47	42	0.1138	48	21	C.1131
51	36	C.1027	52	67	0.0988	53	49	0.0945
56	10	0.0902	57	7	0.C872	58	20	0.0349
61	54	C.0638	62	37	0.0637	63	60	0.0573
66	45	0.C526	67	4	C.C519	68	76	0.0516
71	13	0.0385	72	31	0.0382	73	74	0.0179

RANK	DOC	CORR	RANK	DOC	CORR
4	39	0.5177	5	71	0.4966
9	61	C.4333	10	41	0.4082
14	16	0.2655	15	12	0.2624
19	6	0.2366	2C	73	0.2339
24	65	0.2149	25	15	0.2117
29	48	0.1910	3C	19	C.1901
34	26	0.1567	35	75	0.1544
39	81	0.1459	40	50	C.1455
44	32	0.1204	45	38	0.1195
49	25	0.1057	50	43	0.1054
54	24	C.0944	55	57	C.0936
59	46	C.0743	60	59	0.0730
64	35	0.0547	65	5	C.0541
69	72	C.0432	70	3	0.0418

The Correlation of Centroid 2 with all Unclustered Documents

Fig. 3

ITEM 2 CENTROIDCOCC0002 ACIABTH CCCS COUUU002

CCN	WT	CON	WT	CON	WT	CON	WT	CCN	WT
1	24	3	24	4	6C	5	144	7	36
19	84	20	12	21	18	22	30	23	18
3C	12	37	12	41	12	42	24	43	12
54	12	57	12	61	132	62	12	63	12
77	36	81	48	85	12	87	24	91	24
126	24	128	12	129	72	135	24	136	12
147	24	15C	12	155	12	158	12	162	12
193	12	196	12	199	24	212	24	213	12
231	12	232	12	243	12	248	12	259	12
282	12	284	12	286	24	287	48	291	24
321	12	444	12	455	36	481	6	530	12

CON	WT	CON	WT	CON	WT	CON	WT
8	120	12	60	13	24	17	48
25	48	26	12	32	48	33	72
46	24	48	60	50	12	51	72
64	12	66	48	67	24	72	36
98	36	99	12	115	12	116	108
138	12	140	12	141	12	143	12
171	12	180	60	184	12	191	12
217	12	219	36	223	12	225	12
266	24	267	12	271	36	275	12
293	12	294	12	296	12	315	12

THE 95 CONCEPTS ABOVE HAVE A SUM OF ABSOLUTE WEIGHTS = 2640

WITH A ROOT SUM OF SQUARED WEIGHTS = 377.00

THE 11 RELEVANT -- 2 64 27 39 71 68 18 29 61 41 30

The Completed Cluster 2

Fig. 4

ment numbers, and correlation coefficients are listed in Fig. 2. In the example, at least 10 documents (N_1) must have a correlation greater than 0.15 (p_1), and at least 5 documents (N_2) must have a correlation greater than 0.25. The correlation of document 2 is larger than 0.15 for 19 other documents, and for 5 other documents the correlation exceeds 0.25. Document 2 therefore passes the density test. M_1 in this example is 5, and therefore p_{min} is calculated by finding the greatest correlation difference between adjacent documents, starting with the document of rank 5 (at least M_1 documents must be included) and checking differences up to M_2 documents (in this case 15 documents). The largest gap occurs between ranks 7 and 8 — therefore p_{min} is taken to be 0.2475.

The classification vector (called the centroid) is formed by merging the document vectors of documents having correlations above p_{min} (0.2475). The centroid, composed of concepts and weights, is shown in Fig. 4. This centroid is then correlated with all previously unclustered documents (Fig. 3). A second cutoff correlation p_{min} is calculated to determine which documents belong in cluster 2. Here the greatest correlation difference (starting at M_1 and checking until M_2) occurs between the documents ranked 11 and 12. Therefore p_{min} becomes 0.3859, and the top 11 documents are included in cluster 2. These documents are listed as the "11 Relevant" in Fig. 4.

DCLSTR uses a variation of Doyle's Algorithm. The following description of the algorithm covers the main points. [5] Assume that the document set is arbitrarily partitioned into m clusters, where S_j is the set of documents in cluster j. Associated with each set S_j is a corresponding concept vector C_j and frequency vector F_j. The concept vector consists of all the concepts occurring in the documents of S_j, and the frequency vector specifies the number of documents in S_j in which each concept occurs.

Every concept in C_j is assigned a rank according to its frequency; i.e., concepts with the highest frequency have a rank of 1, concepts with the next highest frequency receive a rank of 2, etc. Given an integer b (base value), every concept in D_j is assigned a rank value equal to the base value minus the rank of that concept. The vector of rank values is called the profile P_j of the set S_j. Fig. 5 illustrates the concept and frequency vectors, and the corresponding profiles for a sample document collection.

Starting from a partition of the document set into m clusters, the profiles are generated as described. Every document d_i in the document space is now scored against each of the m profiles by a scoring function g, where $g(d_i,P_j)$ equals the sum of the rank values of all the concepts from d_i which occur in C_j. Fig. 5 shows the results of scoring the documents in the sample collection against the profiles from Fig. 5.

A new partition of the document set into $m+1$ clusters is then made by the following formula:

$$S_j = \{d_i \mid g(d_i,P_j) \geq T_i\} \qquad 1 \leq j \leq m$$

$$T_i = \begin{cases} H_i - [a \cdot (H_i - T)] & \text{if } H_i > T \\ T & \text{otherwise} \end{cases}$$

where

$$H_i = \max(g(d_i,P_j))$$

$$0 \leq a \leq 1$$

$$T = a \quad \text{is the given cutoff value .}$$

Those documents which do not fall into any of the m clusters S_j are called loose documents, and they are assigned to a special class L. The process is

d_1	d_2	d_3	d_4	d_5	d_6	d_7
c_1	c_1	c_1	c_1	c_1	c_3	c_6
c_2	c_2	c_7	c_2	c_8		c_8
c_5	c_4	c_8	c_3			
	c_5		c_5			

a) Documents

S_1	C_1	F_1	P_1	S_2	C_2	F_2	P_2	S_3	C_3	F_3	P_3
d_1	c_1	3	5	d_2	c_1	2	5	d_6	c_3	1	5
d_3	c_2	1	3	d_4	c_2	2	5	d_7	c_6	1	5
d_5	c_5	1	3		c_3	1	4		c_8	1	5
	c_7	1	3		c_4	1	4				
	c_8	2	4		c_5	2	5				

b) Initial Clusters, Profiles, and Frequencies

Construction of Profiles from Documents
(base value = 6)

Document	Profile of Highest Score	Score
d_1	2	15
d_2	2	19
d_3	1	12
d_4	2	19
d_5	1	9
d_6	3	5
d_7	3	10

S'_1	S'_2	S'_3	L
d_3	d_1	d_7	d_5
	d_2		d_6
	d_4		

b) Resulting Clusters

One Iteration of Doyle's Classification Algorithm
(cutoff = 10)

Fig. 5

now repeated after replacing P_j by P'_j . The iteration continues until S_j satisfies the termination condition that $S'_j = S_j$ (actually $S^{*\prime}_j = S^*_j$, where S^*_j is the subset of S_j consisting of all those documents that score highest against profile P_j).

Basically, this algorithm matches documents to existing clusters by computing a document-cluster score for each document with respect to each cluster, and placing a document into those clusters for which a sufficiently high score is obtained. The clusters are then updated to include the new documents. In each iteration all the documents are correlated with all the clusters, and the clusters are updated until further updating does not alter the group of documents in each cluster. This updating is shown in list form in Figs. 6 and 7. The 12 profiles (clusters) of Fig. 6 are matched against the documents, and updated to become the profiles of Fig. 7.

It should be noted that the document clustering process can be extended to the clustering of clusters. That is, if one of the two clustering algorithms generates m groups of documents, these m groups could be grouped together, as if they were documents, into n clusters, where $1 \leq n \leq m$. These n clusters could then be grouped together, and so on, until a hierarchical cluster tree is formed as shown in Fig. 8. At present no routines for automatically constructing such multi-level cluster trees exist in the SMART system, although such an algorithm is planned for implementation in the near future. Both CLUSTR and DCLUSTR generate the first level of the cluster trees, thus representing special cases of more general tree construction routines.

C) The Selection of Documents to be Searched

The search process consists of four steps. First a search query is

```
      THE DOCUMENTS IN PROFILE  1 ARE
 67    71    80    81    82    83    84    87   100   102   128

      THE DOCUMENTS IN PROFILE  2 ARE
 20    64    65    66    68    70    85    86   103   122   124   169   196

      THE DOCUMENTS IN PROFILE  3 ARE
 45    74    75    76    77    79   112   135   154   161

      THE DOCUMENTS IN PROFILE  4 ARE
  9    17    23    62   116   117   134   146   147   151   180   197

      THE DOCUMENTS IN PROFILE  5 ARE
  3    61    90    93    94   110   113   120   181

      THE DOCUMENTS IN PROFILE  6 ARE
 18    25    41    63    69   111   114   115   121   183   192   193   195

      THE DOCUMENTS IN PROFILE  7 ARE
  2    19    39   101

      THE DOCUMENTS IN PROFILE  8 ARE
  4    30    31    57    58   187   188

      THE DOCUMENTS IN PROFILE  9 ARE
 23    26    29    43    72    78    91    92    95   104   118   132   133
149   152   153   155   156   158   159   179   185

      THE DOCUMENTS IN PROFILE 10 ARE
 13    15    16    56    59    60   136   141   150   160   176   182   184
189   198
      THE DOCUMENTS IN PROFILE 11 ARE
  8   162   163   164   165   166   167   168

      THE DOCUMENTS IN PROFILE 12 ARE
 46    47    48    49    50    52
```

Original Profiles (Clusters)

Fig. 6

THE DOCUMENTS IN PROFILE 1 ARE
67 71 81 83 84 87 100 128

THE DOCUMENTS IN PROFILE 2 ARE
64 65 66 70 86 124

THE DOCUMENTS IN PROFILE 3 ARE
74 75 76 77 112 154 161

THE DOCUMENTS IN PROFILE 4 ARE
9 62 116 147 151 197

THE DOCUMENTS IN PROFILE 5 ARE
3 61 90 110 113 120 181

THE DOCUMENTS IN PROFILE 6 ARE
18 25 41 114 115 121 183 192 195

THE DOCUMENTS IN PROFILE 7 ARE
2 19 39

THE DOCUMENTS IN PROFILE 8 ARE
4 31 57 58 187 188

THE DOCUMENTS IN PROFILE 9 ARE
28 95 132 133 158 159 179 185

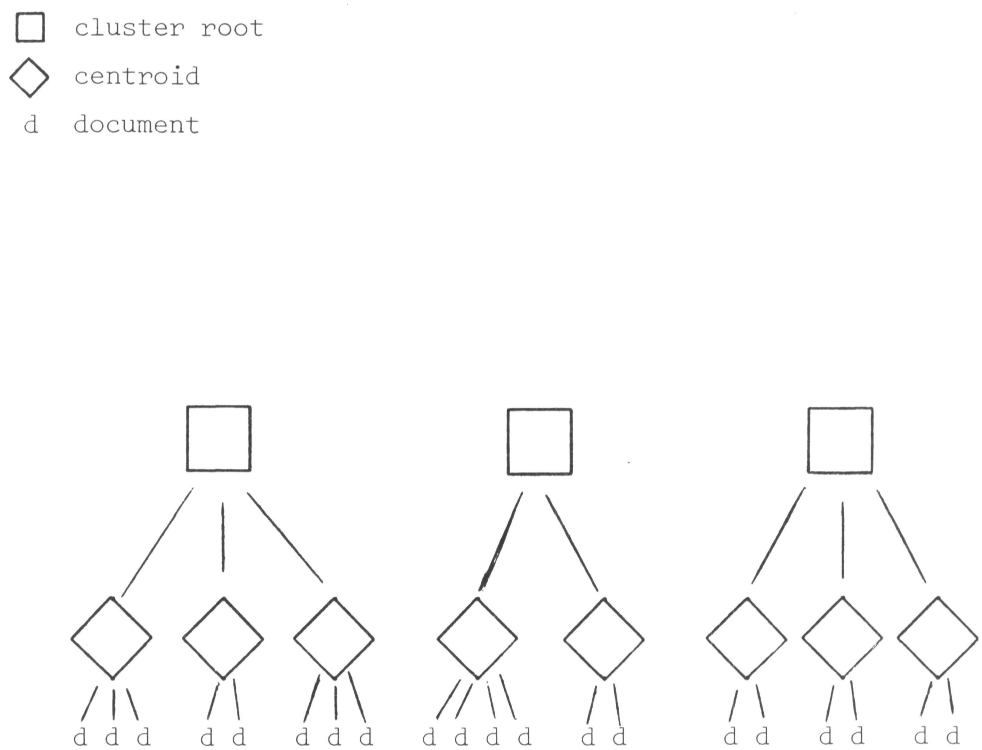
THE DOCUMENTS IN PROFILE 10 ARE
56 60 150 160 182 184 189 198

THE DOCUMENTS IN PROFILE 11 ARE
8 162 163 164 165 166 167 168

THE DOCUMENTS IN PROFILE 12 ARE
46 47 48 49

Updated Profiles (Clusters)

Fig. 7



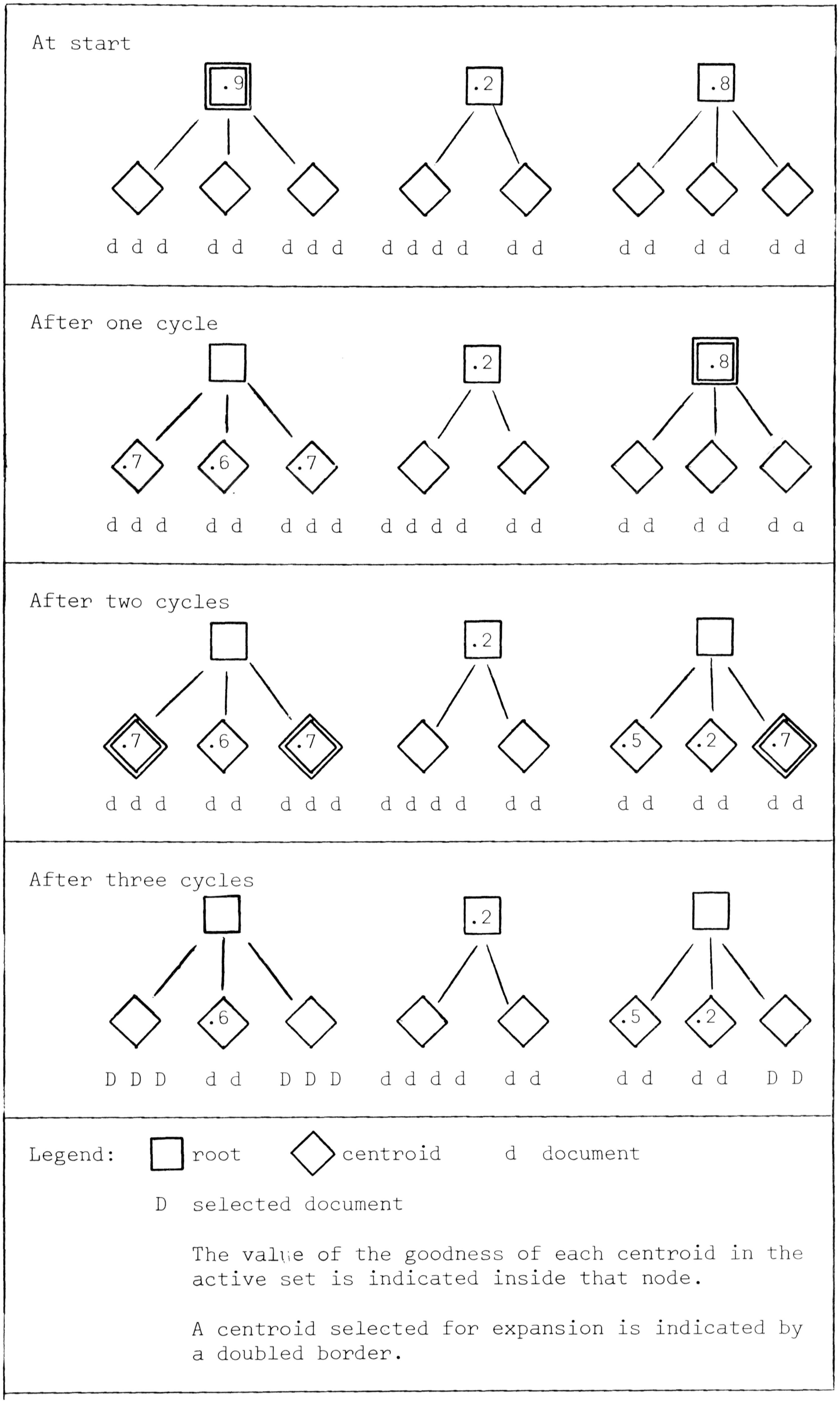
A Hypothetical Cluster Tree

Fig. 8

defined, either using the author's original query, or a modification of the original query, in numeric concept vector form. One important modification consists in using documents judged relevant by the author to modify the original query vector. This process is known as relevance feedback and is discussed in part D of this section.

Once a search query is defined, the set of documents to be correlated with the query is selected. SMART provides two options; either a full search or a tree search may be made. In a full search every document in the retrieval base is correlated with the query. In this case the selection of the documents to be searched is trivial — all documents in the collection are searched.

In the tree search [6], a set of documents is selected by a cyclic process, using a tree such as that pictured in Fig. 9. At any one time, a set of active nodes exists in the tree; initially this is the set of roots (the highest level of clusters). Each node in the active set is compared with the search query. The "goodness" of each node is defined from the relatedness of a query to a node, and from other information about the structure of a tree; the nodes of the "active" set are then ordered by this value of a "goodness". A subset of active nodes is selected as being most promising. The corresponding nodes are deleted from the active set, and the sons of these nodes (if centroids) are correlated with the query and become a part of the active set. Those sons which represent documents are then entered onto a list of documents to be used in subsequent correlations with the query. The active set is cyclically reordered and another group of nodes is selected to have its sons examined until some desired number of documents are located. The process used to obtain a list of specific docu-



Searching a Hypothetical Tree

Fig. 9

ments to be directly compared to a query is represented in Fig. 9.

The listing reproduced in Fig. 10 shows an example of input to the cluster searching routine. The first iteration (Iteration 0) uses a full search instead of a tree search. The second iteration (Iteration 1) represents a tree search on the cluster collection "CENTROID NO MORE" using the "COSINE" correlation. The desired number of documents to be selected for correlation with the query is given by "WANTED", where "WANTED" is defined as:

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

where

CORDOC	implies that at least "CORDOC" documents will be correlated in this iteration;
TIMALL	"TIMALL" times "ALLOF" (for this iteration) documents are additionally correlated in this iteration;
TIMREL	"TIMREL" times the number of relevant documents not yet retrieved are additionally correlated in this iteration;
TIMNMR	"TIMNMR" times "NOMOR" (for this iteration) documents are additionally correlated in this iteration.

"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

PARAMETERS FOR TREE SEARCHING

ITERATION 0 A FULL SEARCH RATHER THAN A TREE SEARCH IS BEING DONE.

ITERATION 1	COLLECTION	CENTROID	NO MORE		CORRELATION	COSINE		
WANTED	CORDOC	2	TIMALL	2	TIMREL	0	TIMNMR	0
GOODNESS	MCN	1.00	PCN	1.00	MLV	0.10	PLV	1.00
	MALL	0.0	PALLUF	0.0	PALLCN	0.0	PALLLV	0.0
	MCFCN	0.0	PFCFCN	0.0	PNCFCN	0.0		
	MCNLV	0.0	PNCNLV	0.0	PVCNLV	0.0		
	MCFLV	0.0	PFCFLV	0.0	PVCFLV	0.0		
SELECTION	MINNOD	1	MAXNOD	3	GAP	0.10	EPSLON	0.05
REJECTION	MNGOOD	0.10	MNCORR	0.05	PERCOL	0.0 %	TIMWAN	0.0

Parameters for Tree Searching

Fig. 10

AUTHOR	QUERY	ITER	DOCUMENTS WANT	DOCUMENTS HAVE	DOCUMENTS NEED	CENTROID	GOODNESS	CROWNS CROWN	CROWNS LEVEL	CORRELATION
BATCH	34-1	1	12	8	4	5	0.3157	21.	1.00	0.3157

AUTHOR	QUERY	ITER	WANT	HAVE	NEED	CENTROID		SON	SON	SON	SON	SON
BATCH	34-1	1	12	29	-17	5						
							21 DOCUMENT SONS -	2	4	7	8	9
								11	12	17	19	23
								24	25	38	40	42
								46	48	67	69	70
								71				

Selection and Expansion of Third Set of Nodes for Query 34

Fig. 14

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 "goodness" of each node (the parameter value used to rank the nodes) may also be controlled by the user through 17 parameters as follows.

$$\begin{aligned}\text{"GOODNESS"} = {} & \text{"COEF"} + \text{"MCN"} \times \text{"CROWN"}^{\text{"PCN"}} + \text{"MLV"} + \text{"LEVEL"}^{\text{"PLV"}} \\ & + \text{"MCFCN"} \times \text{"COEF"}^{\text{"PFCFCN"}} \times \text{"CROWN"}^{\text{"PNCFCN"}} \\ & + \text{"MCNLV"} \times \text{"CROWN"}^{\text{"PNCNLV"}} \times \text{"LEVEL"}^{\text{"PVCNLV"}} \\ & + \text{"MCFLV"} \times \text{"COEF"}^{\text{"PFCFLV"}} \times \text{"LEVEL"}^{\text{"PVCFLV"}} \\ & + \text{"MALL"} \times \text{"COEF"}^{\text{"PALLCF"}} \times \text{"CROWN"}^{\text{"PALLCN"}} \times \text{"LEVEL"}^{\text{"PALLLV"}}\end{aligned}$$

where "COEF" is the correlation value (usually cosine) between the node and the query, "CROWN" is the number of nodes that are the sons of the node, and "LEVEL" is the level of the node. For example, the node in Fig. 9 with a "goodness" of 0.9, has a "CROWN" of 11 and a "LEVEL" of 3.

It should be noted that the formula for "GOODNESS" 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 "GOODNESS" being usually set to zero.

The size of the subset of active nodes to be expanded (after all active nodes are ranked by "goodness") is determined by additional parameters specified by the user, as printed in the listing (Fig. 10) under "SELECTION" and "REJECTION".

MINNOD MAXNOD	At least "MINNOD" nodes and not more than "MAXNOD" nodes are to be expanded for this iteration;
GAP	If there exist two nodes between "MINNOD" and "MAXNOD" which have a difference greater than "GAP", all nodes above that gap are expanded;
EPSLON	Any nodes within "EPSLON" of the last node selected for expansion are also to be expanded;
MNGOOD	Any node with a "GOODNESS" of less than "MNGOOD" is not to be retained for expansion;
MNCORR	Any node with a correlation less than "MNCORR" with the query is not retained for expansion;
PERCOL	Only nodes whose combined "CROWN" is greater than "PERCOL" percent of the size of the collection being searched need be retained for expansion.
TIMWAN	Only nodes whose combined "CROWN" is greater than "TIMWAN" times the number of documents to be correlated with are retained for expansion.

The selection of the documents using the parameters from Fig. 10 is shown in Figs. 11, 12, and 13. The queries are processed as a batch, and queries 31, 32, 33, and 34 are shown as examples. The queries are first matched against the "roots" of the centroid tree consisting of centroids 1, 2, and 3. The results of the matching and other useful statistics are shown in Fig. 11. The query number, iteration number, number of documents wanted and found, centroid used to match, "goodness" of the matching, statistics of the centroid, and the cosine correlation are given for each query match against all the roots. The "REJECTION" parameters are used here to eliminate centroids before any ranking is done on "goodness". A "MNGOOD" of 0.10 causes centroid 2 to be dropped from the active set of query 33, and centroid 3 to be dropped

TREE SEARCHING -- SELECTING THE DOCUMENTS WITH WHICH 17 QUERIES WILL BE CORRELATED.

AUTHOR	QUERY	ITER	DOCUMENTS WANT	HAVE	NEED	CENTROID	GOODNESS	CROWNS CROWN	LEVEL	CORRELATION	
BATCH	31-1	1	12	0	12	1	0.2375	26.	2.00	0.2375	
BATCH	32-1	1	12	C	12	1	0.1039	26.	2.00	C.1039	
BATCH	33-1	1	12	C	12	1				0.C	TO BE DROPPED (BY MINCOR)
BATCH	34-1	1	12	C	12	1	0.1419	26.	2.00	0.1419	
BATCH	31-1	1	12	0	12	2	0.2540	41.	2.00	0.2540	
BATCH	32-1	1	12	C	12	2	0.1111	41.	2.00	0.1111	
BATCH	33-1	1	12	C	12	2	0.0222	41.	2.00	C.0222	TO BE DROPPED (BY MNGOOD)
BATCH	34-1	1	12	C	12	2	0.2842	41.	2.00	0.2842	
BATCH	31-1	1	12	C	12	3	0.0689	25.	2.00	C.0689	TO BE DROPPED (BY MNGOOD)
BATCH	32-1	1	12	C	12	3	0.0502	25.	2.00	C.0502	TO BE DROPPED (BY MNGOOD)
BATCH	33-1	1	12	0	12	3				0.0	TO BE DROPPED (BY MINCOR)
BATCH	34-1	1	12	C	12	3	0.5314	25.	2.00	0.5314	

AUTHOR	QUERY	ITER	DOCUMENTS WANT	HAVE	NEED	CENTROID	GOODNESS	CROWNS NODES	EST.CUM	
BATCH	31-1	1	12	C	12					
						2	0.2540	41	41	TO BE EXPANDED
						1	0.2375	26	67	TO BE EXPANDED
BATCH	32-1	1	12	C	12					
						2	0.1111	41	41	TO BE EXPANDED
						1	0.1039	26	67	TO BE EXPANDED
BATCH	34-1	1	12	C	12					
						3	0.5314	25	25	TO BE EXPANDED
						2	0.2842	41	66	TO BE RETAINED
						1	0.1419	26	92	TO BE RETAINED

First Selection of Nodes to be Expanded

Fig. 11

AUTHOR	QUERY	ITER	WANT	HAVE	NEED	CENTROID		SON	SON	SON	SON
BATCH	31-1	1	12	0	12	1	3 CENTROID SONS -	-7	-6	-4	
BATCH	32-1	1	12	0	12	1	3 CENTROID SONS -	-7	-6	-4	
BATCH	31-1	1	12	0	12	2	3 CENTROID SONS -	-10	-9	-5	
BATCH	32-1	1	12	0	12	2	3 CENTROID SONS -	-10	-9	-5	
BATCH	34-1	1	12	0	12	3	3 CENTROID SONS -	-12	-11	-8	

AUTHOR	QUERY	ITER	DOCUMENTS WANT	HAVE	NEED	CENTROID	GOODNESS	CROWNS CROWN	LEVEL	CORRELATION	
BATCH	31-1	1	12	0	12	4	0.0462	10.	1.00	0.0462	TO BE DROPPED (BY MNGOOD)
BATCH	32-1	1	12	0	12	4	0.1048	10.	1.00	0.1048	
BATCH	31-1	1	12	0	12	5	0.2480	21.	1.00	0.2480	
BATCH	32-1	1	12	0	12	5	0.0380	21.	1.00	0.0380	TO BE DROPPED (BY MNGOOD)
BATCH	31-1	1	12	0	12	6	0.2472	6.	1.00	0.2472	
BATCH	32-1	1	12	0	12	6	0.0865	6.	1.00	0.0865	TO BE DROPPED (BY MNGOOD)
BATCH	31-1	1	12	0	12	7	0.3085	10.	1.00	0.3085	
BATCH	32-1	1	12	0	12	7	0.0734	10.	1.00	0.0734	TO BE DROPPED (BY MNGOOD)
BATCH	34-1	1	12	0	12	8	0.2180	7.	1.00	0.2180	
BATCH	31-1	1	12	0	12	9	0.2688	6.	1.00	0.2688	
BATCH	32-1	1	12	0	12	9	0.0581	6.	1.00	0.0581	TO BE DROPPED (BY MNGOOD)
BATCH	31-1	1	12	0	12	10	0.2240	14.	1.00	0.2240	
BATCH	32-1	1	12	0	12	10	0.2078	14.	1.00	0.2078	
BATCH	34-1	1	12	0	12	11	0.6764	8.	1.00	0.6764	
BATCH	34-1	1	12	0	12	12	0.2084	10.	1.00	0.2084	

Expansion of First Set of Active Nodes

Fig. 12

AUTHOR	QUERY	ITER	DOCUMENTS WANT	DOCUMENTS HAVE	DOCUMENTS NEED	CENTROID	GOODNESS	CROWNS NODES	CROWNS EST.CUM	
BATCH	31-1	1	12	0	12					
						7	0.3085	10	10	TO BE EXPANDED
						9	0.2698	6	16	TO BE EXPANDED
						5	0.2480	21	37	TO BE EXPANDED
						6	0.2472	6	43	TO BE EXPANDED
						10	0.2240	14	57	TO BE EXPANDED
BATCH	32-1	1	12	0	12					
						10	0.2078	14	14	TO BE EXPANDED
						4	0.1048	10	24	TO BE RETAINED
BATCH	34-1	1	12	0	12					
						11	0.6764	8	8	TO BE EXPANDED
						2	0.2842	41	49	TO BE EXPANDED
						8	0.2180	7	56	TO BE RETAINED
						12	0.2084	10	66	TO BE RETAINED
						1	0.1419	26	92	TO BE RETAINED

AUTHOR	QUERY	ITER	WANT	HAVE	NEED	CENTROID		SON	SON	SON	SON	SON
BATCH	34-1	1	12	0	12	2	3 CENTROID SONS -	-10	-9	-5		
BATCH	31-1	1	12	21	-9	5	21 DOCUMENT SONS -	2	4	7	8	9
								11	12	17	19	23
								24	25	38	40	42
								46	48	67	69	70
								71				
BATCH	31-1	1	12	27	-15	6	6 DOCUMENT SONS -	1	14	27	37	65
								80				
BATCH	31-1	1	12	37	-25	7	10 DOCUMENT SONS -	5	13	24	52	53
								56	59	67	79	81
BATCH	31-1	1	12	43	-31	9	6 DOCUMENT SONS -	1	18	33	55	73
								77				
BATCH	32-1	1	12	14	-2	10	14 DOCUMENT SONS -	1	2	11	16	22
								29	32	39	47	50
								51	63	66	62	
BATCH	34-1	1	12	8	4	11	8 DOCUMENT SONS -	6	15	28	30	34
								41	58	61		

Selection of Second Set of Nodes to be Expanded
and Expansion of These Nodes

Fig. 13

from the active set of queries 31 and 32. Similarly, a "MNCORR" of 0.05 causes centroid 1 and centroid 3 to be dropped from the active set of query 33. The centroids remaining in the active set of each query are the [illegible]ked and the "SELECTION" parameters used to select nodes to be expanded (Fi[illegible]1). Note that query 33 has no active set remaining and therefore is dropped from further searching (a more careful set of parameters for "goodness" would have eliminated this problem).

The "SELECTION" parameters indicate that at least 1 centroid should be expanded, and up to 3 centroids may be expanded until a gap of 0.10 in "goodness" occurs. Query 34 exhibits such a gap between centroids 3 and 2; hence only centroid 3 is selected for expansion.

The expansion of centroids is shown in Fig. 12. Query 31 and 32 both now have an active set of 6 centroids; query 34 has 3 centroids in its active set. Again the active sets are matched against their respective queries and the "REJECTION" parameters are applied. This time one centroid (centroid 4) is dropped from the active set of query 31, 4 centroids (centroids 5, 6, 7, and 9) are dropped from the active set of query 32, and no centroids are dropped from the active set of query 34.

Fig. 13 shows the selection of centroids to be expanded from among the active sets. Again applying the "SELECTION" parameters, no gap greater than 0.1 occurs within the first 3 centroids (centroids 7, 9, and 5) for query 31; furthermore, centroids 6 and 10 have a goodness within the "EPSLON" of 0.05, and hence are also selected for expansion. A gap greater than 0.1 occurs between the first and second centroids (centroids 10 and 4) for query 32; therefore only centroid 10 is to be expanded. Query 34 has a gap in goodness greater than 0.1 between centroids 2 and 8; thus only centroids

11 and 12 are expanded.

The expansion for query 31 produces 43 document sons, easily satisfying the need for 12 documents. Query 32 finds 14 document sons during expansion, and the need for 12 documents is again satisfied. Query 34, however, finds only 8 document sons and 3 centroid sons on expansion; it thus becomes necessary to search further to find the additional four documents. The selection and expansion of a third set of nodes for query 34 is shown in Fig. 14. Here, expansion of centroid 5 produces 21 document sons for query 34, thus filling the total requirement of 12 documents.

It should be noted that the selection of documents to be searched is not equivalent to final searching. For example, query 31 must be processed in a regular search against the 43 documents selected (instead of performing a full search using the entire document collection). The averaged results of the searching runs are shown in section E.

D) The Searching of the Document Groups

Once the documents to be searched in a given iteration are selected, the query used in the search process is constructed. The search query is generated using concept numbers from four distinct sources: the author's original query, documents which the author considers relevant before the search is started, specific concepts and weights which the author would like to add to the query, and relevance feedback information from previous search iterations (if any exists). Information from the first three sources is contained for iteration 0 (first search hence no feedback information) in Fig. 15. The following information is given for each query: the authors, the query number, the iteration number, the sources of the query and the corresponding document numbers, multipliers for these documents (all weights of

```
AUTHOR      QUERY#   ITER  SOURCE                DOC#  MULT     DOC#  MULT     DOC#  MULT     DOC#  MULT     DOC#  MULT

BATCH          1-1    0
                           ORIGINAL QUERY           (    1)
                           AUTHOR SUPPLIED         3(    2)     10(    2)
                           USER SUPPLIED          CON#  WGHT    CON#  WGHT     CON#  WGHT     CON#  WGHT     CON#  WGHT
                             CONS&WGHTS            2(   14)

BATCH          2-1    0
                           ORIGINAL QUERY           (    1)
                           AUTHOR SUPPLIED        12(    2)     29(    3)     47(    2)     61(    2)     76(    2)

                           USER SUPPLIED          CON#  WGHT    CON#  WGHT     CON#  WGHT     CON#  WGHT     CON#  WGHT
                             CONS&WGHTS            2(   79)

BATCH          3-1    0
                           ORIGINAL QUERY           (    1)
                           AUTHOR SUPPLIED        10(    2)
                           USER SUPPLIED          CON#  WGHT    CON#  WGHT     CON#  WGHT     CON#  WGHT     CON#  WGHT
                             CONS&WGHTS            1(   12)     2(   12)      3(   12)
```

First Construction of Search Queries

Fig. 15

```
SEARCH--CHECKING AND PRINTING OF CONTROL CARDS FOR SEARCH PARAMETERS.

OPTIONS FOR SEARCH              0 ARE:

ORIG MULT  PREV MULT  MIN CORR      TYPE CORR     NORMAL     ITEMS&MULTS  CONS&WGHTS    FREEZE/FLUID  UNITVC     WGHTS DRPED  PER DRPED
    1          0      0.0150        COSINE        ABNORMAL       YES          YES          FLUID      BY WORD    POS.AVE.       99.90

POS MULT   NEG MULT   POS RANK CUT  NEG RANK CUT  POS CORR CUT  NEG CORR CUT  POS ATLEST  NEG ATLEST  POS NOMORE  NEG NOMORE
   0          0           0             0            1.0000        1.0000          0           0           0           0

UNLESS     STOPALL    PREC CUTOFF   PDEFIN
   C          NO         0.0           ILS

OPTIONS FOR SEARCH   EHW         1 ARE:

ORIG MULT  PREV MULT  MIN CORR      TYPE CORR     NORMAL     ITEMS&MULTS  CONS&WGHTS    FREEZE/FLUID  UNITVC     WGHTS DRPED  PER DRPED
    0          1      0.0           COSINE        ABNORMAL       NO           NO           FLUID      BY WORD    POS.NON.       0.0

POS MULT   NEG MULT   POS RANK CUT  NEG RANK CUT  POS CORR CUT  NEG CORR CUT  POS ATLEST  NEG ATLEST  POS NOMORE  NEG NOMORE
   1         -1           5             2            1.0000        1.0000          0           0           5           2

UNLESS     STOPALL    PREC CUTOFF   PDEFIN
   2          YES        0.0           ILS
```

Options for Searching

Fig. 16

concepts in a given document are multiplied by the specified multiplier), and the query concept numbers and their weights. For example, query one uses the concepts from the original query (with all weights multiplied by one), plus all the concepts from documents 3 and 10 (all weights in both documents being multiplied by 2), plus concept 12 with a weight of 14. Query 1 is then defined by the combination of all these concepts and their weights.

Following the initial search query set-up, further modifications can be made before the search is started. The user specifies the type of modification to be made by introducing parameters as shown in Fig. 16.

The options for the query modification are listed in Fig. 16, one section being devoted to each iteration. The parameters are defined as follows:

ORIG MULT	Multiplier of original query.
PREV MULT	Multiplier of query used for previous iteration.
MIN CORR	Parameter controlling retrieval. Any document with a correlation less than (or equal to) "MINCOR" is not shown to 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. As usual, a blank field is equivalent to zero.
TYPE CORR	The type of correlation to be used. If blank, the correlation of the previous iteration is used. If blank for the zeroth iteration, 'COSINE' is substituted. At present, "COSINE" is the only available correlation.

NORMAL	If this field contains the word 'NORMAL' for each definition, "RMULT" is divided by the number of relevant used in that definition. "NMULT" is likewise divided by the number of nonrelevant used in feedback.
ITEMS & MULTS	This field contains 'YES' if specific items and multipliers are given for each and every query in this iteration.
CONS & WGHTS	This field contains 'YES' if a specific vector of concepts and weights is supplied for each and every query in this iteration.
FREEZE/ FLUID	If this field contains 'FREEZE', the items seen by the user defining the query are frozen in the order seen. Otherwise, all rank positions are available and all documents are correlated.
UNITVC	If this field contains the words "BY WORD", the weights of a given vector are not normalized. 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 produces the same weight for all documents, 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), the multipliers must be set at a high value 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.

WGHTS DRPED	This field is of the form '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 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, say 12. The latter method permits dropping all weights less than a certain percentage of the average weight. For example, if all concepts less than 90% of the average weight are dropped from normal composites, 75% of the concepts are deleted, but only 40% of the weight of the composite is lost.
PER DRPED	(See above. This is a floating point number and must be punched with a decimal point.)

The second line of Fig. 16 covers parameters used for relevance feedback, and not for the initial iteration, although the values are printed. Definitions for the second and third lines are covered in the discussion for the second iteration.

For query 1, the user-supplied parameters call for a multiplier of the query of 1, nonnormalized vectors, additional items and multipliers, additional concepts and weights, no normalizing of weights ("UNITVC" = "BY WORD"), and dropping of all concepts whose absolute weight is less than

(99.9 * sum of the absolute weights) / (100.0 * the number of unique concepts).

= 99.9 * 962 / 1000 * 23

In query 1, concepts with a weight smaller than 40 are dropped in accordance with the specifications of Fig. 16. The display of Fig. 17 for query 1 shows that the original 23 concepts (formed by combining the concept vectors of the orignal query plus author-supplied documents and author-supplied concepts and weights) are reduced to the six concepts shown in the figure.

These modified queries are then correlated with every document in the group previously selected (in this case a full search of the entire collection is made). After the queries are correlated with all the documents, documents having a correlation greater than 0.015 ("MIN CORR" for this iteration) are ranked. The top 30 documents retrieved are listed in Fig. 18. The first two lines of the listing contain the titles for the iteration, the query title and the relevant items for the query. (At present, the document relevance is pre-judged and held constant for all runs using a given query collection.) The major section of the page contains the correlation and rank of the documents retrieved for each iteration. The recall and precision values (defined in section E) obtained after retrieval of the given document are also given. For example, document 69 is the first document retrieved for query one in the first iteration (iteration 0). The correlation coefficient of this document with the query is 0.3924, and the recall and precision values after the retrieval of document 3 are 0.0 and 0.0 respectively. Similarly, document 17, a relevant document, is retrieved with rank 2, and its correlation with the query is 0.3430. The recall and precision values after retrieval of document 17 are 0.333 and 0.333 respectively.

```
FOR COMPOSITE          1, WEIGHTS WERE DROPPED BY (POS.AVE., 99.9).   ORIGINALLY,   23 CONCEPTS HAD A WEIGHT SUM OF    962.
                          WEIGHTS WERE TESTED AGAINST   40.         THERE ARE NOW    6 CONCEPTS AND A WEIGHT SUM OF    600.


ITEM    1    A01 TITLES PROB IN MAKING DESCRIPT - DIFF IN AUTO RETR  R

              CON   WT     CON   WT     CON   WT     CON   WT     CON   WT     CON   WT
                1  168      10  120      11   48      15   72      93   72     533  120

             THE   6 CONCEPTS ABOVE HAVE A SUM OF ABSOLUTE WEIGHTS =  600 WITH A ROOT SUM OF SQUARED WEIGHTS = 264.00


FOR COMPOSITE          2, WEIGHTS WERE DROPPED BY (POS.AVE., 99.9).   ORIGINALLY,   61 CONCEPTS HAD A WEIGHT SUM OF   2971.
                          WEIGHTS WERE TESTED AGAINST   47.         THERE ARE NOW   22 CONCEPTS AND A WEIGHT SUM OF   1987.


ITEM    2    A02 FACT   PERTINENT DATA RETR. AUTO IN RESPONSE TO REQ E

              CON   WT     CON   WT     CON   WT     CON   WT     CON   WT     CON   WT     CON   WT     CON   WT     CON   WT
                1  204       2   79       4   72       5  132       8   72       9   96      11   72      18   96      19  132
               46   72      48  132      81  144     116  192     126   48     134   48     135   72     147   48     154   48
              180   72     291   60     304   48     530   48

             THE  22 CONCEPTS ABOVE HAVE A SUM OF ABSOLUTE WEIGHTS = 1987 WITH A ROOT SUM OF SQUARED WEIGHTS = 474.01


FOR COMPOSITE          3, WEIGHTS WERE DROPPED BY (POS.AVE., 99.9).   ORIGINALLY,   11 CONCEPTS HAD A WEIGHT SUM OF    444.
                          WEIGHTS WERE TESTED AGAINST   39.         THERE ARE NOW    4 CONCEPTS AND A WEIGHT SUM OF    300.


ITEM    3    A03 INFORM WHAT IS I SCIENCE - GIVE DEFINITIONS

              CON   WT     CON   WT     CON   WT     CON   WT
                1   96      10   84      15   48      93   72

             THE   4 CONCEPTS ABOVE HAVE A SUM OF ABSOLUTE WEIGHTS =  300 WITH A ROOT SUM OF SQUARED WEIGHTS = 154.14
```

The Construction of the Vectors
for the First Iteration

Fig. 17

SMART--TEST OF SEARCH OF ENTIRE ADI COLLECTION PAGE 55 04/17/69 04:43:21.51 33.4300

LEGEND RUN 0 - SCH1 RUN 1 - SCH2 RUN 2 - SCH3 RUN 3 - SCH4

QUERY 1 - A01 TITLES PROB IN MAKING DESCRIPT - DIFF IN AUTO RETR R THE 3 RELEVANT ITEMS BEING 17 46 62

CORRELATIONS 0	1	2	3	RANK	DOCUMENTS 0	1	2	3	RECALL 0	1	2	3	PRECISION 0	1	2	3
0.3924	0.9259			1	69	17R			0.0	0.333			0.0	1.000		
0.3430	0.3250			2	17R	4			0.333	0.333			0.500	0.500		
0.3087	0.2577			3	4	40			0.333	0.333			0.333	0.333		
0.2858	0.2561			4	27	71			0.333	0.333			0.250	0.250		
0.2654	0.2407			5	11	46R			0.333	0.667			0.200	0.400		
0.2259	0.2331			6	71	69			0.333	0.667			0.167	0.333		
0.2146	0.2250			7	47	25			0.333	0.667			0.143	0.286		
0.2123	0.2169			8	46R	68			0.667	0.667			0.250	0.250		
0.1945	0.2117			9	57	30			0.667	0.667			0.222	0.222		
0.1788	0.2085			10	19	47			0.667	0.667			0.200	0.200		
0.1778	0.1978			11	62R	11			1.000	0.667			0.273	0.182		
0.1742	0.1963			12	23	66			1.000	0.667			0.250	0.167		
0.1742	0.1865			13	30	24			1.000	0.667			0.231	0.154		
0.1617	0.1853			14	81	56			1.000	0.667			0.214	0.143		
0.1358	0.1852			15	2	26			1.000	0.667			0.200	0.133		
0.1345	0.1838			16	70	1			1.000	0.667			0.188	0.125		
0.1324	0.1797			17	1	43			1.000	0.667			0.176	0.118		
0.1286	0.1787			18	39	12			1.000	0.667			0.167	0.111		
0.1213	0.1771			19	22	16			1.000	0.667			0.158	0.105		
0.1196	0.1718			20	75	49			1.000	0.667			0.150	0.100		
0.1180	0.1702			21	15	22			1.000	0.667			0.143	0.095		
0.1151	0.1679			22	14	45			1.000	0.667			0.136	0.091		
0.1078	0.1657			23	77	59			1.000	0.667			0.130	0.087		
0.0990	0.1622			24	64	28			1.000	0.667			0.125	0.083		
0.0980	0.1622			25	34	19			1.000	0.667			0.120	0.080		
0.0947	0.1571			26	25	81			1.000	0.667			0.115	0.077		
0.0857	0.1558			27	61	53			1.000	0.667			0.111	0.074		
0.0835	0.1411			28	28	23			1.000	0.667			0.107	0.071		
0.0808	0.1382			29	55	21			1.000	0.667			0.103	0.069		
0.0797	0.1361			30	51	38			1.000	0.667			0.100	0.067		
	0.0768			56		62R				1.000				0.054		

	DOC.CORR	CENT.CORR	DROP DOC	CORR.RANK	OLD.DOC	OLD RELDOC	NEW DOC	POS.FEED	NEG FEED	QUERY CORR	REC.CEIL
RUN 0	82	0	23	59	0	0	0	1	0	1.0000	0.0
RUN 1	82	0	5	77	0	0	5	2	1	0.5336	0.3333

Retrieval Results for Query 1

Fig. 18

The last section of Fig. 18 contains various statistics for the run. These are defined as follows:

DOC. CORR	The total number of document-query correlations performed in the given iteration.
CENT. CORR	The total number of centroid-query correlations performed in the given iteration.
DROP DOC	The number of documents with a query-document correlation of less than "MIN CORR".
CORR. RANK	The number of documents with a query-document correlation of greater than or equal to "MIN CORR".
OLD. DOC	The total number of documents previously seen by the user.
OLD RELDOC	The total number of relevant documents previously seen by the user.
NEW DOC	The total number of documents (relevant and nonrelevant) shown to the user in this iteration.
POS. FEED	The number of items in the definition of the query with a positive multiplier for feedback.
NEG FEED	The number of items in the definition of the query with a negative multiplier for feedback.
QUERY CORR	The correlation of the query used in the present iteration with the original user query.
REC. CEIL	The recall ceiling seen by the user.

The listing of the retrieved relevant documents completes the first iteration of the search. At this point, the user makes relevance judgments, or, alternatively, prejudged relevance decisions are registered, and a new search query is constructed using information about the retrieved documents. The user-supplied instructions specifying what information is to be used, and how the new query is to be constructed are taken from the input parameters (shown in Fig. 16 in the second two lines under options for SEARCH 1). The definitions of the parameters are as follows:

POS MULT	All weights of the relevant documents used in feedback are multiplied by this number.
NEG MULT	All weights of the nonrelevant documents used in feedback are multiplied by this number. To signify that negative feedback is not desired "NEG MULT" is blank or zero.
POS RANK CUT	All relevant items with iteration ranks above "POS RANK CUT" according to the ordering of the previous iteration are used in defining the new query.
NEG RANK CUT	All nonrelevant items with iteration ranks above "NEG RANK CUT" according to the ordering of the previous iteration are used in defining the new query.
POS CORR CUT	All relevant items with a correlation above this value are also used. (This value must include a decimal point.) If "POS CORR CUT" is zero or blank, no relevant are selected due to this parameter.
NEG CORR CUT	All nonrelevant items with a correlation above this value are also used.
POS ATLEST	At least "POS ATLEST" relevant will be fed back (if they exist — i.e., more remain to be found).
NEG ATLEST	At least "NEG ATLEST" nonrelevant will be fed back.

POS NOMOR	However, no more than "POS NOMOR" items will be searched to provide the "POS ATLEST" relevant documents.
NEG NOMOR	However, not more than "NEG NOMOR" nonrelevant will be used. Note that only documents scanned in an attempt to locate relevant documents for positive feedback are used in attempting to find nonrelevant for negative feedback.
UNLESS	Negative feedback is done except when "UNLESS" relevant documents are found. If "UNLESS" are found, no negative feedback at all is done. To signify that no negative feedback is desired, "NEG MULT" should contain blanks or a zero. Should 'UNLESS' be left blank or set to zero, negative feedback is attempted regardless of the number of relevant actually used in positive feedback.
STOPALL	"STOPALL" is set to '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 satisfied. The default is 'NO'.
PREC CUTOFF	If the precision after "POS RANK CUT" documents is over "PREC CUTOFF", and if the precision after more items are judged drops below "PREC CUTOFF", the judging of documents ceases.
POEFIN	'SILENT' if search queries are not to be printed; 'STANDARD' if search queries are to be printed; 'DETAILS' if details of the search query definition process are to be printed (used only for debugging).

Using iteration 2 as an example, the new search query Q_{i+1} is defined by the following equation:

$$Q_{i+1} = (1)Q_i + (1)\sum_{j=1}^{n_r}(r_i)_j - (1)\sum_{j=1}^{n_s}(s_i)_j$$

where $(r_i)_j$ designates the concepts and weights of relevant document $(r_i)_j$; $(s_i)_j$ designates the concepts and weights of nonrelevant document $(s_i)_j$; Q_i is the previous query (for iteration i), and n_r and n_s are defined by the number of relevant documents retrieved and the number of nonrelevant documents retrieved, respectively.

In iteration 2, $n_r \leq 5$; therefore, only the top five documents are retrieved, and not all of them will be relevant (in most cases). If at least two relevant documents are retrieved among the top five documents, no negative feedback will be done ($n_s = 0$). If fewer than two relevant documents are found, any nonrelevant retrieved among the top two documents will be used for feedback ($n_s \leq 2$). This condition is stipulated by an "UNLESS" of 2 and a "NEG RANK CUT" of 2.

The newly defined search queries are shown in Fig. 19. For query 1, one relevant and four nonrelevant documents are retrieved in the top 5. The relevant document (17) and the only nonrelevant in the top two retrieved (69) are used to construct the new query. Query 2 retrieves 1 relevant, but the top 2 retrieved are both nonrelevant, so both (27 and 33) are used, together with the one relevant found during feedback. Query 3 finds 3 relevant in the top five retrieved; hence no negative feedback is used. The new query vectors (Fig. 20) are used for searching, and the results are shown in Fig. 18, second iteration.

AUTHOR	QUERY#	ITER	SOURCE	DOC#	MULT	DOC#	MULT	DOC#	MULT	DOC#	MULT	DOC#	MULT
BATCH	1-1	1											
			PREVIOUS QUERY	(	1)								
			REL & NON-REL USED	69(	-1)	17(	1)						
BATCH	2-1	1											
			PREVIOUS QUERY	(	1)								
			REL & NON-REL USED	27(	-1)	33(	-1)	71(	1)				
BATCH	3-1	1											
			PREVIOUS QUERY	(	1)								
			REL & NON-REL USED	60(	1)	43(	1)	3(	1)				

Redefinition of Search Query

Fig. 19

ITEM 1 A01 TITLES PROB IN MAKING DESCRIPT - DIFF IN AUTO RETR R

CON	WT	CON	WT	CON	WT	CON	WT	CON	WT	CON	WT	CON	WT	CON	WT	CON	WT
1	12	5	12	6	12	9	12	21	36	22	6	23	18	25	24	59	12
67	12	72	12	78	12	104	12	115	12	156	12	171	12	185	12	228	12
229	12	276	24	277	24	297	12	529	12								

THE 23 CONCEPTS ABOVE HAVE A SUM OF ABSOLUTE WEIGHTS = 336 WITH A ROOT SUM OF SQUARED WEIGHTS = 76.37

ITEM 2 A02 FACT PERTINENT DATA RETR. AUTO IN RESPONSE TO REQ E

CON	WT	CON	WT	CON	WT	CON	WT	CON	WT	CON	WT	CON	WT	CON	WT	CON	WT
3	12	5	12	8	12	17	36	21	30	25	12	26	12	36	12	41	12
43	12	62	12	63	12	64	12	72	12	158	12	171	12	193	12	196	12
223	12	266	12	271	24	284	12	291	12	294	12	444	12	481	6		

THE 26 CONCEPTS ABOVE HAVE A SUM OF ABSOLUTE WEIGHTS = 360 WITH A ROOT SUM OF SQUARED WEIGHTS = 77.30

ITEM 3 A03 INFORM WHAT IS I SCIENCE - GIVE DEFINITIONS

CON	WT	CON	WT	CON	WT	CON	WT	CON	WT	CON	WT	CON	WT	CON	WT	CON	WT
1	144	5	12	9	36	10	108	11	24	15	12	22	12	26	12	31	24
34	36	38	12	52	12	58	12	108	12	134	12	162	12	163	12	171	12
193	12	195	12	211	12	217	12	225	24	240	12	253	12	260	12	274	12
291	12	363	12	427	36	465	12	533	72								

THE 32 CONCEPTS ABOVE HAVE A SUM CF ABSOLUTE WEIGHTS = 780 WITH A ROOT SUM OF SQUARED WEIGHTS = 215.67

Construction of the Vectors
for the Second Iteration

Fig. 20

E) Search Evaluation

Several different evaluation measures are used in the SMART system, all based on the concepts of recall and precision. The definitions of these measures are the following:

$$\text{Recall} = \frac{a}{b}$$

$$\text{Precision} = \frac{a}{c}$$

where

a = the number of relevant documents retrieved

b = the number of relevant documents in the collection

c = the number of documents retrieved.

These measures are usually computed at a specified point during retrieval, usually either after a given number of documents have been retrieved, or after a given recall has been obtained.

Two types of averaging graphs, and four types of overall recall and precision averages are generated by the SMART system and listed in Figs. 21, 22, 24, and 25. Fig. 21 shows one type of graph and all four overall averages. At the top of the listing the runs being evaluated are identified (in this case a full search run (run 0) and a centroid search run (run 1)). Below are listed the recall levels being used, and the precision achieved at each recall level. The number of queries used in the averaging at each point is also given. For example, at recall level 0.10, run 0 shows a precision of 0.4948, but for only 2 queries a relevant document had been retrieved at that recall level.

RECALL--LEVEL AVERAGES

LEGEND: RUN 0 -- 35 QUERIES (PLUS 0 NULLS) -- SMART FULL SCH DOCUMENTATION RUN
RUN 1 -- 35 QUERIES (PLUS 0 NULLS) -- SMART CENT SCH DOCUMENTATION RUN

	RUN 0		RUN 1	
RECALL	NQ	PRECISION	NQ	PRECISION
0.0	0	0.4948	0	0.4813
0.05	1	0.4948	1	0.4813
0.10	2	0.4948	1	0.4803
0.15	5	0.4734	4	0.4620
0.20	13	0.4282	11	0.4197
0.25	18	0.4222	12	0.4148
0.30	18	0.4179	12	0.4073
0.35	24	0.3812	16	0.3797
0.40	24	0.3791	15	0.3680
0.45	24	0.3690	15	0.3599
0.50	31	0.3666	19	0.3599
0.55	31	0.2901	15	0.2900
0.60	31	0.2876	15	0.2895
0.65	31	0.2661	15	0.2800
0.70	30	0.1961	13	0.2154
0.75	29	0.1946	13	0.2154
0.80	29	0.1918	13	0.2102
0.85	25	0.1756	13	0.2102
0.90	24	0.1636	13	0.2034
0.95	24	0.1636	13	0.2034
1.00	28	0.1636	14	0.2034

	RUN 0	RUN 1
NORM RECALL	0.7024	0.4920
NORM PRECISION	1.0000	1.0000
RANK RECALL	0.2014	0.2208
LOG PRECISION	0.3435	0.3458

SYMBOL KEYS: NQ = NUMBER OF QUERIES USED IN THE AVERAGE NOT DEPENDENT ON ANY EXTRAPOLATION.
NORM = NORMALIZED.

Recall-Level Averages

Fig. 21

Below the recall-level averages the four overall averages are listed. These are described more extensively in reference [3] (chapter 8) and are briefly defined below: [7]

$$\text{Normalized Recall} = 1 - \frac{\sum_{i=1}^{n} r_i - \sum_{i=1}^{n} i}{n(N-n)}$$

$$\text{Normalized Precision} = 1 - \frac{\sum_{i=1}^{n} \log r_i - \sum_{i=1}^{n} \log i}{\log \frac{N}{(N-n)!n!}}$$

$$\text{Rank Recall} = \frac{\sum_{i=1}^{n} i}{\sum_{i=1}^{n} r_i}$$

$$\text{Log Precision} = \frac{\sum_{i=1}^{n} \log i}{\sum_{i=1}^{n} \log r_i}$$

where

n = number of relevant documents

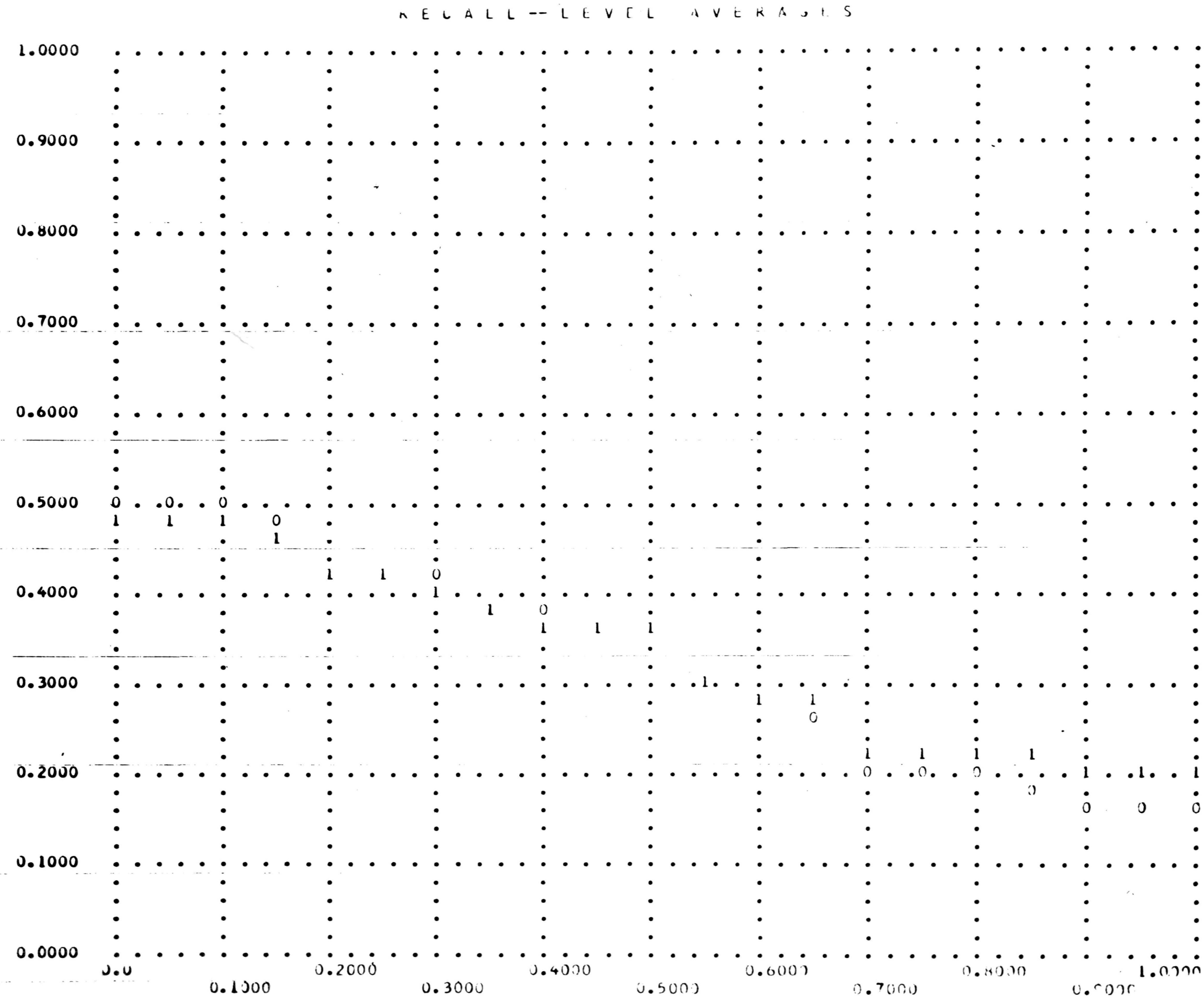
N = number of documents in collection

r_i = rank of i^{th} relevant document

i = ideal rank positions for the i^{th} relevant item.

Fig. 22 shows a computer-generated graph of the recall and precision averages previously given in Fig. 21.

In this type of graph, the precision is recorded for a given recall level.



THE Y-AXIS INCREMENT IS 0.2000E-01
Y-AXIS = PRECISION

THE X-AXIS INCREMENT IS 0.1000E-01
X-AXIS = RECALL

SYMBOL KEYS: 0 = RUN 0 1 = RUN 1

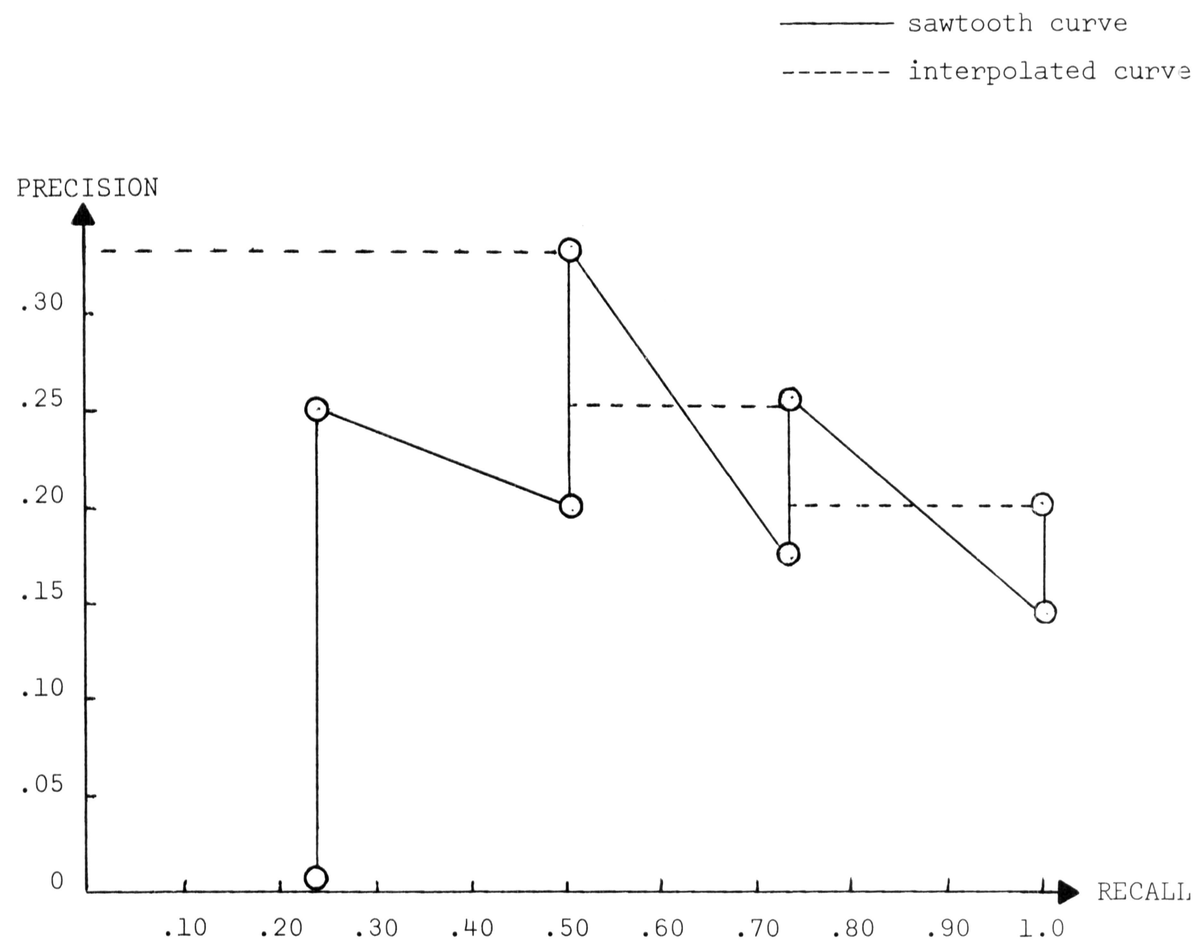
Recal-Level Averages

Fig. 22

For example, at a recall of 0.10 (10 percent of the relevant documents retrieved) run 0 shows a precision of 0.4948, and run 1 a precision of 0.4803. These precision values are the averages of the precision, at a given recall level, for all the queries searched. It should be noted that interpolation methods are needed to produce the averages, since all queries do not possess an exact precision value at each given recall level.

The graph of Fig. 23 shows the necessary interpolation for a hypothetical query with four relevant items. The relevant documents are assumed to be retrieved with ranks of 4, 6, 12 and 20. Thus, at 25 percent recall, the precision is 0.25; at 50 percent recall, the precision is 0.33, and so on. However, these values correspond actually to the highest possible precision points, since they are calculated just after a relevant document is retrieved. In this example, after 3 documents are retrieved, the precision is 0, after 5 documents, the precision is 0.20, and so on. This range of precision for each recall level is indicated by the top and bottom points in Fig. 23 at 25%, 50%, 75%, and 100% recall. The solid sawtooth line connecting these points is not used for interpolation; it is intended to indicate the drop in precision between the actual recall levels for this query, as more nonrelevant documents are retrieved.

The interpolation method actually used by the SMART system is based on the dashed lines shown in Fig. 23 where a horizontal line is led leftward from each peak point of precision, up to a point where a higher point of precision is encountered. This new curve (the dashed line in Fig. 23) does not lie above the sawtooth curve at all points. When the precision drops from one recall level actually achieved to the next, an immediate drop in precision after the first point to the level of the next point is



An Illustration of the Interpolation Method Used by the "Neo-Cleverdon" Recall-Precision Averages

Fig. 23

indicated. For example, in Fig. 23 the precision value at 0.50 recall is 0.33; but at 0.55 recall, the interpolated value used for the new averages is 0.25 precision. When the precision rises from one recall level to the next, however, the first precision point actually achieved is ignored for purposes of interpolation. The achieved precision of 0.25 at 0.25 recall in the example of Fig. 23 is ignored, and an interpolated precision of 0.33 is used for the averages for all recall levels from 0 to 0.50.

The second kind of average graph also generated is shown in Figs. 24 and 25. In this graph, the recall and precision are recorded and averaged after the retrieval of a given number of documents. For example, after one document has been retrieved in run 0, the average recall (over all the queries) is 0.0903, and the average precision is 0.3714. The recall and precision are averaged for 24 different cutoff points, and for 6 different percentage points, such as after 10 percent of the collection has been retrieved, etc. Three other statistics (besides the recall and precision) are measured and listed for each cutoff point. The first (NR) is the number of relevant documents retrieved at the given cutoff, and the second (CNP) is the cumulative number of relevant documents retrieved by this point. These values are included to aid in the proper evaluation of runs, since the document-level averages are not plotted at equal levels of recall for each query (as are the recall-level graphs). Also listed is the number of queries used to obtain the average at each point.

The final part of the evaluation process consists of tests of the significance of the differences between runs. Three basic statistical tests, the sign test, the T-test, and the Wilcoxon signed rank test, are calculated for each pair of search runs. All three statistical tests indicate whether

DOCUMENT--LEVEL AVERAGES

LEGEND: RUN 0 -- 35 QUERIES (PLUS 0 NULLS) -- SMART FULL SCH DOCUMENTATION RUN
RUN 1 -- 35 QUERIES (PLUS 0 NULLS) -- SMART CENT SCH DOCUMENTATION RUN

RANK	RUN 0 NR	RUN 0 CNR	RUN 0 NQ	RUN 0 RECALL	RUN 0 PRECISION	RUN 1 NR	RUN 1 CNR	RUN 1 NQ	RUN 1 RECALL	RUN 1 PRECISION
1	13	13	35	0.0903	0.3714	13	13	24	0.0903	0.3714
2	11	24	35	0.1927	0.3429	11	24	24	0.1927	0.3429
3	7	31	33	0.2265	0.3095	7	31	21	0.2265	0.3095
4	4	35	33	0.2389	0.2714	4	35	21	0.2389	0.2714
5	7	42	33	0.2831	0.2657	7	42	21	0.2831	0.2657
6	4	46	33	0.2977	0.2476	5	47	19	0.2986	0.2524
7	4	50	33	0.3170	0.2347	5	52	18	0.3236	0.2429
8	3	53	33	0.3258	0.2214	2	54	17	0.3315	0.2276
9	1	54	33	0.3315	0.2048	3	57	17	0.3417	0.2188
10	1	55	33	0.3324	0.1914	2	59	17	0.3457	0.2090
11	6	61	33	0.3674	0.1935	7	66	17	0.3887	0.2139
12	1	62	32	0.3710	0.1853	3	69	13	0.4086	0.2117
13	2	64	32	0.3783	0.1805	3	72	11	0.4247	0.2103
14	5	69	32	0.4018	0.1826	4	76	11	0.4397	0.2111
15	4	73	32	0.4336	0.1824	4	80	10	0.4556	0.2124
16	2	75	31	0.4717	0.1791	4	84	9	0.4692	0.2141
17	4	79	30	0.4950	0.1796	1	85	7	0.4739	0.2105
18	2	81	29	0.5015	0.1774	3	88	6	0.4918	0.2105
19	4	85	29	0.5238	0.1784	0	88	4	0.4918	0.2069
20	1	86	28	0.5286	0.1752	2	90	4	0.4997	0.2066
30	20	106	24	0.6571	0.1657	3	93	0	0.5243	0.1953
50	28	134	13	0.8120	0.1626	0	93	0	0.5243	0.1835
75	21	155	2	0.9430	0.1603	16	109	0	0.5434	0.1837
100	15	170	0	1.0000	0.1636	61	170	0	1.0000	0.2034
	0	170				0	170			
10.0%	53	53	33	0.3258	0.2214	54	54	17	0.3315	0.2276
25.0%	33	86	28	0.5286	0.1752	36	90	4	0.4997	0.2066
50.0%	38	124	16	0.7576	0.1643	3	93	0	0.5243	0.1874
75.0%	22	146	6	0.8862	0.1619	0	93	0	0.5243	0.1803
90.0%	7	153	2	0.9145	0.1601	12	105	0	0.5364	0.1826
100.0%	17	170	0	1.0000	0.1636	65	170	0	1.0000	0.2034

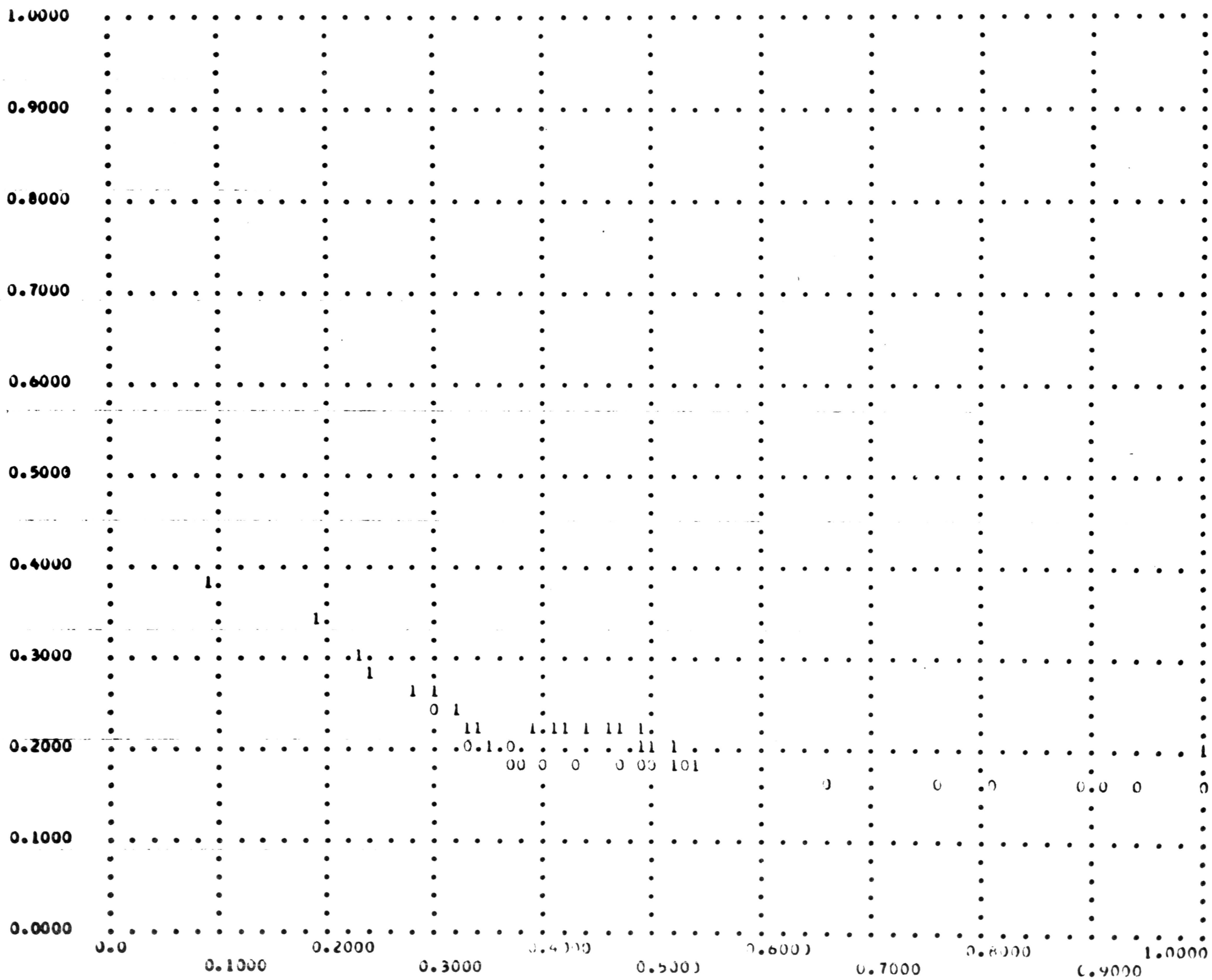
SYMBOL KEYS: NR = NUMBER OF RELEVANT.
CNR = CUMULATIVE NUMBER OF RELEVANT.
NQ = NUMBER OF QUERIES USED IN THE AVERAGE NOT DEPENDENT ON ANY EXTRAPOLATION.
% = PERCENT OF TOTAL NUMBER OF ITEMS IN COLLECTION.

Document-Level Averages

Fig. 24

SMART—DOCUMENTATION RUN PAGE 131 06/16/69 14:19:10.73 100.2900

D O C U M E N T -- L E V E L A V E R A G E S



THE Y-AXIS INCREMENT IS 0.2000E-01
Y-AXIS = PRECISION

THE X-AXIS INCREMENT IS 0.1000E-01
X-AXIS = RECALL

SYMBOL KEYS: 0 = RUN 0 1 = RUN 1

Document-Level Averages

Fig. 25

a given difference in two averages is likely to have occurred by chance.

A one-sided test is designed to compare a supposedly better sample B, with a given standard sample A. Specifically, one proposes two hypotheses H_0 and H_1. H_0 states that two samples A and B are produced by the same distribution; H_1 states that sample B is statistically better than sample A. H_1 is accepted if it is unlikely, under H_0, that a difference between samples as great as, or greater than, that observed would occur by chance.

A two-sided test similarly compares two samples under the same H_0, but with the alternate hypothesis H_1 being that samples A and B are from different distributions. Here again H_1 is accepted if the probability, under H_0, is low that a difference between two samples is as great as, or greater than, that observed would occur by chance.

The T-test assumes that the differences d_i between the two measures, a_i and b_i, are distributed normally. Explicitly, it is assumed that d_i has mean $\overline{d}$, and standard deviation σ_d. Note that $\overline{d}$ and σ_d are computable for any distribution, including also the normal distribution. In particular, it is known that many sets of differences are not normally distributed. (For further discussion of the T-test and sign test, see reference [7], page 12, also [3] chapter 8).

The sign test assumes that a result is equally likely to favor either sample A or sample B. Thus, it measures the probability of a more extreme distribution favoring B, or favoring either A or B.

The Wilcoxon signed rank test postulates that a greater difference between paired samples is more significant, but only as the numbers affect the ranking of the differences. For example, differences of -1, 2, -3, 4, and 20 are equivalent to differences of -1, 2, -3, 4, and 5 since only the rank of the ordered differences favoring a sample is important (not the actual values of the differences). The Wilcoxon test assumes that the two samples come from the same family of distributions, i.e., either two normal distributions, or two binomials, etc.

The three tests are performed for eleven points of the recall-level averages, and for the four overall measures of recall and precision of the document level averages; in addition, the tests are also performed for the 17 cost statistics. The three listings for the three different test procedures (Fig. 26 — 28) cover only the first option (eleven points of the recall-level averages plus the four overall measures).

For the T-test (Fig. 26), the following values are given for each of the fifteen statistics: the mean and standard deviation of the statistic for each of the two searches (A and B); the mean and standard deviation of the differences between the statistics for A and B; and a value T, which is defined as

$$T = \frac{(\overline{A}-\overline{B}) * \sqrt{N}}{\sigma_{A-B}}$$

where N is the number of degrees of freedom (which is one less than the number of queries being tested). The one-sided and two-sided probabilities (indicating whether a difference between the two samples as great as, or greater than, that observed would occur by chance) is also listed. Finally, the fifteen one-sided tests are statistically combined into a single measure also listed.

The sign test (Fig. 27) gives the number of queries favoring search A, favoring search B, and tied; the normal deviate ignoring ties (computed by using the binomial normal approximation); and the one-sided and two-sided

SMART--FEEDBACK SEARCHES ON CRANFIELD 200 WORDFORM   PAGE 65   09/10/69   23:53:15.96   87.0200

T TEST

TESTING COLLECTION B FOR PERFORMANCE BETTER THAN COLLECTION A (1-SIDED) OR UNEQUAL TO COLLECTION A (2-SIDED)
A (FILE 0), 42 QUERIES: CRN2ST FULL   FEEDBACK SEARCHES ON CRANFIELD 200 COMBINATION OF WORDFORM AND THESAURUS
B (FILE 1), 42 QUERIES: CRN2ST FEED1   FEEDBACK SEARCHES ON CRANFIELD 200 COMBINATION OF WORDFORM AND THESAURUS

ON OPTION 1, 15 MEASURES -- RANK RECALL, LOG PRECISION, NORMALIZED RECALL, NORMALIZED PRECISION, AND RECALL LEVEL AVERAGES

STATISTICS	MEAN A	SD A	MEAN B	SD B	MEAN A-B	SD A-B	T	1-SIDED PROB	2-SIDED PROB
RANK R	0.8778	0.1358	0.9184	0.1142	-0.0407	0.0788	-3.3437	0.0003	0.0007
LOG P	0.7035	0.2168	0.7448	0.2026	-0.0413	0.0771	-3.4713	0.0001	0.0003
NORM R	0.3231	0.3233	0.3757	0.3046	-0.0526	0.1348	-2.5273	0.0058	0.0116
NORM P	0.4961	0.2647	0.5304	0.2618	-0.0343	0.0739	-3.0063	0.0015	0.0029
R-L-A .0	0.6541	0.3735	0.6676	0.3622	-0.0135	0.0434	-2.0238	0.0213	0.0426
REC 0.1	0.6541	0.3735	0.6676	0.3622	-0.0135	0.0434	-2.0238	0.0213	0.0426
0.2	0.6131	0.3678	0.6330	0.3563	-0.0199	0.0583	-2.2116	0.0133	0.0267
0.3	0.5626	0.3697	0.5952	0.3550	-0.0325	0.0758	-2.7820	0.0029	0.0057
0.4	0.5439	0.3668	0.5800	0.3509	-0.0361	0.0805	-2.9061	0.0020	0.0040
0.5	0.5028	0.3612	0.5486	0.3372	-0.0458	0.0994	-2.9829	0.0016	0.0032
0.6	0.4095	0.3481	0.4974	0.3245	-0.0879	0.1483	-3.8414	0.0003	0.0006
0.7	0.3690	0.3386	0.4190	0.3222	-0.0499	0.1569	-2.0633	0.0193	0.0387
0.8	0.3150	0.3198	0.3551	0.3022	-0.0401	0.1462	-1.7787	0.0376	0.0751
0.9	0.2784	0.3202	0.3305	0.3041	-0.0521	0.1288	-2.6205	0.0045	0.0090
1.0	0.2774	0.3205	0.3305	0.3041	-0.0530	0.1299	-2.6461	0.0042	0.0084

COMBINED SIGNIFICANCE -- TOTAL CHI SQUARE WITH 30 DEGREES OF FREEDOM
THE PROBABILITY OF A CHI SQUARE LARGER THAN THE OBSERVED 170.5000 IS 0.0000

SYMBOL KEYS: SD -- STANDARD DEVIATION

T Test

Fig. 26

SMART--FEEDBACK SEARCHES ON CRANFIELD 200 WORDFORM   PAGE 66   09/10/69 23:53:16.10   87.1600

S I G N   T E S T

TESTING COLLECTION B FOR PERFORMANCE BETTER THAN COLLECTION A (1-SIDED) OR UNEQUAL TO COLLECTION A (2-SIDED)
A (FILE 0), 42 QUERIES: CRN2ST FULL   FEEDBACK SEARCHES ON CRANFIELD 200 COMBINATION OF WORDFORM AND THESAURUS
B (FILE 1), 42 QUERIES: CRN2ST FEED1   FEEDBACK SEARCHES ON CRANFIELD 200 COMBINATION OF WORDFORM AND THESAURUS

ON OPTION 1, 15 MEASURES -- RANK RECALL, LOG PRECISION, NORMALIZED RECALL, NORMALIZED PRECISION, AND RECALL LEVEL AVERAGES

STATISTICS	FAVORING METHOD A	FAVORING METHOD B	TIED	NORM DEV IGN TIES	1-SIDED PROB	2-SIDED PROB	NORM DEV USING TIES	1-SIDED PROB
RANK R	7	19	16	2.3534	0.0153	0.0306	-0.6172	0.7800
LOG P	8	18	16	1.9612	0.0387	0.0774	-0.9258	0.8598
NORM R	7	19	16	2.3534	0.0153	0.0306	-0.6172	0.7800
NORM P	8	18	16	1.9612	0.0387	0.0774	-0.9258	0.8598
R-L-A .0	1	7	34	2.1213	0.0385	0.0770	-4.3205	1.0000
REC 0.1	1	7	34	2.1213	0.0385	0.0770	-4.3205	1.0000
0.2	1	8	33	2.3333	0.0226	0.0451	-4.0119	1.0000
0.3	1	13	28	3.2071	0.0018	0.0036	-2.4689	0.9955
0.4	1	15	26	3.5000	0.0006	0.0011	-1.8516	0.9778
0.5	2	17	23	3.4412	0.0007	0.0014	-1.2344	0.9174
0.6	3	20	19	3.5447	0.0004	0.0007	-0.3086	0.6784
0.7	8	16	18	1.6330	0.0767	0.1534	-1.5430	0.9552
0.8	9	17	16	1.5689	0.0851	0.1702	-1.2344	0.9174
0.9	8	18	16	1.9612	0.0387	0.0774	-0.9258	0.8598
1.0	8	18	16	1.9612	0.0387	0.0774	-0.9258	0.8598
COMBINED	73	230	327	9.0194	0.0000	0.0000	-6.7730	1.0000

COMBINED SIGNIFICANCE -- TOTAL CHI SQUARE WITH 30 DEGREES OF FREEDOM
IGNORING TIES -- THE PROBABILITY OF A CHI SQUARE LARGER THAN THE OBSERVED 131.4126 IS 0.0000
USING TIES -- THE PROBABILITY OF A CHI SQUARE LARGER THAN THE OBSERVED 3.4689 IS 1.0000

SYMBOL KEYS: NORM DEV IGN TIES -- STANDARD NORMAL DEVIATE CALCULATED IGNORING TIES
NORM DEV USING TIES -- STANDARD NORMAL DEVIATE CALCULATED USING TIES

Sign Test

Fig. 27

probabilities for the test ignoring ties. The normal deviate and the one-sided probability using ties (based on a method developed by Cathy May [8]) are also computed and listed. The one-sided tests are again statistically combined into overall figures.

The Wilcoxon signed rank test (Fig. 28) gives the sum of ranks favoring search A and favoring search B; the number of degrees of freedom (specifically, the number of untied pairs); the normal deviate (computed using the Wilcoxon-normal approximation); and the resulting one-sided and two-sided probabilities. A statistically combined significance value is also listed.

3. Access to the SMART System

The SMART system exists at Cornell as a private library system, located on a disk, which 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 of those routines in the sequence requested.

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

initiates SMART routines	//JOB . . . . . . (parameters). . . . . . /*SMART
sets up document groups for a collection already on file	CLUSTR . . . . . (parameters). . . . . (parameters). . . . .
performs retrieval runs using methods called for by the parameter cards	SEARCH . . . . . (parameters). . . . . (parameters). . . . .

W I L C O X O N   S I G N E D - R A N K   T E S T

TESTING COLLECTION B FOR PERFORMANCE BETTER THAN COLLECTION A (1-SIDED) OR UNEQUAL TO COLLECTION A (2-SIDED)
A (FILE 0), 42 QUERIES: CRN2ST FULL FEEDBACK SEARCHES ON CRANFIELD 200 COMBINATION OF WORDFORM AND THESAURUS
B (FILE 1), 42 QUERIES: CRN2ST FEED1 FEEDBACK SEARCHES ON CRANFIELD 200 COMBINATION OF WORDFORM AND THESAURUS

ON OPTION 1, 15 MEASURES -- RANK RECALL, LOG PRECISION, NORMALIZED RECALL, NORMALIZED PRECISION, AND RECALL LEVEL AVERAGES

STATISTICS	SUM OF RANKS FAVORING A	SUM OF RANKS FAVORING B	NDF	NORMAL DEVIATE	1-SIDED PROB	2-SIDED PROB
RANK R	51.5	299.5	26	3.1494	0.0009	0.0019
LOG P	55.0	296.0	26	3.0605	0.0013	0.0026
NORM R	72.0	279.0	26	2.6287	0.0045	0.0091
NORM P	57.0	294.0	26	3.0097	0.0015	0.0030
R-L-A .0	4.5	31.5	8	1.8904	0.0342	0.0685
REC 0.1	4.5	31.5	8	1.8904	0.0342	0.0685
0.2	4.5	40.5	9	2.1325	0.0189	0.0378
0.3	10.0	95.0	14	2.6680	0.0043	0.0086
0.4	12.0	124.0	16	2.8957	0.0022	0.0045
0.5	23.0	167.0	19	2.8974	0.0022	0.0044
0.6	23.0	253.0	23	3.4977	0.0001	0.0002
0.7	93.0	207.0	24	1.6286	0.0533	0.1066
0.8	108.0	243.0	26	1.7144	0.0444	0.0888
0.9	80.0	271.0	26	2.4255	0.0079	0.0157
1.0	80.0	271.0	26	2.4255	0.0079	0.0157

COMBINED SIGNIFICANCE -- TOTAL CHI SQUARE WITH 30 DEGREES OF FREEDOM
THE PROBABILITY OF A CHI SQUARE LARGER THAN THE OBSERVED 157.3448 IS 0.0000

SYMBOL KEYS: NDF -- NUMBER OF DEGREES OF FREEDOM

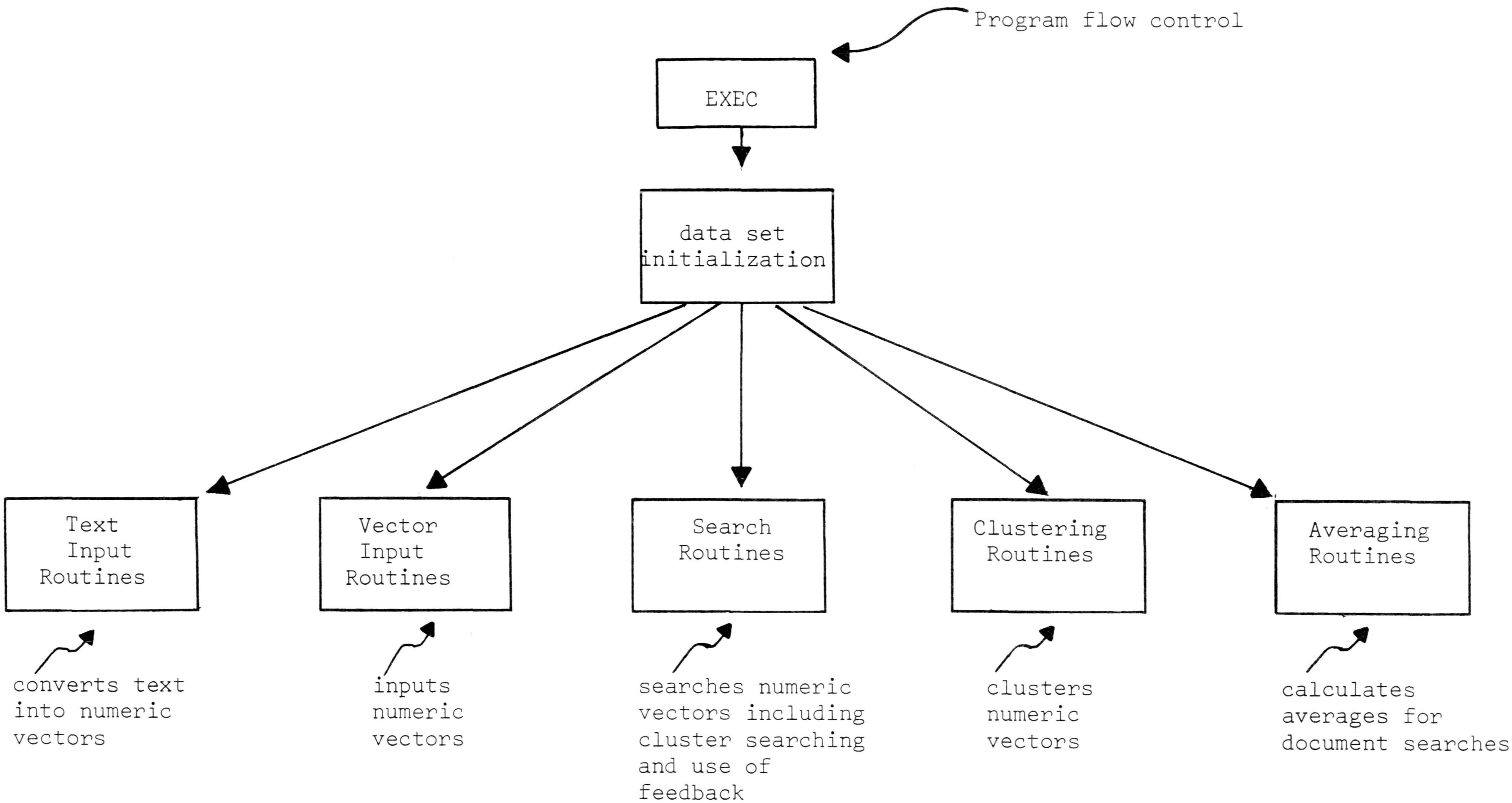
Wilcoxon Signed-Rank Test

Fig. 28

performs statistical averages for the previous search	{	AVERAG . . . . . (parameters). . . . . (parameters). . . . . . .
signals end of job	[	STOP

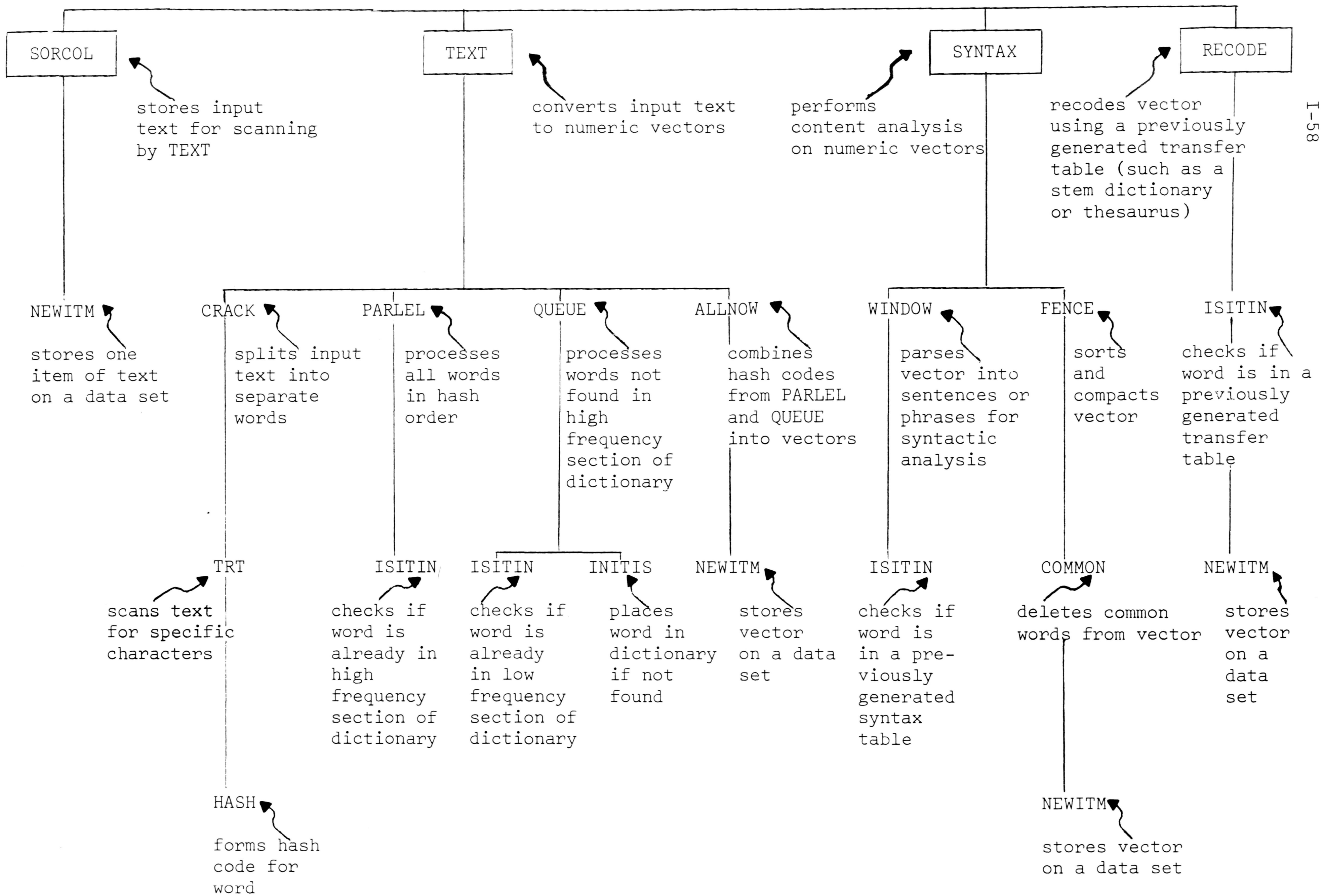
4. Basic SMART System Flowchart

The SMART routines fall into two categories: routines that can be called with control cards, and routines that can only be called by other routines. The latter set in interconnected by means of complex internal vectors, designed to make the most efficient use of in-core storage. A flowchart is produced in Figs. 29 — 33. The routines which can be called by control cards are enclosed by boxes.



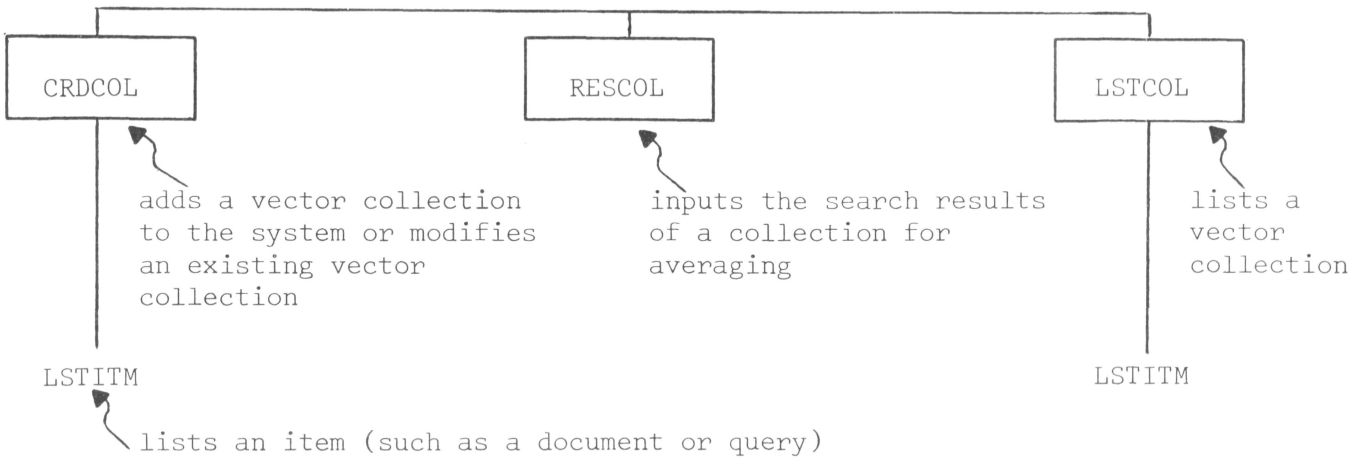
SMART Systems Chart

Fig. 29



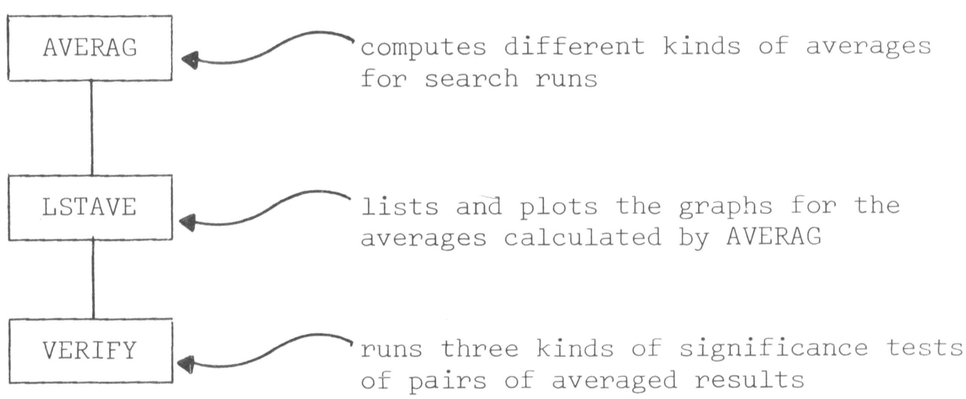
SMART System Chart — Text Input Routines

Fig. 30



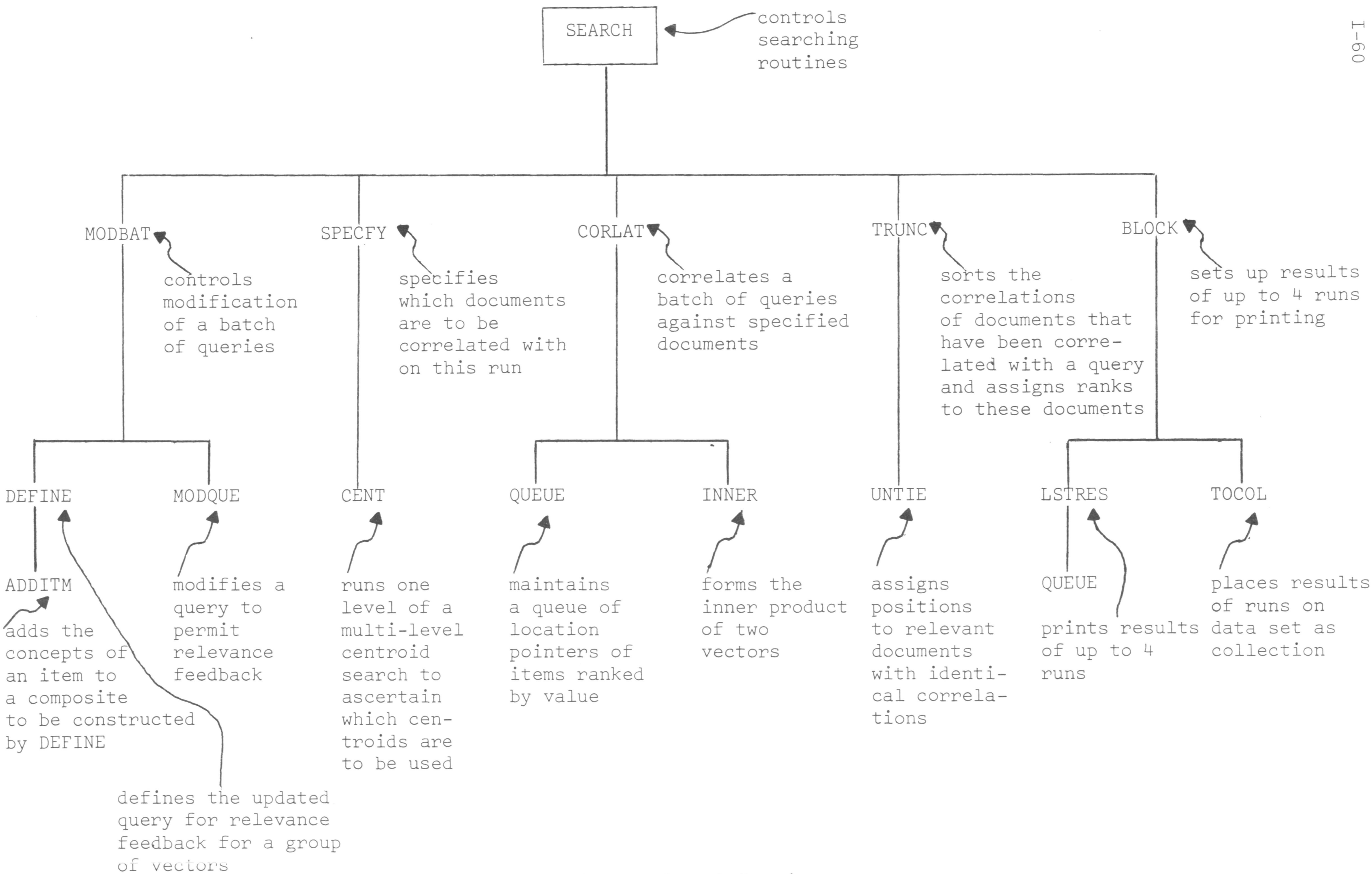
SMART Systems Chart — Vector Handling Routines

Fig. 31



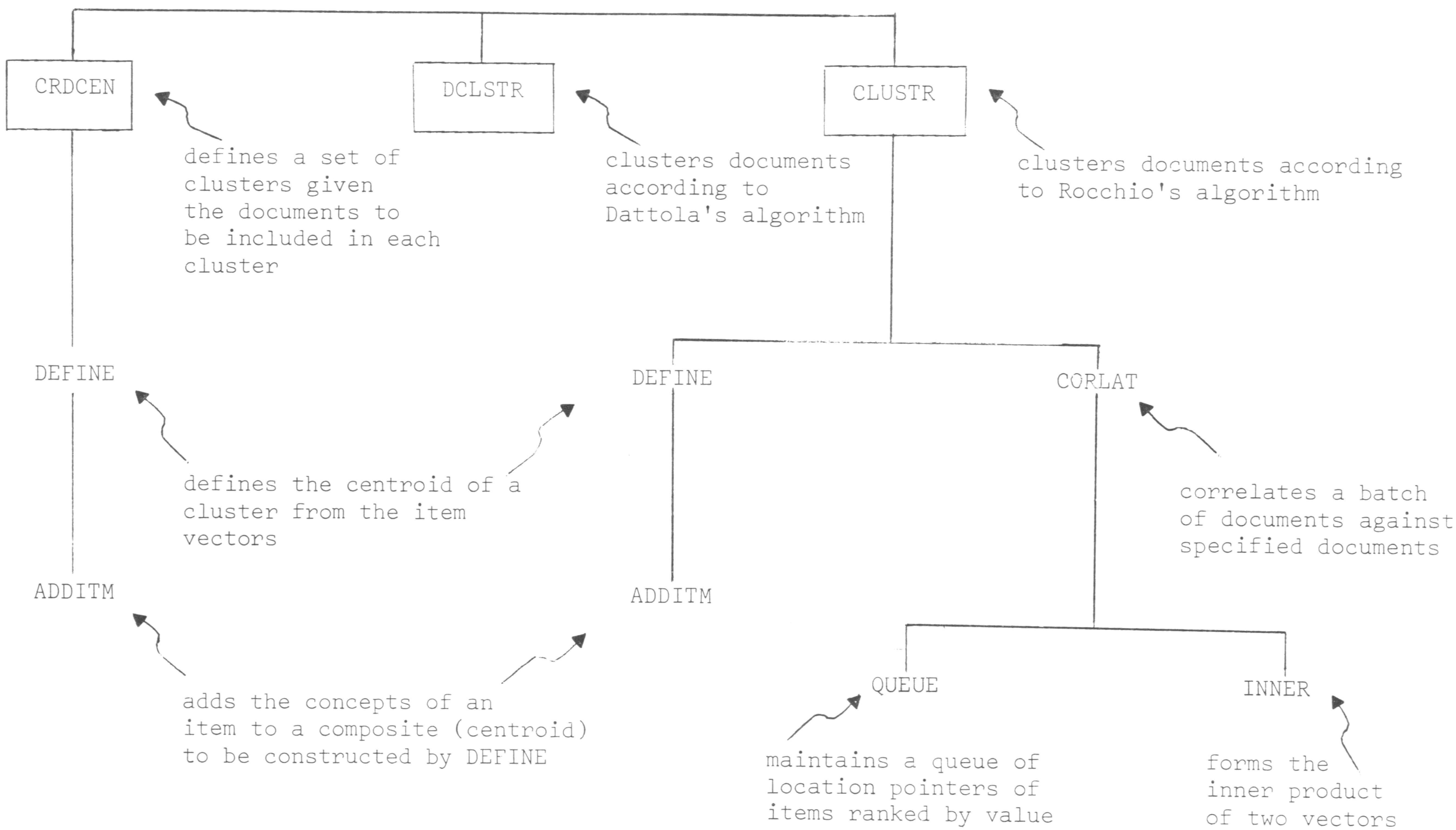
SMART System Chart — Averaging Routines

Fig. 32



SMART Systems Chart — Search Routines

Fig. 33



SMART Systems Chart — Clustering Routines

Fig. 34

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] D. Williamson, The Cornell Implementation of the SMART System, Information Storage and Retrieval, Report ISR-14 to the National Science Foundation, Section II, Department of Computer Science, Cornell University, October 1968.

[3] G. Salton, Automatic Information Organization and Retrieval, McGraw Hill Book, Co., New York, 1968, chapter 3.

[4] J. J. Rocchio, Jr., Document Retrieval Systems — Optimization and Evaluation, Harvard Doctoral Thesis, Information Storage and Retrieval, Report ISR-10 to the National Science Foundation, Harvard Computation Laboratory, Cambridge, March 1966.

[5] R. T. Dattola, A Fast Algorithm for Automatic Classification, Information Storage and Retrieval, Report ISR-14 to the National Science Foundation, Section V, Department of Computer Science, October 1968.

[6] R. Williamson, Centroid Searching (Tree Searching), unpublished paper, May 1968.

[7] G. Salton and M. E. Lesk, Computer Evaluation of Indexing and Text Processing, Information Storage and Retrieval, Report ISR-12 To the National Science Foundation, Section III, Department of Computer Science, Cornell University, June 1967.

[8] Cathy May, Evaluation of Search Methods in an Information Retrieval System, Term Report written for Computer Science 435, Cornell University, May 1968.