THE EVALUATION OF LARGE INFORMATION RETRIEVAL SYSTEMS WITH APPLICATION TO MEDLARS

A Thesis

submitted to the

UNIVERSITY OF NEWCASTLE UPON TYNE

for the Degree

of

DOCTOR OF PHILOSOPHY

W.L. Miller, M.A.
(May, 1970)

ACKNOWLEDGEMENTS

The work described in this thesis was carried out in the Computing Laboratory of the University of Newcastle upon Tyne.

I wish to express my thanks to Professor E.S. Page, Dr. A.J. Harley, and Miss E.D. Barraclough for their invaluable advice and guidance throughout the research project and in the preparation of the text.

I also wish to thank the many users of the MEDLARS system who helped in the evaluation of the various search techniques. My contacts with them were through Dr. A.J. Harley of the National Lending Library for Science and Technology, Boston Spa, Miss M. King of Newcastle University, Mr. R. Dannatt of Strathclyde University, Glasgow, and Dr. P.A. Jacobs of the Medical Research Council Clinical and Population Cytogenetics Research Unit, Edinburgh, to all of whom I am especially indebted.

The MEDLARS system was programmed by Miss E.D. Barraclough who freely made available programs and subroutines from the standard system.

During most of the period of this research (1965-68) I was in receipt of a Science Research Council Studentship, and a Postgraduate Bursary of the University of Edinburgh.

SUMMARY

The MEDLARS system consists of a large file of indexed references with facilities for retrieving all references indexed by a specified combination of index terms. Were the file small the retrieval performance of the system could be evaluated by asking a number of system users to inspect the entire file and to compare the references they selected as relevant to their needs with the references selected by a MEDLARS search. Since the file is large this is not possible. MEDLARS performance is therefore evaluated by measuring the change in the users' literature awareness induced by the system, and by comparing the standard system with three alternative retrieval techniques.

To measure the change in literature awareness a measure, the Extension Ratio, is developed which is approximately equal to the ratio of the number of relevant references known by a user after a MEDLARS search, to the number known before it. It is shown that this measure does not necessarily vary with the number known before. Most MEDLARS users can be divided into two groups; those knowing less than 13 relevant references and those knowing between 13 and 65. For the latter the average Extension Ratio is between 2 and 3 times, and is not affected by the number of known relevant references. For the others the average Extension Ratio is higher and varies inversely with the number known before.

The standard MEDLARS retrieval facility, Boolean Searching, is first compared with an alternative search technique, Probabilistic Searching, which also uses index terms for retrieval. This technique has the advantage that the number of references to be retrieved can be predetermined. It also retrieves more of the relevant references even when its output size is the same as for the standard MEDLARS search. A method is given which extends some of the advantages of the Probabilistic technique to the Boolean.

The standard search is then compared with two techniques which do not use index terms. The MEDLARS file contains, in addition to index terms, the reference titles both in the vernacular and in (American) English. An efficient method of searching these English titles is given based on a Key-Letter-In-Context index to each title. The output size of this search can be fixed in advance.

This program takes 5 times as much computer time as the standard Boolean search program, and retrieves fewer (about 5:6) of the relevant references than the Boolean when their output sizes are the same. However this method does not require an indexing operation, and when it is used with an output size larger than the Boolean, it retrieves more of the relevant references.

The second method of retrieval without indexing uses the links between references created when authors cite other papers. MEDLARS does not contain information about these links, and a relatively small Citation File had to be specially constructed. Citation Searching, using this file, is compared with the best of the three other methods, Probabilistic Searching. The Probabilistic search retrieves more of the relevant references when the output sizes are the same.

The average extension of the users' literature awareness, as measured by the Extension Ratio is an index of the system's retrieval performance. The comparative tests described in this thesis suggest that its retrieval performance could be improved by using the Probabilistic technique, and that comparable, though not quite so good, retrieval could be obtained without indexing, by making use of titles. The detailed results also show that all retrieval techniques retrieve very much less than 100% of relevant references, and that the use of several search methods together greatly improves the retrieval of relevant references.

CONTENTS

Chapter 1. Measures of Retrieval Performance for Large Systems and an Overall Evaluation of MEDLARS.

1.1 Information Retrieval Systems

1.2 The Evaluation of Large Systems

1.3 Relevance and Recall Estimates

1.4 The Extension Ratio

1.5 Properties of the Extension Ratio

1.6 An Evaluation of MEDLARS in Britain

Chapter 1. Measures of Retrieval Performance for Large Systems and an Overall Evaluation of MEDLARS

1.1 Information Retrieval Systems

MEDLARS [1] is one of a number of large Information Retrieval Systems, each of which attempts to cover all current literature in its area. Systems exist for a wide variety of subject areas, including Chemistry [2], Electrical Engineering [3], and Nuclear Science [4]. MEDLARS (Medical Literature Analysis and Retrieval System) attempts to include all medical research papers, indexing more than 180,000 journal references per year, without restriction on language or country of origin. Amongst other services provided by these systems is the 'Demand Search' service: the production of special bibliographies on demand, usually, but not always, on a topic of interest to a particular research worker. The technical difficulties associated with Demand Searches can be grouped under two headings:-

(i) The problem of processing the very large quantities of data at a sufficiently high speed to make searches economically possible, even if expensive.

(ii) The problem of supplying the user with a bibliography which is so small as not to overload his data processing ability, yet which contains a large amount of useful information. This will be called the problem of achieving good Retrieval Performance.

There is a distinction between Information Retrieval Systems of the MEDLARS type, which process journal references, and Data Retrieval Systems. An example of the latter is a system which handles the specifications of All Electric Motors commercially available [4]. It is possible to put a precise question to a Data Retrieval system, and to get a precise answer e.g. all electric motors giving

0.75 h.p. at 2500 r.p.m. on 110 v. The MEDLARS system is not designed to answer such queries. It can provide a list of references which may have relevance for a user's research interest e.g. computer simulation of systems containing Biogenic Amines (MEDLARS search no. 3923). Such questions are not precise, nor can the answers be so. Even the user may have doubts as to whether a reference is relevant, or useful, for his research, and he will probably revise the judgements he does make as his research progresses.

1.2. The Evaluation of Large Systems

Large Information Retrieval Systems are so expensive to construct and operate, that there are seldom two or more systems for the same subject area (although several systems may overlap). Retrieval Performance cannot be evaluated by comparing the system with others, for the literature and the users vary from subject to subject in ways that can affect Retrieval Performance e.g. the titles of papers on Chemistry may be better descriptions of the content of these papers than are the titles of papers on Medical subjects. Each large system must therefore be evaluated by comparing it only with variants of itself, or by directly measuring the benefits conferred on its users. Both approaches will be used here. This Chapter develops a measure of the benefits to users of a Retrieval System, and applies the measure to MEDLARS. Subsequent Chapters develop variants of the standard MEDLARS Demand Search and compare them with the standard form.

Much of the published research on the Evaluation of Retrieval Systems has involved specially constructed files of the order of 200 to 1000 references [5]. The methods of evaluation developed for these small files cannot be applied to large systems; where the file contains several hundred thousand references. Where the file is small, the system can be tested by making a request for information, and then assessing the 'Relevance' of each reference in the file to the request. A

2 x 2 table can be formed:-

	Number of References supplied by system in answer to query	Number of References in file but not supplied in answer to query	
Number of references judged relevant by questioner	x	y	(x + y)
Number of references judged irrelevant by questioner	z	t	(z + t)
	(x + z)	(y + t)	

From this table many measures of retrieval performance have been defined [6], but the most widely used are Recall Ratio = $\frac{x}{x+y}$ and Precision Ratio = $\frac{x}{x+z}$ Recall measures the ability of the system to provide all relevant references, and Precision measures its ability to provide only relevant references e.g. [7]. The values of x and z are easy to obtain since they are found when the questioner evaluates the references supplied to him by the system, and in general the sum x + z is not too large for inspection. The value of y is quite different. It cannot be found unless the questioner assesses every reference in the file. Thus, although Recall = $\frac{x}{x+y}$ can be measured when the file is small, it cannot when the file is large. Recall is thus a useful concept for discussion of retrieval techniques, but not for measurement of their effectiveness. Estimates of Recall can be used to evaluate large systems only when the reliability of the estimation procedure can be demonstrated.

1.3 Relevance and Recall Estimates

The concept of 'relevance' has been widely criticised. The assessment of the relevance of a reference to a request varies from judge to judge [8]. Since a single index term can be regarded as a request, experiments showing indexing

inconsistency [9, 10, 11] also show relevance-assessment inconsistency. One experiment on the repeated indexing of 171 references reported that "55 papers were indexed so differently (on the second occasion) that a correlation between the two instances was impossible." [12]. In spite of valid criticism of relevance, the consensus appears to be that "no practical alternative to the use of relevance exists at the present time". [13]

Relevance judgements should be made by the users of the system, not by system personnel nor by 'impartial judges' appointed by them. There is some evidence that system personnels' judgements correlate well with each other, but not with the users' judgements [14], and it is important to distinguish "relevant to the query as stated" from "relevant to the user's information need" [15]. Only the user can assess the latter and this is the relevance which matters. Failure to elicit the true need is at least partly the fault of the system. The results quoted later in this chapter for the MEDLARS system are based on answers which users gave to questionnaires. Two types were used: one asked for relevant/not relevant judgements (Newcastle University searches), the other asked that references be judged relevant only if the user intended to consult the full text (N. L. L. searches). Judgements were based on a print of Title, Author, Journal Reference, and Index Terms. No abstracts or texts were provided, but relevance assessments based on abstracts or titles may not be very different [16, 17]. Degrees of Relevance were not used. All the measures of retrieval effectiveness discussed below can be calculated using degrees of relevance if such data is available. For each measure the numbers of relevant references would be replaced by sums of relevance-weights.

Recall has been described as an "absolute" measure of retrieval effectiveness, dependent only on the retrieval technique and the file. In fact this is not so, for the relevance-assessments determine Recall and they are best made by system-users.

For large files only an estimate of Recall is possible, and then the value obtained depends upon the estimation procedure also. The burden of proof that an estimation procedure is reliable rests on those who use it. Some of the procedures used are reviewed below.

Method 1. Source Document Method [18, 19, 20, 21]

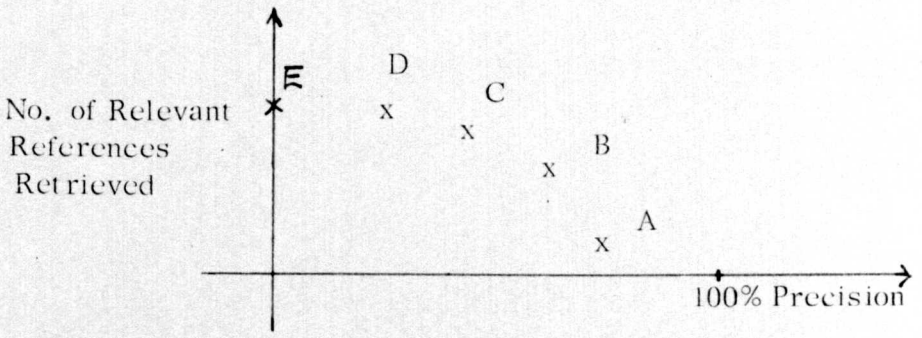
Take 100 references from the file, and for each reference devise a query to which the reference is relevant. These references are known as "source documents". Perform the 100 searches. The number of searches which retrieve their source document is the percentage Recall. This method has been criticised in detail [22, 23, 24, 25, 26]. The important objection is that questions based on particular references are different in character from queries that arise from information needs. The source documents are more closely related to the questions than is normal. The Recall estimate is likely to be an over estimate of the performance of the system with genuine questions.

Method 2. Single-term match [3]

Assume that any relevant reference will have at least one index-term in common with the user's query. Then the value of y (= no. of relevant references not retrieved by the system) can be found by assessing only a subset of the file. This method ignores the effect of adding further index-terms to the user's query to widen its scope. Even if the search has been well-formulated, additional terms may retrieve many relevant references. These terms might well have been omitted from the query in the expectation that they would retrieve too many irrelevant references. The estimate is clearly an upper-bound to the true recall. The degree of over estimation is unknown.

Method 3. Extrapolated Single-term match [27]

Repeat each search several times, gradually widening its scope by relaxing the structure (e.g. replacing logical and's by or's) but not adding any new index terms, until the search becomes a single-term match between references and question. Plot a graph of the number of relevant references retrieved against the corresponding Precision Ratio. e.g.



A is the point obtained for the original search. As its scope is widened points B, C. D are obtained. D represents a single-term match.

The points are joined by a smooth curve which is extrapolated (by eye) back to the zero-Precision line. The point of intersection E with theRelevant references axis gives the total number of relevant references in the system.

An immediate objection to this procedure is that the extrapolation of curves of experimental data requires assumptions about the shape of the curve in the region of extrapolation. The important objection, as with Method 2, is that relaxing the structure of a query is not the only means of widening its scope.

Method 4. Sample File [28]

Use a sample, preferably a random sample, for the evaluation. Unfortunately this requires sample sizes which may still be too large for the user to assess. For example, MEDLARS has a file of approximately 750,000 references. Supposing that of these references, 750 are relevant to a query, the expected number of relevant references in a random sample of 1000 is only one. Thus even

a sample of 1000 references would give an insensitive estimate of Recall, and if it contained no relevant references it could give no estimate.

Method 5. Union of Outputs [29]

When comparing search techniques, assume that all the relevant references are retrieved by one or other technique. Again this gives an upper-bound to the true Recall. It produces an especially high over-estimate when all the techniques do badly.

Method 6. User-augmented Recall [30,31]

Assume that all relevant references are either known already to the system user, or are found by the system. This can lead to a gross over-estimate of Recall. In the experiment cited, three quarters of users knew no relevant references other than those retrieved by the system. This contributed to an average Recall estimate of 94%.

Method 7. Recall-base [32]

This is superficially similar to method 6 but does not require the assumption that the system user is superlatively well-informed. For each search a "Recall base" is formed of relevant references already known by the user or found for him by librarians. The percentage of the Recall base retrieved by the system is taken as an estimate of Recall. There is no assumption that the Recall base contains all relevant references. It is assumed that the proportion of the Recall base retrieved is equal to the proportion of all relevant references retrieved. This estimate of Recall will be called the "Consistency Ratio", for all the references known to the user should be retrieved unless the system user and the indexer are inconsistent with each other in their use of index terms, or the system-user fears that the use of a term would give low Precision.

Thus attempts to estimate Recall involve assumptions which may not be true, and publication of performance figures including Recall estimates does not end speculation about the performance of the system, bat starts speculation about the reliability of the estimation procedure. The Precision Ratio can always be found, but by itself neither Precision nor Recall is important. Retrieval is the process of filtering the few relevant references out of a file which is mainly irrelevant. To evaluate the process the yield and concentration of relevant references in the product must be considered together.

1.4 The Extension Ratio

Even when Recall can be measured it ignores the relationship between the file and the system-users. Retrieval of all relevant references in the file only provides information for the user if the coverage is high, or the document collection arises from literature which the user does not normally check. Some index of the extension of the users knowledge is necessary. A 2 x 2 table can be defined:-

	No. of Relevant references found by system	No. of Relevant references not found by system	
No. of Relevant references known to user before search	a	m	x= a + m
No. of Relevant references not known to user before search.	b	?	?

(1) $\dot{x}$ has the same value as in the 2 x 2 table of 1.1

(2) ? is used for quantities which cannot be found

One possible measure of the extension of the user's knowledge is the ratio of the number of relevant references he is aware of after the search to the number he knew before, i.e. $e = \frac{a + b + m}{a + m}$. (It is interesting to note that the inverse of e is an estimate of the user's Recall using method 6 of 1.2. Thus e is both an arithmetic and logical inverse of the Recall estimate obtained by method 6).

In general, the higher the value of e, the greater is the service given to the user but there is an important exception. Queries are sometimes put by users who know no relevant references and hope they are not going to find any, e.g. to check that there is no reported instance of a given drug causing a particular effect. The use of $e = \frac{a + b + m}{a + m}$ as an index of extension suffers from the disadvantage that e is undefined when no relevant references are known or found, and e is infinite when none are known but some are found. This difficulty may be avoided by adding one to numerator and denominator. Since the ratio is always greater than or equal to one it is also convenient to subtract one from it. Then a new index E can be defined as:-

$$E = \frac{a + b + m + 1}{a + m + 1} - 1 = \frac{b}{a + m + 1} \qquad \text{equn. (d1)}$$

E is not a monotonic transformation of e, since the values of e for small $(a + m)$ are reduced by a greater factor than for large $(a + m)$. When $a + m = 0$, E takes a value equal to the number of relevant references found by the system. When $(a + m)$ is large, E is approximately the ratio of new relevant references found to the number known already. It measures the ability of the system to multiply the relevant references known to the users. This proposed measure was mentioned in a report on the British MEDLARS system [33] and a similar measure called the Novelty Ratio, in an evaluation of the American MEDLARS system [32]. The importance of the Novelty Ratio was not recognised. It was assumed that its value was high when users knew few relevant references, and low when they knew

many. The properties of E are not so simple.

1.5 Properties of the Extension Ratio E

It is natural to expect that the ratio $E = \frac{b}{a + m + 1}$ will be large when $(a + m)$ is small and conversely.. In so far as $(a + m)$ is an index of the user's awareness of the literature, the larger it is the smaller the multiplication of his reference list. However, it is desirable that E should have some measure of invariance to variations in the stringency of the user's assessment of relevance which is also reflected in the value of $(a + m)$. To investigate the properties of E a simplified model is used.

Property I: Invariance to Variations in the Stringency of Relevance Assessment

Consider two users, R and Q, with the same interests. The same search can be performed for both. After the search let R look through the references judged relevant by Q (whether found by the system or by Q). Let R's more stringent view of relevance be expressed as a constant probability $p \leq 1$ that he will accept as relevant a particular reference judged relevant by Q.

Then the Extension Ratio for Q is E_Q, where

$$E_Q = \frac{B}{F + 1} \qquad \text{equn } (a2)$$

where Q knew F relevant references before the search and the search found another B relevant references. No reference counting towards B counts towards F or vice versa. Similarly let the Extension Ratio for R be E_R, where

$$E_R = \frac{b}{f + 1} \qquad \text{equn. } (a3)$$

then b and f are random variables and are independent. Thus the Expected Value of E_R. $\mathcal{E}(E_R)$ is

$$\mathcal{E}(E_R) = \mathcal{E}(b)\,\mathcal{E}\left(\frac{1}{f + 1}\right) \qquad \text{equn. } (a4)$$

$$= Bp. \sum_{k=0}^{k=F} \binom{F}{k} \frac{p^k (1 - p)^{F-k}}{k + 1} \qquad \text{where } \binom{F}{k} \text{ is the binomial coefficient}$$

$$= \frac{B}{F+1} \left[1 - (1-p)^{F+1} \right]$$

$$\therefore \mathcal{E}(E_R) = E_Q \left[1 - (1-p)^{F+1} \right] \qquad \text{equn } (a5)$$

Thus $\mathcal{E}(E_R) \longrightarrow E_Q$ as $p \longrightarrow 1$, and also as $F \longrightarrow \infty$. The Expected Value of E_R is equal to KE_Q where the value of K for different combinations of p and F is given in the table:-

Values of F \ Values of p	p = 0.25	p = 0.50	p = 0.90
F = 1	0.4374	0.7500	0.9900
F = 2	0.5780	0.8750	0.9990
F = 5	0.8220	0.9844	1.0000
F =10	0.9577	0.9995	1.0000
F =20	0.9976	1.0000	1.0000
F =50	1.0000	1.0000	1.0000
F =100	1.0000	1.0000	1.0000

Table of K where $\mathcal{E}(E_R) = K.E_Q$

The Variance of E_R can also be calculated. It is σ_R^2 where

$$\sigma_R^2 = \left[Bp + B(B-1)p^2 \right] \left[\frac{1}{p(F+1)} \right] \left[\sum_{i=o}^{F} \frac{(1-p)^i - (1-p)^{F+1}}{F+1-i} \right] - \frac{B^2}{(F+1)^2} \left[1-(1-p)^{F+1} \right]^2$$

equn. (a6)

An upper bound can be found to the sum $\sum_{i=0}^{F} \frac{(1-p)^i}{F+1-i}$ by replacing it with a geometric progression having a common ratio equal to the maximum of the ratio of adjacent terms in the original sum. Thus,

$$\sum_{i=o}^{F} \frac{(1-p)^i}{F+1-i} < \frac{1}{F+1} \cdot \sum_{i=o}^{F} 2^i (1-p)^i < \frac{1}{F+1} \cdot \frac{1}{2p-1} \qquad \text{equn } (a7)$$

$$\text{Hence, } \sigma_R^2 < E_Q^2 \left[\left(\frac{1-p}{B} + p \right) \frac{1}{2p-1} - \left(1 - (1-p)^{F+1}\right)^2 \right] \qquad \text{equn } (a8)$$

The case of $B = O$ is trivial since then $E_Q = E_R = O$ and $\sigma_R^2 = O$. So B can be assumed $\geqslant 1$. Then

$$\sigma_R^2 < E_Q^2 \left[\frac{1}{2p-1} - \left(1 - (1-p)^{F+1}\right)^2 \right] \qquad \text{equn } (a9)$$

Thus, $\frac{\sigma_R}{E_Q}$, the coefficient of variation tends to zero as p tends to 1, and tends to $\frac{2(1-p)}{2p-1}$ as F tends to ∞. The apparent infinity at $p = \frac{1}{2}$ is not significant, for it results from the gross approximation in equn (*a*7). Another equally valid upper bound suggests an infinity at $p = 1/3$ etc. A table of values based on equn (*a*9) is given, but needs careful interpretation since the values of the bounds on $\frac{\sigma_R}{E_Q}$ are gross over-estimates.

Values of F \ Values of P	p = 0.70	p = 0.80	p = 0.90
F = 1	1.300	0.8632	0.5195
F = 2	1.247	0.8262	0.5119
F = 5	1.226	0.8166	0.5000
F = 10	1.225	0.8166	0.5000
F = 50	1.225	0.8166	0.5000
F = 100	1.225	0.8166	0.5000

Table of σ_R / E_Q

From the tables of $\mathcal{E}_{(E_R)/E_Q}$ and ∇_{R/E_Q} it can be seen that for p larger than 0.5 and F larger than 2 the expected value of the Extension Ratio for R is close to the Ratio for Q. Low p and F give fairly high relative standard deviations. Thus, for several users with the same interests, variation in the stringency of relevance assessment should not cause any consistent variation in E, but when there are few relevant references known before the searches, or large differences in stringency, there may be a scatter of values of E.

Property II: Inverse relationship to user's knowledge of the literature

Let several users denoted by subscript i (i = 1, 2, ...) have identical relevance judgements but varying knowledge of the literature. If a single search is performed, and the results submitted to each user, they must agree on which references are relevant, and upon the total, T, of relevant references. The Extension Ratio for the i th user is

$$E_i = \frac{B_i}{A_i + M_i + 1}$$

$$= \frac{T - A_i.}{A_i + M_i + 1} \qquad \text{equn } (a\ 10)$$

where A_i = no. of relevant references retrieved, which user knew already

B_i = no. of relevant references retrieved, which user did not know

M_i = no. of relevant references known to user, which the system failed to retrieve.

The Consistency Ratio, used as an estimate of Recall in method 7 is C_i, where

$$C_i = \frac{A_i}{A_i + M_i} \qquad \text{equn } (a\ 11)$$

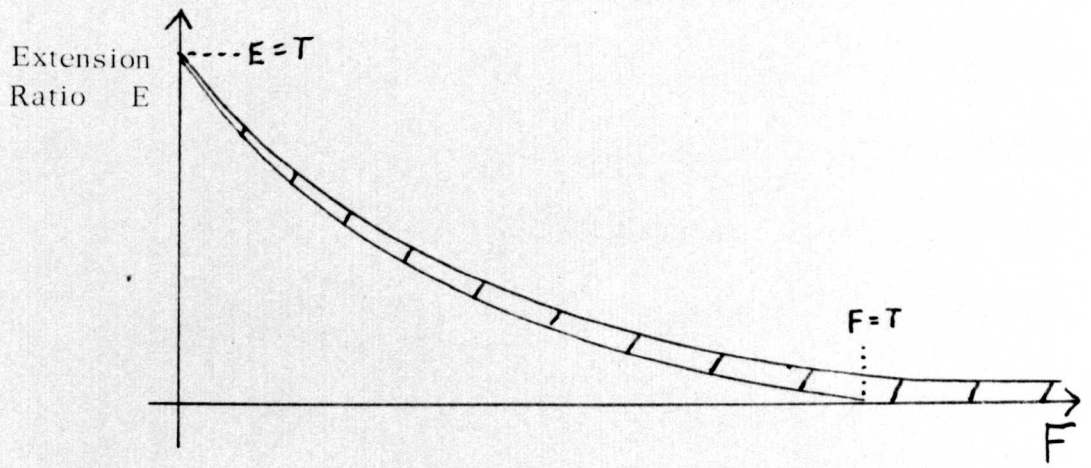
Putting $F_i = A_i + M_i$ the number of relevant references known before the search, and combining equns (*a* 10) and (*a* 11), gives

$$E_i = \frac{T - C_i F_i}{F_i + 1} \qquad \text{equn } (a\ 12)$$

The maximum value of C_i is the minimum $\left\{ 1, \ T/F \right\}$ and the minimum value of C_i is zero. Thus the points (E_i, F_i) lie between the curves

$$E = \frac{T}{F + 1} \qquad \text{equn } (a\ 13)$$

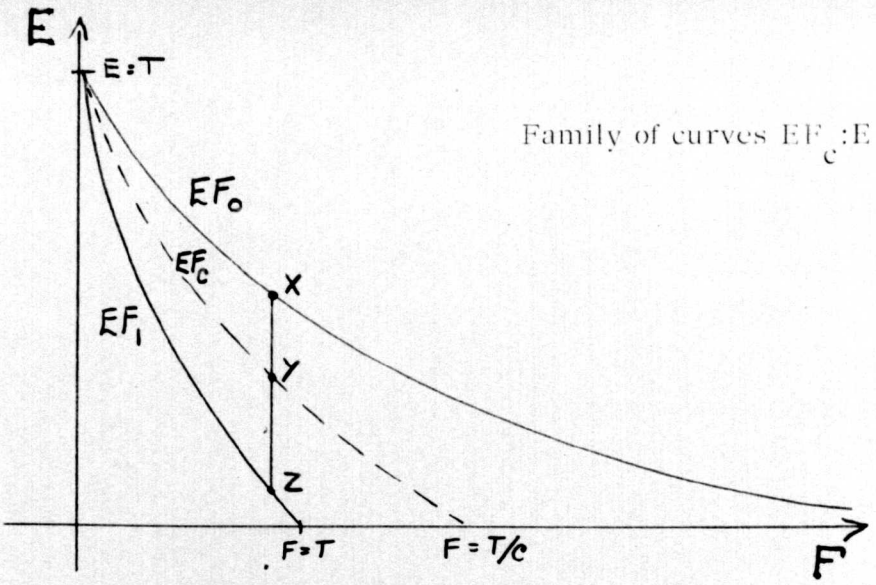
and $$E = \max \left\{ \frac{T - F}{F + 1}, \ O \right\} \qquad \text{equn } (a\ 14)$$



All points (E_i, F_i) lie within the shaded area.

The shaded band is widest at F = T when the (vertical) width is $\frac{T}{T + 1}$. For MEDLARS the median value of T is ~35. The band is narrow relative to its maximum height above the F axis. Thus for users with the same concept of relevance the Extension Ratio is inversely proportional to the number of references known before the search.

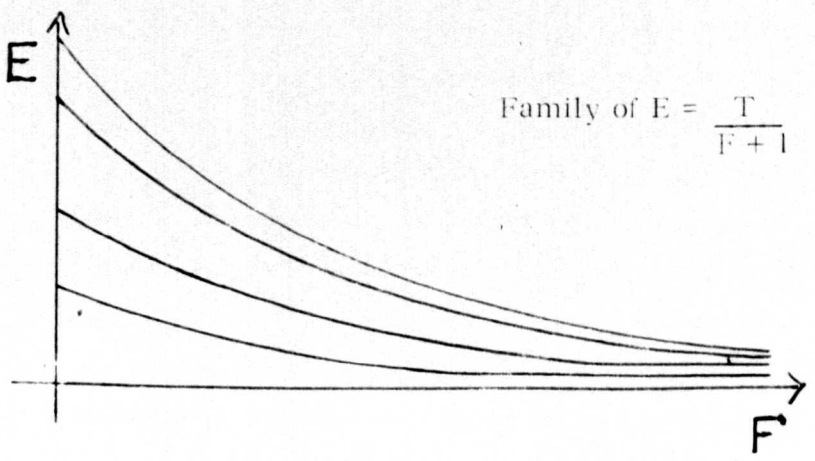
The relation between the Consistency Ratio and the Extension Ratio can be seen from equn (*a* 12). Each value of C. the Consistency Ratio, defines an E - F curve, EF_c, lying between the two boundary curves, and the fraction of the vertical line above EF_c is C, i.e. $C = \frac{XY}{XZ}$ in the figure.



Family of curves $EF_c: E = \frac{T-CF}{F+1}$

Thus, for users who agree on relevance, and who know the same number of references before a search, the higher the Consistency Ratio (Recall Estimate), the lower the Extension Ratio.

Property I of the Extension Ratio suggests that were the users of a system equally well informed, then the values of their Extension Ratios would exhibit no systematic dependence upon the number of relevant references that each user knew before his search. The system would be more successful on some occasions than on others, causing the (E, F) points to be scattered about a horizontal line. If users were equally stringent in their definitions of relevance, Property II suggests that (E, F) points would lie close to the curve $E = \frac{T}{F+1}$ (where T is the total number of relevant references retrieved by a search). Since values of T would differ for different searches, the (E, F) points would each lie near their particular $E = \frac{T}{F+1}$ curve.



Family of $E = \frac{T}{F+1}$ curves for various T.

Since the users have equally stringent notions of relevance, the value of T for a particular search depends only on the system performance, not on the particular user who makes the relevance assessments. Thus the values of T should not be systematically related to the values of F which do depend on the users. Thus the probability-distribution for T, P(T) depends upon the characteristics of the system but is the same for each user, whatever the value of F. Hence the Expected value of the Extension Ratio for a particular value of F is $\mathcal{E}(E_F)$ where

$$\mathcal{E}(E_F) = \mathcal{E}\left(\frac{T}{F+1}\right) = \frac{\mathcal{E}(T)}{F+1} \qquad \text{equn (} \lambda 15\text{)}$$

and the probability distribution of E for a particular value of F is $P(E_F)$ where

$$P(E_F) = P\left(\frac{T}{F+1}\right) = \frac{P(T)}{F+1}$$

Thus the distribution of (E, F) points for equally stringent users should be a wedge shaped scatter sloping down as F increases. The main interest in a plot of experimental data onto an E-F graph, apart from its description of system-performance, is to determine which effect is predominant - equally stringent (wedge shaped) or equally well informed (rectangular shape).

1.6 An Evaluation of MEDLARS in Britain

MEDLARS is a typical large-scale information retrieval system. Each year approximately 180,000 references covering the whole of Medicine are indexed at the National Library of Medicine, Washington D.C. The primary purpose is the production of Index Medicus, a printed listing of titles, each title being listed under an average of three Subject Headings. As a by-product magnetic tapes are produced listing an average of nine index terms (Subject Headings) against each title. The computer version thus has a much greater depth of indexing than the printed Index

Medicus, but the extra Headings are, in general, less important content indicators than the printed Headings. These magnetic tapes are searched at computer installations in several countries [33, 34]. The data presented below is for searching performed at Newcastle University for British users of the system. The standard search technique compared each reference with a logical (or Boolean) expression of index terms e.g.

Milk and (Goats or Cows)

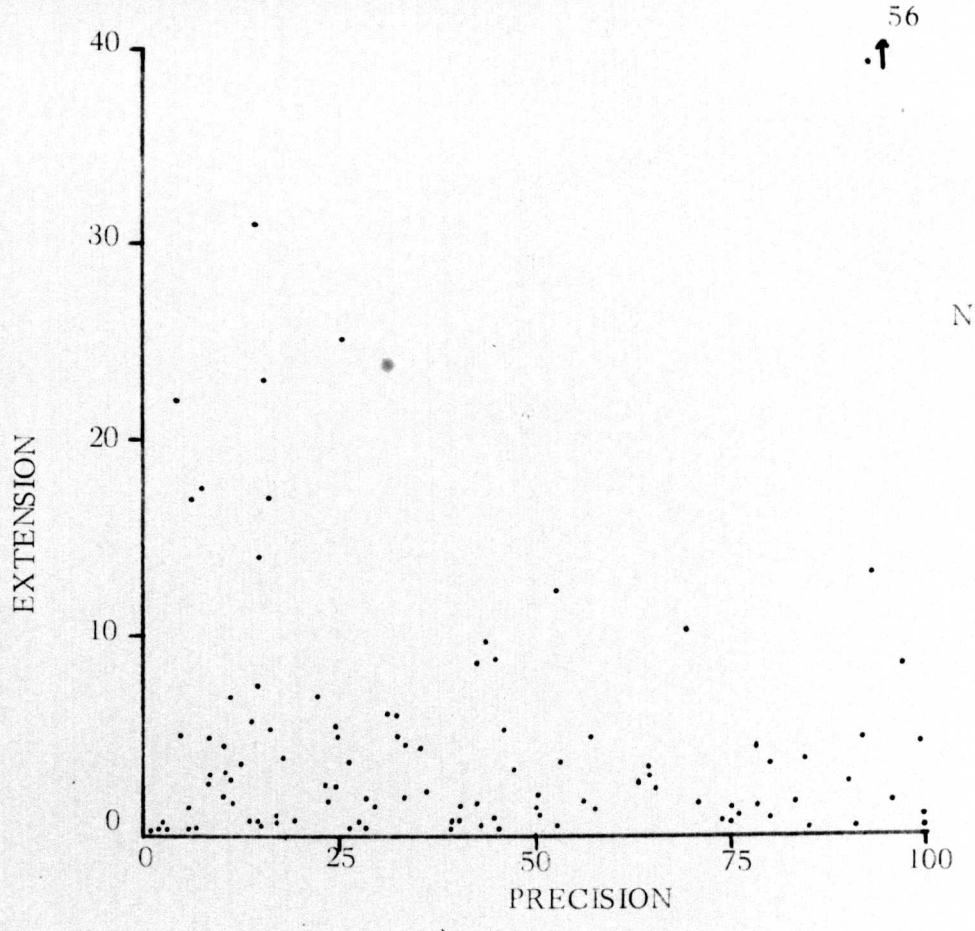
would retrieve references on cows milk and on goats milk. All results in this Chapter are for such "Boolean" searches.

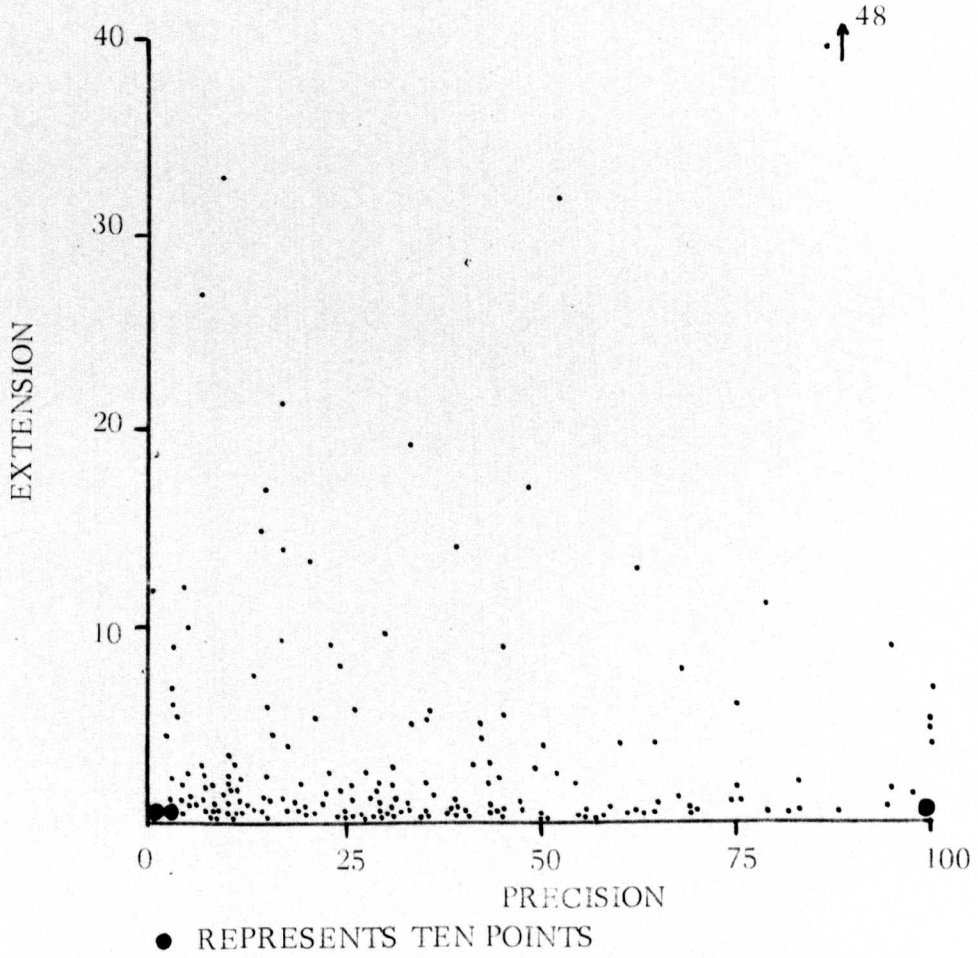
Results from 211 NLL MEDLARS searches and 104 Newcastle University MEDLARS searches are shown. (National Lending Library (NLL) searches were based on requests received by telephone and letter. Newcastle University searches were formulated at interviews, each interview taking up to one hour).

Graph 1. Extension Ratio v Precision Ratio. It might have been expected that high Extension would be achieved in searches having low Precision. In fact no such inverse relationship exists. Nor is there a positive relation.

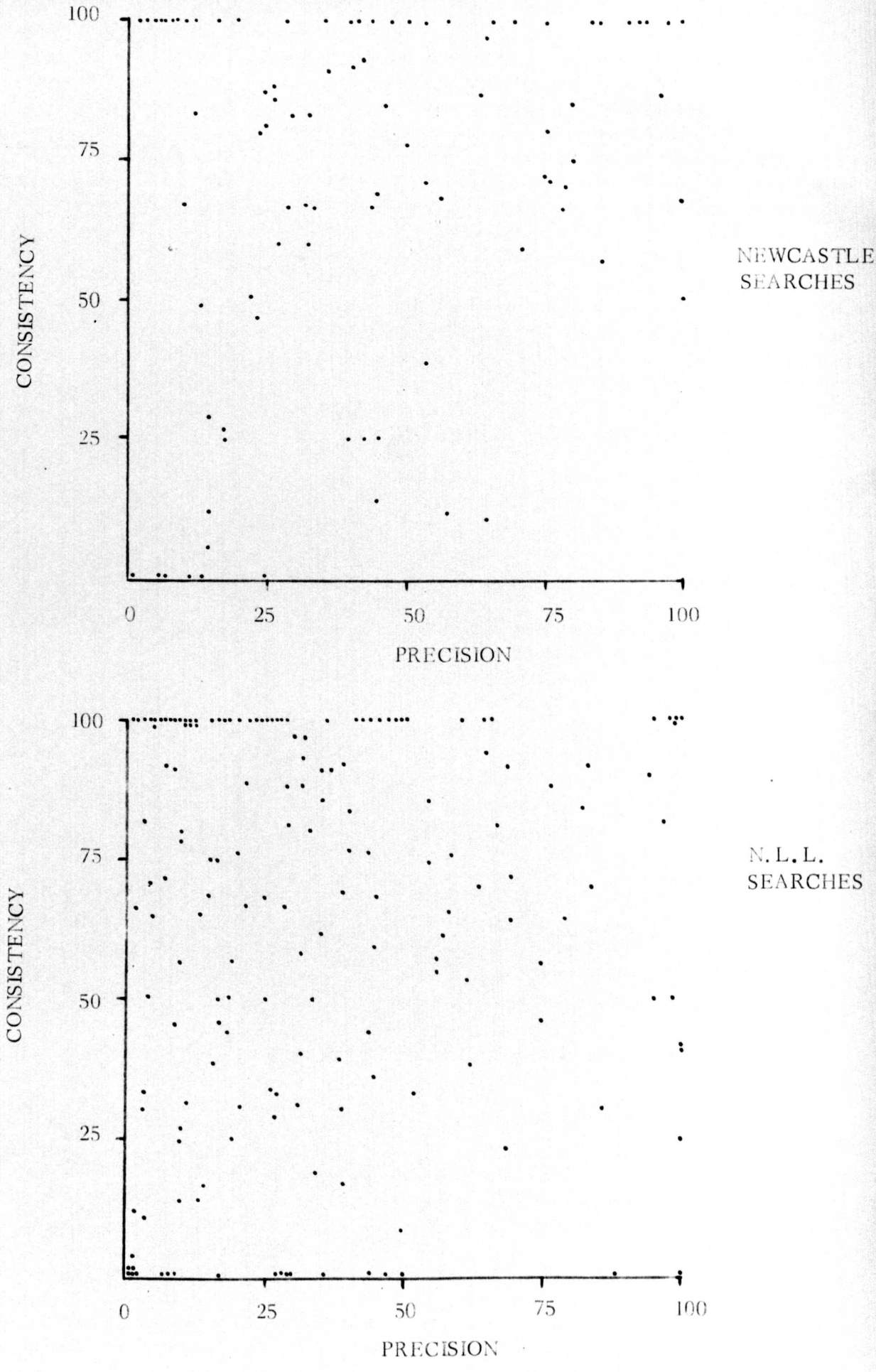
Graph 2. Consistency Ratio (Recall Estimate) v Precision Ratio. This is a scatter diagram for individual searches and is equivalent to Fig. 14 (p. 129) of the American MEDLARS test [32]. It has been frequently suggested [e.g. in 35] that an "inverse relation exists between Recall and Precision whatever the variable may be that is changed". This is almost a tautology if the only variation permitted is the strict narrowing or widening of the search formulation. If the search has been well-formulated so that in its narrowest form it achieves its highest Precision, then the statement quoted is tautologous for variations in the search specificity.

MEDLARS permits three subsearches for each search, each strictly narrower than its predecessor. By averaging results over all first-subsearches, then over all second-subsearches, etc., three (Recall, Precision) points can be plotted





GRAPH 1 EXTENSION - PRECISION



GRAPH 2. CONSISTENCY (RECALL EST.) - PRECISION

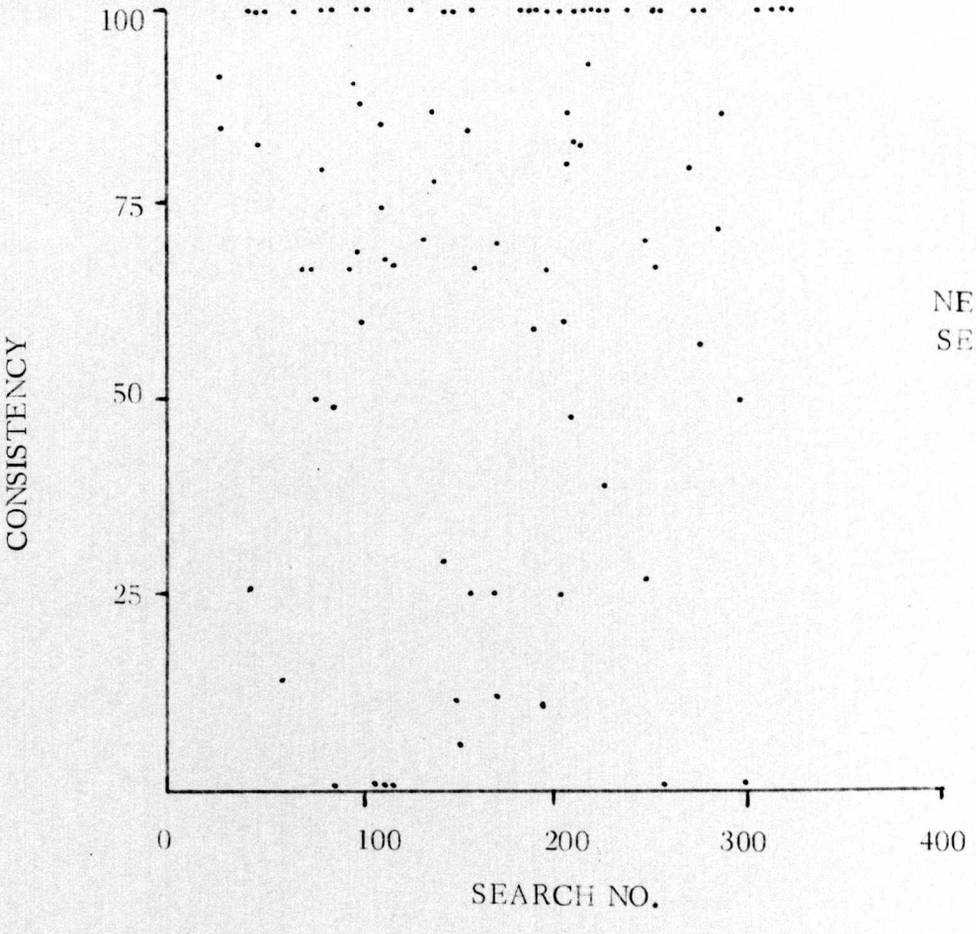
(fig. 7 page 78 of [32]), which exhibit an inverse relation. This, however, merely shows that on average the American MEDLARS searches were well-formulated.

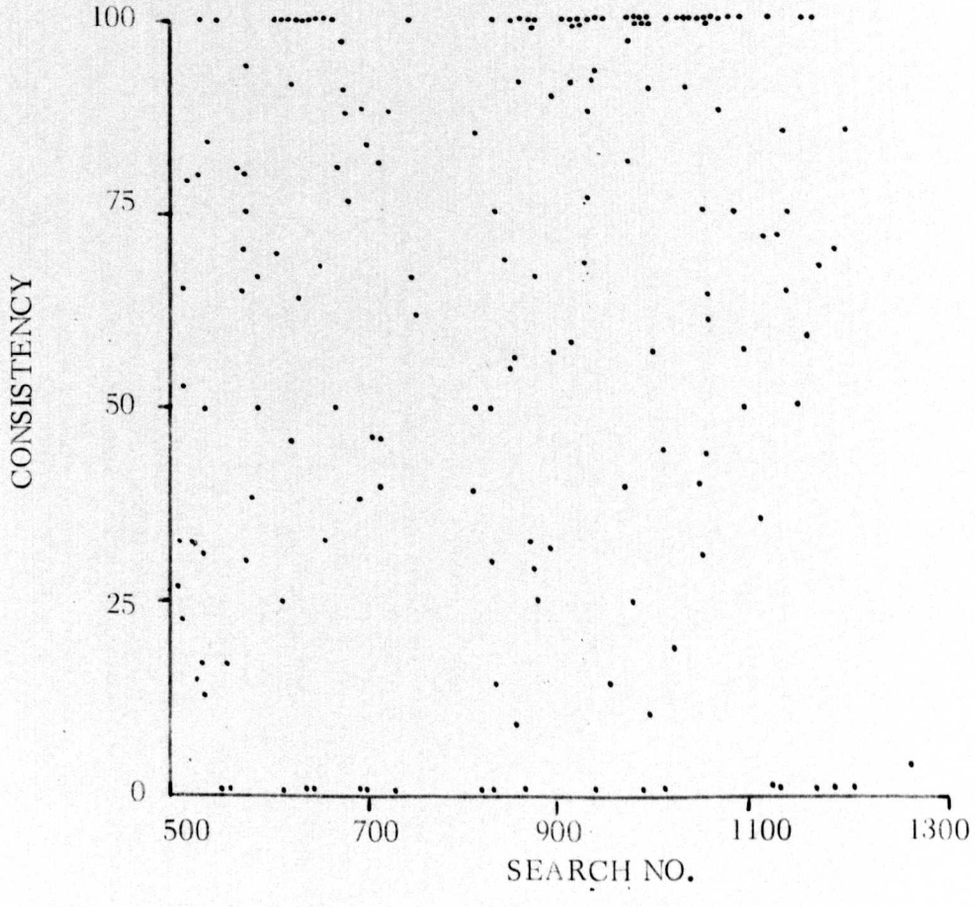
It would be of greater interest and importance if it could be shown that searches achieving a high Recall did so at the expense of low Precision and vice versa. In so far as Consistency is a good estimate of Recall, this effect, if it exists, would show in Graph 2. It does not. Indeed, the lack of points in the high Precision - low Consistency quarter indicates a loose positive relation between Consistency and Precision.

Thus, either Consistency is not a good estimate of Recall, or there is no "trade-off" between Recall and Precision, or both hypotheses are wrong. It was argued above, on theoretical grounds, that Consistency was not a good estimate of Recall. The "trade-off" hypothesis may well also be wrong. Except in the circumstances where the hypothesis is tautologous or nearly so i.e. where the only permitted variation is in search specificity, a positive relation is reasonable. For example, for one search the vocabulary may be more suitable than for another or the search writer may understand the problem better. In such cases, the variation between searches can produce a positive relation between Recall and Precision.

Since for large operational systems Recall is never known, any relation involving Recall must remain a matter for conjecture. Certainly Graph 2 can not prove one exists. However, Graph 2 does suggest a positive relation between Consistency and Precision.

Graph 3. Consistency v Search Number. There was no marked improvement in Consistency performance during the period of the test.





GRAPH 3. CONSISTENCY (RECALL EST.) - SEARCH NUMBER

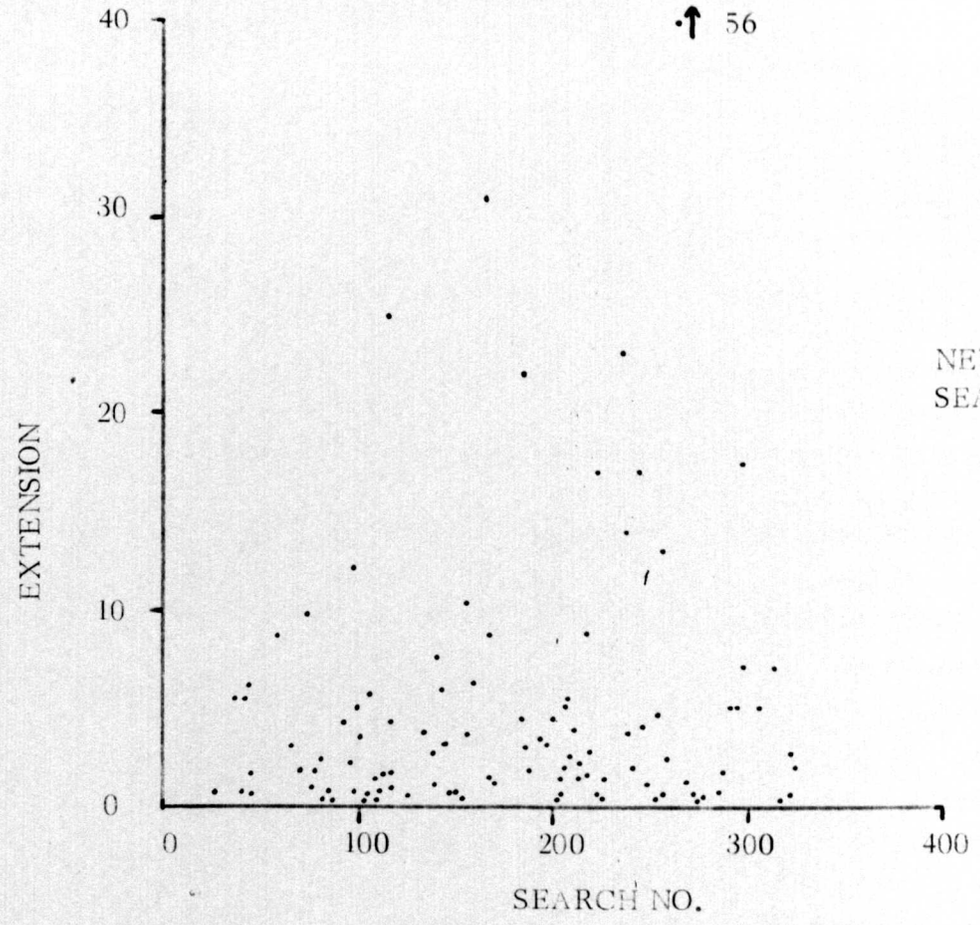
Graph. 4. Extension v Search Number. There was no marked improvement in Extension performance during the period of the test.

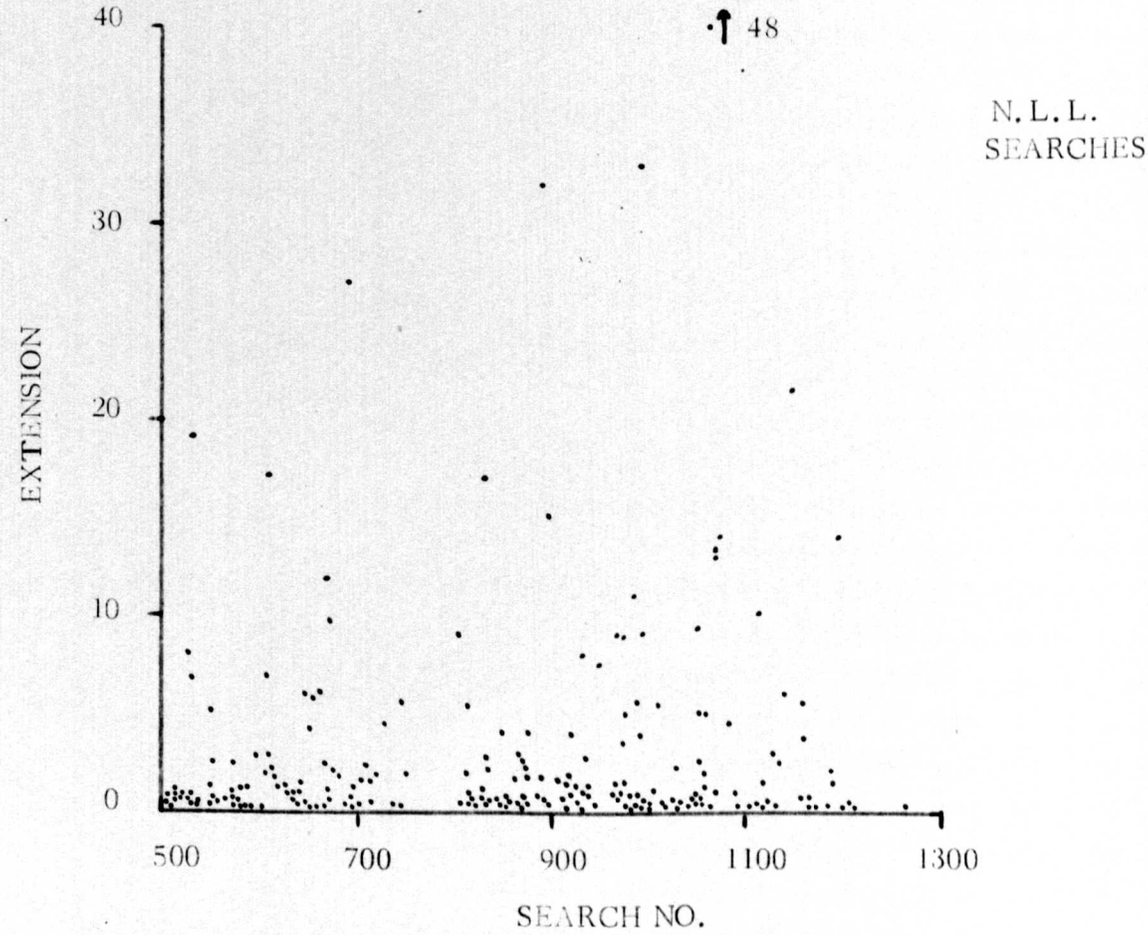
Graph 5. Extension Ratio v Consistency Ratio (Recall Estimate)

Extension measures the ability of the system to provide new relevant references. Consistency measures its ability to retrieve known relevant references. If the Extension Ratio rose and fell with the Consistency Ratio, the latter could be regarded as a good index of retrieval efficiency, not because it was a good estimate of Recall, but because it reflected, in a single measure, both the Extension and the Consistency. The graph shows that this is not so, nor is there any obvious trade-off relation. The two ratios appear to measure fundamentally different aspects of retrieval performance.

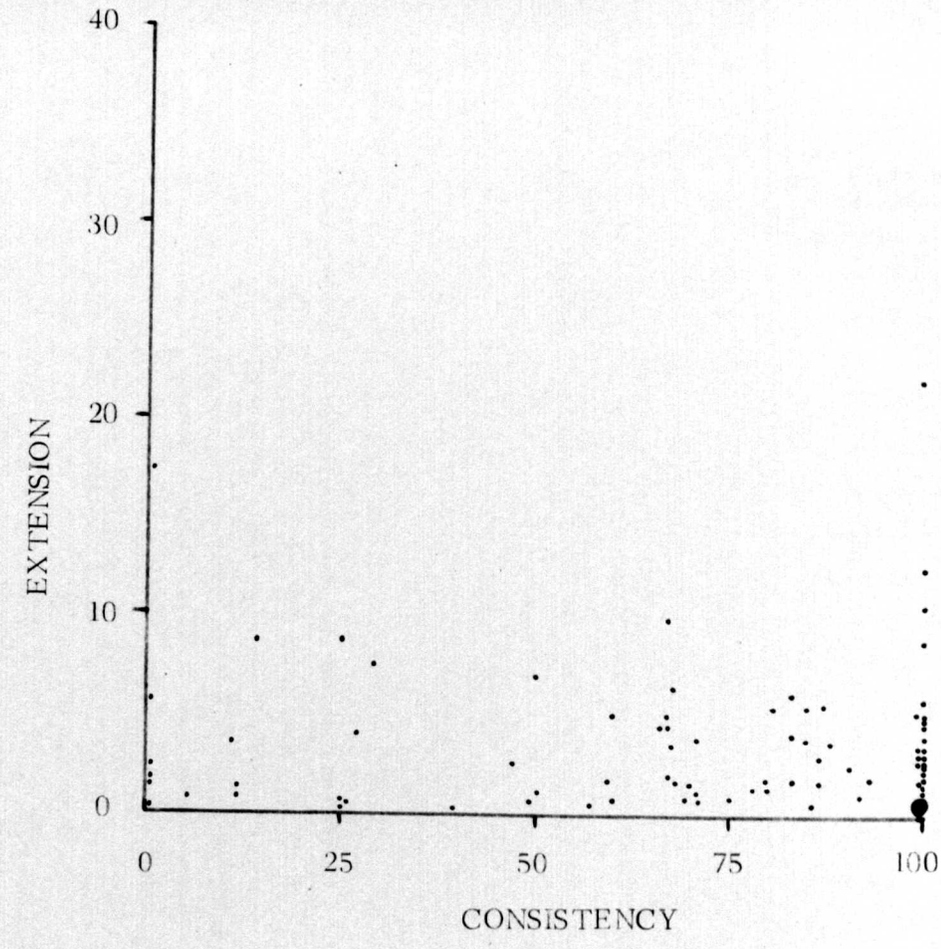
A measure of retrieval performance inevitably carries with it an implication that the measure should be maximised. Total Recall is obviously better than partial Recall in the absence of any accompanying disadvantages such as higher costs or lower Precision. Extension and Consistency as measures, must be evaluated in this light. The achievement of higher Consistency Ratios is a useful exercise in the training of search writers, but it confers no benefits on the users. The usefulness of a retrieval system to its users should be measured by some formula, which shows how much better off they are after using the system. Extension is one such measure. MEDLARS performance was therefore evaluated in terms of Extension. Since no trade-off between Extension and Precision was evident in Graph 1, it is possible to quote the Extension Ratio without qualifying it with a Precision Ratio. It follows that MEDLARS retrieval performance can be assessed by inspecting Graph 6 alone.

Graph 6. Extension Ratio (E) v No. of Relevant References known to User before Search (F). For purposes of clarity this Graph is drawn to one scale for $F \leqslant 5$ and to another for $F \geqslant 5$. This retains as many individual points as possible to avoid any loss of information.

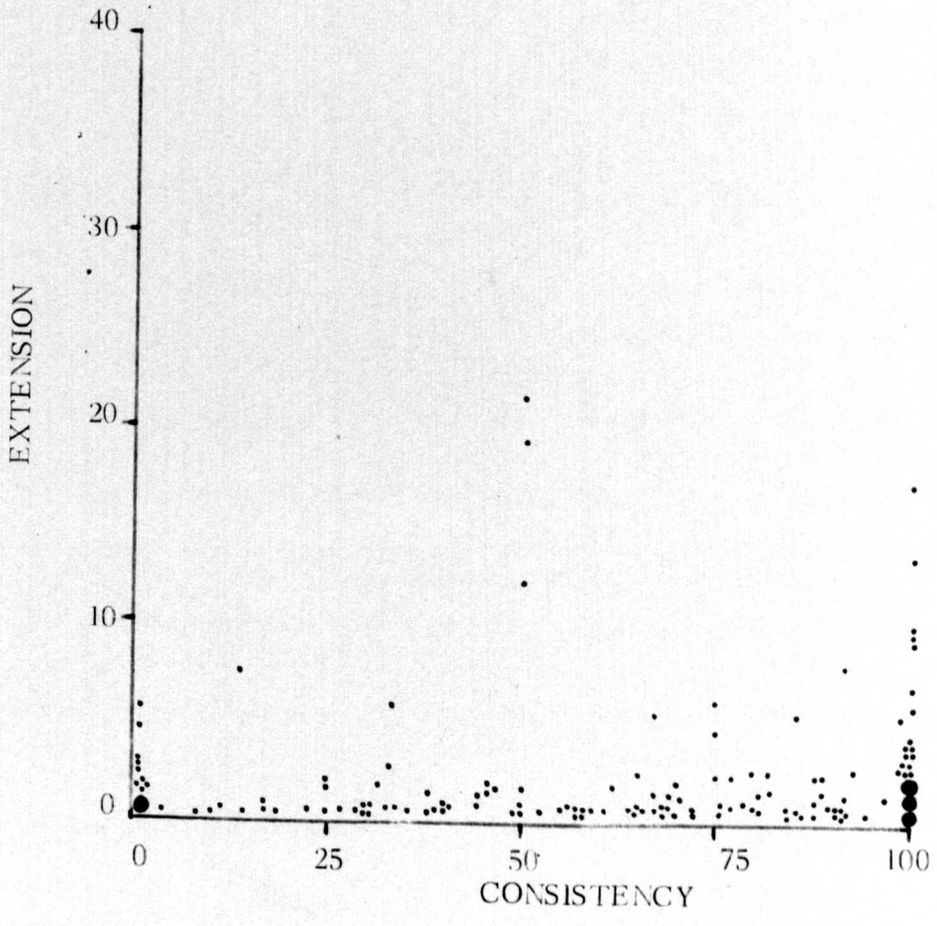




GRAPH 4. EXTENSION - SEARCH NUMBER



NEWCASTLE SEARCHES



N.L.L. SEARCHES

● REPRESENTS TEN POINTS

GRAPH 5. EXTENSION - CONSISTENCY (RECALL EST.)

To interpret Graph 6 it is useful to restate briefly the theoretical points made earlier in this Chapter. These were:

(i) For users with equally stringent notions of relevance, the probability distribution of E for a particular value of F is $P(E_F)$ where

$P(E_F) = \frac{P(T)}{F+1}$ and P(T) is the probability distribution of T, the total number of relevant references retrieved, and in particular the expectation

$\mathcal{E}(E_F) = \frac{\mathcal{E}(T)}{F+1}$

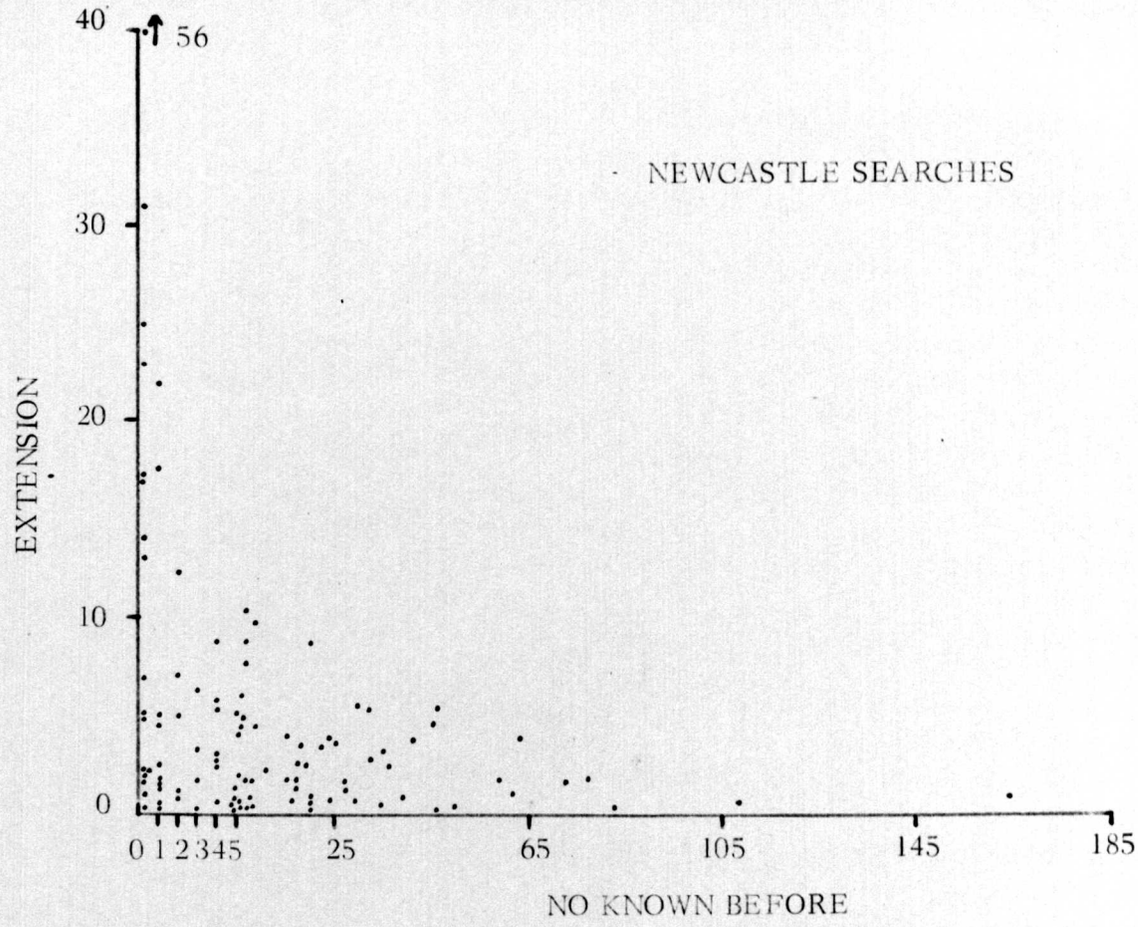
(ii) For users who are equally well-informed, the expected value of E is approximately invariant under changes in F so that

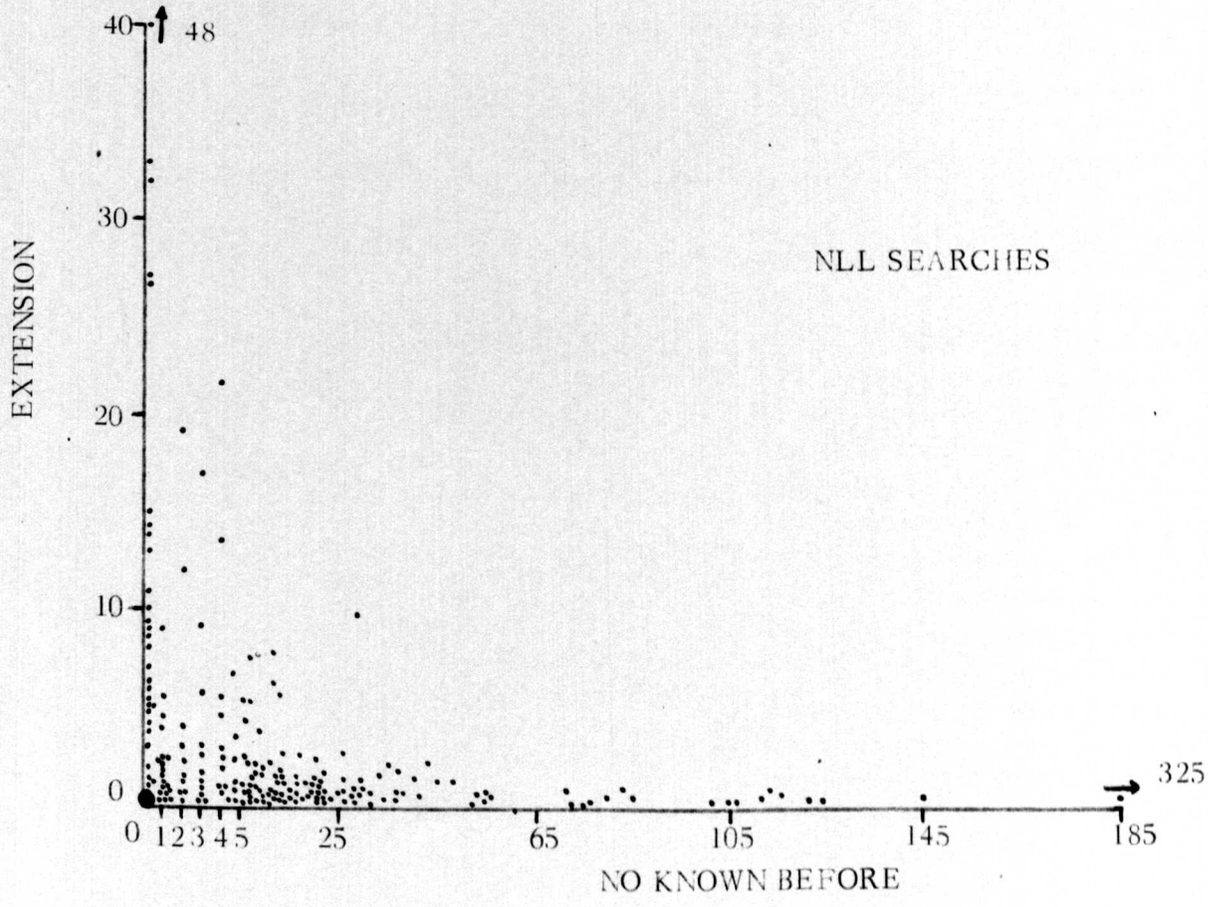
$\mathcal{E}(E_F) =$ constant.

In both circumstances the variance may be high. These two effects are both operating, and Graph 6 shows a scatter of points which does not fit either of these hypothetical situations but lies somewhat between i.e., the Graph points are such that the average E for particular F, $A(E_F)$ declines as F increases but not so fast as to be inversely related to F + 1. The average extension is given below for bands of F. Bands are chosen to be wide enough to contain sufficient points. To the nearest 0.5 the averages are

Band of F	0	1	2	3	4	5→9	9→13	13→17	17→21	21→25	25→45	45→65	65→85	85→105	105→125
Average NLL searches	8.5	2	4	4	5.5	2	2	1.5	1	1	1	1	0.5	0.5	0.5
Newcastle searches	14	6	5.5	3	4	3	5.5	2	2.5	2.5	2.5	2.5	1	0.5	0.5

If the "equally stringent" effect were the only one present the average E would be cut to one sixth of its initial value by the time F = 5. This value $A(E_5)$ would itself be cut to $^1/_{10}$ of its size by F = 60 i.e. $A(E_{60}) = {}^1/_{10}\, A(E_5)$





GRAPH 6. EXTENSION - NO. OF RELEVANT REFERENCES KNOWN TO

and similarly $A(E_{120}) = \frac{1}{2} A(E_{60})$ etc.

From the table it can be seen that in the range $0 \leq F \leq 5$ the average does follow an approximately inverse curve especially for Newcastle-formulated searches. In this range the predominant effect is the "equally stringent" effect. The users' literature-awareness seems to dictate how many relevant references they know before the search. The very high extension ratios in this range thus reflect the users ignorance as well as the capability of the system.

For $F \geq 5$, $A(E_F)$ declines much more slowly than would an inverse curve. The "equally stringent" effect is much less marked although still evident. Over the range $5 \leq F \leq 100$ the value of $A(E_F)$ declines to about $1/5$ of its initial value. An inverse curve would decline to $1/17$ of its initial value. Thus for $F \geq 5$ the number of references known by the user before the search does not measure directly his awareness of the literature.

Another approach to assessing the relative importance of the two effects is to calculate a least squares regression line

$E = \alpha + \frac{\beta}{F+1}$ where α and β are constants to be chosen by the least squares method. The values of α and β indicate respectively the relative importance of the "equally informed" and "equally stringent" effects. This uses the least squares method to produce a summary statistic (the curve $E = \alpha + \frac{\beta}{F+1}$). Since there is no reason for the Graph points to be on or near a line, it is meaningless to calculate measures of the goodness-of-fit (or significance) of this regression line. Regression lines were calculated for several subsets of data points and are displayed in the table.

All searches

$$E = 1.280 + \frac{8.277}{F+1} \quad (2, 87)$$

All NLL searches

$$E = 1.045 + \frac{7.015}{F+1} \quad (2, 88)$$

All Newcastle searches

$$E = 1.624 + \frac{10.91}{F+1} \quad (6, 87)$$

$F \leqslant 5$

$$E = 1.784 + \frac{6.088}{F+1} \quad (36, 77)$$

$F \geqslant 5$

$$E = 0.598 + \frac{12.09}{F+1} \quad (7, 77)$$

$F \leqslant 5$

$$E = 0.00 + \frac{13.53}{F+1} \quad (100, 100)$$

$F \geqslant 5$

$$E = 1.77 + \frac{16.58}{F+1} \quad (5, 61)$$

The figures in brackets show the minimum and maximum percentage of E attributable to the inverse term in the appropriate range of F. These regression lines can be misleading. For example, the line for Newcastle searches with $F \geqslant 5$ has a constant term of 1.77 and thus declines asymptotically to that value. This value is three times the corresponding one for the NLL searches. However, no conclusions should be drawn from this as the higher value of the asymptote is produced partly by the scarcity of data points from Newcastle for which $F > 80$. In fact for $F > 80$ neither Newcastle nor NLL achieved Extension ratios higher than one. The regression lines must be considered with the data points they summarise.

Simply by considering averages $A(E_F)$ it was seen that the "equal stringent" effect was predominant for F less than about 5. For the Newcastle searches, the constant term in the regression (for $F \leqslant 5$) is zero, which also shows that this effect predominates. For the NLL searches the regression line gives less clear evidence, the constant term for the $F \leqslant 5$ line being non-zero, and in fact greater than the constant term for the $F \geqslant 5$ line. The percentages of E attributable to the inverse term also give a guide to the relative importance of the two effects. Again the inverse effect is much more important in the range $0 \leqslant F \leqslant 5$ than in $F \geqslant 5$.

The results for the Newcastle searches show higher Extension Ratios than for the NLL searches. The differences between the two groups of searches, which could affect the Extension Ratio are given in the table:

Newcastle Searches	NLL Searches
(1) Contact between user and search-writer was by personal interview.	(1) Contact between user and search-writer was by post and by telephone.
(2) Questionnaire asked that new "relevant" references found by MEDLARS be marked on an evaluation sheet provided.	(2) Questionnaire asked that new "relevant" references be marked, only if the user intended to obtain the full text.
(3) MEDLARS service easily available, being operated on the campus.	(3) Contact with MEDLARS only after user had consulted his own librarian.

All three differences tend to produce higher Extension Ratios for Newcastle searches. Difference (1) tends to make Newcastle searches better formulated, (2) depresses the number of relevant references found by MEDLARS for NLL users, without depressing the number of relevant references they already know, and (3) tends to make the NLL users more aware of the literature before their MEDLARS search.

MEDLARS retrieval performance is summarised in the table below, which shows the average Extension Ratio for various bands of F (Extension Ratio to nearest 0.5).

F	0→0	1→4	5→12	13→65	65 +
Newcastle	14.0	5.0	3.5	2.5	0.5
NLL	8.5	3.5	2.0	1.0	0.5

MEDLARS EXTENSION RATIOS

References from Chapter 1.

1 "The MEDLARS Story at the National Library of Medicine," National Library of Medicine, Washington, D.C., 1963.

2 "An Experiment in Selective Dissemination of Information", The Chemical Society Research Unit, University of Nottingham, 1966.

3 Aitchison, T.M., Clague, P., Report SDI/1 Nat. Elect. Res. Counc., 1966.

4. "Proceedings of the First Cranfield Conference on Information Retrieval," Pergamon Press, London, 1968.

5 Salton, G., Amer. Doc., 16, 1965, 209.

6 Swets, J.A., Science, 141, 1963, 245.

7 Cleverdon, C., Mills, J., Amer. Doc., 15, 1964, 4.

8 Dake Gull, C., Amer. Doc., 7, 1956, 320.

9 Painter, A.F., "An Analysis of Duplication and Consistency of Subject Indexing involved in Report Handling at the Office of Technical Services (O.T.S.), U.S. Dept. of Commerce", O.T.S., Washington D.C., 1963.

10 Rodgers, D.J., "A Study of Inter-indexer Consistency", and "A Study of Intra-indexer Consistency," both General Electric Co., Washington, D.C., 1961.

11 Bryant, E.C., King, D.W., Terragno, P.J., "Analysis of an Indexing and Retrieval Experiment for the Organometallic File of the U.S. Patent Office", Report, WRA-PO-10, Westat Research Analysts, Denver, Colorado, 1963.

12 MacMillan, J.T., Welt, I.D., Amer. Doc., 12, 1961, 27.

13 Rees, A.M., "The Evaluation of Retrieval Systems", Report CSL:TR-5, Center for Documentation and Communication Research, Western Reserve Univ., Cleveland, Ohio, 1965.

14 Barhydt, G., "Parameters of Information Science", Spartan Books, Washington, D.C., 1964, 383.

15 Doyle, L.B., "Automation and Scientific Communication", American Documentation Institute, Washington, D.C., 1963, 199.

16 Resnick, A., Science, 134, 1961, 1004.

17 Resnick, A., Savage, T.R., Amer. Doc., 15, 1964, 93.

18 Aitchison, J., Cleverdon, C.W., "Report on a Test of the Index of Metallurgical Literature at the Western Reserve University," Cranfield College of Aeronautics, England, 1963.

19 Cleverdon, C.W., "Report on the Testing and Analysis of an Investigation into the Comparative Efficiency of Indexing Systems", Cranfield College of Aeronautics, England, 1962.

20 "Conclusions" in "Information Retrieval in Action", Western Reserve University Press, 1963.

21 Cleverdon, C.W., Library Quarterly, 35, 1965, 121.

22 Swanson, D.R., Library Quarterly, 35, 1965, 1.

23 Kent, A., "Information Processing 1962", North-Holland Publ. Co., Amsterdam, 1963.

24 Rees, A.M., "Review of a Report of the ASLIB-Cranfield Test of the Index of Metallurgical Literature of the Western Reserve University", Western Reserve University, 1963.

25 Sharp, J.R., J. Doc., 20, 1964, 170.

26 Rees, A.M., "Search Results" in "Information Retrieval in Action," Western Reserve University Press, 1963.

27 Rolling, L.N., J. Doc., 22, 1966, 93.

28 Fels, E.M., Amer. Doc., 14, 1963, 28.

29 Goffman, W., Newill, V.A., I.S.R., 3, 1966, 19.

30 Rogers, F., Bull. Med. Lit. Assoc., 54, 1966, 1.

31 Rogers, F., Bull. Med. Lib. Ass., 54, 1966, 316.

32 Lancaster, F.W., "Evaluation of the MEDLARS Demand Search Service", National Library of Medicine, Washington, D.C., 1968.

33 Harley, A.J., "Results of the first year's operations and Evaluation Studies", U.K. MEDLARS Service, National Lending Library for Science and Technology, Boston Spa, 1967.

34 King, M., "Report on the operation of the MEDLARS Service in the Newcastle region", Report to O.S.T.I., London, 1968.

35 Cleverdon, C.W., Mills, J., Keen, M., "Report on the Factors Determining the Performance of Indexing Systems," Cranfield College of Aeronautics, England, 1966.

Chapter 2. Probabilistic Searching

2.1 Coordinate Indexing

2.2 Boolean Searching using a Coordinate Index

2.3 Scoring Searching

2.4 The Information Content of a Coordinate Index Term

2.5 Some differences between MEDLARS indexing and Pure Coordinate Indexing.

1. Pre-coordinated Terms
2. Category Numbers
3. Subheadings
4. The "most-specific term" indexing convention
5. Index Medicus Headings

2.6 Modification of the Scoring technique for Categories.

2.7 Modification of the Scoring technique for Equivalent terms.

2.8 A KDF9 Program for Probabilistic Searching of the MEDLARS file Input-Store-Restrictions-Retrieval Action - Time.

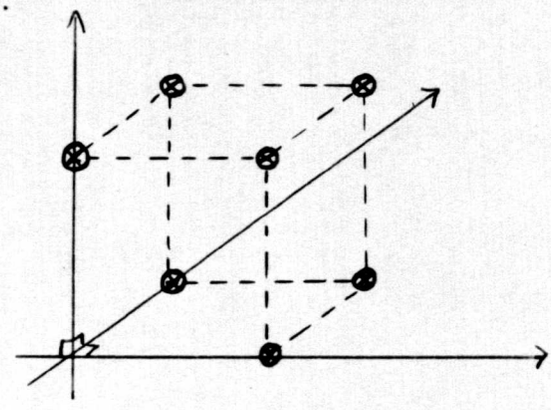
2.9 A Comparison of Boolean and Probabilistic Searching of the MEDLARS file: Tests and Results. Test 1 - Results of Test 1 - Conclusions from Test 1. Test 2. - User-categories - Time - Results of Test 2. Output size v performance - Causes of failures - Conclusions from Test 2.

Chapter 2. Probabilistic Searching

2.1 Coordinate Indexing

In its simplest form, Coordinate Indexing locates each reference at a unique point in a space whose coordinate directions represent index terms. This space may be finite dimensioned ("use a restricted Vocaublary") or infinite dimensioned (e.g. free-language indexing). The coordinates are orthogonal, i.e. the application of one index term to the reference does not have any implications as to the application of other terms.

When index terms can only be applied or not applied, i.e. when weighting of the index terms is not permitted, the only points at which a reference may be located are the vertices of a hypercube - one of whose vertices is at the origin. This may be represented in a simple diagram which will be useful later in the description of deviations from pure Coordinate Indexing.



Diag. 1

Diag. 1 shows the possible locations of references in a Coordinate Index which uses only three index terms.

2.2 Boolean Searching using a Coordinate Index

A Boolean Expression is a set of index terms linked by the operators "and, or, not" and which takes the value "true" or the value "false" for a particular reference when the index terms are replaced by true or false according as they do or do not appear in the reference. Formally Boolean Algebra is an isomorphism of binary arithmetic under the transformation

and	X
or	+
not	1- i.e. not 0 = 1, not 1 = 0
true	1
false	0

with + defined by 1 + 1 = 1, 1 + 0 = 0 + 1 = 1, 0 + 0 = 0. Less formally a Boolean Expression may be illustrated by example. Thus,

B = milk and (cow or goat)

takes the value true for any reference indexed with (milk, cow) or (milk, goat) and takes the value false otherwise. A Boolean Search consists of the retrieval of all references for which some Boolean Expression takes the value "true".

The presence of "not" in Boolean Expressions used for information retrieval has been much criticised [1, 2, 3, 4]. It can only be justified if the presence of an index term is a sufficient reason for not retrieving a reference, irrespective of what other index terms are attached to it. The subject of interest to the system-user, and the subject represented by the negated term must be mutually exclusive in the index. This can never happen in a Coordinate Index. The alternative to negation is praeternegation [1], i.e. the listing of acceptable alternatives. This is always possible. The most general form of a Boolean Search Statement is thus a positive Boolean Expression, i.e. one which does not contain the operator not.

In systems based on Boolean retrieval the operator not may be available. This is true of the MEDLARS system [5]. It is available because the list of acceptable alternatives may, on occasion, be long and there is a danger that some may be overlooked. Furthermore the cost of the search (in computer time) may depend upon the number of index terms in the search statement. Then negation has the advantage of economy, if not of retrieval-efficiency. Again,

this is true of MEDLARS as implemented on KDF 9.

2.3 Scoring Searching

Since a Boolean Search divides the file into just two categories - retrieved / not retrieved, it is very sensitive to the omission of an important index term from a reference, from the Boolean Search Statement, or from the Vocabulary. Such omissions are to be expected. For want of time or of subject-knowledge, either the indexer or the (professional) search-formulator may fail to include a relevant term. Indeed, there is no reason to suppose that the indexer and searcher will be more consistent than a pair of indexers, and several experiments have indicated that the consistency of indexing is low [6]. But this understates the problem for the interest of a reference to a particular user may not lie in its main topic, and minor topics may not be exhaustively indexed. Occasionally, when a restricted vocabulary is used, the obvious term to describe the reference may not exist in the vocabulary, and indexer and searcher may choose different substitutes, e.g. in the MEDLARS system the term "cell wall" was not available at one time [7]. The term "cell membrane" though apparently close was rejected as entirely different by users with good subject knowledge.

Scoring search techniques are designed to use imperfect indexing by assigning scores to references rather than retrieved/not-retrieved indicators [8]. For ease of discussion and without loss of generality, these scores can be assumed positive numbers, the references with numerically higher scores being retrieved in preference to those with lower scores. A Boolean Search is an example of the crudest possible Scoring technique i.e. one where only two distinct scores are possible. Except in this degenerate case, the assignment of scores has the advantage that the lack of a particular search-term may reduce the score of a reference without (by itself) preventing the retrieval of that reference.

Two criteria for retrieval are convenient. Either a reference's score must exceed some threshold value T, or its score must be one of the top N scores. Especially with the latter criterion, it makes sense to present those references which are retrieved in order of decreasing score in the hope that the user will find the information required without examining all the retrieved references. This is a possibility if the user is after a fact, rather than a complete survey of the literature, e.g.

"What is being done in paediatric post-operative care?"

MEDLARS users, at least in the U.K., tend toward the latter type of query.

The reference's score may be calculated from any of its attributes, e.g. author's name, citation links to other references, title, index terms, language, the journal in which it appears. A user may well prefer references which are easily accessible and thus prefer major journals in his own language. However, if the score is calculated purely from the index terms, there is a fundamental difference between retrieval by the threshold criterion and retrieval by the second criterion. In that case, retrieval by the threshold criterion is equivalent to Boolean Searching. The equivalent Boolean Search Expression is easily written down. It is,

B = (x and y and .....) or (a and b and) or .....

where each combination of terms producing a score higher than the threshold is written down in turn, and each combination is linked to the next by "or". It follows that only a Scoring Search using the second criterion of retrieval is different from Boolean Retrieval, and that a Scoring Search can only do better (or worse) than a Boolean Search when it uses that criterion.

When the score is a function of index-terms only, it may be represented by V = f(S) where S is the set of index-terms attached to a reference. A simple example is:-

$$f(S) = \sum_{i \in S} Wi$$

where a weight W_i is associated with each term in the vocabulary. In a normal search nearly all the W_i would be zero. The non-zero W_i would be associated with the "search terms".

The function f(S) must have a property corresponding to the restriction of Boolean Search Expressions to positive Boolean Expressions. Where one set of terms includes (strictly) another the score for the former should be at least as high, i.e.

$$S^+ \supset S \text{ implies } f(S^+) \geqslant f(S).$$

since the addition of a further index-term to a reference cannot make it any the less relevant to the topics represented by its other index terms.

With this one restriction, the formula used to calculate the score may be whatever is (i) convenient to calculate in a given system and (ii) gives a good retrieval performance in that system. The intuition of those familiar with the system may well suggest a more effective formula than a reasoned theory, particularly if the theory makes statistical assumptions. The only valid justification of a retrieval technique is experimental. But statistical arguments may justify a test, even if the test does not justify the statistics.

2.4 The Information Content of a Coordinate Index Term

The basic requirement of a Scoring Search formula is that the scores calculated by it be estimates of the probabilities of the references being relevant, or are estimates of some monotonic transformation of these probabilities.

Let $P(R/T_1 T_2 \ldots . T_k)$ represent the probability of relevance (event R), conditional on the occurrence of index terms 1 to k (events T_1 to T_k). Then the unconditional

probability:-

$$P(R\,T_1T_2\ldots T_k) = P(R)\,P(T_1/R)\ldots\ldots P(T_k/R\,T_1T_2\ldots T_{k-1})$$

and also

$$P(R\,T_1T_2\ldots T_k) = P(T_1)\,P(T_2/T_1)\ldots P(R/T_1\,T_2\ldots T_k)$$

whence by division

$$P(R/T_1T_2\ldots T_k) = P(R)\left[\frac{P(T_1/R)}{P(T_1)} \ldots\ldots\ldots \frac{P(T_k/R\,T_1\ldots T_{k-1})}{P(T_k/T_1\ldots T_{k-1})}\right] \text{equn } (\beta 1)$$

In a Pure Coordinate Index as defined in 2.1, the application of one index term cannot alter the probability of application of another. This "probability" is with respect to a population of "all references to which the indexing system could be applied." This is a hypothetical set and not merely the set of existing references. Thus, in a Pure Coordinate Index,

$P(T_i/T_j) = P(T_i)$ for all i and j, and by repeated application,

$P(T_i/T_jT_k\ldots\ldots) = P(T_i)$ for all i.

If this same property of term-independence holds for the much more restricted population of "all possible relevant references" (again a hypothetical set) then

$P(T_i/RT_j) = P(T_i/R)$ for all i and j and Equation ($\beta 1$) can be very much simplified to Equation ($\beta 2$):-

$$P(R/T_1T_2\ldots T_k) = P(R) \prod_{i=1}^{i=k} \frac{P(T_i/R)}{P(T_i)}$$

In any operational retrieval system it is most unlikely that this property would hold exactly either in the total population or in the relevant set. The concept of a Pure Coordinate Index is difficult to realise. The effect of MEDLARS deviations is considered in the following sections.

The value of P(R) may vary from search to search, but for any one search it can be taken as constant over the references in the file. (This is reasonable when only index-terms are considered. If data is available on varying P(R) it can be incorporated in an obvious manner in the following equations).

Thus the formula S_1,

$$S_1 = \sum_{i=1}^{i=k} \frac{P(T_i/R)}{P(T_i)}$$

gives a set of reference scores which is an order-preserved transformation of the set given by $P(R/T_1 T_2 \ldots T_k)$, and so for purposes of information retrieval S_1 is equivalent to the conditional probability.

The formula S_1 can be made the basis of a practical retrieval technique if the probabilities $P(T_i/R)$ and $P(T_i)$ can be estimated. It is usual in retrieval systems such as MEDLARS which employ a restricted vocabulary, to do a considerable amount of research on the terms in the vocabulary to ensure that they are well chosen. A bye-product of this work is a list of "tallies". A term's "tally" is its frequency of use as an index term. This frequency p_i, say, provides an estimate of the probability $P(T_i)$. This estimate is far from perfect, particularly if terms are periodically added to the vocabulary or deleted from it, but it is readily available, and is an estimate based on a sample consisting of all the references which have actually been indexed by the system, e.g. in 1969 the MEDLARS sample comprised over three quarters of a million references.

The estimation of $P(T_i/R)$, i.e. of the probability of application of the i th term to a relevant reference can be made by the system user either alone or in consultation with a professional search-formulator. This requires a personal judgement and again the estimate may be far from perfect. Let it be w_i.

Then the formula S_2,

$$S_2 = \prod_{i=1}^{i=k} \frac{w_i}{p_i}$$

provides an estimate of the formula S_1. It can be used to calculate the score of a reference indexed by terms 1 to k. The system user will only be interested in a few of the terms in the vocabulary, and explicit values of W_i will only be available for these "search-terms". However, by expressing no interest in a term j he is implicitly stating that $P(T_j/R) = P(T_j)$

i.e. $\frac{w_j}{p_j} = 1$ so that S_2 can be written

$$S_2 = \prod_{i=1}^{k} \left(\frac{w_i}{p_i} \right)^{\delta_{is}}$$

where the terms i(i = 1, 2, ...k) index the reference, and $\delta_{is} = 1$ if i is a search term $= 0$ otherwise.

Obviously S_2 can also be written

$$S_2 = \prod_{j=1}^{S} \left(\frac{w_j}{p_j} \right)^{\delta_{jr}}$$

where now the product is over all the search terms and δ_{jr} is 1 or 0 according as the search-term j does or does not appear against the reference.

An order preserving transformation of S_2 is the log-transform $S_3 = \log S_2$

i.e. $$S_3 = \sum_{j=1}^{s} \delta_{jr} \log \frac{w_j}{p_j}$$

and $$S_3 = \sum_{j=1}^{s} \delta_{jr} W_j \quad \text{where } W_j = \log \frac{w_j}{p_j}$$

that is, the score of a reference is the sum of the "weights" of the search-terms indexing it. The weights are assigned by the user and equal $\log \frac{w_j}{p_j}$ for the j th

search term. A retrieval technique based on the scoring formula S_3 interprets the Coordinate Index as a device for transmitting information about the relevance of the references to the user. For the individual user each index term has an "information content" of $W_j = \log \frac{w_j}{p_j}$ and these information contents are additive. This is a fundamentally different approach from Boolean Seraching and merits a brief discussion.

A standard definition of "information content" is as follows [9]:—

If a set of inputs to an information transmitting device is denoted by $\{x_k\}$ with associated a priori probabilities $\{P(x_k)\}$, and the set of outputs is $\{y_i\}$ with probabilities $\{P(y_i)\}$ the problem of information transmission is "If y_i is the output from the device, how much information does its occurrence give about the input x_k ?". The standard measure of the information given is the logarithm of the ratio of the a posteriori probability $P(x_k/y_i)$ to the a priori probability $P(x_k)$. This measure $\log \left[\frac{P(x_k/y_i)}{P(x_k)} \right]$ is the information content of y_i with respect to x_k .

If we regard the Coordinate Index as an information transmitting device with inputs relevant R, and not relevant $\overline{R}$, and outputs $\{T_i\}$(the occurrence of index terms) then the information content of term i with respect to relevance is I_i , where

$$I_i = \log \left[\frac{P(R/T_i)}{P(R)} \right]$$

Now $P(RT_i) = P(R)\, P(T_i/R) = P(T_i)\, P(R/T_i)$

whence

$$I_i = \log \left[\frac{P(T_i/R)}{P(T_i)} \right]$$

Thus $W_i = \log \frac{w_i}{p_i}$ the user-assigned weight of a search term is an estimate of the information content I_i of the i th term with respect to the relevance of a reference to the particular user.

It is possible for I_i to be negative, namely when $P(T_i/R) < P(T_i)$

i.e. when $P(R/T_i) < P(R)$

i.e. when the occurrence of an index term can indicate that the reference is less likely to be relevant than its other index terms would suggest. This is analogous to Boolean negation and by choosing values of w_i less than the known p_i the user can "weight-against" an index term, but this should not be done in a Coordinate Index system, since the arguments against Boolean negation also apply here.

The presence of "log" is important since the formula S_4.

$$S_4 = \sum_{j=1}^{s} \delta_{jr} \frac{w_j}{p_j}$$

is not an order-preserved transform of S_3.

$$S_3 = \sum_{j=1}^{s} \delta_{jr} \log \frac{w_j}{p_j}$$

and would not give the same retrieval results. On purely intuitive grounds, S_4 is a fairly obvious scoring formula to test [10] but the difference between the intuitively-derived and the theoretically derived formulae is significant.

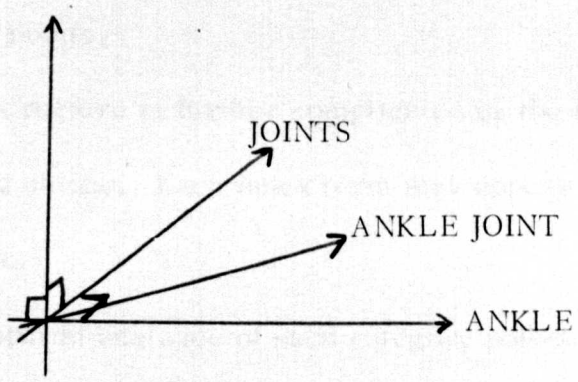
2.5 Some differences between MEDLARS Indexing and Pure Coordinate Indexing

The last section presented an information retrieval technique for use with a Pure Coordinate Index as defined in 2.1. Any operational retrieval system is unlikely to conform exactly to such a simple type. For a system which is a variant of a Coordinate Index the method developed in 2.4 is not justified by the arguments of that section, but may still be suggested by them. The method must be tested to show whether deviations from Coordinate Indexing so reduce the performance of the method that its advantages over Boolean Searching disappear.

If the major systematic deviations from Pure Coordinate Indexing are known, the method can perhaps be modified to take account of, and even make constructive use of, these differences. MEDLARS has at least five systematic differences from Pure Coordinate Indexing.

(1) Pre-coordinated Terms

The spatial analogue of a Pure Coordinate System given in 2.1 was a set of mutually orthogonal vectors representing index terms. The MEDLARS vocabulary MeSH [7] contains "pre-coordinated" terms whose analogues are vectors which are not orthogonal to all other term vectors. For example MeSH contains "JOINTS", "ANKLE JOINT", and "ANKLE" which may be represented as in the diagram,



with "ANKLE JOINT" lying in the plane defined by "JOINTS" and "ANKLE". Such a situation could be made compatible with 2.4 by using instead of "ANKLE JOINT" the individual terms and indicating in the indexing of a particular reference that for that reference, the terms were linked. The important difference that this would make is to the tallies (frequencies of occurrence) of the terms. The application of "ANKLE JOINT" would then count as an occurrence of "ANKLE JOINT", and of "ANKLE" and of "JOINT". Since this is not done the tallies p_i do not provide such good estimates of $P(T_i)$ as they could.

This example is very easy to appreciate but the lack of emphasis on orthogonality can produce terms which are effectively coordinations of others without being obviously so, e.g. the term "SEASONS" is largely, though perhaps not entirely, a pre-coordination of "WEATHER" and "PERIODICITY".

(2) Category Numbers

Superimposed on MeSH is a hierarchical category structure. There are three levels of generality and it is not a simple hierarchy in that there are many points at the most general level, not one. For example, ANKLE JOINT at category point A2.48.4 is linked to JOINTS at A2.48 which in turn is linked to MUSCULOSKELETAL SYSTEM at A2. The standard MEDLARS Boolean Search can use category numbers such as A2.48 as well as index terms. In a Boolean Search Formulation the use of A2.48 is equivalent to

"JOINTS or ANKLE JOINT or ELBOW JOINT or ....."

i.e. to the logical sum of all index terms placed at the point A2.48 or at more specific points.

The structure is further complicated by the location of a term in the hierarchy not being unique. Each index term may appear at 1, 2, 3 or 4 points in the category structure.

The spatial analogue of such category points is obvious. Each point represents a vector space which is a subspace of the full space spanned by the full set of MeSH terms. Categories can be used in a modified form of the technique of 2.4 and the modifications necessary are given in the next section (2.6).

(3) Sub-headings

A number of "Subheadings" are available, e.g. "METABOLISM", "CYTOLOGY". Subheadings are terms which can only be used when linked to one of the normal MeSH index terms. The purpose is to distinguish e.g. a reference

Ref. 1 on A (as a therapy), B (as a poison)

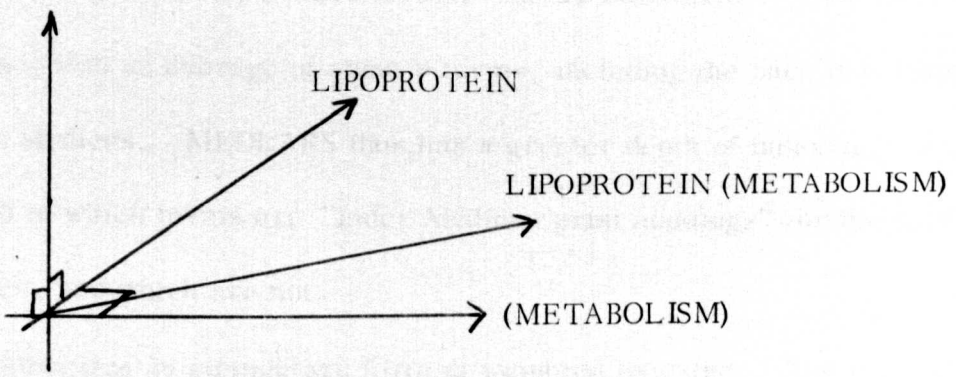
from a second reference

Ref. 2 on A (as a poison), B (as a therapy)

Although the subheadings can only be used to index a reference when linked to an index term, it is possible when performing a Boolean Search to detect the

occurrence of a Subheading irrespective of which term it is associated with, e.g. it is possible to specify that the subheading "METABOLISM" should appear or "LIPOPROTEINS (METABOLISM)" i.e. the term "LIPOPROTEINS" linked to the subheading "METABOLISM".

Indexing by "term-plus-subheading" is thus similar to using a pre-coordinated term, but with the important difference that the pre-coordination can be removed and the two component terms treated individually. Providing tallies are kept for terms, subheadings, and all combinations of term-plus-subheadings the method of 2.4 can be used. (In the U.K. MEDLARS system these tallies are not available).



The subheadings can be regarded as additional index terms, represented by additional vectors as shown in the diagram.

(4) The "most-specific-term" indexing convention

Since MeSH contains terms at different levels of generality there could be some ambiguity as to whether to use e.g. "MICE" or "RODENTS" to index a reference about mice. There is therefore a convention that only the most specific term possible should be used. Thus a reference on mice is indexed by "MICE" but not by "RODENTS". It is possible that a reference be indexed by both "MICE" and "RODENTS". This would occur when the reference was about mice (hence "MICE") and also about some rodent e.g. chipmunk which was not a MeSH term. Since chipmunk did not exist in MeSH the most specific term available would be "RODENT". This does not cause any problems either in Boolean or Probabilistic (2.4) searching provided that the meaning of the term "RODENT"

is clearly understood. If what is required is "anything on rodents" then the category point B2.72.58 should be used in either search formulation. This category point is equivalent to the logical sum of all the MeSH rodent terms - "RODENT", "CHINCHILLA", "GERBIL", "GUINEA PIG" ..... etc.

(5) Index Medicus Headings

The entire MEDLARS system is a bye-product of the printing process of the publication "Index Medicus" which lists the titles of all references as they appear. In Index Medicus each title is listed under an average of about 2 or 3 "Headings". These Headings are MeSH index terms. In the MEDLARS system each reference is indexed with an average of about 9 terms, including the ones it is listed under in Index Medicus. MEDLARS thus has a greater depth of indexing. It also has a record of which terms are "Index Medicus print headings" for the particular reference, and which are not.

This provides an elementary form of weighted indexing. The method of 2.4 can be modified to use this additional information. A simple means is to modify S_3,

$$S_3 = \sum_{j=1}^{s} \delta_{jr} \log \frac{w_j}{p_j}$$

by having two values of p_j, the frequency of occurrence. A record is not kept of the frequencies of occurrence of terms as print headings. As a substitute for this, it can be arbitrarily assumed that one third of each term's occurrences are as print headings ($\frac{1}{3} \doteqdot \frac{3}{9}$). This is obviously an approximate procedure. Then for a particular reference p_j or $p_{j/3}$ can be used according as the term is not, or is a print heading.

2.6 Modification of Scoring Technique for Categories

The use of a MeSH category number in a Boolean Search Formulation is equivalent to the logical sum of the index terms designated by that number. This is equivalent to the collapsing of the subspace spanned by those terms into a single vector, to be used wherever any of the individual terms were previously applied. All that is necessary for the use of category numbers in the scoring search technique (2.4) is values w_c and p_c which are estimates of $P(T_c/R)$ and $P(T_c)$, where T_c represents the event:-

"the reference is indexed by a term having the specified category number".

The probability $P(T_c)$ can be calculated from the equation,

$$P(T_c) = 1 - \prod_{j=1}^{j=N} (1 - P(T_j)) \qquad \text{(equn } \beta 3\text{)}$$

where the terms j = 1 to N have the category number c (or, of course, a more specific category number). Equation ($\beta 3$) is valid when the event "a reference is not indexed by term i" is statistically independent of the event "not indexed by j". This condition is met for a Pure Coordinate Index. More important, it is the same condition as is necessary to validate the method of 2.4. It thus brings in no further conditions or assumptions.

From Equation ($\beta 3$) it is reasonable to estimate p_c by

$$p_c = 1 - \prod_{j=1}^{j=N} (1 - p_j)$$

replacing probabilities by estimates.

Since MeSH terms may appear at up to 4 points in the category structure, equation (β3) is an oversimplification. The probability, $P(T_j)$, of application of the j th term which has a category number representing a point within the search category, should ideally be replaced by the smaller probability, $P_c(T_j)$, that the term be applied with a category number within the search category. This second probability is smaller since not all the terms' category numbers need be within the search category. The formula given for p_c is thus an over-estimate, but the frequency counts necessary for a more accurate estimate are not available. Should the scoring technique appear promising when tested, there are no conceptual or practical obstacles to prevent the recording of tallies for category points. No estimation would then be necessary.

The modified form of S_3 is now

$$S_3 = \sum_{j=1}^{s} \delta_{jr} \log \frac{w_j}{p_j} + \sum_{c=1}^{s^1} \delta_{cr} \log \frac{w_c}{p_c}$$

with the probability $P(T_c/R)$ being estimated directly by the system user, as for $P(T_j/R)$. The first sum is over search terms, the second over search categories.

2.7 Modification of Scoring Technique for Equivalent Terms

The MeSH terms have been assigned to category points by the system. These are the 'official' category points. This may not satisfy a particular user. He may be interested in a 'category' of terms which is a category for him, i.e. he regards the terms as substitutes for each other, but is not a category of MeSH.

Alternatively a user may class a group of terms as substitutes for each other, but imperfect substitutes in that he has a scale of preferences. The group may or may not be a MeSH category.

One solution to this problem is to partition the list of search terms into these "user-categories" and in the calculation of S_3,

$$S_3 = \sum_{j=1}^{s} \delta_{jr} \log \frac{w_j}{p_j} + \sum_{c=1}^{s^1} \delta_{cr} \log \frac{w_c}{p_c}$$

to select only the largest information content $\log \frac{w_j}{p_j}$ provided by the terms in a particular group. This is equivalent to assuming that the information given about relevance by the occurrence of two alternative terms is not more than is given by the single term which is the preferred alternative. In algebraic terms the assumption is that $P(R/T_1 T_2) = \text{Max}\,[P(R/T_1), P(R/T_2)]$ when terms 1 and 2 are alternatives. With this modification S_3 becomes:

$$S_3 = \sum_{\substack{\text{all groups} \\ \text{of search} \\ \text{terms}}} \text{Max} \left\{ \left\{ \delta_{jr} \log \frac{w_j}{p_j} \right\} , \left\{ \delta_{cr} \log \frac{w_c}{p_c} \right\} \right\}$$

This partitioning approach has been used elsewhere [11].

Another solution is to combine Boolean and Scoring Searching. This is described in the next chapter, but it is only a partial solution to the problem of this section since it makes little provision for imperfect substitution and a scale of preferences.

2.8 A KDF9 Program for Probabilistic Searching of the MEDLARS file

The standard MEDLARS system was implemented in the U.K. by E.D. Barraclough on an E.E.L.M. K.D.F.9 computer. The quantities of data to be handled were large for a machine of this size and speed and machine code programming was necessary. Standard E.E.L.M. Magnetic Tape Handling Routines were used.

The program to perform a Scoring search was therefore written in K.D.F.9 machine code, partly because of the same problem of large-scale data handling, and partly to make it compatible with the Magnetic Tapes

containing the MEDLARS file which had been set up by the standard programs. Adoption of machine code and E.E.L.M. tape routines throughout avoided much duplication of effort. In particular the program used for printing the retrieved references in an easily readable format was the standard MEDLARS print.

Program Input

(1) The MEDLARS file of indexed references held on magnetic tapes in reference order.

(2) A separate magnetic tape containing the MEDLARS vocabulary MeSH and giving for each term its category numbers and tally (frequency).

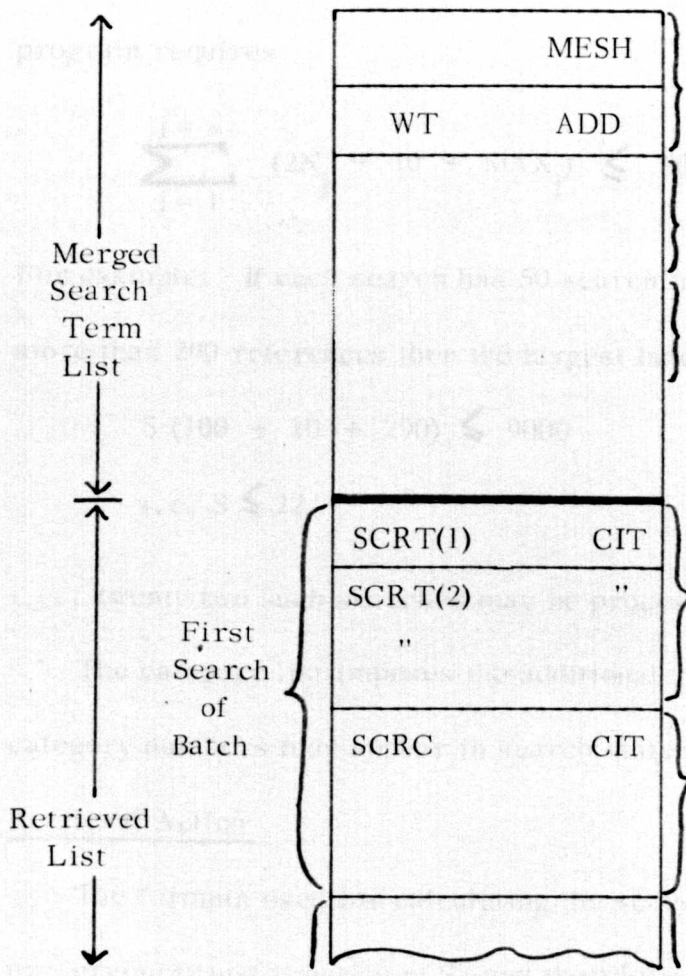
(3) A paper tape containing a batch of search formulations. Each formulation consisting of up to ten user-categories, each user-category being a list of terms and MEDLARS category numbers with associated w_i provided by the user. For each search formulation the maximum number of references to be retrieved. Restrictions on batch size etc. are given below.

Layout of the KDF9 high speed store

The program reads the paper data tape and combines all search terms (from whatever search) into a single list sorted in term order. One pass up the MeSH tape provides all the p_i and p_c required and the search of the file commences. From this point onwards the layout of the high speed store is as follows:

The KDF9 store consists of 16K 48-bit words. When system software, program and magnetic tape buffer stores are in store some 10,000 words remain. These are split between a "Category list" (1,000) and a joint "Search Term and Retrieved Reference List". The category list is merely a category version of the search term list and no further description is necessary. The other list is laid out as in the diagram:-

Search Term and Retrieved Reference List



Two words for each search term. MESH is an integer identifying the term in the vocabulary. WT = $\frac{p_i}{w_i}$ and ADD is the address of the word containing the information content of the user-category of the term.

Provision for up to 10 user-categories

Provision for MAX references to be retrieved

SCRT(a) : the $\frac{p_i}{w_i}$ value of the most preferred term from the a th user-category which is present in the indexing of the reference being considered.

SCRC : the score of a retrieved reference

CIT : the identification number of a reference (note, not all the CIT's in the diagram are the same number)

Size Restrictions

From the diagram it is clear that for a batch of S searches with N_j terms used in the j th search, and with an output limit MAX_j on the j th search, the program requires

$$\sum_{j=1}^{j=s} (2N_j + 10 + MAX_j) \leq 9000$$

For example: If each search has 50 search terms, and each user requires no more than 290 references then the largest batch size possible is S where

$$S\,(100 + 10 + 290) \leq 9000$$

$$\text{i.e. } S \leq 22$$

i.e., twenty two such searches may be processed simultaneously.

The category list imposes the additional restriction that not more than 333 category numbers may appear in search statements.

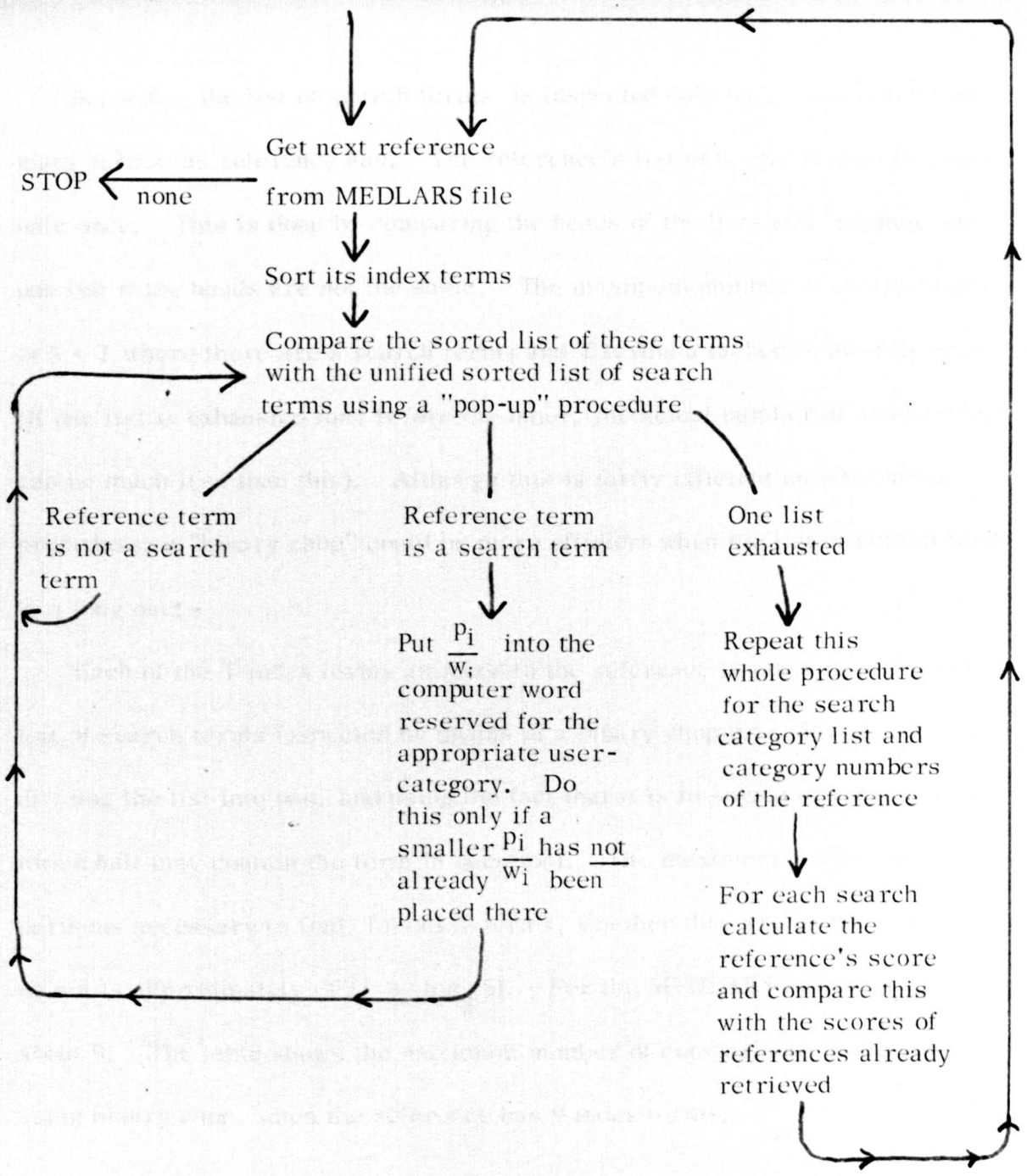
Retrieval Action

The formula used for calculating the score of a reference is an order-preserving transformation of S_3 and therefore equivalent to it. It is,

$$\text{Score} = -\prod_{\text{all user-categories}} \left(\text{Min} \left\{ \left\{ \left(\frac{p_j}{w_j} \right)^{\delta_{jr}} \right\} \cdot \left\{ \left(\frac{p_c}{w_c} \right)^{\delta_{cr}} \right\} \right\} \right)$$

This formula should be compared with that given in 2.7. When terms appear as Index Medicus Headings the p_j values are modified as described in 2.5.

The retrieval action can be described by a very simplified "flow diagram".



This retrieval action has two important features. First, the comparison of the score of the current reference with the scores of the references already retrieved: If the number already retrieved is equal to the maximum output desired by the user, the current reference can only be added to the retrieved set at the expense of rejecting one which has already been retrieved. This is done if the score of the current reference exceeds the smallest score in the set. The program is so designed that a reference whose score is less than that minimum is rejected after only one comparison. This situation is likely to be very frequent.

Secondly, the list of search terms is inspected only once, no matter how many terms the reference has. The reference's list of terms is also inspected only once. This is done by comparing the heads of the lists and "popping-up" one list if the heads are not the same. The maximum number of comparisons is S + T where there are S search terms and T terms attached to the reference. (If one list is exhausted long before the other, the actual number of comparisons can be much less than this). Although this is fairly efficient an alternative procedure, a "binary chop" could be more efficient when the list of search terms is a long one:-

Each of the T index terms attached to the reference is taken in turn and the list of search terms inspected by means of a binary chop procedure (successively dividing the list into two, and using the fact that it is in sorted order to decide which half may contain the term in question). The maximum number of comparisons necessary to find, for all T terms, whether they are on the search list or not is approximately $T(1 + \log_2 S)$. For the MEDLARS system, T averages about 9. The table shows the maximum number of comparisons using pop-up and using binary chop, when the reference has 9 index terms.

No. of Search Terms	No. of Comparisons using (i) Pop-up	(ii) Binary chop
2	11	18
128	137	72
256	265	81
512	521	90

The table shows the very considerable superiority of the binary chop procedure for large numbers of search terms. The superiority is exaggerated by the table, in that large batches of searches often result in repeated search terms and some

modifications are necessary to the simple binary chop method. Nevertheless, for batches of 500 search terms, the table suggests that the speed of the program could be improved fourfold.

Time Taken

The processor time taken is spent partly in overheads such as sorting, and partly in comparing index terms or category numbers with the references. As a simple approximation,

$$\text{Time} = V + t\,T + cC$$

where V is the overhead
t is the number of index terms
c is the number of category numbers
T is the time per term
C is the time per category number.

Four searches were run over 34,000 references with results as in the table:

Time	t	c
20 mins	265	23
20	359	3
24	430	7
18	198	19

Taking V as 3 minutes, T as 2.67 seconds and with each category number taking 4.7 times as long as an index term gives estimated times of 20, 20, 24 and 19 minutes respectively. The comparison time per reference is thus 0.8×10^{-4} seconds per index term and 3.75×10^{-4} seconds per category number. These estimates are compared with the results of a large number of searches reported in the next section.

2.9 A Comparison of Boolean and Probabilistic Searching of the MEDLARS file: Tests and Results

Two tests of retrieval performance were made. In each test a number of searches were performed by the Boolean Search strategy, by the Probabilistic method, and also by Title Searching (described in Chapter 4). Two useful measures for evaluation are Precision and Relative Recall, where

$$\text{Precision} = \frac{\text{No. of relevant references retrieved}}{\text{Total no. of references retrieved}}$$

and

$$\text{Relative Recall} = \frac{\text{No. of relevant references retrieved by a particular technique}}{\text{Total no. of relevant references retrieved using all techniques}}$$

As noted in Chapter 1, it is not strictly correct to average Relative Recall figures over several searches and then compare strategies, since the average is a function of the distribution of performance of the combined methods. But Relative Recall for individual searches can be validly used.

Test 1. All MEDLARS users participating in the MEDLARS Monthly Selection Service were offered the option of parallel searches by Boolean and other methods. In return they were to evaluate the references retrieved by classing them as "relevant" or "not-relevant". There were 50 such users at the time of the offer. Some did not take up the offer and others did not evaluate the retrieved references. As a result only 23 searches were available for use in comparing Boolean and Probabilistic strategies.

To avoid imposing an unacceptable burden of work on users, the Probabilistic Search Statements were formulated by MEDLARS staff who based them on the Boolean Search Statements. These had been formulated in consultation with users. No attempt was made to group search terms into user-categories and the modification of the Scoring Technique given in 2.7 was not used.

The output sizes were arbitrarily set at 30 or 75 depending on whether the Boolean Search Statement seemed specific or general.

Each "search" was over 34,000 references.

Results of Test 1.

(1) The Boolean Technique did overwhelmingly better. In 10 of the 23 searches the Boolean was better on both Precision and Relative Recall. The Probabilistic Technique was better on both counts in only 2 of the 23 searches.

(2) Of the relevant references retrieved the Boolean Technique retrieved 70% and the Probabilistic 46%, although these figures are complicated by the fact that not all techniques were used for all searches. This was the result of giving the users the option of any combination of the three available techniques.

Conclusions from Test 1. It was obvious from the results of this first test that further testing would be necessary. Interpretation of the results was difficult because the response was less than half the original sample (23 out of 50) and this could well be a biased sub-sample.

In spite of this the results did suggest that the Probabilistic Technique, as used in this test, was considerably inferior to the Boolean.

Test 2. The criteria for this test were that it:

(a) Should provide fully comparable results for the search techniques insofar as was possible without making the test unrealistic. To this end no option was given, all searches were performed by all techniques. In addition the Boolean Search was performed first and when its output size was known, this size was set as the Probabilistic Search's maximum output. However it was held to be unrealistic to set this maximum at less than 10.

(b) Should provide sufficient data to give statistically significant results, and preferably at a high level of significance. The response rate should be close to 100% to avoid the problem of a biased sub-sample, and the number of searches should be as high as resources permitted.

To get a high response rate a group of users was chosen at a single geographic location - Strathclyde University. This made it possible to obtain relevance assessments at personal interviews instead of relying on postal communication. All retrieved references were evaluated. Only two users were unavailable. Both had left the University. In one case the evaluation was carried out by a colleague, in the other by the user's supervisor.

The group was chosen from those members of the University who had already had a MEDLARS search run. Some senior staff, Readers and Professors, were eliminated as having too little time to cooperate on evaluation.

The Boolean Formulations of the remaining 27 searches were then considered. One was rejected as unsuitable for a Title Search. The remaining 26 formed the test base. In the course of the test a typing error was made in the specification of one Boolean Statement and to avoid results biased against that method only 25 formulations are quoted below.

Using only the MeSH terms found in the Boolean Search Formulations, the corresponding Probabilistic Search Statements were formulated. Each Formulation was then searched over 6 MEDLARS file tapes, each tape containing 35,000 references.

For clarity the terms "Search Formulation" and "Search" will be distinguished. The test used 25 Boolean Search Formulations, 150 Boolean Searches, and the same for the Probabilistic technique.

This test format was somewhat unrealistic in that e.g. it involved repeating a previously run search, but as a test method it had the advantages,

(i) All searches represented genuine requests for information,

(ii) Users' judgements on relevance were final.

(iii) The Boolean Search Formulations were devised by staff of the University Library in consultation with users, and were not merely the product of MEDLARS staff.

(iv) The same index terms and category numbers were used in the Probabilistic Search Formulation. It would be most unrealistic to assume that such a radically different pair of techniques would use the same index terms but the introduction of other terms would have made the analysis of results more difficult, and would have required more work of the users.

(v) Because of a time lag between the original search and the test, the test did not, in general, consist of searches over a set of references already searched.

The use of User-categories in Scoring Search Formulations

The original test, although inconclusive, had suggested that the Scoring Technique as used in that test, was much inferior to the Boolean. Since the standard MEDLARS system had been operating for several years it seemed possible that Boolean Formulations were more expertly constructed than the Scoring Formulations. The latter were examined for any systematic errors. Estimation of $P(T_i/R)$ i.e. w_i the estimated fraction of relevant references indexed by term i, should ideally be estimated by the users but this would have required more cooperation from them than was available. They were therefore estimated by the author and a subject specialist. No great improvement could be expected on this count.

A big improvement did seem possible by making considerable use of User-categories. One search for example was titled "Hyaluronidase and Dental Caries". The Boolean Search Statement was:-

B = HYALURONIDASE and (Sum)

where Sum represents fifteen dental terms or - ed together. These were,

1 DENTAL CALCULUS
2 DENTAL CARIES
3 DENTAL CARIES SUSCEPTIBILITY
4 DENTAL PROPHYLAXIS
5 DENTAL CEMENTUM
6 DENTAL DEPOSITS
7 DENTAL PLAQUE
8 DENTAL ENAMEL
9 DENTAL ENAMEL PROTEINS
10 DENTAL ENAMEL SOLUBILITY
11 DENTRIFICES
12 DECALCIFICATION TECHNIC
13 PERIODONTIUM
14 SALIVA
15 CALCIUM

The use of the MEDLARS category number for TOOTH (A3.54.49) would have given many but not all of these and would have included some terms not listed here. These fifteen terms quite obviously form a User-category and very poor results would be given if the Scoring Search Formulation did not recognise this, for references indexed with several dental terms would be retrieved in preference to those with one dental term plus HYALURONIDASE.

This is an extreme example but 21 of the 25 Boolean Formulations were of the form

B = (Sum 1) and (Sum 2) and ( ) .....

i.e. a set of User-Categories "and" ed together. Accordingly the User-category facility was extensively used, indeed in every Scoring Search Formulation. As the example on HYALURONIDASE showed, this could be expected to produce a marked improvement.

Computer Time Taken. Both Boolean and Scoring programs handled all 25 searches in one batch. The time taken to search each tape of 35,000 references was about 8 to 10 minutes for the Boolean and 30 to 32 for the Scoring program. The number of index terms used was 482, and 33 category numbers were specified. Using the values calculated in the last section the estimated time for the Scoring Search is

$$\text{Est. Time in minutes} = 3 + \left(\frac{0.8 \times 10^{-4} \times 482}{60} + \frac{3.75 \times 10^{-4} \times 33}{60} \right) 35,000$$

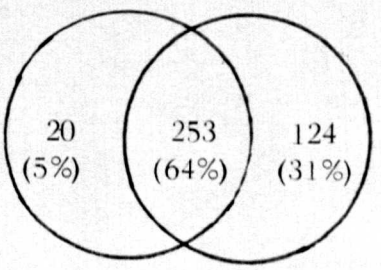
$$= 31 \text{ minutes.}$$

This estimate agrees very closely with the actual times taken.

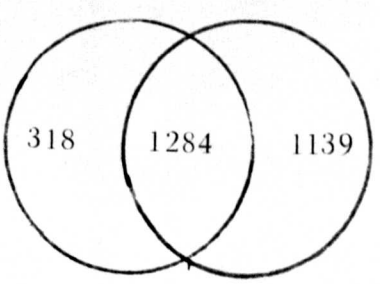
The three or four to one ratio of Scoring and Boolean Search times reflects the fact that the Scoring program examined all the index terms, while the Boolean took advantage of the "and" s to reject references quickly. The use of a binary chop method, described above, would have reduced or eliminated this difference.

Results of Test 2. The results can be clearly presented in a diagram of two overlapping circles, each corresponding to references retrieved by one technique.

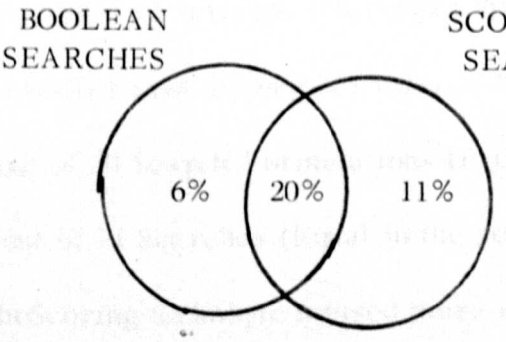
The overall results for the 150 searches were:



NO. AND % OF RELEVANT REFERENCES RETRIEVED (150 SEARCHES)
% IS RELATIVE RECALL



TOTAL NO. OF REFERENCE RETRIEVED (150 SEARCHES)



PRECISION (%)

(150 SEARCHES)

(It can be misleading to quote overall Relative Recall figures, though they are given in the first diagram).

The pattern of results indicates that the Scoring technique retrieved more relevant references than the Boolean, and did so without unduly sacrificing Precision.

When individual searches are examined the same pattern is maintained. Of the 20 relevant references missed by the Scoring technique, 13 were missed by just one search formulation, and examination of that formulation showed why. The title of the search was "The Evaluation of Analgesics and also the Effects of Analgesics on Electroencephalography in Animals and Man." The Boolean Search Formulation used was

B = (Sum of Electroencephalography terms)

and (Sum of Pain terms and drug terms)

The Scoring Search Formulation used divided the terms into 3 User-categories: (1) Electroencephalography terms (2) Pain terms (3) Drug Terms. As a result the Scoring Searches selected a number of references indexed by Pain and Drug terms but without Electroencephalography terms, and this understated the importance of Electroencephalography to the user.

This explanation does not mean that the Scoring Search did well on all formulations, but does give the cause of failure.

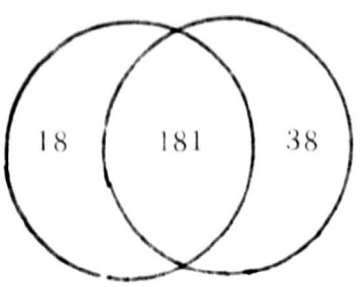
The number of relevant references missed by the Scoring technique was strictly smaller than the number missed by the Boolean in:

19 out of 20 Search Formulations (Equal in the remaining 5 cases) and 64 out of 71 Searches (Equal in the remaining 79). Of the 7 searches where theScoring technique missed more relevant references, 6 were with the formulation discussed above. The first result is statistically significant at the 0.002% level, the second at the 0.000001% level. But for the calculation of statistical significance to be meaningful, the sample of 25 search formulations from the total of MEDLARS formulations, would have to be an unbiased sample, and this it may not be.

Output Size and Retrieval Performance

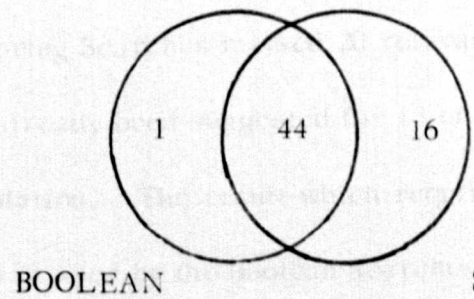
The Precision Figures presented above showed that references retrieved by the Scoring technique alone were, on average, more relevant than those retrieved by the Boolean Search alone (6% to 11%). However, because of different output sizes the references retrieved by the Scoring Technique were, on average slightly less relevant than those retrieved by the Boolean Searches (17% to 15.5%). This suggests that the superior performance of the Scoring Technique was due to its ability to widen the scope of the search without losing too much Precision, rather than to other factors such as generally better performance. To examine this possibility more closely, three diagrams are given of retrieval performance at different output levels. Each diagram shows the number of relevant references retrieved.

(1) Equal Output: Boolean and Scoring Searches retrieve the same number of references in each search (Data from 41 searches using 9 formulations)



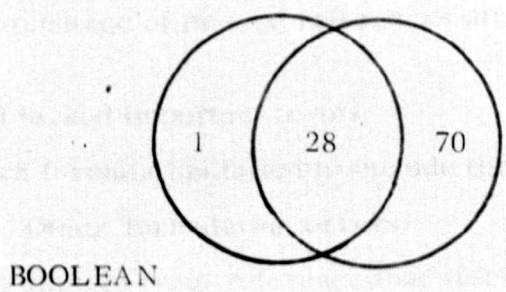
Scoring performance was strictly better in 19 out of 26 searches (15 equal). Of searches in which the Boolean was better, 6 out of 7 used the single formulation discussed above. This also accounted for 13 of the 18 references missed by the Scoring technique.

(2) Scoring output larger: Boolean output in the range 5 to 9 inclusive, Scoring output equal to its minimum of ten (Data from 25 searches, 13 formulations).



Scoring performance was strictly better in 8 out of 8 searches (17 equal).

(3) Scoring output double: Boolean output in the range 0 to 4 inclusive, Scoring output equal to its minimum of ten. (Data from 84 searches, 19 search formulations).



Scoring performance was strictly better in 37 out of 37 searches (47 equal).

These diagrams are reduced to percentage form in the table below.

	Boolean	Both	Scoring	
Equal Outputs	8	75	16	100%
Greater Output	2	72	26	100%
Double Output	1	28	70	100%

This table shows that the Scoring technique gave better retrieval performance than the Boolean even when output size was the same. This result was statistically significant at a very high level, but the magnitude of the superiority was small. When the output size of the Scoring Search was enlarged the magnitude of the superiority was greatly increased.

The Causes of Recall Failures

The Scoring Searches missed 20 relevant references found by the Boolean. A cause has already been suggested for 13 of these failures, namely a mis-judged search formulation. The result which requires explanation is the large number of references missed by the Boolean Searches.

F.W. Lancaster in [12] reported on the causes of Recall failure using MEDLARS in the U.S.A. The results were based on 88 searches and 633 relevant references. Each missed reference was inspected by MEDLARS staff. The causes suggested, and the percentage of missed references attributed to each cause were:-

(i) MeSH lacked important terms	10%
(ii) Search formulation failed to include terms	21%
Other formulation errors	11%
(iii) Term omitted from reference (but reference contained information on topic)	30%
Other indexing errors	6%
(iv) Misunderstanding of users' information needs or other user/system interaction failures	25%
Total	93%

These should be regarded as the definitive percentages rather than results given here, since the prime objective of this chapter was to consider alternative search techniques. However, the author did inspect all Boolean retrieval failures, and the results are given:-

Since tallies were not available for Subheadings, they could not be used in the Scoring Search Formulations, and to ensure comparability of test results they were deleted from Boolean Formulations. Had this not been done, the Boolean Searches would have retrieved 5% of the missed references. Had Subheadings been used the Scoring Searches might also have done better.

Boolean Retrieval Failures

(i)	Subheadings not used	5%
(ii)	Index terms too general or specific	1%
(iii)	"Parallel" term used in indexing	11%
(iv)	Search terms not in reference	83%

By a "parallel" term is meant an obvious substitute for one of the Search terms, e.g. one search formulation specified.

INFANT or INFANT, NEWBORN but some references assessed relevant by the user were indexed by CHILD.

The fourth cause, Search terms missing, comprises many of the causes listed under (ii), (iii) and (iv) of Lancaster's classification and without much additional work the individual failures cannot be authoritatively assigned to one or other, but some examples may be illuminating. It was not surprising that Psychologists who had specified their information needs by "HUMAN" or "CHILD" could not avoid an interest in papers on "RATS" and "DOLPHINS". A reference entitled "On the Characteristics of Autonomic Innervated Striated Musculature. The Inner Ocular Musculature of the Chicken" was marked relevant by the user whose request was titled "Physiology, Pharmacology and Biochemistry of Bird Muscle". The Boolean Search Statement was of the form:

B = (Sum Bird terms) and (Sum Muscle terms) and (Sum Technique terms)

Bird and muscle terms were present in the indexing of the reference, but none of the 12 technique terms quoted in the Search Statement.

Conclusions from Test 2

Some of the merits of this test were listed above. Briefly, the original information requests and the evaluations of the individual references retrieved came from genuine users not from MEDLARS system staff. Attempts were made to get comparable results by using the same index terms in Boolean and Scoring Searches, and where possible, the same output size. But the demerits of the test must also be listed. These include:-

(i) Only 25 formulations were used. These came from only a few departments of one University. These consisted of several departments associated with Pharmacology, Pharmaceutical Technology etc., and a number of others including Biochemistry, Food Science, Bioengineering and Psychology. They did not include any Clinical Medicine departments since Strathclyde had no Medical Faculty. The results given thus have more bearing on the satisfaction of the information needs of medical scientists than of practising doctors. Obviously other categories of MEDLARS users could give different results. Against this, Strathclyde was, at least for the first year, the largest single (institutional) user of MEDLARS in Britain, and the test results were remarkably consistent as shown by the high statistical significance levels achieved.

(ii) Relevance assessments were based on the standard MEDLARS print of author, title, journal, index terms and subheadings. It is possible that relevance assessments were not only subjective, which they should be, but uninformed, which they should not. But the Boolean and Scoring (and Title Search) outputs were merged, duplicates deleted,and then sorted into a bibliographic order chiefly determined by journal of publication. Users thus did not know which, or how many, search techniques had been used to retrieve individual references. This should have cut down any consistent bias to Scoring or to Boolean retrievals.

(iii) It could be argued that the test was biased towards comparability at the expense of realism; that it constrained both techniques to work at less than their optimal efficiency. One advantage of the Boolean technique is that it can automatically adjust the output size to retrieve as many or as few references as seem likely to be relevant (i.e. satisfy the Boolean Expression). Without the foreknowledge of the Boolean Output size the maximum to be retrieved by the Scoring Search might well have been set higher when the Boolean retrieved few, thus perhaps retrieving many more irrelevant references, and lower when the Boolean retrieved many, thus perhaps missing relevant references. The extent of this effect is unknown but its existence is certain.

In a similar manner, the Scoring technique was not used to maximum advantage, since only those index terms appearing in Boolean formulations were used. By requesting that another user-category of index terms be present, no reference would be rejected except in favour of some reference with a higher score. Thus many more index terms could have been used in Scoring formulations without there being any possibility of reducing the number of references retrieved. This could have led to more powerful Scoring Search Formulations using more user-categories than in the test, where no more than 5 were used in any one search.

In spite of these faults, the test shows that, for the scientific users of MEDLARS, the Scoring Technique retrieves marginally more relevant references than the Boolean when the output sizes are the same. In addition, when there are few references satisfying the Boolean formulation, the Scoring technique retrieves many more relevant references by automatically widening the scope of the search. This conclusion holds with somewhat less certainty for MEDLARS users in general, and with even less certainty for Coordinate Index Systems in general. This test does mean that the Boolean technique should not be adopted on other Coordinate systems without a thorough comparative test of it against a Scoring technique, possibly the particular example developed here.

Many of the requirements of the Probabilistic Scoring formula of this chapter were not met by the MEDLARS indexing system. Were they to be met, the performance of the Probabilistic Technique could well improve. The low Precision percentages achieved by both techniques show that there is much room for an improvement.

References Chapter 2.

1 Soergel, D, I.S.R., 3, 1967, 129.

2 Goffman, W., C.A.C.M., 4, 1961, 557.

3 Goffman, W., C.A.C.M., 4, 1961, 594.

4 Goffman, W., C.A.C.M., 7, 1964, 439.

5 "The MEDLARS Story at the National Library of Medicine," Nat. Lib. Med., Washington, 1963.

6 MacMillan, J.T., Welt, I.D., Amer. Doc., 12, 1961, 27.

7 "Medical Subject Headings: MeSH", Nat. Lib. Med., Washington, annually.

8 Maron, M.E., Kuhns, J.L., J.A.C.M., 7, 1960, 216.

9 Fano, R.M., "Transmission of Information", M.I.T., 1961.

10. Needham, R.M., "Applications of the theory of Clumps", C.L.R.U., Cambridge, 1964.

11 "Search Manual", Chemical Society Research Unit in Information Dissemination and Retrieval, Nottingham, 1967.

12 Lancaster, F.W., Amer. Doc., 20, 1969.

Chapter 3. Further Applications of Scoring Techniques

Chapter 3. Further Applications of Scoring Techniques

3.1 Automatic Reformulation of Search Statements

When a search fails to retrieve sufficient relevant references, or retrieves too many irrelevant, it is often uneconomic to repeat the search with a modified search statement. But if the retrieval service is a Selective Dissemination of Information, or S.D.I., Service this is not so. In an S.D.I. Service each search is repeated at regular intervals (e.g. every week or month), but, on each occasion, only the most recent acquisitions are inspected, not the whole file. In this environment the modification of search formulations is useful, not only to improve the original formulations, but also to recognise the users' changing interests.

If the first search fails to retrieve any relevant references the formulation can only be modified by human intervention. Otherwise the evaluation of individual references as relevant/not relevant provides data for automatic procedures.

Reformulation of Boolean Searches: Suppose that the first Boolean Search, B, retrieved two relevant (and some non-relevant) references, and that the first reference was indexed by terms X_1 to X_N, the second by terms Y_1 to Y_M. Then the Boolean Statement B^1,

$$B^1 = (x_1 \text{ and } x_2 \ldots\ldots \text{ and } x_N) \text{ or } (y_1 \text{ and } y_2 \ldots\ldots \text{ and } y_M)$$

would retrieve all the relevant references in the file which were retrieved by the original formulation B. In addition B^1 is a tighter, or more specific, formulation than B,

i.e. $B^1 \subset B$

so that B^1 would retrieve no more irrelevant references than B, and possibly fewer.

Unfortunately it is possible that if B and B^1 were used to search another set of references, e.g. the next month's acquisitions in an S.D.I. system, then B^1 might retrieve fewer relevant references than B. However, by using methods

presented later in this chapter, B^1 could be automatically widened again. This widening of B^1 would be in a different direction than towards the original formulation B, and the net result would be a different rather than a tighter Boolean search.

The generalisation to the case of more than two relevant references retrieved on the first round is obvious.

Reformulation of Probabilistic Searches: For a Pure Coordinate Index as defined in 2.1, the Search Statement of a Probabilistic Search consists of a list of index terms, each with an associated w_i which is the user's estimate of $P(T_i/R)$, the probability that a relevant reference is indexed with term i. To reformulate such a statement it is only necessary to add terms to the list, delete unwanted ones, and re-estimate each w_i. This re-estimation can be done by an automatic calculation utilising the user's relevance judgements. The relative frequency of term i in the relevant references retrieved is w_i^1, where

$$w_i^1 = \frac{r_i}{R}$$

and r_i of the R relevant references are indexed by term i. The term i is not a suitable search term unless $P(T_i/R) > P(T_i)$ and so term i can be taken as a search term with

$$w_i^1 = \frac{r_i}{R} \quad \text{provided} \quad w_i^1 > p_i$$

where p_i (as before) is the relative frequency of use of the term i by the indexing system. For w_i^1 to be a good estimate of $P(T_i/R)$ the relevant references retrieved should be an unbiased sample of all possible relevant references, where unbiased means unbiased with respect to the frequency of occurrence of term i.

A variant of this procedure is to take

$$w_i^1 = \lambda\ w_i + \mu \frac{r_i}{R} \qquad \text{where } \lambda + \mu = 1, \quad \lambda > 0, \quad \mu > 0$$

and λ, μ are weighting parameters.

A program was written to calculate the $w_i^!$ but the two retrieval tests of Chapter 2 indicated that MEDLARS was sufficiently different from a Pure Coordinate Index to make the use of User-categories crucially important. In such a situation the Probabilistic Search Statement does not consist of a single list of terms and the method of Boolean Searching by the Probabilistic Technique (3.6) is more appropriate, as a means of reformulating the search.

3.2 Super-Boolean Searching

A search, in which the minimum requirement for retrieval is the satisfaction of a Boolean Search Statement, will be called a Super-Boolean Search.

As an attempt to control the size of the output of a Boolean Search, several Boolean Search Statements B_1, B_2 .... may be used, each strictly tighter than its predecessor so that

$$B_1 \supset B_2 \supset B_3 \ldots\ldots$$

Before references are printed and sent to the user, the sizes of these classes B_1, B_2, ... can be inspected and a decision taken on which to print (an expensive part of the whole retrieval system). MEDLARS for example permits up to 3 such statements per search. On search no. 1 the sizes were,

B_3 1 reference retrieved, 1 relevant

B_2 25 references retrieved, 15 relevant

B_1 485 references retrieved, 93 relevant.

The minimum requirement for retrieval was satisfaction of B_1, searches B_2 and B_3 representing Super-Boolean Searches which were themselves Boolean. There is no compelling reason why the Super-Boolean Searches should not be Scoring Searches. Had they been so, many more different levels of output would have been possible than 1, 25 or 485. There might be an economic advantage in doing the first search using the Boolean program, (it is quick) with a wide search formulation and then ranking the output by a Scoring program. But this would only be worthwhile if

the Scoring Search did give good rankings i.e. rankings with a marked relevance gradient. Over the full range of references in the file the Probabilistic Technique did give good rankings as was shown by its retrieval performance (Chapter 2). But it does not follow that it necessarily produced marked relevance gradients over small sections of the total ranking, and this property must now be investigated.

3.3 Ranking Performance of the Probabilistic Search

The ranking performance of the Probabilistic technique over the range of references retrieved by a wide Boolean Search could be examined directly by rankings the outputs of a number of Boolean Searches. Alternatively, data from the test reported in Chapter 2 can be used.

In 41 of the 150 searches in that test, the Boolean and Scoring searches retrieved the same number of references and 80% of Boolean retrievals were retrieved by the Scoring Search. The ranking in these Scoring searches is an approximation to the ranking performance of the Scoring technique over the range of references retrieved by Boolean searches. It is also of interest for itself.

The formula used to calculate reference scores was an order-preserving transform of S, where

$$P\ (\text{cond}) = P.S$$

P (cond) was the estimated probability of a reference being relevant given that it was indexed by certain terms. P was the unconditional probability of a reference in the file being relevant to the particular user. For any one search P was taken as constant and S became an order-preserving transform of P(cond). When several searches are under consideration P cannot be assumed constant between searches. It follows from this that in evaluating ranking performance, the performance for each search must be evaluated separately before there can be any combination of results from several searches. In particular, there is no way of coalescing the individual rankings to smooth the data.

For each search the average rank of a relevant reference was divided by the highest possible rank number. Ranks started at rank zero. For random rankings (of the references retrieved) the Expected Value of this statistic K, say, is obviously one half (by symmetry). A count of the number of searches in which it is less than one half gives the significance level of the hypothesis that the rankings are better than random. The results were

$K < 0.5$ in 26 searches

$K > 0.5$ in 12 searches

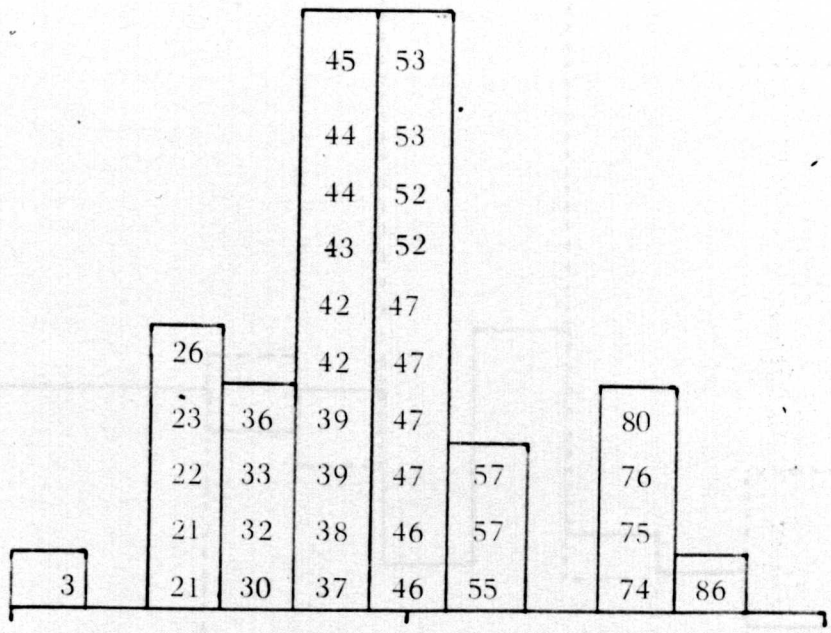
K not defined (no relevant references) in 3 searches

(Total 41 searches)

The success rate of 26 out of 38 makes the hypothesis statistically significant at the 2.0% level. One search formulation, used for 6 of these 41 searches appeared, on inspection in Chapter 2, to have been badly formulated. Of the 6 searches, 3 retrieved no relevant references and for the other 3 the K values were 0.86, 0.57 0.76. Without these searches the hypothesis would have achieved the 0.3% significance level.

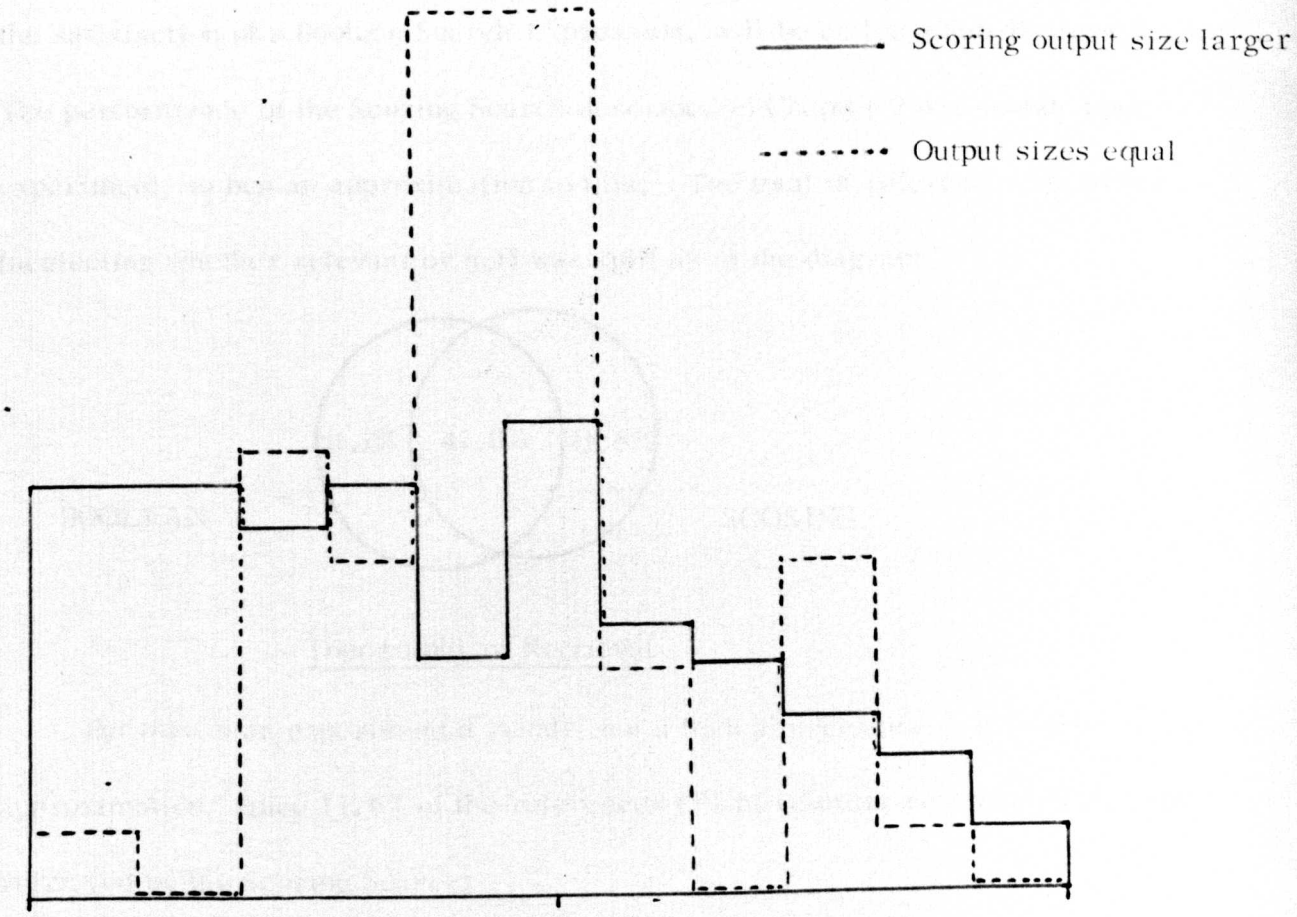
The mean K was 0.45 which is close to 0.5. The distribution of K is given in the figure which shows the actual values of K achieved in individual searches. They are arranged in columns corresponding to bands of 0.09 and the heights of the columns form a histogram:-

Table of K-values for individual searches. K expressed as a percentage, and displayed in eleven bands of 9% ranging from 0.5% to 99.5%.



This table suggests that although the ranking was better than random, it was not much better. This result, although disappointing, is consistent with the results of Chapter 2, where it was shown that there were many relevant references not retrieved by the Boolean Searches which were retrieved by Scoring searches with larger output size. Scoring Searches with output size equal to Boolean retrieved only marginally more relevant references. This indicated that as the output size of the Scoring Search was increased the frequency of occurrence of relevant references declined fairly slowly, and this same property implies K values close to 0.5 for output sizes of the order of a Boolean output.

Of the 150 searches 109 had Scoring Search outputs larger than the Boolean outputs. Of the 109, 39 retrieved no relevant references. The K values for the remaining 70 were calculated. The mean was 0.40, noticeably better than for the previous set. The success rate was only 41 out of 65 (5 equal) however. The histogram of K values for these searches, and for the previous set are shown in the diagram, the histograms being normalised to have unit area under each.



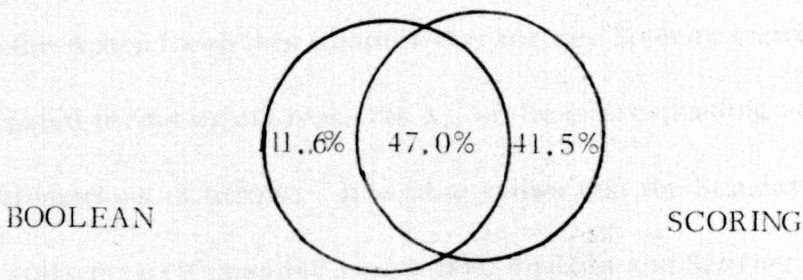
Distribution of K values

As can be seen from the diagram, the distribution of K values becomes much more skewed towards the K=O line, when the output size is increased beyond Boolean output size.

From these results it appears that the ranking of Boolean outputs by the Scoring formula would only be useful if the Boolean Statements used were wider than those normally used for Boolean Searching.

3.4 Sub-Boolean Searching

A search in which a sufficient, but not necessary, condition for retrieval is the satisfaction of a Boolean Search Expression, will be called a Sub-Boolean Search. The performance of the Scoring Search described in Chapter 2 was found, upon experiment, to ben an approximation to this. The total of references retrieved (neglecting whether relevant or not) was split as in the diagram.



Topography of Retrieval

But this is an experimental result, not a logical necessity. It is also only an approximation, since 11.6% of the references (5% of relevant references) were not retrieved by the Scoring Search.

A search procedure which retrieved every reference satisfying a Boolean Expression, and then used the methods of the Scoring Search to retrieve other references would enable the professional search formulator, or the user, to avoid choosing one technique rather than another, and would require only one search formulation. Such a search technique is developed below.

3.5 Standard Structure for Boolean Search Statements

Any Boolean expression B can be written in Conjunctive Normal Form. Conjunctive Normal Form consists of several groups of terms, the groups "and" ed together, the terms within a group "or"ed.

e.g. $B = (x_{11}$ or x_{21} or ...) and $(x_{12}$ or x_{22} or...) and .....

For the purposes of this Chapter, this form will be called the Standard form, or Standard structure, of a Boolean expression. Computer programs exist to put any Boolean expression into its Standard form e.g. [1].

It is convenient to use the symbols "+" for "or" and "." for "and", i.e. to use arithmetic symbols. The standard form of a Boolean expression is then

$$B = \prod_j \left(\sum_i x_{ij} \right)$$

In general the x_{ij} are either affirmations or negations of the terms of the original Boolean expression of which B is the Standardised version. It is shown in the Appendix to this Chapter that for any Boolean expression which has no negated terms or clauses, the x_{ij} in the corresponding standard form are all affirmations of terms. It is also shown that the Standard form is unique. These results make it possible to combine Boolean and Scoring Searching into a single retrieval technique.

Since the x_{ij} are index terms (more strictly, affirmations of the presence of index terms), each $\sum_i x_{ij}$ is an automatically generated User-category, i.e. a set of terms which, for the purpose of the particular search, are alternatives. The transformation of a Boolean Search Statement to Standard Form provides a Scoring Search formulation involving user-categories. It is obvious that any reference satisfying the original Boolean expression must have at least one index term from each user-category. Conversely, any reference failing to satisfy the original expression must lack the terms of at least one user-category. It follows that if each user-category is assigned a weight W_u and the score of a reference is calculated by S_4,

$$S_4 = \sum_u \delta_{ru} W_u$$

(where $\delta_{ru} = 1$ if the reference has a term from user-category u and $\delta_{ru} = 0$ otherwise).

then all references satisfying the original Boolean expression have a score S_{max},

$$S_{max} = \sum_u W_u$$

and all other references have strictly lower scores. This is true no matter what

values are assigned to the W_u provided that they are strictly positive

i.e. $W_u > 0$

The formula S_4 thus provides a method for Sub-Boolean Searching.

3.6 Boolean Searching by a modified Probabilistic Technique

To obtain a unique weight W_u for a user-category u it is necessary to calculate an "information content" for each category, which does not depend upon which term from the category appears in a reference. This may be done by treating the user-category as MEDLARS categories were treated in Chapter 2 (not like user-categories in Chapter 2). The estimated probability of occurrence of u is then p_u,

$$p_u = 1 - \prod_i (1 - p_i)$$

where the p_i are tallies of the terms in u. This p_u is an estimate of P(U). Since a term from each category must be present to satisfy the Boolean expression, the request for Boolean retrieval implies that the estimate of P(U/R), the probability of the category's occurrence in relevant references, is unity.

Thus, $\log \frac{P(U/R)}{P(U)}$ may be estimated by $\log \frac{1}{p_u}$ and, when this is substituted for W_u in formula S_4, S_4 has the form of the Scoring formula of Chapter 2.

i.e. $S_4 = \sum_u \delta_{ru} \log \frac{1}{p_u}$

The scores calculated by S_4 cannot be interpreted as an order-preserved transformation of the probabilities of the relevance of references, since the user-categories u are not orthogonal. One index term may appear in several of the u. This will always happen unless the original Boolean expression is already in Standard form. But the formula S_4 does have a number of useful properties including:-

(1) All references satisfying the original Boolean expression receive the same score. This score is strictly greater than the score of any other reference.

(2) S_4 gives the correct ranking with respect to the addition of a further search-term to a reference.

Proof: Another index term cannot reduce the number of user-categories satisfied by the reference, and may increase the number thereby increasing the score.

(3) S_4 gives the correct ranking with respect to the frequency of individual terms in the file.

Proof: If the presence of a term i has any effect at all, it puts the reference into one or more additional user-categories. The corresponding $\log \frac{1}{p_u}$ values are added to the score. Each of these p_u is of the form

$$p_u = 1 - K(1 - p_i) \qquad (0 < K \leq 1)$$

since i is in U.

The K is the $\prod_{k \in U} (1-p_k)$ i.e. the product of the $(1 - p_k)$ for the other index terms in the user category U.

If a term j had been used in the original search expression in place of the term i, it would have appeared in exactly the same user-categories U, but the p_u would have been different

$$p_u = 1 - K(1 - p_j)$$

Since no other terms are changed the K here is the same as before.

Now if $p_i > p_j$ then

$$1 - K + Kp_i > 1 - K + Kp_j \quad \text{since } K > 0$$

whence $$\frac{1}{1 - K(1-p_i)} < \frac{1}{1 - K(1-p_j)}$$

and $$\log \frac{1}{1 - K(1-p_i)} < \log \frac{1}{1 - K(1-p_j)}$$

i.e. the higher the value of p_i the lower the information content of the categories in which it appears. This proves the required result.

(4) When a term i appears in n of the N user-categories then up to (N - n) other search terms are necessary for retrieval. The ratio $\frac{n}{N}$ is thus one measure of the importance of a term in a Boolean expression. With this measure of importance:-

S_4 gives a ranking on which the effect of a term's presence is positively related to the importance of the term in the original Boolean expression.

Examples of the Importance Ratio

(a) $B = z + x \cdot y$ has standard form, B^1,

$B^1 = (z + x)(z + y)$

and the importance ratio for z is 1.00 signifying that it is, by itself, sufficient for retrieval.

(b) $B = t(xy + yz + zx)$ has Standard form, B^1,

$B^1 = t(x + y)(y + z)(z + x)$

and t requires only 2 other terms for retrieval although the importance ratio for t is only 0.25.

Proof of property:

The effect of the presence of a term of importance $\frac{n}{N}$ is to add n of the N possible $\log \frac{1}{p_u}$ values into the score. The larger $\frac{n}{N}$ is the greater the increment to the score. Again there is the implied condition that "other things are equal". This time the other things are the relative magnitudes of the $\log \frac{1}{p_u}$ and the satisfaction of some of the n categories by other terms present. As before, this condition does not affect the conclusion.

The formula S_4 thus has several useful properties in addition to its performance as a Sub-Boolean search technique. The only modification necessary to convert the program of Chapter 2 to perform Sub-Boolean Searches is to calculate and store the appropriate p_u in the place of the p_i. Once this initial stage of the program has been passed the algorithm is as before. The transformation of Boolean expressions

to Standard form would have to be done manually, or by use of another program e.g. that of Hieber [1].

References Chapter 3

1 Hieber, L.J., "The Minimisation of Boolean Functions". M.Sc. Thesis, University of Newcastle upon Tyne, 1965.

Appendix to Chapter 3

Proof that the Standard Form of a Boolean Search Statement is unique and contains no negated terms

The Standard form of a Boolean expression $B = B(x_1, \ldots x_n)$, where B is a function of index terms 1 to n, can be obtained by a two stage process, as follows.

Stage 1. Obtain the "implicants" of B, by writing

$$B = (B(0,0,\ldots.0) + x_1 + \ldots.+x_n)\,(B(0,0,\ldots.0,1) + x_1 + x_2 + \ldots. + \overline{x}_n)\,(\ldots..$$

where

(i) $\overline{x}_i$ means "not x_i"

and (ii) e.g. $B(0,0,\ldots.0,1)$ is the truth value of B for x_1 to x_{n-1} false, and x_n true,

and (iii) there is a pair of brackets on the right hand side for every combination of truth values of x_1 to x_n i.e. 2^n pairs of brackets in all. These are called the "implicants" of B.

Stage 2. Reduce the implicants to "prime implicants".

By using the identity $(a + b)(a + \overline{b}) \equiv a$ repeatedly, the implicants can be reduced to "prime implcants".

A "literal", y, is an affirmation or negation of one of the terms x_1 to x_n. A "prime implicant" is a logical sum of literals P, with the properties that

(i) $B \subset P$

(ii) if any literal is removed from P then B is no longer contained within the sum of the remaining literals.

In general the set of Prime Implicants is not an irredundant set, i.e. B is equivalent to the logical product of a subset of the Prime Implicants. This means that the Standard form of a Boolean expression is not, in general, unique.

A Boolean expression used for information retrieval must contain no negated terms (2.2). With this restriction it can be proved that the literals of prime implicants are all affirmations, and that the Standard form is unique.

Proof (i): The literals of the prime implicants of a Boolean Search statement, B, are all affirmations

Suppose that P is a prime implicant of B, and that $P = \bar{x}_1 + \Sigma$, where Σ is a sum of literals other than x_1 or $\bar{x}_1$. Then by part (i) of the definition of a prime implicant,

$$B \subset \bar{x}_1 + \Sigma$$

and by part (ii) of the definition,

$$B \not\subset \Sigma$$

Thus there must be at least one point (or reference) z in B but not in

i.e. $z \in \bar{x}_1 + \Sigma$

and $z \notin \Sigma$

$\therefore$ $z \in (\bar{x}_1 + \Sigma)\bar{\Sigma} = \bar{x}_1 . \bar{\Sigma}$

Now $\bar{\Sigma} = x_2^* . x_3^* . \cdots x_n^*$ where the * denotes the affirmation, negation or omission of a term, and not all of the terms x_2 to x_n are omitted.

Thus $z \in \bar{x}_1 . x_2^* . x_3^* . \cdots . x_n^*$

There is thus a combination of the presence or absence of terms x_2 to x_n which is sufficient for retrieval, if and only if x_1 is absent. This is a contradiction since B is a Search Statement and therefore does not involve the operator not. The required result has thus been proved.

Proof (ii): The Standard form of a Boolean Search Statement is unique

Lemma. No prime implicant of a Search Statement contains another unless they are duplicates.

Suppose P_1, P_2 are prime implicants of B, and that $P_1 \subset P_2 = \sum_{i=1}^{i=K} x_1$

(without loss of generality),

and that $P_1 \neq \sum_{i=1}^{i=K} x_1$

Then, since P_1 is a sum of affirmed terms, and is included in P_2,

$$P_1 = \sum_{i=1}^{i=L} x_i \quad \text{with } L < K$$

(again there is no loss of generality in taking the first L terms).

But P_1 is a prime implicant, and so

$$B \subset \sum_{i=1}^{i=L} x_i \text{ by part (i) of defn}$$

But P_2 is a prime implicant, and so

$$B \not\subset \sum_{i=1}^{i=L} x_i \text{ (for } L < K\text{) by part (ii) of defn.}$$

This is a contradiction and the lemma is proved.

Main Proof: Suppose P_1 to P_m are prime implicants, and that P is another.

Suppose P is redundant since it includes the product of the others,

$$\text{i.e. } P_1 P_2 \ldots\ldots P_m \subset P = \sum_{i=1}^{i=K} x_i$$

Since $P_1 \neq P$ and $P_1 \not\subset P$ and $P_1 \not\supset P$ (by lemma) then P_1 is a sum of terms (by proof (i)), one, at least, being different from x_1 to x_k. Let it be z_1.

$$\text{Then } z_1 \not\subset P = \sum_{i=1}^{i=K} x_i \quad \text{but } z_1 \subset P_1.$$

Similarly z_2 to z_m can be found in P_2 to P_m but not P.

Then $z_1 z_2 \ldots z_m \subset P_1 P_2 \ldots P_m$ and $z_1 z_2 \ldots z_m$ is not the empty set since each z is an affirmation.

But $z_1 z_2 \ldots z_m \not\subset P$

This is a contradiction, and proves the required result.

Chapter 4: Retrieval without Indexing - 1. A Title Searching Program

4.1 Alternatives to Indexing

4.2 Some Comparisons of Indexing with Titles

4.3 A Scoring Search based on Titles

4.4 Categories in Title Searches

4.5 A KDF9 Program for Title Searching of the MEDLARS file

Input - Layout of store - Size restrictions - Retrieval action - Inspection of Titles for Search Fragments - Time taken

4.6 A Comparison of Boolean and Scoring Searches using Index Terms with Scoring Searches using Titles

Test 1 - Description and Results

Test 2 - User categories - Computer Time taken - Results of Test 2 - Titles and Index Terms - Non-Boolean Terms used as Search Fragments - MEDLARS Subheadings - Conclusions from Test 2.

Chapter 4: Retrieval without Indexing - 1. A Title Searching Program

4.1 Alternatives to Indexing

Authors themselves provide several potential means of retrieving their publications. These sometimes include author - assigned index terms but are more often restricted to;—

(i) The full text

(ii) The abstract

(iii) The title

(iv) A list of other publications related in various ways to the content of the text.

The transformation of text into index terms is performed by the retrieval system staff. This compression from full text to a few index terms reduces the cost of the system's storage and retrieval operations, but almost inevitably reduces the amount of information about the content of the reference. Similarly, the abstract or the title contain less information than the full text. Authors, particularly of Medical references, are notorious for producing titles such as "Can this be yours?" which contain very little information at all. Nonetheless the author is better placed than almost everyone else to produce a truly informative title.

In the MEDLARS system the titles are stored with the index-terms which makes it possible to compare the reductions of the full text produced by indexers, with those produced by authors.

An index based system requires a senior staff of subject experts to do the indexing, and a junior staff to convert the index terms into machine readable form, e.g. by keypunching. A titles based system only requires the junior staff. If, as in MEDLARS, the titles are deemed necessary for display purposes, the index-based system requires considerably more junior staff, as well as senior staff. It follows that an index-based system has to demonstrate considerably better

retrieval performance to justify its greater costs. (Translation of titles into a standard language, e.g. English, could affect the relative costs).

4.2 Some Comparisons of Indexing with Titles

Studies of a number of indexes in a wide range of subject areas have shown that a large percentage of the index terms (or synonyms for them) were to be found in the titles. Some examples may be quoted from published papers:-

(i) A legal index: Examination of 3228 titles showed that 64.4% of titles contained all the corresponding index terms, and only 10.5% contained none. [1]

(ii) Physical Review: 25 titles in Physical Review were indexed by the editors of Physics Abstracts and Chemical Abstracts. 63% of titles contained all index terms [2]

(iii) Chemical Abstracts Subject Index: 84 titles were examined, 57% contained all index terms. [3]

(iv) Three Medical Indexes: 50 titles were chosen from each of 3 Medical Indexes. The percentages of index terms appearing in titles were 32%, 54%, and 26% [4]. Although these figures are not directly comparable with those in (i), (ii) and (iii), they suggest that Medical titles may have a relatively low information content.

An estimate of the relative usefulness of MEDLARS titles and MEDLARS indexing for retrieval is given by two further studies:-

(v) MEDLARS: Two samples, of 4770 titles and of 451 titles, were compared with Index Medicus. In each sample 86% of index terms appeared in titles. [5]

The Index Medicus terms are those MEDLARS index-terms under which a reference is printed in Index Medicus. On average each reference is printed under only a quarter of its index terms, the other terms being of less importance for that particular reference. Thus 86% of these 'print headings' occurred in titles, not 86% of all the MEDLARS terms indexing the references.

(vi) MEDLARS: F.W. Lancaster in [6] attempted to estimate the Recall and Precision performance of MEDLARS as operated at Washington D.C. The method used to estimate Recall was to set up a 'Recall base' of relevant references before a search, and to find what percentage of this base was retrieved by the MEDLARS search. This was called the 'Consistency Ratio' in Chapter 1, to avoid the implication that it estimated Recall. It is, however, a reasonable measure of one aspect of retrieval performance (albeit not the most important aspect). The Consistency and Precision were 60% and 52% respectively for MEDLARS Boolean Searching using all MEDLARS index terms. When searches were performed using 'print headings' only, the percentages changed to 44% and 60% respectively, i.e. the percentage of the recall base retrieved fell from 60 to 44, but the percentage of the output that was relevant rose from 52 to 60.

Since (v) showed a high degree of similarity between MEDLARS titles and 'print headings', the results of (vi) are an approximate indication of the retrieval performance using titles instead of index terms.

The performance figures for searching using only the 'print headings' are surprisingly good when compared with those for the full indexing, and this requires some explanation. Experimental studies of indexing have indicated that as the depth of indexing is increased there is less and less agreement on the appropriateness of the additional terms [7]. The relationship between indexer and search-formulator is akin to that between two indexers. The consistency between searcher and indexer, separated by time and space, is likely to be less than between two indexers taking part in an indexing experiment.

One study, using 8 indexers to index 20 references found that the main source of the small proportion of words on which the indexers were agreed was the titles and sub-titles. [8]

All these studies have looked for the appearance in the title of a subset of the index terms. In comparing the information contents of titles and indexing, it is also necessary to look at the percentage of important title words which appear in the indexing, e.g. the MEDLARS search no 10040 on 'Streptococcal Cell Walls' was inhibited by lack of the term 'Cell Wall' in the vocabulary, although the term appeared in titles. (The missing term was later added to the vocabulary). It is thus possible for titles to have a greater information content than the print headings, while only containing 86% of these headings.

In systems using a restricted vocabulary, the lack of a term in the vocabulary can force both indexer and searcher to use index terms which do not describe the content of the reference or of the user's interests, e.g. in the above example, 'CELL MEMBRANE' might have been substituted for 'CELL WALL' although these are not the same. Thus of the 14% of print headings which did not appear in titles, an unknown percentage could simply be incorrect descriptors. If indexer and searcher each chose identical substitutes for the true descriptor, only a lack of Precision would result, but if the substitutes were different Recall and Precision would both be reduced.

4.3 A Scoring Search based on Titles

The Washington MEDLARS results indicated that a Boolean title search of MEDLARS would retrieve less of a given recall base, but would gain in Precision. By using a Scoring technique to search the titles the output size of the search can be fixed in advance. By fixing output size, any increase in Precision automatically gives an increase in Recall and vice versa. This can be seen by expressing Recall as a function of Precision:-

Recall = $\frac{r}{R}$ where r of the R relevant references are retrieved

Precision = $\frac{r}{O}$ where O is the output size.

Then Recall = Precision x $\frac{O}{R}$

But R is fixed for any search (although not known), and in a Scoring search O is set arbitrarily before the search commences. It follows that greater depth of indexing which is likely to give a Boolean search higher Recall and lower Precision, does not have such easily predictable effects on a Scoring search with fixed output size. Although MEDLARS titles contain far fewer terms than there are index terms, a Scoring search can make use of the greater applicability of those that do appear.

The simplest form a Title Scoring search can take is a list of search terms, the score of a reference being

$$S_5 = \sum_{j=1}^{S} \delta_{jr}$$

where $\delta_{jr} = 1$ if the jth search term does appear in the title of the reference, and $\delta_{jr} = O$ if it does not.

Provision may be made for search terms to be weighted by modifying S_5 to

$$S_5 = \sum_{j=1}^{S} \delta_{jr} W_j$$

where W_j is the weight of the jth search term. If the frequencies of occurrence of the jth term in titles generally, and in relevant titles, can be estimated then, by taking

$$W_j = \log \frac{w_j}{p_j}$$

with w_j = estimate of relative frequency in relevant titles

p_j = estimate of relative frequency in all titles

the Title Scoring Search has the same form as the Probabilistic Search of Chapter 2.

Although the user can be asked to estimate w_j, no equivalent of index term tallies is available for MEDLARS titles and the estimation of p_j would be difficult. More importantly, the terms in titles are not terms from a Coordinate Index system, and the Probabilistic formula does not apply. The values W_j are thus left entirely at the user's discretion.

Synonyms: The selection of the search terms is of great importance, as in an index term search, but the problems of selection are different. With a vocabulary of index terms one problem was to select from the vocabulary substitutes for some desired term which was not available. For a title search there is the complementary problem of listing not only the desired term, but all reasonable synonyms, e.g. in MEDLARS. English and American titles are left unaltered, and all other languages are translated into American. This means that the list of search terms need not contain French terms but must contain both English and American spellings ('SULPHUR', 'SULFUR'). Standard abbreviations such as 'CNS' for "CENTRAL NERVOUS SYSTEM' or 'EEG' for 'ELECTROENCEPHALOGRAPHY' are also 'synonyms' for Title Search purposes. Different areas of study habitually use different terminology, e.g. 'VITAMIN C' and 'ASCORBIC ACID'. In particular, chemicals are often given one name by chemists and several others by doctors and manufacturers. Standard lists of cross-references, 'see also lists', are obviously helpful, if they are available.

Another category of synonyms can be produced by punctuation e.g. 'SE 52' could conceivably be recorded in various titles as 'SE-52', SE(52)', 'SE/52' or 'S.E.52'. To avoid having to specify all of these variants individually, a single symbol standing for any punctuation character is necessary. In the input to the program described below, a space represents any punctuation character.

4.4 Categories in Title Searches

Language Categories of Title Terms. In an indexing system with a controlled vocabulary it is possible to classify the terms to make the search easier to formulate, and faster to perform. e.g. the MEDLARS category number D6.48.12 represents 'BARBITURATES' and nine specific barbiturates:- 'AMOBARBITAL', 'HEXOBARBITAL' etc. If the Title Search technique is such that titles can be inspected for fragments of words, as well as complete words, the Title Search can make use of the categorisation provided by ordinary language. In the above example, a Title Search for 'BARBIT' would retrieve titles containing 8 of the 10 barbiturate terms. Only 'THIAMYLAL' and 'THIOPENTAL' would be missed. Of course the search for 'BARBIT' might also retrieve references on other topics than barbiturates. The linguistic classification could be accidental and have no logical foundation, e.g. a search for 'Prosthetic Appliances for the Hip and Leg' retrieved a number of references on the legal aspects of medicine due to the listing of 'LEG' as a search fragment. In spite of this counter example, the ability to search for fragments of words is most important. Retrieval can be restricted to complete words by specifying a 'fragment' whose first and last characters are spaces e.g. ' LEG '

User-categories. The linguistic categories may not satisfy a particular user. In particular the various synonyms for a term form a category for the user, but need not have a common sequence of letters in them. Synonyms such as 'SULPHUR' and 'SULFUR' are obviously perfect substitutes for each other, but a user may choose to regard two terms as imperfect substitutes in that he has a preference. A group of imperfect substitutes may or may not be a linguistic category e.g. the user may request references on any barbiturate but with a preference for papers on 'PHENOBARBITAL" One solution is to list 'BARBIT' and 'PHENO' separately. Another is to make use of user-categories in the calculation of S_5, by selecting from each user-group a maximum of one weight, the weight selected corresponding

to the most preferred term from that group in the title, i.e.

$$S_5 = \sum_{\substack{\text{all groups} \\ \text{of search} \\ \text{terms}}} \underset{\substack{\text{within} \\ \text{group}}}{\text{Max.}} \left\{ \delta_{jr} \; W_j \right\}$$

In the example on barbiturates a user-category of two terms could be used -

'BARBIT' with weight 1

'PHENOBARBIT' with weight 2

A reference on 'PHENOBARBITAL' would receive, from this user-category, a weight of 2.

4.5 A KDF9 Program for Title Searching of the MEDLARS file

The standard MEDLARS system was implemented in the U.K. by E.D. Barraclough on an E.E.L.M. KDF9 Computer. The quantities of data to be handled were large for a machine of this size and speed, and machine code programming was necessary. Standard E.E.L.M. Magnetic Tape Handling routines were used.

The program to perform a Title Scoring search was therefore written in KDF9 machine code, partly because of the same problem of large-scale data handling, and partly to make it compatible with the Magnetic Tapes containing the MEDLARS file which had been set up by the standard programs. Adoption of machine code and E.E.L.M. tape routines throughout avoided much duplication of effort. In particular the program used for printing the retrieved references in an easily readable format was the standard MEDLARS print.

Program Input

(1) The MEDLARS file of references held on magnetic tape in reference order.

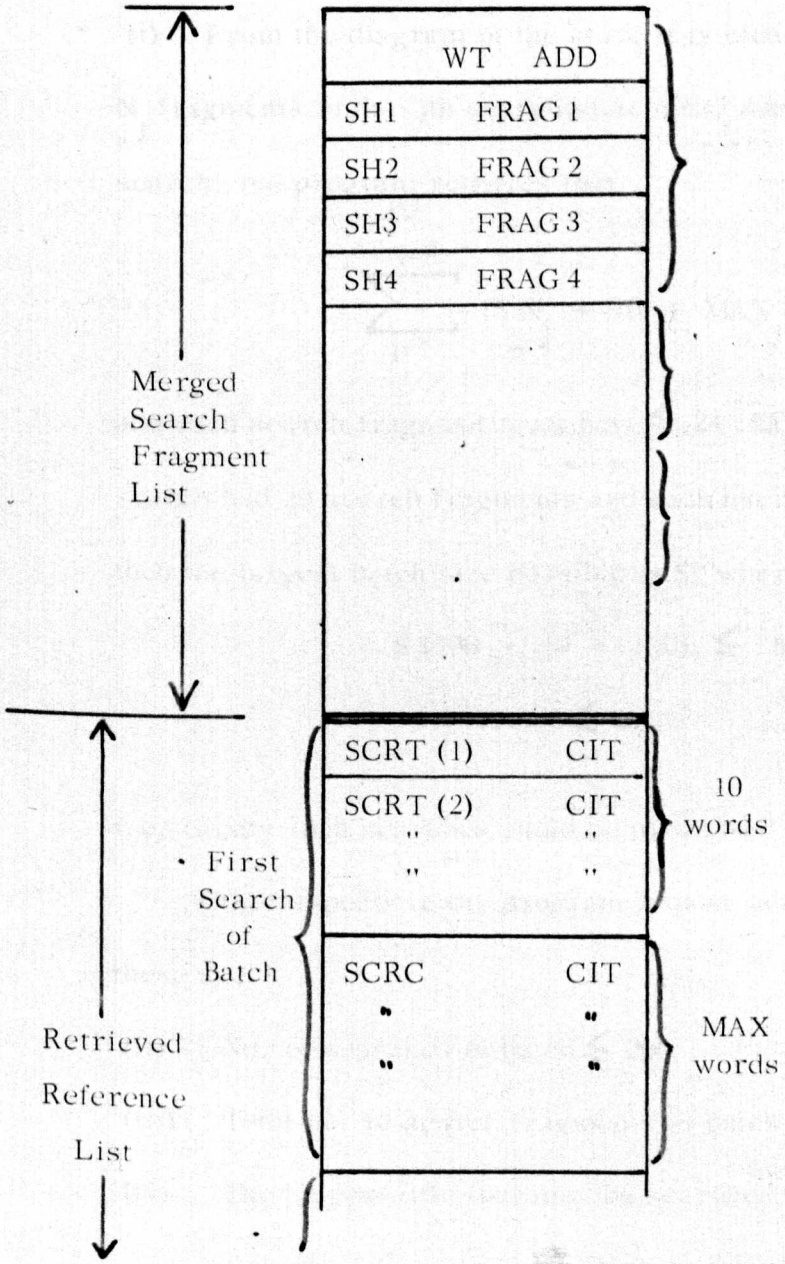
(2) A paper tape containing a batch of search formulations. Each formulation consisting of up to ten user-categories, each user-category being a list

of search fragments, with associated W_i provided by the user. For each search formulation the maximum number of references to be retrieved. Restrictions on batch size etc. are given below.

Layout of the KDF9 high-speed store

The program reads the paper data tape and combines all search terms (from whatever search) into a single list sorted in search fragment order. The 'sort' is alphanumeric, with a sorting order which places space first, followed by digits, and then by letters. No other characters are acceptable in search fragments. Before a title is compared with the search list all non-alphanumeric characters are replaced by spaces. This ensures compatability between titles and search fragments (some 'junk' characters in MEDLARS titles are eliminated altogether. They are only of use in the printing of Index Medicus) The KDF9 store consists of 16K 48-bit words. When system software, program, and magnetic tape buffer-stores are in store, some 11000 words remain. Of these, 4000 are used in processing the titles, and 6000 are used for a joint Search Fragment and Retrieved Reference list. This list is laid out as in the diagram.

Search Fragment and Retrieved Reference List



Five words for each search fragment. WT is the W_i assigned by the user. ADD is the address of the word for the score of the appropriate user-category.

The next four words contain the fragment, six characters in each word (FRAG 1 to FRAG 4). The first twelve bits of each word isa a 'shift' value which is used for masking (SH1 to SH4). The shift value is 48-6C where the computer word contains C characters, e.g. a fragment 'NEUROMUSC' would have 6 characters in the first word, 3 in the second, and none in the third and fourth.

For each search there is provision for up to 10 user-categories. SCRT(k) is the score of the kth user category.

SCRC is the score of a retrieved reference. CIT indicates a reference identification number. Up to MAX references may be retrieved.

Size Restrictions

(i) From the diagram of the store it is clear that for a batch of S searches with N_j fragments in the jth search statement, and an output limit of MAX_j on the jth search, the program requires that

$$\sum_{j=1}^{j=S} (5\,N_j + 10 + MAX_j) \leq 6000$$

and each search fragment must have $\leq$ 24 characters. For example: If each search had 20 search fragments and each user required no more than 190 references then the largest batch size possible is S, where

$$S\,(100 + 10 + 190) \leq 6000$$

$$\text{i.e. } S \leq 20$$

i.e. twenty such searches could be processed simultaneously.

Other aspects of the program impose additional restrictions, however, and these are -

(ii) No. of searches in batch $\leq$ 200

(iii) Total no. of search fragments in batch $\leq$ 800

(iv) The longest title that may be searched must be less than 2000 characters in length. At e.g. 100 characters printed across a line printer page this represents 20 lines, i.e. a short abstract. The longest MEDLARS title seen by the author was about 500 characters long.

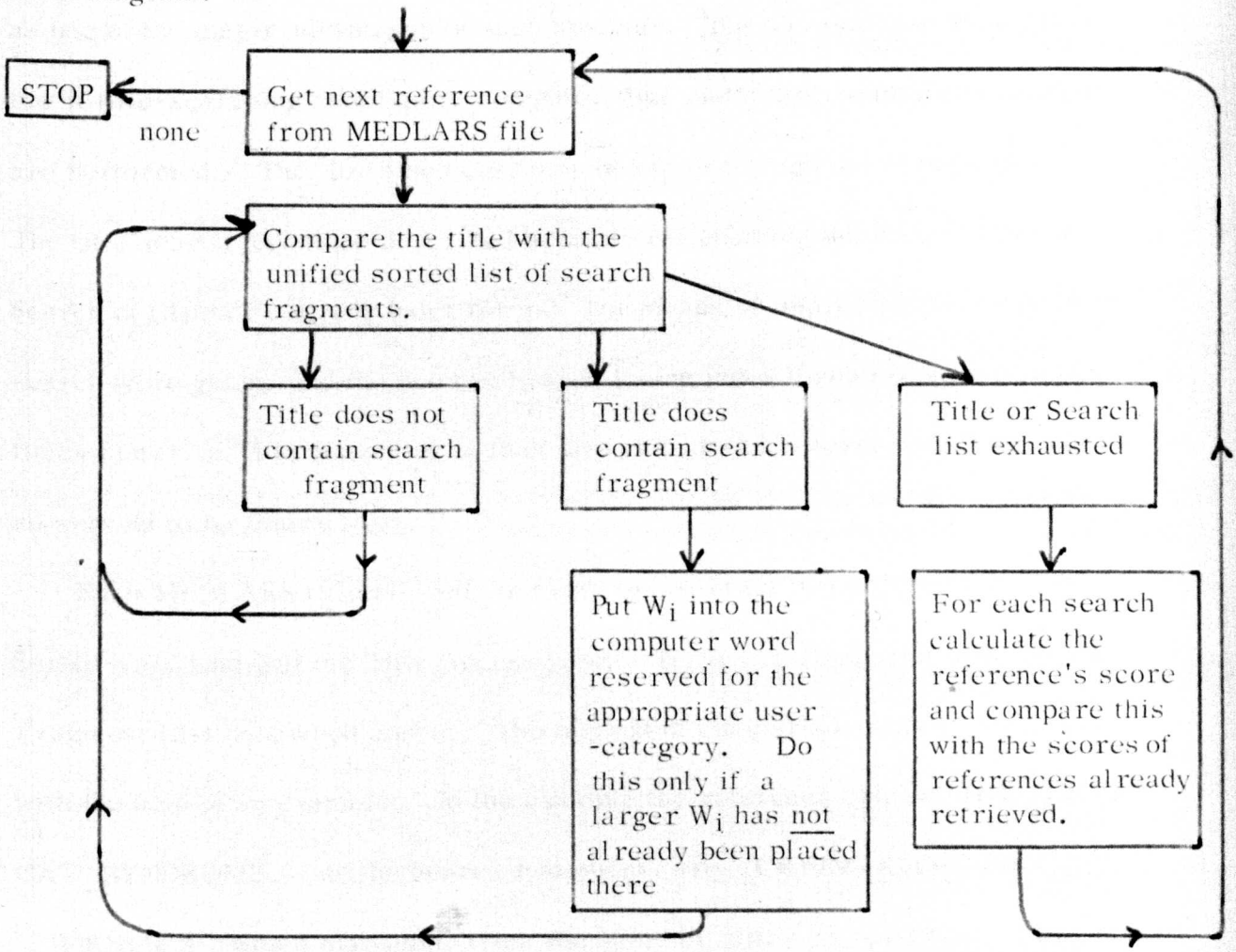
Different combinations of restrictions would be possible, but the number of search fragments must always be less than 2/5 th of the number of characters in a title since the store areas used to hold titles during the search, are used to sort the search fragments before the search begins.

Retrieval Action

The formula used for calculating the score of a reference is S_5, where

$$S_5 = \sum_{\text{all user-categories}} \left\{ \max_{\text{within category}} \left\{ \delta_{jr} W_j \right\} \right\}$$

as developed in 4.3. The retrieval action can be described by a very simplified flow diagram.



If the number of references already retrieved for a search already equals the maximum for that search, the current reference can only be added to the retrieved set at the expense of rejecting one which has already been retrieved. This is done if the score of the current reference exceeds the smallest score in the set. The program is so designed that a reference whose score is less than that minimum is rejected after only one comparison. This situation is likely to be frequent.

The most interesting part of the retrieval action is the inspection of the title for the presence of search fragments. The method used is very efficient and needs a detailed description.

Inspection of Titles for Search Fragments

The low cost of input to a Title-based retrieval system was given in 4.1 as one of the major advantages of such systems. But the total cost may not be low if title-searching takes more computer time and a large number of searches are performed. The low fixed cost may be offset by high operating costs. The title search technique described here almost equalled the speed of the Scoring Search of Chapter 2 (using index terms), but means of improving the speed of that search were given, and the Boolean Search (using index terms) was three to four times faster. Title Searching is thus slower than index-term searching, but not so slow as to be impractical.

Each MEDLARS title is used, in turn, to construct two ordered lists, the Sorted Title List and the Title Access List. These are compared with the Search Fragment List described above. The method of comparison is best described with the help of an example. In the example the reference title is 'THE_CRYING_CAT_SYNDROME.' and the Search Fragments are '_CRYING_CAT_' and '_TURNER_S_' which may come from the same or different searches. (The dashes are used here in the text to represent spaces). At the time of search the three lists would appear as:-

Search Fragment

WT 1			ADD 1			
12		C	R	Y	I	N
12	G		C	A	T	

WT 2			ADD 2			
12		T	U	R	N	E
24	R		S			

Sorted Title

28						
29						
0						T
1					T	H
2				T	H	E
3			T	H	E	
15		C	A	T		S
8		C	R	Y	I	N
19		S	Y	N	D	R
4		T	H	E		C
17	A	T		S	Y	N
16	C	A	T		S	Y
9	C	R	Y	I	N	G
23	D	R	O	M	E	
27	E					
7	E		C	R	Y	I
14	G		C	A	T	
6	H	E		C	R	Y
12	I	N	G		C	A
26	M	E				
22	N	D	R	O	M	E
13	N	G		C	A	T
25	O	M	E			
24	R	O	M	E		
10	R	Y	I	N	G	
20	S	Y	N	D	R	O
18	T		S	Y	N	D
5	T	H	E		C	R
11	Y	I	N	G		C
21	Y	N	D	R	O	M

Title Access

					T	0
				T	H	1
			T	H	E	2
		T	H	E		3
	T	H	E		C	4
T	H	E		C	R	5
H	E		C	R	Y	6
E		C	R	Y	I	7
	C	R	Y	I	N	8
C	R	Y	I	N	G	9
R	Y	I	N	G		10
Y	I	N	G		C	11
I	N	G		C	A	12
N	G		C	A	T	13
G		C	A	T		14
	C	A	T		S	15
C	A	T		S	Y	16
A	T		S	Y	N	17
T		S	Y	N	D	18
	S	Y	N	D	R	19
S	Y	N	D	R	O	20
Y	N	D	R	O	M	21
N	D	R	O	M	E	22
D	R	O	M	E		23
R	O	M	E			24
O	M	E				25
M	E					26
E						27
						28
						29

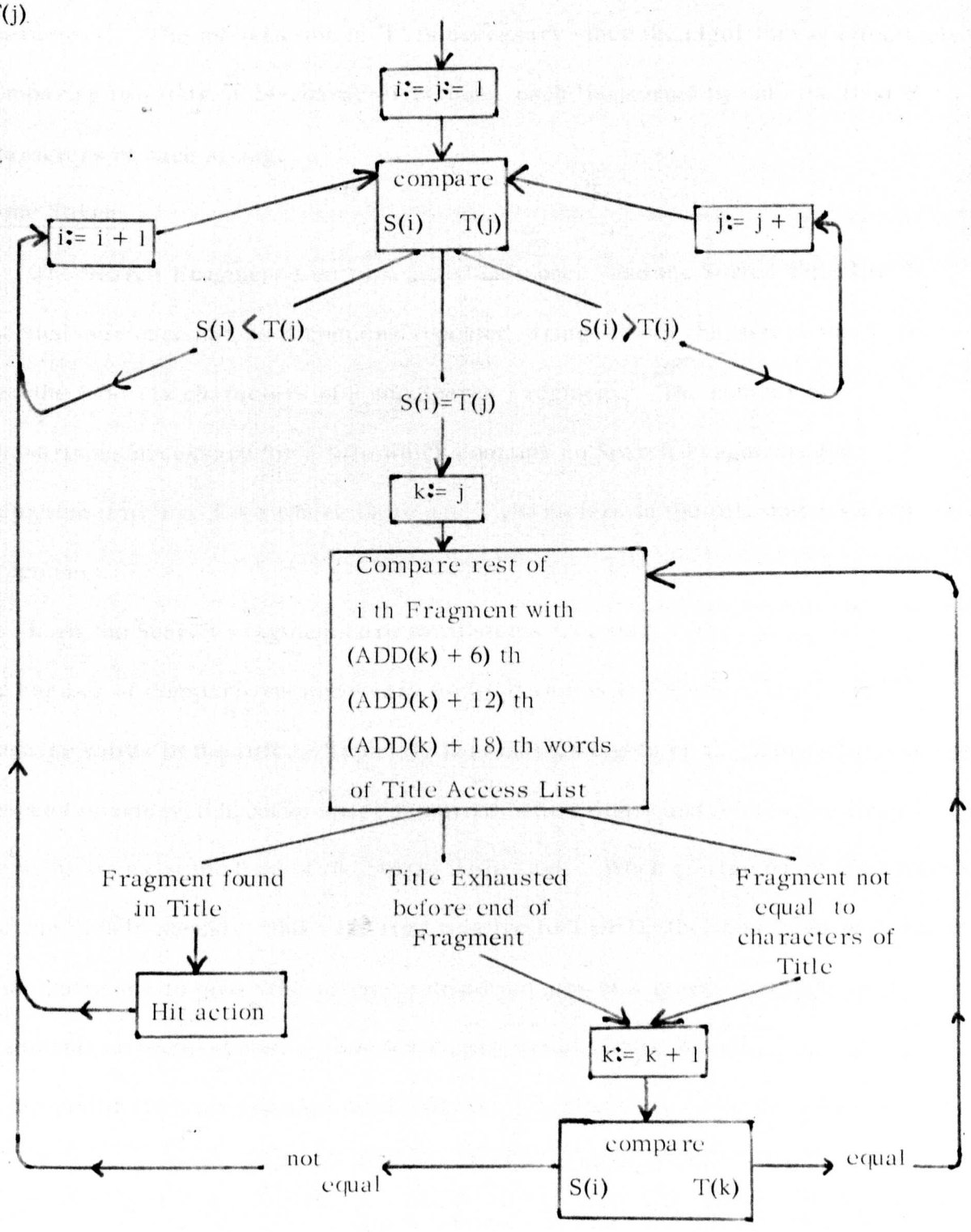
Relative Address of Computer word in Title Access List

The Title Access List contains one computer word for each character in the title. Each computer word contains six consecutive characters from the title. If the first character in each computer word is read by scanning the eye down the page, the title is immediately visible. The Sorted Title List contains the same computer words but rearranged into dictionary order (with space before digits before letters). Stored in each word of the Sorted Title List is the address of that same word in the Title Access List. The Search Fragment List has been described already. It is in an order determined by the first six characters of each Fragment.

Since the Search Fragment and Sorted Title Lists are both in a sorted order, the title can be inspected, for the occurrence of the first six characters of a Search Fragment, by a 'pop-up' procedure. When the first six characters are found in a title, a much slower algorithm inspects it for the rest of the Search Fragment. The Sorted Title word contains ADD, its address in the Title Access List. The remaining characters of the Search Fragment are compared with those in the (ADD + 6)th, (ADD + 12)th, (ADD + 18)th words of the Title Access List. Since the Search Fragment and Sorted Title Lists are only sorted on a key of six characters the algorithm for detecting 24 character Search Fragments is not a simple 'pop-up' procedure, but the more complicated algorithm is faster than a multi-computer-word sort followed by a simpler comparison procedure. The comparison algorithm used in the program is given in the flow diagram below:-

COMPARISON OF SEARCH FRAGMENTS WITH TITLES

Words in Search Fragment List denoted by S(i) (strictly S(i) is the first six characters of the i th Fragment). Words in Sorted Title List denoted by T(j)



For simplicity the flow diagram does not show the stopping criteria (S or T lists exhausted), nor the masking procedures for comparing Search Fragment List words of less than 6 characters with the Title words, all of which contain six characters. The introduction of 'k' is necessary since the algorithm is effectively comparing two lists of 24-character strings, each list sorted by only the first 6 characters of each string.

Time Taken

The Search Fragment List is scanned only once, and the Sorted Title List is scanned only once unless it contains repeated strings of six characters which are also the first six characters of some Search Fragment. The number of comparisons necessary for a title which contains no Search Fragments (the 'rejection time') is T + S where there are T characters in the title and S Search Fragments.

If all the Search Fragments are word stems i.e. start with a space, then the number of comparisons necessary for rejection is D + S where there are D language words in the title. Typically D is only an eighth of T. This improvement in speed is achieved because space is sorted before digits and letters, putting all the word stems at the head of the Sorted Title List. When a large batch of searches are run simultaneously, and S is large relative to T or D, the improvement in speed is not sufficient to offset the retrieval disadvantages of a restriction of Search Fragments to word-stems. This restriction would reduce Recall.

Once the first six characters of a Search Fragment are found the program leaves the fast pop-up routine and enters the much slower section which examines the remaining characters. The program will operate at a much higher speed if Search Fragments are restricted to less than six characters. Again the restriction will give an improvement in speed at the expense of retrieval disadvantages. This restriction would reduce Precision.

The pop-up routine is left more frequently when Search Fragments occur more frequently in Titles. Thus frequently occurring Search Fragments slow the program down.

If all these complications are ignored (because they are difficult to handle, not because they are unimportant) the computer time taken is split between overheads and time related to the number of Search Fragments. Four searches of 34,000 references using different lists of Search Fragments gave results:-

Time (in mins.)	No. of Search Fragments
48	108
46	68
57	175
39	3

If the overhead is eliminated by subtracting the last result from each of the others, the estimates of the times per search fragment are (in minutes), 0.086, 0.108, 0.104, per 34,000 references. The time per search fragment per reference is approximately 2×10^{-4} seconds. This compares with about 0.8×10^{-4} seconds per search term (index term) in the Probabilistic Scoring Search of Chapter 2. However in the test below, with a large number of search fragments, the title search program took less time than these calculations would predict.

The overhead is large, about 39 minutes. This was verified by searching for the single fragment '-AAA', which again took about 39 minutes. Since such a search involves the comparison of only one search fragment with only about 3 words from the Sorted Title List, the time taken is almost entirely due to the construction of the two title lists - Sorted Title and Title Access. This takes a long time since every character in the title has to be inspected, puctuation characters replaced by spaces, and characters only used for printing Index

Medicus eliminated. The Title Access List is formed during this inspection. Then it is sorted to form the Sorted Title List. If the title search were adopted as a standard search procedure, this overhead could be all but eliminated by processing the titles once only, and storing the two title lists on a special magnetic tape. Then the speed of the title Search program would be about half the speed of the Probabilistic Search program according to these estimates. In the test below, the Title Search went at almost the same speed as the Probabilistic.

4.6 A Comparison of Boolean and Scoring Searches using Index Terms with Scoring Searches using Titles

Two tests of retrieval performance were made. In each test a number of searches were performed by the standard Boolean Search, by the Title Search, and by the Probabilistic method. These tests were described in detail in 2.9, and to avoid needless repetition only the aspects which affected the Title Search are described here.

Test 1: Description and Results

Users of the MEDLARS Monthly Selection Service were given the option of Title Searches. Of the 50 users some did not accept the offer, and some did not evaluate the references retrieved. As a result only 13 searches were available for use in a comparison of Boolean and Title searching. The Title Search Statements were formulated by MEDLARS staff from the terms of the Boolean Search Statements. No use was made of user-categories. Output sizes were set arbitrarily at 30 or 75.

The Boolean Searches did overwhelmingly better. In 8 of the 13 searches the Boolean was better on both Precision and Relative Recall. In 4 of the remaining 5, the Boolean Search was strictly better on one ratio but worse on the other. Of the relevant references retrieved the Boolean Searches averaged 70% and the Title Searches 43%, although these figures are complicated by the

fact that not all techniques were used for all searches.

From this test it was obvious that a further test was required, in which all searches were performed by both techniques, and in which more care was taken in selecting the output size of the Title Searches, in order to get fully comparable results.

Test 2: The criteria which determined the design of the second test are given in 2.9. The design was an attempt to meet the partially contradictory requirements of realism and comparability.

For the Title Search Formulations it was considered unacceptably unrealistic to take as Search Fragments the stems of the MeSH terms used in the corresponding Boolean Searches. Ideally the Title Searches should have been formulated in exactly the same manner as the Boolean i.e. at personal interviews between the Strathclyde Library staff and the users. This could well have exhausted their cooperation before the evaluation stage was reached. The Title Searches were therefore formulated by a Strathclyde Librarian and a subject expert in consultation with the author. They were based on the meaning of the MeSH terms used in the Boolean Search Statements, and the Librarian's memory of the original (Boolean formulation) interviews with users. The titles retrieved by the original Boolean Searches were also available to him. In two cases the Boolean Search formulation consisted almost entirely of MEDLARS category numbers, and the user had to be interviewed to provide Search Fragments.

The Boolean Search Statements used 482 MeSH index terms and 33 category numbers. The Title Searches used 492 Search Fragments, some such as '-AMIN', 'ETHYL', 'CAINE' intended to be the equivalent of MEDLARS category numbers, others corresponding to individual terms. Although the Title Search Fragments had not been generated in a mechanical fashion from the MeSH terms of the Boolean Search the relationship between Fragments and Boolean Searches

was close. Of the 492 fragments, all but 68 appeared in one of

(i) The MeSH terms of the Boolean Statement

or (ii) The MeSH terms represented by the Category numbers in the Boolean Search Statement

or (iii) The Title of the Boolean Search Statement

(The two title search statements formulated by the user are not included). Some effects of the inclusion of these 68 terms are analysed in the results section below. Most, but not all, are obvious synonyms for MeSH terms used in the Boolean Statements,

e.g.

Boolean Statement (MeSH term)	Title Statement (Word Fragment)
ASCORBIC ACID	"VITAMIN - C"
BIRD	"-AVIAN"
COMPUTER	"MACHIN"
DENTAL terms	"TOOTH"
DENTAL terms	"TEETH"

This table shows some of the 68 fragments not found in MeSH terms of the corresponding search.

User-Categories: As with the Probabilistic Search, user-categories were used in every Title Search Formulation. The example quoted in 2.9 of the search on "Hyaluronidase and Dental Caries' is equally pertinent here.

Computer Time Taken: Both Boolean and Title Search programs handled all 25 searches in one batch. The time taken to search each tape of 35,000 references was about 8 to 10 minutes for the Boolean, and about 80 minutes for the Title Search. However about 40 minutes could be ascribed to the 'overhead' function of setting up the title lists. The effective search time for the Title Search

program was thus about 40 minutes. The number of search fragments used was 492. Using the values calculated in the last section estimated time for the Title Search is

$$\text{Est. time in mins.} = 40 + \frac{2 \times 10^{-4} \times 492 \times 35000}{60}$$

$$= 40 + 57$$

$$= 97 \text{ mins.}$$

This estimate is considerably more than the actual time taken. The program apparently becomes more efficient as the number of search fragments is increased. This did not happen with the Probabilistic Scoring program which is structurally similar to the Title Search program. This performance difference is due to the action of the 'pop-up' method for comparing two sorted lists.

Suppose, for the sake of clarity, that a sorted list of search fragments is to be compared with a sorted title list, and that the title does not contain any of the search fragments. Suppose also that the two lists are sufficiently long, or evenly distributed through the alphabet, that the comparison does not stop until both lists are (almost) exhausted. Then the number of individual comparisons that must be made is (T + S) where there are T elements in the Sorted Title list, and S in the sorted Search Fragment list. The value of T + S arises since, after each comparison, one or other list is popped up i.e. one element from one list is rejected and is never considered again. In the Probabilistic Search the number of comparisons may also be written (T + S) where T represents the number of index terms of the reference, and S the number of search terms. In the test, the values of S for both Title and Probabilistic Search were approximately 500, but the values of T were very different. For the Probabilistic Search the average value of T is 9, for the Title Search T is the number of characters in the title, and the average length of titles is in the region of 80 to 100 characters.

The number of comparisons for varying numbers of search terms are set out in the table.

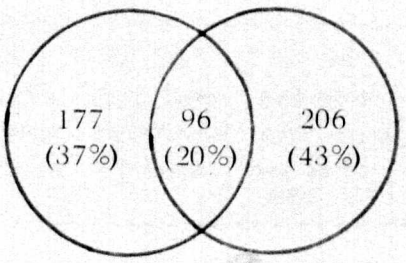
No. of Search Terms	No. of comparisons (approx.) Title Search	Probabilistic Search
100	200	110
200	300	210
300	400	310
400	500	410
500	600	510

As can be seen, the efficiency of the Title Search algorithm is relatively low for small numbers of Search Fragments.

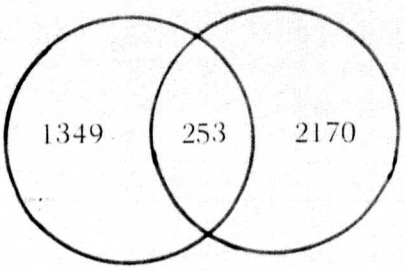
Results of Test 2 The results are given in diagrams of two overlapping circles, each corresponding to the references retrieved by one technique.

The overall results for the 150 searches were:

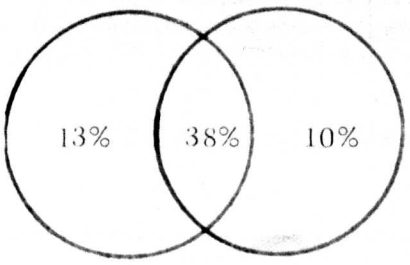
BOOLEAN SEARCHES TITLE SEARCHES



No and % of Relevant References Retrieved (150 searches). % is Relative RECALL



Total no. of references retrieved (150 searches)



Precision (%)

(150 searches)

The precision of retrieval for each technique can be calculated from the first two diagrams, and is 17% for Boolean, 12.5% for Title Searching. While it can be misleading to quote overall Relative Recall figures these are given in the first diagram.

The output size of the Title Search was equal to that of the Boolean when the Boolean retrieved 10 or more references, and was 10 otherwise. The results for different output sizes are given in the table, which shows the number of relevant references retrieved by Title Searches with (a) the same output size as Boolean, (b) Title output 10, Boolean in the range 5 to 9, (c) Title output 10, Boolean in the range 0 to 4. The table also gives the numbers of search formulations, and of searches, on which these retrieval figures are based. The statistical significance of the results can be calculated from the numbers of searches in which the Title Search retrieved more relevant references than the Boolean and these figures are also given in the table.

Boolean output size	No. of Rel. Refs. retrieved by Boolean only	No. of Rel. Refs. retrieved by both	No. of Rel. Refs. retrieved by Title only	No. of search formulations	No. of searches	No. of searches in which Boolean better	No. of searches in which Title better
10	128	71	85	9	41	26	5
5 - 9	28	17	37	13	25	7	10
0 - 4	21	8	84	19	84	4	40

This table is clearer in percentaged form: -

Boolean Output Size	% of relevant references retrieved Boolean only	Both	Title only	Total
10	45	25	30	100%
5 - 9	34	21	45	100%
0 - 4	19	7	75	101%

These tables show that when the output sizes were equal, the Boolean Searches performed better, and that the superiority of performances was consistent (26 out of 31 searches i.e. significant at the 0.01% level). When the Boolean Searches retrieved few references, and the Title output was between one and two times the Boolean output size, the Title Searches retrieved more relevant references, though the superiority was not very consistent (10 out of 17 searches, i.e. significant at the 32.0 % level). When the Title output was very much larger than the Boolean, many more relevant references were retrieved and the superiority of the Title Searches was consistent (40 out of 44 searches i.e. significant at the 0.00001% level).

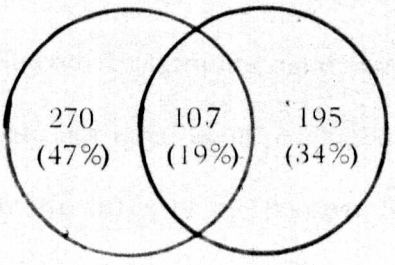
The tables suggest that Title Searching was inferior to Index term searching in retrieval efficiency, but that by using a scoring technique and setting a large output size, searching by titles could produce as many, or more relevant references than the corresponding Boolean Searches. Users might be willing to spend more of their time 'editing' the output, if the reduction in system costs were passed on to them in the form of lower charges for searches.

Titles and Index Terms

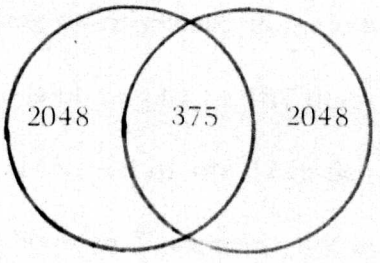
The comparison of the Title Search program with the MEDLARS Boolean Search contributes to the evaluation of MEDLARS, but does not directly evaluate the usefulness of titles for retrieval, since the retrieval techniques differ. By comparing the Probabilistic Scoring search with the Title Scoring search, titles can be compared with MEDLARS indexing, since the search algorithms are basically similar. The results of this comparison have to be treated with caution, however, since there may be interaction between the retrieval technique and the data to which it is applied. e.g. It is possible, even if unlikely, that index terms might give better results than titles when used by a Scoring Search, and worse results when used by a Boolean Search, or vice versa.

In the test 2 described above and in Chapter 2, each search was done using the Boolean, the Probabilistic, and the Title Search program. Since the criterion used to determine the output sizes of the Probabilistic and Title Searches was the same, the output size of each Title Search equalled the output size of the corresponding Probabilistic Search. The two methods can thus be compared by counting the number of relevant references retrieved by each, and the number of searches in which the one retrieved more than the other. The results were ;

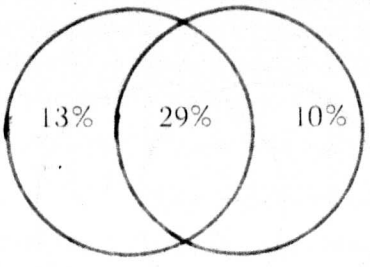
PROBABILISTIC TITLE



Number and % of relevant references retrieved (150 searches)



Number of references retrieved (150 searches)



Precision (%) (150 searches)

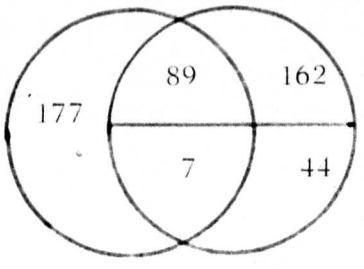
Since the output sizes are the same, the Relative Recall percentages for the two techniques stand in the same ratios to one another as do the Precision Ratios (66% to 53%). The table shows that the index terms are more useful for retrieval, but not overwhelmingly so. Of the 150 searches, the Probabilistic Scorer retrieved more relevant references in 63, less in 41, and the same as the Title Scorer in the remaining 46. The hypothesis that the Probabilistic Scorer is better is significant at at least the 4.0% level.

Non-Boolean terms used as Search Fragments

The construction of the Title Search Statements was described above. Of the 492 Search Fragments used, 68 could be described as 'non-Boolean' in that they could not have been produced by selecting character-sequences from the terms or the titles of the Boolean Search Statements. Of these 68 only 17 contributed to the scores of relevant references retrieved by the Title Search program. The others either did not appear, or appeared with other terms from the same user-group, these other terms having equal or larger weights than the non-Boolean one. The diagram shows how many of the relevant references retrieved by the Title Searches would have received lower weights had the non-Boolean Fragments not been used. They might still have been retrieved by the Title Searches, albeit with lower scores.

NO. OF RELEVANT REFERENCES RETRIEVED

BY BOOLEAN SEARCHES



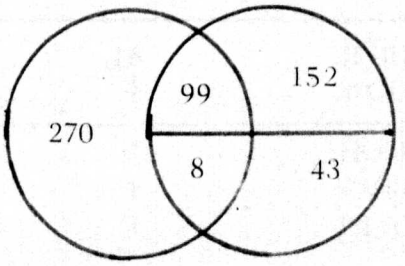
BY TITLE SEARCHES

Score not affected by non-Boolean Terms

Score contains contribution from non-Boolean Terms.

A similar diagram can be constructed to compare the Probabilistic Searches and the Title Searches:-

PROBABILISTIC TITLE



Unaffected

Affected

In each case the non-Boolean terms contributed to the score of about one fifth of the references missed by the other technique. The retrieval performance of the Title Searches does not appear to depend excessively on these terms. The non-Boolean terms which contributed to relevant reference scores can be examined in detail. The table shows the number of references affected, the non-Boolean Term involved, and in the last column, a list of the 'Boolean Fragments' which prompted the use of the non-Boolean term:—

Formulation No	No. of Refs	Non-Boolean Fragment	Associated Fragments
8	1	GINGIVA	12 dental terms plus SALIVA
8	1	ORAL	
9	16	BURN	BANDAGES, DRESSINGS
9	3	WOUND	
10	7	HOSPITAL	MEDICAL RECORDS, INFORMATION RETRIEVAL, STATISTICS, DOCUMENTATION
10	1	PATIENT	
10	3	DIAGNOSIS	
10	1	ELECTRON	COMPUTER PROCESSING
11	4	VITAMIN C	ASCORBIC ACID
13	2	DETERMIN	CHEMICAL STRUCTURE
13	1	STUDY	
15	1	RIBOSOM	MICROSOMES
15	5	ENZYME	ESTERASES, GLUCOSYL-TRANSFERASES
16	1	AVIAN	BIRD MUSCLE
17	1	ANTITHYRO	THYROID ANTAGONISTS
17	1	CATABOL	METABOLISM
25	3	RESPONS	BLOCKING AGENTS, NEUROMUSCULAR FACILITATION

THE EFFECT OF NON BOOLEAN FRAGMENTS

The judgement of whether these terms biased the test in favour of the Title Search program is necessarily subjective. The Search Formulations were obviously not identical with the Index-term Formulations, nor could they have been, since the index-term formulations used 33 MeSH category numbers, and it would have been impractical to have listed all the terms which appeared under those category numbers. The Title Search Formulations did not use more terms than the index term formulations, in spite of the use of the non-Boolean Fragments. The judgement that must be made is whether users and search-writers in consultation would have written better or worse Title Search Formulations than the ones used in the test. In the author's judgement all the non-Boolean terms in the table could have been expected from the dullest of users, except for those used in formulation number nine. The use of BURNS and WOUNDS to retrieve references on the treatment of burns and the treatment of wounds (the title of the search was 'Metallized Dressings'), in addition to BANDAGES and DRESSINGS does seem an improvement of the search Formulation which is not completely obvious. These terms affected 19 references. Although this made them good Search Fragments they did not have a large effect on the total results.

MEDLARS Subheadings. Seven of the relevant references missed by the Boolean Searches and retrieved by Title Searches would have been retrieved by the Boolean Searches had the Formulations contained their original Subheadings in addition to the index terms.

Conclusions from Test 2.

The merits and demerits of the format of the test are listed in 2.9. The scale of the tests - 150 searches using only 25 search formulations - means that the results apply for medical scientists using the MEDLARS system. For other users of MEDLARS, and for other systems, conclusions

based on these results hold with less certainty.

A direct comparison of the efficacy of MEDLARS titles and MEDLARS indexing for retrieval is given by the comparison of the Probabilistic with the Title Searches, since, for every search, the output size was the same, and both used a Scoring technique for retrieval. The index terms retrieved considerably more relevant references than the titles (378 to 303), but the advantage was not so large as to make economic factors unimportant. If a retrieval system based on titles cost very much less for reference-input, the higher costs of searching and the lower Recall achieved might be accepted. More relevant references could be retrieved by setting larger outputs and leaving the users with more sifting to do. The scale of the retrieval system is very important for the success of a title-based retrieval technique. If the system is large enough and important enough within a discipline, its adoption of title based retrieval might encourage authors and journal editors to take care that titles were fully descriptive of the contents of references. The effect, if it occurred, would probably not occur until the system had been in operation for some time.

The comparison of Boolean and Title Searches measures the performance of the Title Search algorithm against the standard MEDLARS system (and v.v.). For Title Searches with output size equal to the Boolean output size the ratio of relevant references retrieved by the two methods was similar to the ratio for Probabilistic to Title. It was 12.75 to 10 for Boolean to Title (at equal output) and 12.50 to 10 for Probabilistic to Title. However, as the size of the Title Search output was increased beyond that of the Boolean Search output the superiority passed to the Title Search. For Title Searches with outputs of 10 against Boolean Searches with outputs of 0 to 4, the ratio of relevant references retrieved was 31 to 10 in favour of the Title Search. This suggests that the

superior performance of index terms was offset by a retrieval technique which automatically widened the search specification, provided that larger than Boolean output sizes were set. The test showed that Titles as a means of retrieval should not be rejected, either on the grounds that title searching would use an impractically large amount of computer time, or on the grounds that it would not give adequate Recall or Precision, unless an extensive test has shown that they gave poor retrieval results in the subject area under study. If no extra costs are incurred by indexing the references, index terms should be used since they give better retrieval. MEDLARS indexing, for example, is produced as a by product of the publication of Index Medicus, and the indexing cost is thus not easy to determine. Even where index terms are available, the construction of the vocabulary may make it unsuitable for particular searches. In that event a Title Search program provides an efficient alternative to an index term search.

References Chapter 4

1 Kraft, D.H., Amer. Doc., 15, 1964, 48

2 Maizell, R.E., Rev. de la Doc., 27, 1960, 126.

3 Ruhl, M.J., Amer. Doc., 15, 1964, 136.

4 O'Connor, J., Amer. Doc., 15, 1964, 96.

5 Montgomery, C., Swanson, D.R., Amer. Doc., 13, 1962, 359.

6 Lancaster, F.W., Amer. Doc., 20, 1969, 119.

7 MacMillan, J.T., Welt, I.D., Amer. Doc., 12, 1961, 27.

8 Rodgers, D.J., "A Study of Inter-indexer Consistency," General Electric Co., Washington, 1961.

Chapter 5. Retrieval without Indexing II. Associative Retrieval on a Citation Network.

5.1 Associative Networks

5.2 Bibliographic Coupling

5.3 Network Transmittance Algorithms for Associative Retrieval

- 5.3.1 Retrieval on a Directed Network
- 5.3.2 Additive Scores
- 5.3.3 Non-additive Scores
- 5.3.4 Computation of Network Transmittance
- 5.3.5 Retrieval on a Citation Network

5.4 The Effect of Incompleteness on Associative Searching using Transmittance Algorithms

5.5 A Computer Representation of a Citation Network based on References from a Card Index maintained by an M.R.C. Unit

5.6 Quantities of Computation required for the Construction of the Network Representation

5.7 A KDF9 Program to Search the Citation Network

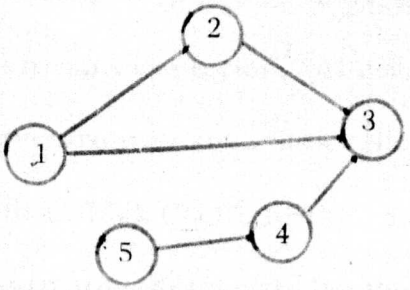
5.8 A Comparison of Scoring Searching using MEDLARS Index Terms with Citation Network Searching

Chapter 5. Retrieval without Indexing.

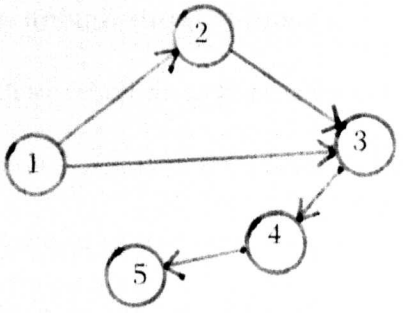
II. Associative Retrieval on a Citation Network

5.1 Associative Networks

An Associative Network is a set of 'objects', any pair of which may be 'linked'. These links may have direction, in which case the network is termed a 'Directed Network'. If the links also have magnitude the network is described as an Associative or Directed Network 'with Link Coefficients'. An alternative terminology is 'Linear Graph' for 'Associative Network' and 'Directed Linear Graph' for 'Directed Network'. Networks can be represented by diagrams such as the following.

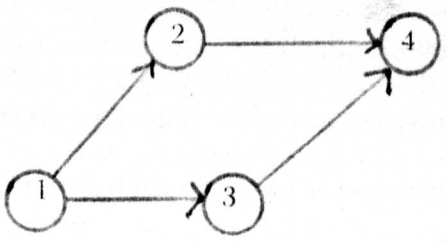


Associative Network

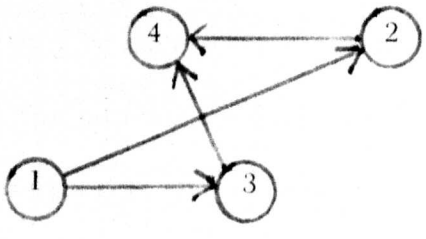


Directed Network

In such diagrams objects are represented by circles, links by lines, directions by arrows. The layout of the circles on the page has no significance. For example, the two diagrams that follow represent the same network.

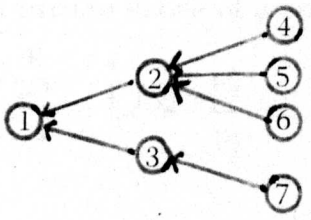


Representation 1.



Representation 2.

A Directed Network in which no object has more than one link leaving it (or in which no object has more than one link entering it) will be called a 'Strict Hierarchy', or a 'Tree'. An example is given in the diagram:



Strict Hierarchy

An example of a network in the standard MEDLARS system is the MeSH [1] Category Structure described in Chapter 2. Each term in the vocabulary can be linked to up to four more-general terms, and to an unlimited number of more-specific terms. Although it can be loosely described as a hierarchical structure it is not a Strict Hierarchy as defined above, but is a Directed Network without Link Coefficients. Such network structures are ubiquitously useful in Information Retrieval, but the discussion in this Chapter will be limited to networks in which the 'objects' are references. These networks of references may either exist naturally or be generated by calculation. An obvious example of a natural network is the Citation Network. Most published research papers quote a number of other published papers and so form a Directed Network without Link Coefficients. Lengthy computer processing may be necessary to produce a representation of this network in a form suitable for information retrieval, but it is only the representation which requires construction, the network itself has already been formed by authors citing each other's papers.

To use a network for retrieval it is necessary to assume that references which are linked are in some way similar. Authors may cite papers for many reasons other than the similarity in content to their own paper, and artificially generated 'similarity' networks could prove more useful for retrieval than

natural ones. For example an associative network can be derived from a coordinate indexed file by using a modified form of the scoring formula of Chapter 2 to calculate the similarity between any pair of indexed references. The formula given for the score of a reference was:

$$S = \sum_{i=1}^{i=K} \delta_i \log \frac{w_i}{p_i}$$

where there are K search terms and

$\delta_i = 1$ if the ith search term is present in the indexing of the reference,

and $\delta_i = 0$ otherwise,

w_i = the user's estimate of the proportion of relevant references that are indexed by the i th search term.

p_i = the proportion of references in the file which are indexed by the i th search term.

This formula can provide a measure of the similarity between two references if the list of index terms attached to one reference is taken as a Probabilistic Search Statement, and the score of the other reference is calculated with respect to this Search Statement. Denoting the similarity between references 1 and 2 by $S_1(1, 2)$, the formula becomes:

$$S_1(1, 2) = S_1(2, 1) = \sum_{i=1}^{i=K} \delta_i^{12} \log \frac{1}{p_i}$$

where there are K distinct index terms used to index the references and

$\delta_i^{12} = 1$ if term i indexes both 1 and 2

and $\delta_i^{12} = 0$ otherwise

and $w_i = 1$ for all i.

The formula is obviously symmetric. The formula gives values of $S_1(1, 2)$ in the range $0 \leq S_1(1, 2) < \infty$. A normalised form giving values in the range $0 \leq S_2(1, 2) \leq 1$ is:-

$$S_2(1, 2) = \frac{\sum_{i=1}^{K} \int_i^{12} \log \frac{1}{p_i}}{\sqrt{\left(\sum_{i=1}^{K} \int_i^{1} \log \frac{1}{p_i}\right)\left(\sum_{i=1}^{K} \int_i^{2} \log \frac{1}{p_i}\right)}}$$

where $\int_i^1$ and $\int_i^2$ are defined in a similar manner to $\int_i^{12}$.

The calculation of similarities $S(\alpha, \beta)$ between each pair of references (α, β) in the file gives an Associative Network with Link Coefficients.

5.2 Bibliographic Coupling

A Citation Network is a Directed Network without Link Coefficients, and can be used directly for information retrieval, but it can also be used to generate other networks, and the generated networks used for retrieval. The retrieval method known as Bibliographic Coupling [2, 3] is one technique for generating a (non-directed) Associative Network. This is done by calculating the similarity between two references, 1 and 2, by the formula:

$$S_3(1, 2) = S_3(2, 1) = \sum_{i=1}^{i=K} \int_i^{12}$$

where the references 1 and 2 cite K distinct references, and $\int_i^{12} = 1$ when both cite the i th cited reference, and $\int_i^{12} = 0$ otherwise. This gives a measure of similarity in the range $0 \leq S_3(1, 2) < \infty$. A normalised form, giving values in the range $0 \leq S_4(1, 2) \leq 1$ is:-

$$S_4(1,2) = \frac{\sum_{i=1}^{K} \int_i^{12}}{\sqrt{\left(\sum_{i=1}^{K} \int_i^{1}\right)\left(\sum_{i=1}^{K} \int_i^{2}\right)}}$$

where $\int_i^1$ and $\int_i^2$ are defined in a similar manner to $\int_i^{12}$.

The form of $S_3(1, 2)$ and $S_4(1, 2)$ is the same as that of $S_1(1, 2)$ and $S_2(1,2)$ respectively, apart from the weighting factors $\log \frac{1}{p_i}$. This correspondence arises from the treatment of citations as though they were index terms.

The Search Statement for a Bibliographic Coupling Search consists of a set of references provided by the user, and known as the Request Set. Some, at least, of these Request Set references must be in the Citation Network. The criterion for the retrieval of a reference is that the sum of its similarities to the Request Set must exceed some threshold value. Alternatively, each reference can be given a score equal to the sum of its similarities to the Request Set, and the N highest scoring references retrieved. (N equals the required output size).

Bibliographic Coupling is particularly appropriate when the representation of the Citation Network consists only of a file of references each with a list of papers cited by it. This representation is close in form to the original printed texts, and is thus relatively inexpensive to construct. In addition, this representation is identical in form to an indexed file of the MEDLARS type, and programs developed for index-term retrieval can perform Bibliographic Coupling Searches. The program described in Chapter 2 for example, only needs to omit the weighting factors $\log \frac{w_i}{p_i}$ to become a Bibliographic Coupling program.

More general Network Retrieval Algorithms proposed in the next section are not related to the representation of the network but to its logical structure. Bibliographic Coupling will be shown to resemble a special case of a Network Retrieval Algorithm.

5.3 Network Transmittance Algorithms for Associative Retrieval

5.3.1 Retrieval on a Directed Network

Let us assume that a Network Search commences when a user specifies a Request Set, and gives each member of that set an initial weight or score. Thereafter references in the network receive scores (or have their scores changed) by virtue of being linked to other references which already have scores. Two conditions are intuitively reasonable:-

(1) A reference should not transmit to other references a higher score than it receives from other references i.e. no reference should act as an amplifier.

(2) The score transmitted by a reference should depend upon the number of references to which a score is sent; the larger that number the smaller the score transmitted. A review article quoting 500 references does not, intuitively, connect those references to each other or to the review as closely as a research paper quoting 5 references connects those 5 to each other or to itself.

Both these conditions are met when the score received by a reference is divided between the references receiving scores from it. This process overfulfills the first condition in that the total of scores transmitted by a reference does not exceed its own score, but it provides a simple criterion for determining the score sent along a link, and it satisfies the required conditions. To construct a retrieval algorithm a criterion for determining the score received by a reference is also necessary. Two alternatives are given in the next sections.

5.3.2 Additive Scores

One criterion for the score of a reference is 'the sum of all scores sent to it'. To prevent spurious amplification of reference scores by e.g. two references citing each other, this criterion must be qualified by the condition that no reference may receive a score which originated from itself, however indirectly.

This criterion taken with that of 5.3.1, defines the score of all references in the network. The calculation of reference scores is the same as the calculation of Causal Effects in Causal Models (e.g. [4][5]), except for the restriction that no reference may receive a score originating from itself. This restriction simplifies the calculation. (Two variables which cause each other do, quite legitimately, amplify each others values, though often to a finite limit).

The summation of scores is appropriate when the user expects that a reference linked to several of the Request Set would have a higher probability of relevance than one linked to only one of the Request Set.

5.3.3 Non-Additive Scores

An alternative criterion for the score of a reference is 'the maximum of the scores sent to it'. This is appropriate when the user expects that a reference closely related to a high scoring member of the Request Set would not have a higher probability of relevance if it were also distantly related to a low scoring member of the Request Set.

The use of this criterion in conjunction with that of 5.3.1, defines the score of all references in the file. Each reference with a non-zero score derives that score from just one member of the Request Set, and lies on a 'path of score assignment' originating at that member. The scores on this path decrease monotonically from the score of the Request Set reference.

5.3.4 Computation of Network Transmittance

When the Network Transmittance criteria are used for retrieval, the scores of the Request Set are initialised by the user. During the course of the computation these initial values may be altered. Transmittances using the summation criterion can be computed as follows:-

The method proceeds in 'rounds', forming on the Kth round all paths of K (or less) links from members of the Request Set. Unless the computation is carried out for (F-1) rounds, where F is the number of references in the file, then the Transmittances computed are only approximate, but since scores decline monotonically along these paths the computation of a few rounds gives good approximations. At the Kth round:-

(i) Take each K-1 length path P_{K-1} found on the last round,

$$P_{K-1} = a_{K-1}\ a_{K-2}\ \cdots\ a_1 a_0$$

where each a represents a reference, and a_0 represents a member of the Request Set.

(ii) Take all references $a_K^{(i)}$ which have a link from a_{K-1}

(iii) Eliminate any $a_K^{(i)}$ which appears in the set $\{a_{K-1}, a_{K-2} \cdots\ a_1, a_0\}$

(iv) Using the remaining $a_K^{(i)}$ form the K length paths

$$P_K^{(i)} = a_K^{(i)}\ a_{K-1}\ \cdots\ a_1\ a_0$$

and associate with each K length path a score $w_K^{(i)}$ defined by

$$w_K^{(i)} = \frac{w_{K-1}}{L}$$

where w_{K-1} is the score associated with P_{K-1}

L is the number of references with a link from a_{K-1}

and w_o is the initial score assigned to a_o by the user.

(v) When the final round has been completed, say the K^1 round, then the score assigned to a reference is the sum of the scores of all paths terminating at it. By the method of computation only paths of length $\leq K^1$ are included in this sum.

The same procedure will calculate Network Transmittance by the second criteria (non-additive scores) if, in (v), the 'sum' is replaced by the 'maximum'.

5.3.5 Retrieval on a Citation Network

A Citation Network is a naturally occurring Directed Network. The original directions of the links are based on whether reference Q1 cited reference Q2 or was cited by it. It is not obvious that these directions are important for retrieval and for the purposes of Network Retrieval Algorithms, the links will be taken as two-way links.

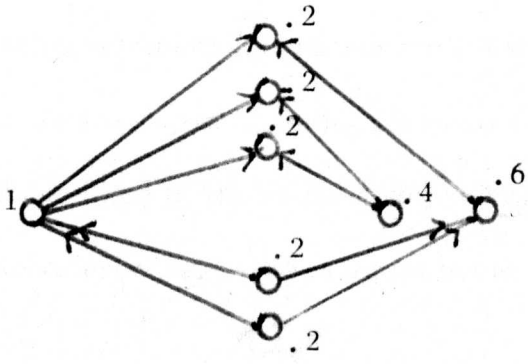
More complex treatments of link directions are possible, and one such treatment, used with the summation criterion for transmittance, gives results similar to Bibliographic Coupling. This will be called Alternating Searching and results from using the computational procedures of 5.3.4, taking the directions of links as:-

(i) from citing reference to cited reference, on first, third and all odd rounds.

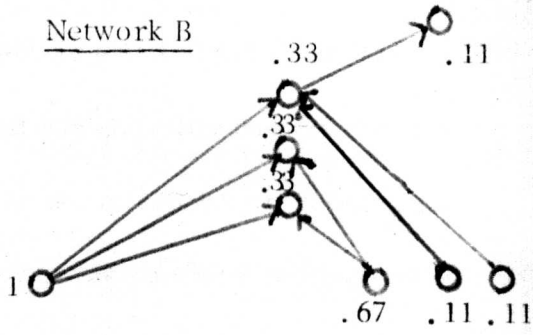
(ii) from cited reference to citing reference on all even rounds.

The diagrams below show the scores assigned by the summation criterion using two-way links, and using alternating links, and also the scores assigned by Bibliographic Coupling. In each case the Request Set consists of the single leftmost reference, and is given a score of 1 by the user. The arrows go from citing reference to cited reference, irrespective of the algorithm used for searching. The values shown are for two rounds of the summation criterion and one round of Bibliographic Coupling.

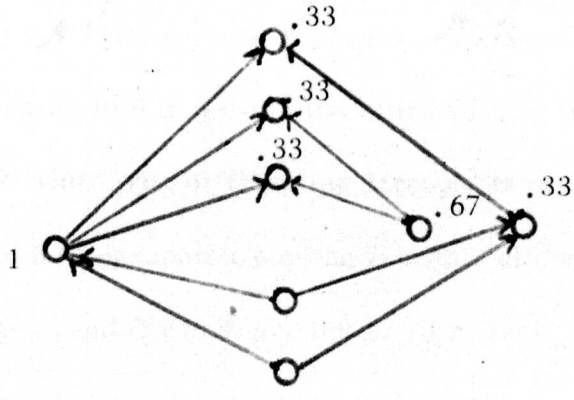
Network A

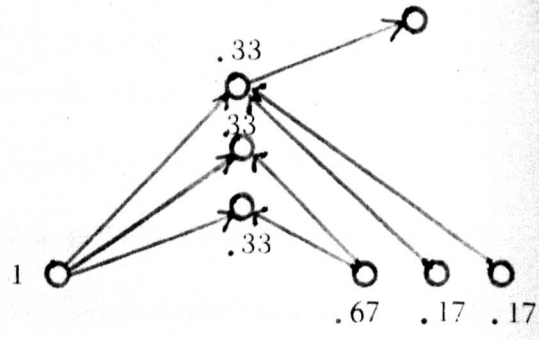


Network B

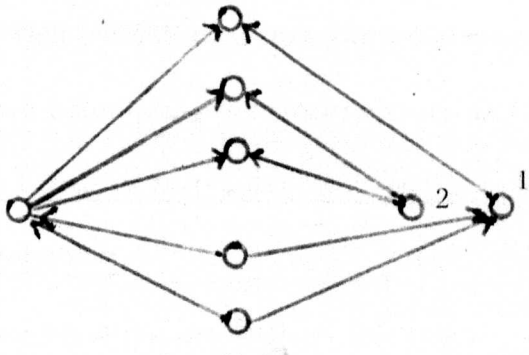


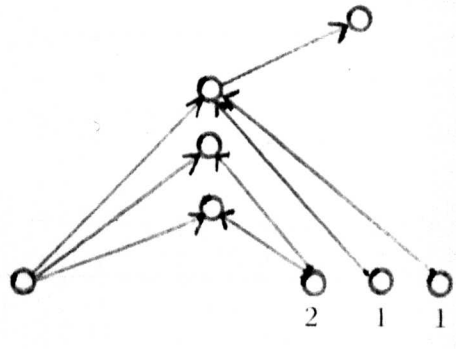
Summation Criterion with two-way links.





Summation Criterion with Alternating links.

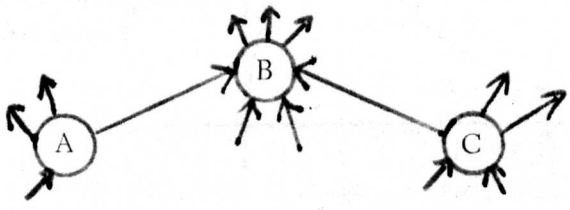




Bibliographic Coupling.

THE WEIGHTS ASSIGNED TO REFERENCES BY THREE RETRIEVAL ALGORITHMS

As can be seen, the method of Bibliographic Coupling gives results similar to those of the summation criterion with Alternating Links. The two differences are that Bibliographic Coupling gives no scores to the intermediate references (i.e. those cited by the request set), and it does not take account of dispersion at these references. The diagram below makes this clear:-

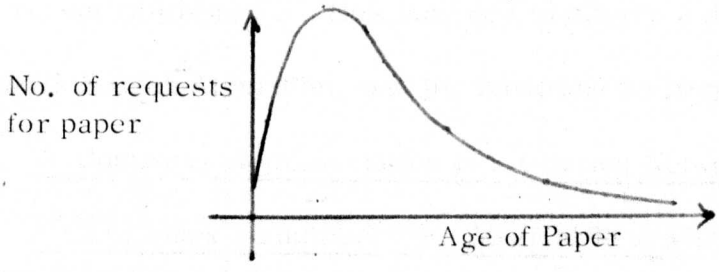


From this diagram, the citing of B by both A and C contributes a value of 1 to the similarity or Coupling Strength between A and C. When a normalized form of Bibliographic Coupling is used, allowance is made for the references cited by A and C which are not cited by both, but no account is taken of other references citing B i.e. Bibliographic Coupling takes account of dispersion at A and C but not at B. The summation criterion takes account of dispersion at A and B when calculating transmittance from A to C, and of dispersion at B and C when calculating transmittance from C to A.

5.4 The Effect of Incompleteness on Associative Searching using Transmittance Algorithms

Any Associative Network of references is necessarily incomplete, if only because next year's publications have not been included, and they may closely link two references in the existing network which are not as yet closely linked. For example, next year ten different papers may each cite a pair of references in the present network which have as yet no path, of any length, linking them. There is no obvious way to estimate which references may be linked by future publications, but the negative effect of future links, the dispersion effect, can be estimated. Most scientific papers have a distribution of use which is high shortly after publication and declines thereafter. Library statistics from the

National Lending Library for Science and Technology, Boston Spa, have been used to estimate the use-distributions of several categories of scientific papers. The distributions of use have the shape of an exponential decay curve as shown below:-



and the 'half-life' of papers in different categories can be estimated from these curves, as the period of time from the publication date to the date by which half of the requests that will ever by made for the paper, have been made [6,7]. For medical literature the half-life is about $3\frac{1}{2}$ years. More generally, the fraction of total use C(x) occurring within x years of publication can be estimated. Assuming that the distribution of citations is of the same form (i.e. equal to the use distribution multiplied by a scale factor), the total number of references linked to (or from) a reference, Q, in a Citation Network can be estimated by:-

$$\text{Total links} = D + G/C\,(f - y)$$

where D is the number of references cited by Q

G is the number of references in the network citing Q

f is the year of the most recent acquisitions represented in the Citation Network

y is the year of publication of Q.

In practice D and G are the numbers of links in the representation of the Citation Network. Since $C(f - y) \lesssim 1$ this estimate scales up to G. It is, however, an underestimate of the total since it takes account of the unpublished references to Q, but not of those which have already been published, but have

not been found and included in the representation.

In the Transmittance Algorithms of 5.3. for a Citation Network the number of links can be modified by the factor $^{1}/C\,(f - y)$ wherever appropriate. If this is not done, the Algorithms are biased in favour of transmittance via more recent references. This may be considered a desirable property of a Network Search Algorithm, and the modification purposely omitted.

5.5 A Computer Representation of a Citation Network based on references from A Card Index maintained by a Medical Research Council Unit.

To test the Network Retrieval Algorithms of this chapter an Associative Network was necessary, and preferably one which was not based on MEDLARS indexing, since the objective was to investigate methods of retrieval which did not require an indexing operation. A small Citation Network was therefore represented on Magnetic Tape for searching using an EELM KDF9 computer.

A simple representation of a Citation Index using a four part bibliographic code could be:-

Code for Journal	Volume No.	Issue No.	Page No.

e.g. CACM00101058

One such entry identifies a reference, and a list of similar codes, identifying the references that it cites, can be attached to it. To use the Algorithms of 5.3 a list of citing references is also required. This can be formed by 'inverting' the original representation, and then merging the original and the inverted representation. Inverted representations, which are also useful for indexed files, consist of lists of references stored under their attributes (index terms, or in this case, cited references).

The disadvantage of this simple representation is that while it is easy to identify journal articles with this code, there is no very satisfactory way of identifying books, research reports, and non-serial literature generally, by means of such a bibliographic code. In order to include such references, the method adopted for the experimental Citation Network was to give each reference a unique identification number, starting with 1, 2, 3..... etc.

The basis of the Citation Network consisted of about 1000 references from one of the card indexes maintained by the M.R.C. Clinical Effects of Radiation Research Unit at the Western General Hospital, Edinburgh. By taking such a compact subject area it was hoped that the inaccuracies caused by the small size of the network would be minimised i.e. that the number of references outside the network, which cited references in the network, would be small compared with other networks based on only 1000 references. The network represented on the KDF 9 included the original thousand references plus all references cited by them. No attempt was made to extend the network backwards in time by including the references cited by these cited references. The representation was formed by the method below, and was a joint project by the author and W.A. Gray [8].

(1) The texts of the references from the card file were found in the National Lending Library for Science and Technology, Boston Spa, and the citations made were microfilmed, and Xerox prints were made from the microfilms.

(2) The citations, and the original references were then coded onto standard sheets, using MEDLARS standard journal codes for all journals indexed by MEDLARS, and using one special code for non-MEDLARS journals, and a second for books or non-serial publications. In addition, MEDLARS journal title abbreviations were recorded for MEDLARS journals, and simple (non-standard) abbreviations of book titles as and when they occurred. The

full information recorded for each reference or citation was:-

Abbreviated Title

Authors

Journal Code (or book code)

Year of Publication

Volume Number

Page Number

These coded sheets were then keypunched onto punched paper tape by staff at Newcastle Computing Laboratory.

(3) Identification numbers were then assigned. The references on the original cards were assigned the numbers 1, 2, 3 ... etc. Thereafter each list of citations was input, and compared (by computer program) with the references already in the file. Those which had similar bibliographic details were classed as 'hits' and printed out for human inspection, the others being automatically added to the file, and taking identification numbers in sequence from the last member of the existing file. A 'hit' with a reference already in the file occurred when:-

(a) the references had the same year of publication

AND { (b 1) Both were non-serials
or
(b2) Both were in the same journal }

AND { (c 1) Both were annonymous
or
(c 2) Both had the same author
or
(c 3) If one had two or more authors, then at least two authors were the same in both references }

To avoid missing references because of small differences in the quotation of authors' names, initials were ignored, since the number of initials quoted even by the author himself, can vary. Authors and journal editors are

perhaps less likely to mis-spell a surname, but it would be very easy for the keypunchers inputting data to MEDLARS or to the Citation File to hit a wrong key. Accordingly the comparison routine allowed a one letter difference between two surnames that it judged the same. No misalignment was permitted however, so that STANFORD was judged the same as STAMFORD but not the same as STAFORD. The method of comparison is discussed in [8].

This procedure of automatically finding 'candidate' matches, and vetting them by inspection before reinput, took a lot of effort but was particularly useful for identifying conference proceedings etc., whose titles were erratically abbreviated, as well as eliminating the effects of incorrect initials and page and volume numbers.

The identification numbers corresponding to each list of citations from one of the original card file were recorded.

(4) When each reference or citation had been assigned an identification number, the file was compared with the MEDLARS file. A program was written to generate from the Citation File, input tapes for the standard MEDLARS Author Search Program. (The Author Search Program was written by E.D. Barraclough of Newcastle Computing Laboratory). These input tapes only included post 1963 references (the MEDLARS file started in 1963), and only those references which had been given a MEDLARS journal code. The Author Search Program performed searches in batches of 50. It permitted initials to differ but surnames had to be exactly matched. The MEDLARS references retrieved by the author searches were stored on magnetic tape but also printed and visually compared with members of the Citation File. Incorrect retrievals were deleted, by program, from the magnetic tape. Once again the human inspection procedure cost a great deal of effort but ensured accuracy.

(5) The representation of the Citation Network as a list of approximately 1000 identification numbers each with a list of cited identification numbers now existed on (paper) code sheets. This was keypunched and inverted by program. The Network representation in original and inverted forms were merged with each other, with the file of bibliographic details, and with the file of MEDLARS retrievals. The resultant representation of the Citation Network consisted of over 10,000 references, each with bibliographic details, lists of cited and citing references, and wherever possible, a list of MEDLARS index terms, category numbers, and a MEDLARS English Language Title. About 4000 of the 10,000 had MEDLARS index terms. The remaining references included those published before the start of MEDLARS in 1963, and papers, reports or books not included in MEDLARS.

This Citation Network is very small compared to the MEDLARS file (at that date, half a million references), but it is relatively large when compared with most files set up for Information Retrieval Experiments, and the compact subject area means that it is not equivalent to a random selection of 10,000 MEDLARS references, but to a much larger randomly selected subset of MEDLARS.

5.6 Quantities of Computation required for the construction of the Network Representation

The number of comparisons of bibliographic details needed for the construction of a Citation Network Representation by the method used by the author was large, and could be prohibitive for all but small experimental files, and the simple bibliographic code method outlined at the beginning of 5.5 would be better if minimisation of cost were more important than the inclusion of non-serial publications.

The file began with the cards from the M.R.C. card index. Let this initial size of file be I. The lists of citations given in these references were then compared with the file, and non-hit references added to the end. Thus the jth of these cited references was compared with

$$I + x_j$$

references where x_j references had been added to the file. The total number of comparisons required to assign identification numbers is

$$\sum_{j=1}^{j=J} (I + x_j)$$

where J references are cited by the original file and x_j is such that

$$x_{j+1} \geq x_j$$

and $$0 \leq x_j \leq F-I \text{ (with both limits attained)}$$

where F is the final size of the file. It can be seen that x_j is monotonic but not strictly monotonic. If the rise in x_j is evenly spread over the j values, then the total number of comparisons is approximately

$$J \cdot \frac{F + I}{2}$$

If V is the average number of citations made by the original references, then the total is:

$$VI\,(F + I)/2$$

and this is more (probably much more) than:-

$$VI^2$$

The process of assigning the identification numbers requires a number of bibliographic comparisons which varies as the square of the original file size. In the work reported here, I = 1000, F = 10,000 (approx.) and VI (F + I)/2 = 5,500,000 V comparisons were necessary. This calculation

ignores the problem of the 'hits' or 'near hits' which in addition to the comparisons counted above, had to be compared visually, and then, if not genuine 'hits', had to be reinput and compared with each other (by machine) for 'hits' and 'near hits'. The iteration of this process was carried no further.

The computation required for the inversion of the file of cited lists is required irrespective of the reference-identification system, unless the search techniques used are to be restricted to variants of Bibliographic Coupling.

The very large amount of MEDLARS searching, undertaken to provide index terms for testing purposes, would not generally be required.

5.7 A KDF9 program to search the Citation Network

To facilitate comparison of Network Search with Index Term searching, the magnetic tape containing the Citation Network was written in a form similar to the standard MEDLARS file tapes. The standard MEDLARS system was implemented in the U.K. by E.D. Barraclough on an EELM KDF9 computer. For the quantities of data to be handled this machine was slow, and had a small high-speed store, and machine code programing was necessary. Standard EELM Magnetic Tape Handling routines were used.

The program to perform the Network Searching was written in KDF9 machine code, partly to make it compatible with the MEDLARS-format magnetic tape, and partly because of the problem of large-scale data handling. The peripherals available did not include any semi-random access store, and the high-speed random access store was small - only 16K 48-bit words. The KDF9 was thus not ideally suited to Network Searches on large networks. A useful facility that it did have was the provision, in the EELM suite of Magnetic Tape Handling Routines, of efficient routines for reading magnetic tapes backwards as well as forwards.

Program Input

(1) The Citation Network held on magnetic tape in a format similar to the MEDLARS file format, but with the addition to each entry of two lists of identification numbers, being the lists of cited and citing references.

(2) A paper tape containing a batch of search formulations. Each formulation consisting of a list of identification numbers, each with an initial value for its Score (these Scores being integers).

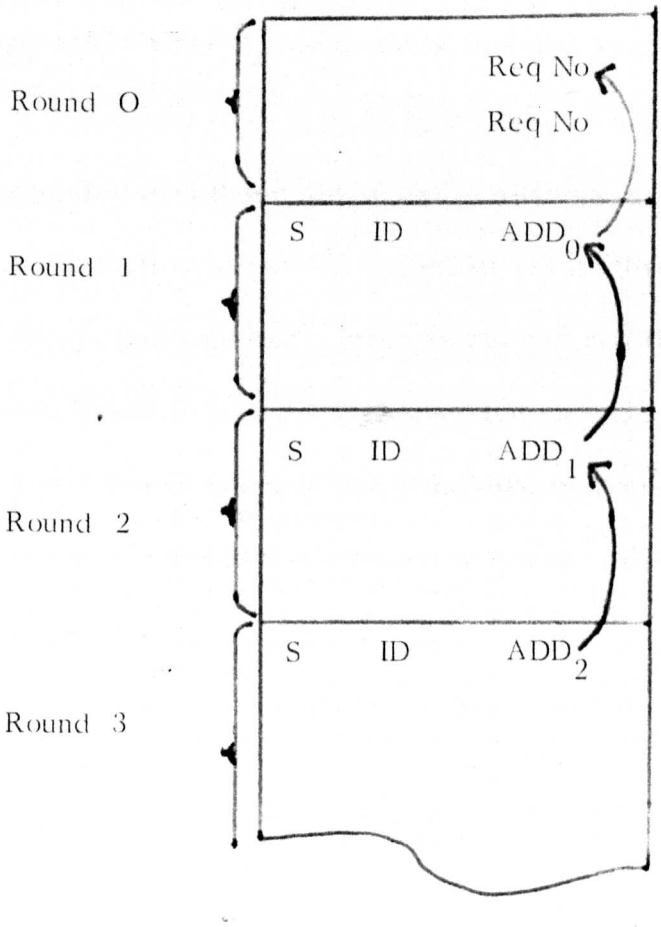
(3) On paper tape, accompanying the search formulations two numbers indicating:-

(a) Two way link/Alternating Searching (see 5.3.5) and

(b) Summation/Single Path Algorithm (see 5.3.2, 5.3.3)

Layout of the KDF9 high-speed store

Almost all the available storage is devoted to a single list of the form:-



Req. No: a Request Number identifying a particular search formulation.

ID: Identification Number of a reference.

S: Score of that reference.

ADD_0: the Address of the appropriate Request Number

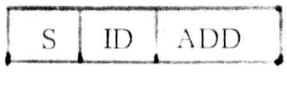
ADD_1: the Address of the first reference on a path.

ADD_2: the Address of the second reference on a path

The references in Round 1 are those provided by the users as input, and the scores are those assigned by them. The references in round N are those which are linked to a member of a Request Set by a path of (N - 1) links. The scores of references in the Nth round are calculated from those in the (N - 1)th round by the method given in 5.3.4. The decision as to which, and how many, references are linked to Q depends upon whether the search mode is Two-way Links or Alternating. Each round corresponds to one pass of the magnetic tape, but since this tape is read in both directions, no rewind time is required.

The initial values of Relevance Scores of the Request Set must be integers and are therefore greater than 1.0. References whose calculated score is less than a threshold value are not included in the list in store. This threshhold is applied to prevent the limited high-speed store being exhausted too quickly. When the program is used for Single Path Searching the effect of this threshhold is such that any references not retrieved are certain to have lower scores than those which are. When the program is being used to search by the Summation Algorithm the effect of this threshhold is not so predictable. It is always possible that a large number of references with very small scores might be linked to a single reference (on some later round) which could then have a non-trivial score. But if the threshhold is not imposed the store is quickly exhausted and this effectively restricts the batch size and/or the number of rounds.

When the required number of rounds has been completed the typical list element is:-

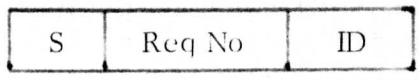


S = score

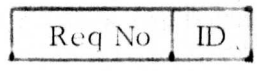
ID = reference identification no.

ADD = address of previous reference on path from Request Set.

This typical entry is replaced by:-



Req No: Search formulation no or 'Request no.'

and the list is sorted using 

as a key. For each request the program then checks how many entries there are for each ID, and if there are more than one the scores of all but one are set to zero, and the score of the remaining entry changed to the maximum of the scores (Single Path Algorithm), or sum of the scores (Summation Algorithm), of entries with that ID. The entries are then sorted by score, and the identification numbers and scores of the N highest scoring references are punched on paper tape, where N is the maximum output size acceptable to the user. References with zero scores are never retrieved and the number of identification numbers punched can thus be less than N.

The punched paper tapes produced by this program are in the same format as those produced by the programs described in Chapter 2 and Chapter 4, and serve as input to the same suite of merging, sorting and printing programs. In particular the standard MEDLARS print program is used to present the retrieved references in an easily readable format.

5.8 A Comparison of Scoring Searching using MEDLARS Index Terms with Citation Network Searching

The Citation Network was based on references from a very specialised card index. The card index was only one of a number of indexes maintained by the M.R.C. Clinical Effects of Radiation Research Unit, and did not cover the interests of even this highly specialised Unit. A test based on genuine information needs (as in Chapters 1, 2 and 4) was not possible, and questions were solicited from members of the Unit. By the time of the test the Unit had changed its name to the M.R.C. Population Cytogenetics Unit, and some changeover of staff had occurred, but five members of the Unit agreed to cooperate and each produced

five requests making 25 requests in all. The requests each consisted of three MEDLARS indexed references fromthe Citation Network plus a single sentence description of the search topic. Such brief search specifications would not normally be acceptable as a basis for formulating MEDLARS Searches. The brevity of the search descriptions does not mean that they were vague. On the contrary, most were quite specific. For example,

Search No 3: Chromosome Studies of Clonal Development in the Acute Leukemias

Search No 7: Mechanism of Action of Phytohaemagglutinin on Lymphocytes

A major difficulty in turning such specifications, even when accompanied by specimen references, into MEDLARS Search Formulations is that the search writer does not know just how specific a formulation is required by the user. When the user has a genuine information need he can be asked to elucidate the degree of specificity required, but not when the search has only been produced for a retrieval test. Network Searching was therefore compared with the Probabilistic Search of Chapter 2 since it was then not necessary to decide on the exact degree of specificity required. In the test reported in Chapter 2 the Probabilistic Search technique gave results similar to, and slightly better, than the Standard MEDLARS Search, when operating at the same output size

Each of the 25 searches was performed using (i) the Probabilistic Search, (ii) the Network Search with additive scores and two-way links, and (iii) the Network Search with additive scores and alternating links. This meant that each user had to evaluate the output from 15 searches, and in consequence that the output sizes had to be small. In all cases the output size was set at 13. Results quoted below are for output sizes of 10. The higher outputs in the programs were set to allow for the exclusion of Request Set references. The Probabilistic Search could only be over the MEDLARS indexed references, 3914 in all. The Network

Searches used all references and links in the file, for the transmission of scores, but were modified at the output stage, to punch not the 13 highest scoring references but the 13 highest scoring MEDLARS references. These 13 always included the Request Set. To ensure full comparability the Request Set references are excluded from the results quoted for the Network Searches, and the results given for the Probabilistic Searches count as retrieved only the ten highest scoring non-Request Set references.

Search Formulations

The search formulations for the Network Searches consisted of the Request Sets selected by users. The formulations of the Probabilistic Searches were constructed by a subject-expert using MeSH and the printed version of the MEDLARS tree structure. All Request Set references were printed in the lists of retrievals sent to users, and were evaluated with the retrievals. The users were aware at the time they formulated their searches that the additive-scores or summation, algorithm would be used for retrieval, and therefore selected sets of 3 references which together defined a topic. Any individual member of a Request Set could therefore be irrelevant to the request. The table below displays the users evaluations of Request Set references against retrieval by the Probabilistic Search (at output size = 13).

Request Set References

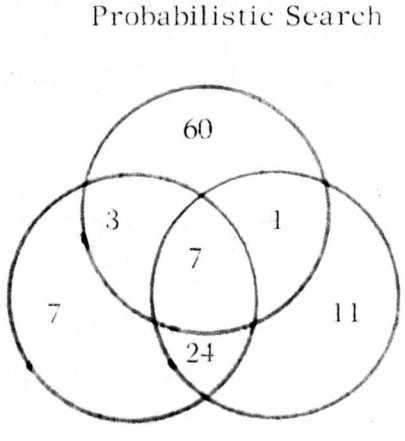
Evaluated by users as:	Retrieved by Probabilistic Search	Not retrieved by Probabilistic Search	
Relevant	28	39	67
Not relevant	0	8	8
	28	47	75

The Consistency Ratio for these searches is thus $^{28}/_{67}$ i.e. 42%. This compares with an Average Consistency Ratio of between 50% and 60% for the Standard Medlars Searches quoted in Chapter 1.

Comparative Retrieval Performance (Output size = 10).

Since the adjusted output sizes are the same for all three search techniques, and no Request Set references are included in any of the adjusted outputs the results can be given in a diagram of three overlapping circles representing the relevant references retrieved by each of the three search methods. The results were:-

Numbers of Relevant References Retrieved



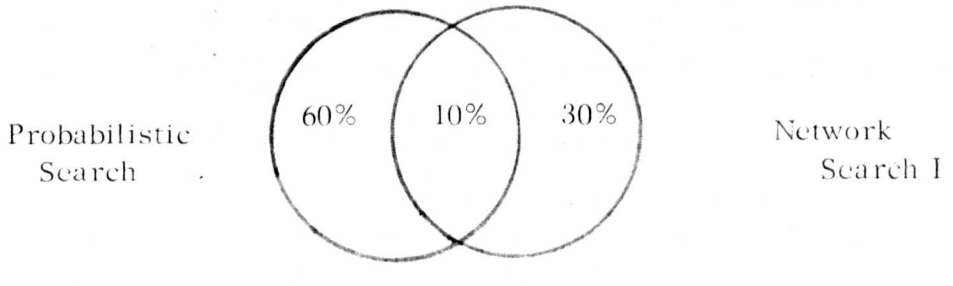
From the diagram it can be seen that the Probabilistic Search retrieved $^{71}/_{113}$ relevant references (63%), and the Network Searches $^{41}/_{113}$ and $^{43}/_{113}$ (36% and 38%). Thus, on the basis of numbers of relevant references retrieved, the Probabilistic Search technique performed considerably better than the Network Searches, and the two Network Searches were about equal.

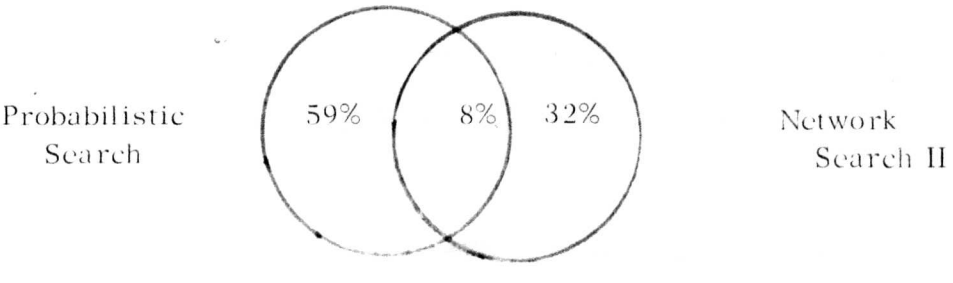
Out of the 25 searches the Probabilistic Search retrieved strictly more relevant references than Network Search I in 14 searches, and less in 7 searches (equal in 4). It retrieved more relevant references than Network Search II in

14 searches, and less in 8 searches (equal in 3). The hypothesis that the Probabilistic Search is best is thus significant at the 7% level. The Network Search II retrieved more relevant references than Network Search I in 4 searches, less in 3, and the same number in 18 searches. It is thus not possible to conclude that one Network Search is better than the other on the basis of this test.

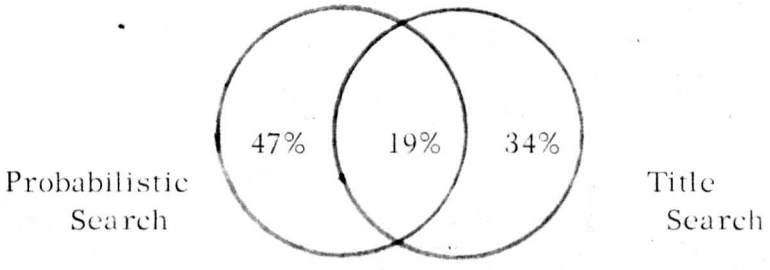
The 25 searches can be divided up according as the Probabilistic Search retrieved 0, 1, or 2 of the Request Set, i.e. achieved Consistency Ratios of 0%, 33% or 67% (none achieved 100%). When Consistency was 0% the Probabilistic Search was best on 3 out of 5 (1 equal) occasions, when Consistency was 33% it was best 3 times out of 8 (2 equal), and when Consistency was 67% it was best 8 times out of 9. These results do not imply a simple relationship between Consistency and superiority of performance, but it is clear that the Network Searches did better relative to Index searches where vocabulary failings or other problems of search specification caused the Index searches to miss known relevant references.

For each of the Network Searches a direct comparison with the Probabilistic Search can be made. Each diagram below shows the number of relevant references retrieved as a percent of the number retrieved by the two search methods being compared.





The equivalent diagram, given in Chapter 4, for a comparison of Probabilistic and Title searching is:-



The Network Searches, like the Title Searches both use the data available with the printed reference, and so represent alternatives to indexing. The overlap between Probabilistic and Title search relevant retrievals is higher than for the Probabilistic and Network searches. This is as expected since a Title search uses a form of author assigned indexing (i.e. titles) and is not so different from the Probabilistic techniques as are the Network searches. Although the results of this Chapter are based on a small number of specially solicited questions, searched over a small experimental file, the Title Search did noticably better against the Probabilistic Search than did the Network searches, retrieving 53% of relevant references rather than 40%. Since the very large expenditure of computer time and human effort needed to construct the Citation Network did not result in better retrieval performance than the less expensive alternative to indexing (i.e. titles), the test suggests that Network Searching on a Citation Network is only justified if a much less expensive representation of the Network can be produced. The system of bibliographic codes

Journal code	Volume no.	Issue no.	Page no.

described in 5.5, in spite of the disadvantages listed there, is such an inexpensive representation. Out of the 25 searches a Network Search was best in 8, and in

these 8 the totals of relevant references retrieved by the three methods were:-

Probabilistic	11
Network I	24
Network II	27

These figures show the value of retrieval techniques which, although inferior to index-term searching, can provide alternative strategies of retrieval where, e.g. the indexing vocabulary is weak, or the search formulation proves difficult.

References from Chapter 5

1. "Medical Subject Headings : MeSH", Nat. Lib. Med., Washington, annually.

2. Kessler, M.M., Amer. Doc., 16, 1965, 223.

3. Cleverdon, C.W., Keen, M. "Factors determining the performance of indexing systems", Cranfield College of Aeronautics, 1966.

4. Mason, S.J., Proc. I.R.E., 41, 1953,1144.

5. Stinchcombe, A.L. "Constructing Social Theories", Harcourt, Brace & World, Inc., N.Y., 1968.

6. Harley, A.J., "Information Retrieval Lectures, 1967", Newcastle University, unpublished.

7. Wood, D.N., Bower, C.A., Bull. Med. Lib. Assoc., 57, 1969, 47.

8. Gray, W.A., Ph.D. Thesis, Newcastle University.

CHAPTER 6

Single and Multiple Retrieval Strategies

6.1 Single Strategies

6.2 Multiple Search Strategies

6.2.1 Disjunctive Strategies

6.2.1 Conjunctive Strategies

Chapter 6: Single and Multiple Retrieval Strategies

6.1 Single Strategies: In the preceding chapters the service given to MEDLARS users is evaluated, and several alternative retrieval techniques tested. The usual method of evaluating retrieval performance by estimating Recall and Precision Ratios is not used, since there is no possibility of obtaining a good estimate of Recall. It is obvious, however, from the existence of relevant references known to users yet not retrieved by MEDLARS, that in the majority of searches MEDLARS certainly does not achieve 100% Recall. To many users this is a serious fault, since performance at anything less than 100% Recall does not give them the assurance that there is no need for a human literature search, and they are obliged to scan the literature as before. The value of MEDLARS does not lie in the elimination of the users' own scanning of the literature. Its value is that in supplementing the users own efforts it provides many relevant references that have been missed by the users. The effect of MEDLARS is thus to increase literature awareness, rather than to eliminate the need for human searching. A measure of this effect the Extension Ratio, is proposed in Chapter 1. The performance of MEDLARS was evaluated using over 300 searches, which were normal searches and not specially solicited for the test.

The results show that the numbers of relevant references known to users are very much increased by MEDLARS searches. Over the main range of searches (between 14 and 65 references known by user before search) MEDLARS doubles or triples the number of relevant references known to users. When the users know less than 13 relevant references before the search, the multiplying effect of MEDLARS is much greater. Above 65 references known before there are very few data points, but the decline is slow e.g. a factor of 1.91 was achieved when 164 references were known. The largest number known before was

325 and the multiplying factor was 1.09.

The alternative techniques were evaluated in tests over relatively few searches. An alternative search technique, which uses the same indexing as the standard MEDLARS search, and one which uses the English-language titles of references, were tested using 25 MEDLARS searches. In this test each search was over 6 magnetic tapes, each of 35,000 references, not the full MEDLARS file. The performance of each search over each tape is recorded separately. Both of these alternative search techniques, the Probabilistic Search, and the Title Search, assign a score to each reference inspected, and retrieve the N best-scoring references. It is thus possible to predetermine the output size of these searches. When the output size is the same as that of the standard MEDLARS search (Boolean Search), the Probabilistic Search retrieves the largest number of relevant references, and the Title Search the smallest. As the output sizes of the alternatives are extended beyond the output size of the standard search, the pattern changes, the Probabilistic Search still retrieving the largest number of relevant references, but the standard search the smallest. On the basis of the limited test described, the Probabilistic Search appears an improvement on the standard MEDLARS search. An estimate of the relative cost of the Probabilistic Search depends upon the cost of computer time, clerical workers' time and indexers' time. In the test of retrieval efficiency equal numbers of search terms and categories were used in Boolean and Probabilistic Searches. For 25 Search Formulations the Probabilistic Search took four times as long on the KDF9 i.e. about 1 minute per thousand references searched, instead of about $^1/_4$ minute per thousand. The use of a 'binary chop' instead of a 'pop-up' comparison technique (see Chapter 2) would reduce the difference considerably. The cost of printing the retrieved references is the same for each method, as are the costs of indexing the file.

The costs of specifying the searches cannot be determined from the test since the Boolean Formulations were used as the basis for both Boolean and Probabilistic specifications, but an interview to determine the users' information needs and to examine the vocabulary for suitable terms should not a priori take longer for one method than for the other. The total costs are thus very similar. The criterion of retrieving the N highest scoring references, rather than applying a threshhold score, does raise difficulties where a very large output is required. One user of the MEDLARS Monthly Selection Service specified a Boolean search which retrieved 2000 out of the 35,000 references on one tape, i.e. almost 6% of the file. He expressed satisfaction at the volume of output and confirmed the search specification for next month. Such an output size implies that a reference's score, once calculated might have to be compared with 2000 others before the next reference could be considered. This would take an excessive amount of computer time. For such large outputs the Scoring technique could only be used in conjunction with a threshold.

The Title Search performance is not sufficiently good for general use, and it is relatively expensive in computer time (~2 min/1000 but half of this spent reformatting titles). . Where suitable MEDLARS index terms are not available it is an efficient alternative to the standard search. It does provide a level of performance which is comparable with the standard MEDLARS system, without the expense of indexing, but all titles have to be translated into a single standard language.

Two search techniques which use a Citation Network were tested using a specially constructed Citation Network which had 3914 references in common with MEDLARS. For this test the questions were specially solicited from research workers with interests in the subject matter of this experimental file.

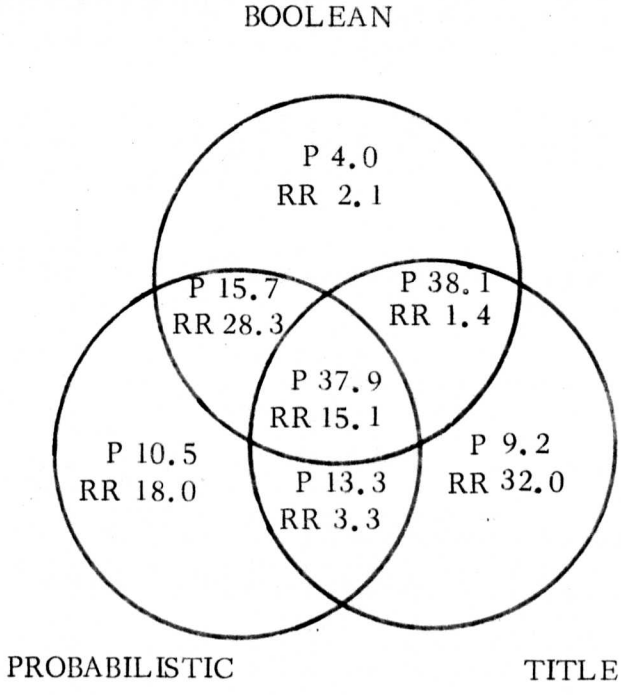
The Network Searches were compared with the Probabilistic Search. Their performance is not sufficiently good for general use, but like the Title Search they provide a useful alternative or supplement to the index term searches. An inexpensive representation of the network is necessary since Title searching already provides a relatively simple,effective, and inexpensive supplement to index term searches.

6.2 Multiple Search Strategies

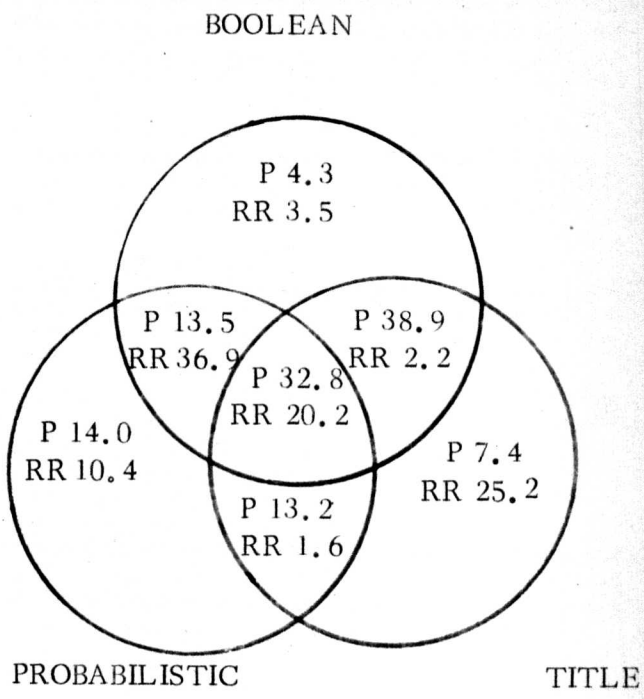
In addition to the individual search techniques considered in 6.1, search strategies using combinations of two or more individual methods are possible and may give retrieval performances which are in some respects better than the performances of individual methods. For example, a strategy which retrieves all references retrieved by either of two individual methods cannot fail to give better Recall than either of them. The diagrams below show the Precision percentage, and the Relative Recall percentage for various strategies. Relative Recall is the percentage, of all the relevant references retrieved, that are retrieved by a particular strategy. It is an overestimate of true Recall and values are only comparable within the one diagram.

6.2.1 Disjunctive Strategies

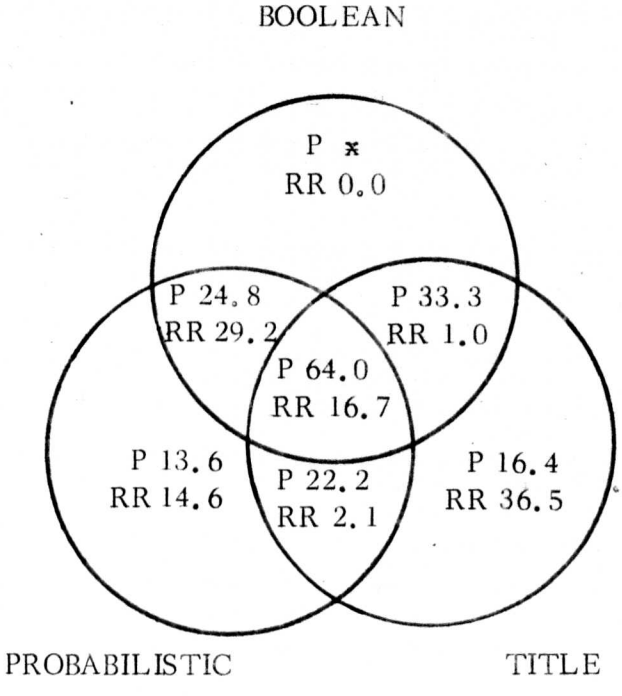
The set of four diagrams below show every disjunctive combination of Boolean, Probabilistic, and Title Searching. The figures given in an area overlapped by two circles are the performance figures for a strategy which retrieves only those references retrieved by both the individual techniques but not by the third. Figures in an area covered by one circle are for a strategy which retrieves only the references retrieved by one individual search, which were not also retrieved by other searches. The diagrams show results for All Searches, Equal Output, Boolean Output Smaller, and Boolean Output Much Smaller, these being the same divisions as in 2.9 and 4.6. (RR = Relative Recall, P = Precision)



I ALL SEARCHES

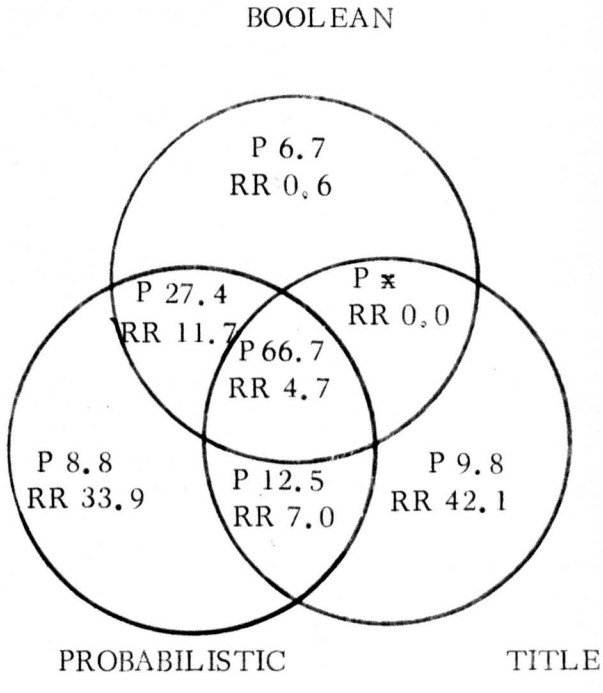


II EQUAL OUTPUT



(*, no retrievals, Precision not defined)

III BOOLEAN OUTPUT SMALLER



(* no retrievals, Precision not defined)

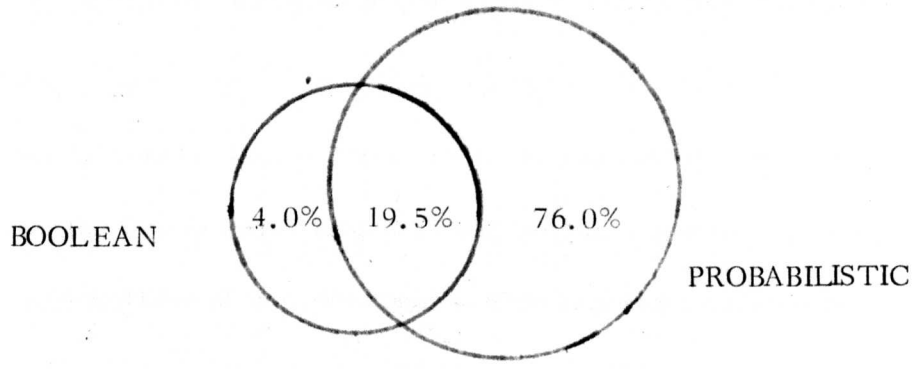
IV BOOLEAN OUTPUT MUCH SMALLER

DISJUNCTIVE COMBINATIONS OF RETRIEVAL STRATEGIES

The ability of the two Scoring Searches to automatically widen a search has been discussed in Chapters 2 and 4. To examine the performance of multiple search strategies, unbiased by this widening effect, the most useful diagram is the Equal Output one, diagram II.

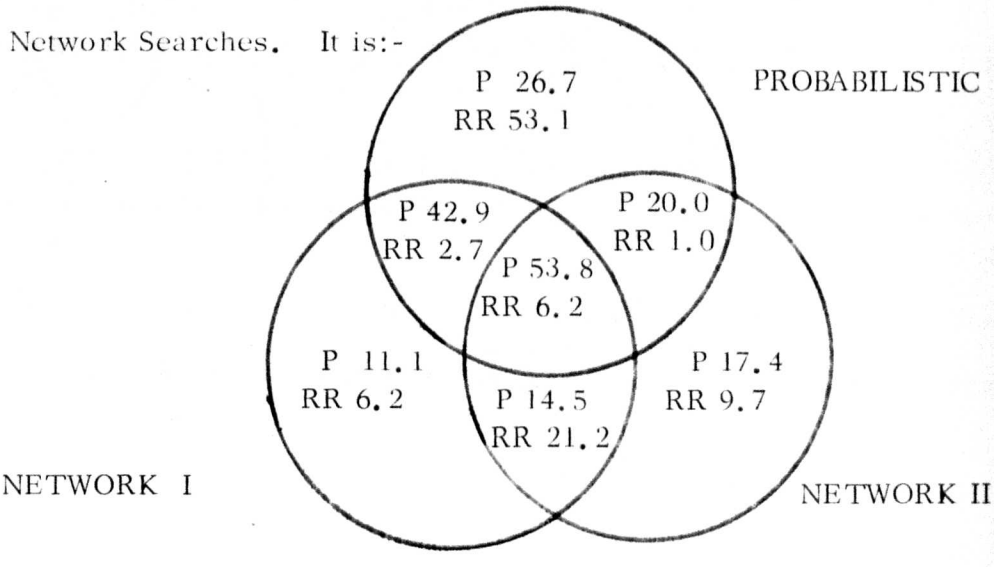
Multiple search strategies, which retrieve references only when two (or more) individual techniques would retrieve them, are used in attempts to increase Precision without unacceptably large decreases in Recall. From the diagram it is clear that the only areas of especially high Precision are the areas of overlap of 'Boolean and Title', and of 'Boolean, Probabilistic and Title', i.e. references retrieved by both Title and Boolean Searches are especially relevant, irrespective of whether they are also retrieved by the Probabilistic Search. However, although the frequency of relevant references in this area of overlap is over twice the frequency in any other area, only 22.4% of the relevant references retrieved are included in the overlap. The cause of the high Precision in areas overlapped by Boolean and Title searches may be that they have very different criteria for retrieval, whereas the Boolean and Probabilistic Searches when operating at equal output sizes have relatively similar criteria for retrieval. Of a total of 1334 references retrieved by each technique, 1061 are retrieved by both Boolean and Probabilistic, but only 213 by both Boolean and Title searches. Thus the strategy of retrieval by both 'Boolean and Probabilistic' is not very different from a single strategy search. While the Probabilistic search uses the same materials as the Boolean, namely index terms, it has a similar logic to the Title search, and the 'Probabilistic and Title' overlap does not have the high Precision of the 'Boolean and Title'.

In diagrams III and IV the area of high Precision includes the same areas as in II. but also includes the 'Boolean and Probabilistic' area. When the Probabilistic output is larger than the Boolean, the pattern of retrieval is:-



The Boolean retrievals are then approximately a subset of the Probabilistic retrievals and diagrams III and IV show that references in this subset are more frequently relevant than in the rest of the Probabilistic retrievals. When the output size of the Probabilistic search is larger than the Boolean, they do have different criteria for retrieval, and the nature of the difference is that the Probabilistic search uses a wider, less specific, criterion, resulting in a lower Precision rate. The different pattern of high Precision in diagrams III and IV compared with diagram II is thus the result of the output size effect operating in III and IV but not in II.

A diagram similar to diagrams I to IV can be drawn for the Probabilistic Search and the two Network Searches. It is:-



V DISJUNCTIVE STRATEGIES USING PROBABILISTIC AND NETWORK SEARCHES

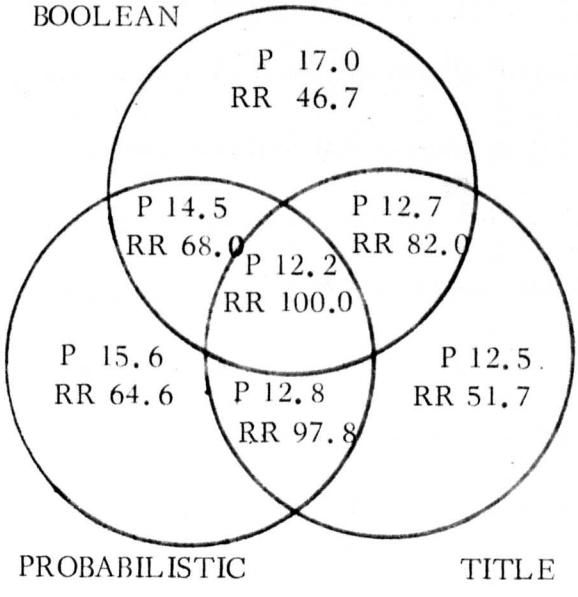
From this diagram it can be seen that the Network searches, being similar, have a Precision in the overlap which is not particularly high. The combination of Index searching and Network searching does give a large improvement in Precision. As before the Relative Recall in the areas of high Precision is low - about 9%. This is still lower than in diagram II, and is due to the wide difference in composition of the Index term and Network search outputs.

Because of the low Relative Recall ratios for the areas of high Precision, disjunctive combinations of the individual search techniques tested do not provide practical retrieval methods.

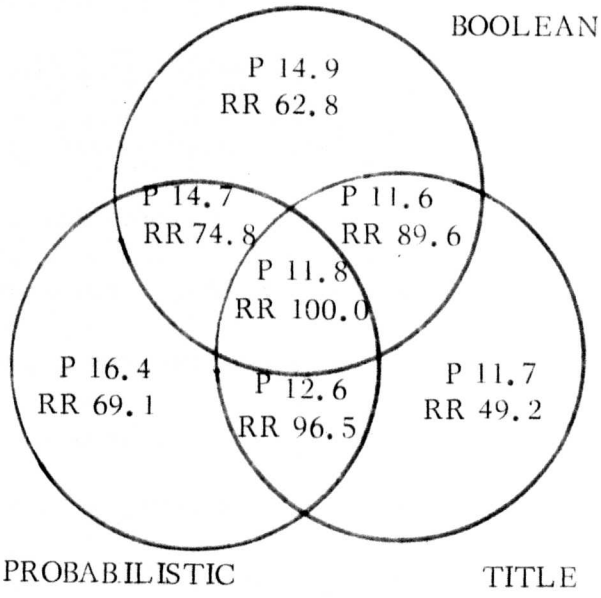
6.2.2 Conjunctive Strategies

The set of four diagrams below show every conjunctive cominbation of Boolean, Probabilistic and Title searching, i.e. the figures given in an area overlapped by two circles are the performance figures for a strategy which retrieves all references retrieved by either of the individual techniques, and perhaps by others also. The diagrams are similar to those of 6.2.1.

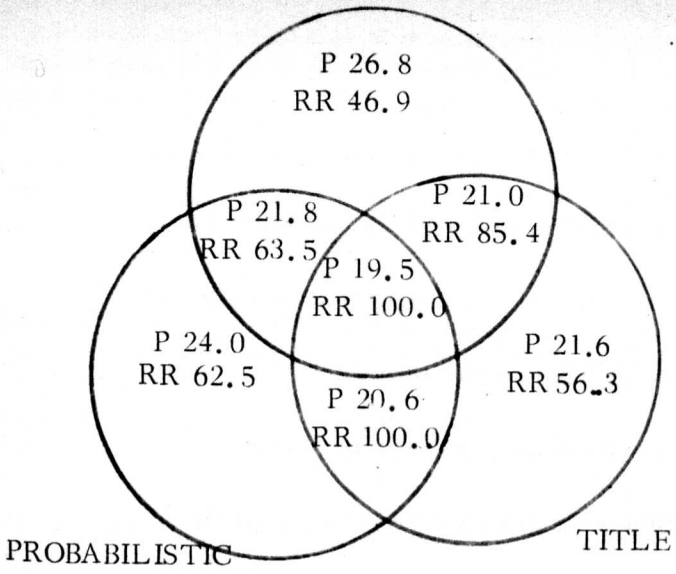
CONJUNCTIVE COMBINATIONS OF RETRIEVAL STRATEGIES



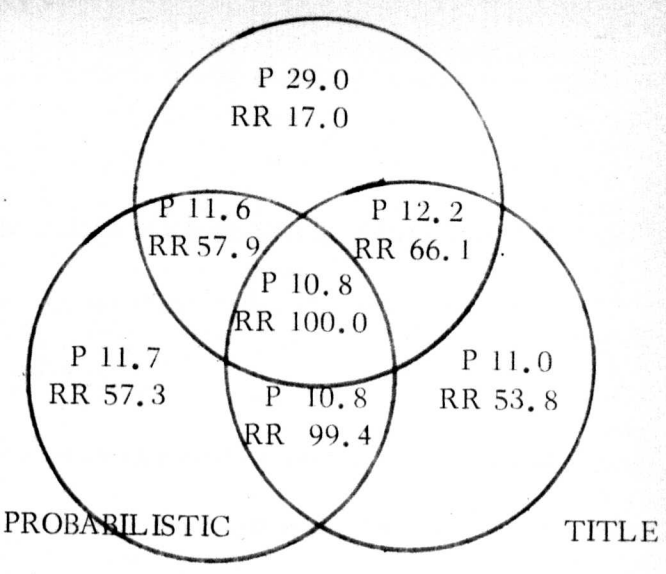
I ALL SEARCHES



II EQUAL OUTPUT



III BOOLEAN OUTPUT SMALLER



IV BOOLEAN OUTPUT MUCH SMALLER

As before, diagram II, Equal Output, is the most useful for an examination of the performance of multiple strategies unbiased by the automatic widening capabilities of the two Scoring searches. Multiple search strategies which retrieve all references retrieved by any ofthe individual strategies are used in attempts to improve Recall. Since Recall cannot be measured, the diagrams show Relative Recall, and consequently a 100% Relative Recall figure appears at the centre of each diagram. Were true Recall figures given the highest figure might not be 100%, and the differences between figures for different strategies would then be less than the differences between figures in the diagrams. The ratio of Recall figures for two different strategies would, however, be the same as the ratio of the two Relative Recall figures shown.

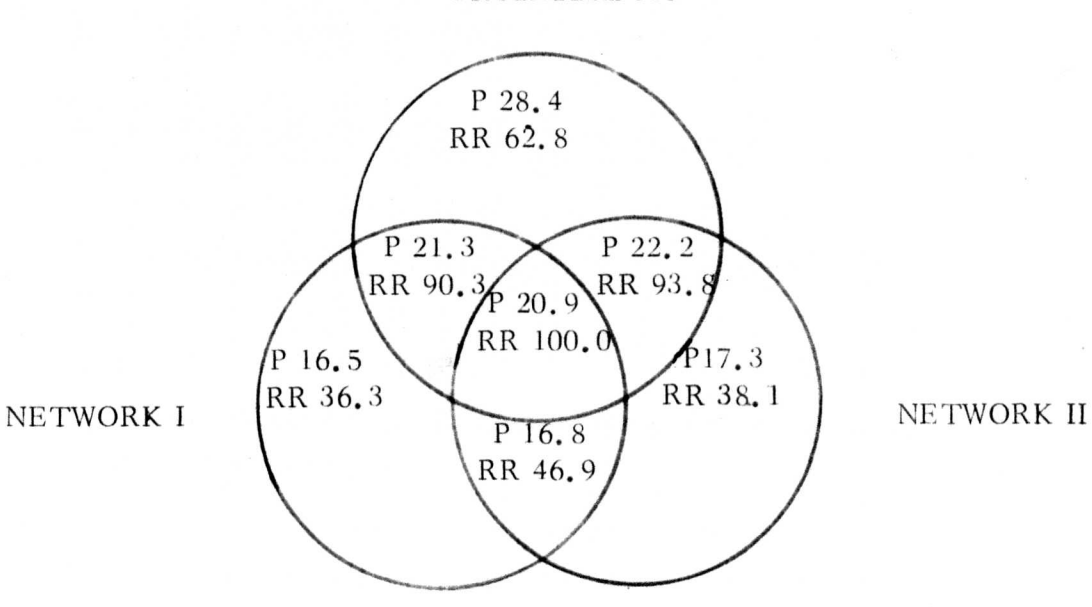
From diagram II it can be seen that the Probabilistic search retrieves most relevant references, 69.1% of all relevant references retrieved. The combination 'Probabilistic or Title' retrieves 96.5% however, and the fall in Precision from 16.4% to 12.6% which accompanies the rise in Recall is not such as to significantly alter the amount of work required of the user.

Diagrams III and IV show that when the Probabilistic and Title searches operate at larger than Boolean output sizes, the Relative Recall of the Probabilistic search falls to under 60%, but the combination of 'Probabilistic or Title' retrieves over 99%, and the losses in Precision from 24.0 to 20.6 and from 11.7 to 10.8 are

even less noticeable than in the Equal Output case. The use of the strategy 'Probabilistic or Title' to achieve high Recall is thus justified, and there is no need to perform searches by all three techniques unless the omission of even a few relevant references is unacceptable.

The use of the 'Probabilistic or Title' strategy rather than a combination of all three search techniques would not save the user much effort, since the double strategy retrieves 2435 out of a total of 2690 retrieved by all three searches at Equal Output, and 466 of 493, and 1570 of 1585 at unequal output sizes. This is reflected in the closeness of the Precision figures for the 'Probabilistic or Title' and 'Probabilistic or Title or Boolean' strategies. The economy of the double strategy is in computer time.

The diagram of conjunctive combinations of Probabilistic and Network searches is:-



V CONJUNCTIVE STRATEGIES USING PROBABILISTIC AND NETWORK SEARCHES

The diagram shows that the Probabilistic Search is best with the highest Relative Recall (63%) and Precision (28%), but that for a drop in Precision from 28% to 22%, the Relative Recall can be increased from 63% to 94% (and the true Recall increased in the same ratio). The use of a combination of Index term and Network searching is thus justified if high Recall is required.

APPENDICES

The following five appendices give details of the three non-standard search programs used. They make use of a number of standard sub-routines. These are:-

P2800	Read Character
P2801	Read Number
P2802	Print Character
P2803	Print Number
L34	Mag Tape Forwards Read - Normal File
L35	Mag Tape Forwards Read - Inverted File
L36	Mag Tape Backwards Read - Inverted File
L78	Check Mag Tape 16 Word Label
L79	Mag Tape Parity Fail
L82	Internal One Word Sort
L89	Type Message

These routines are not listed.

The programs are written in EELM KDF9 User Code, a mnemonic machine code whose arithmetic and logical operations take place in a 16 word push-down stack known as the Nesting Store.

Subroutines P118 and P109 are taken from the standard MEDLARS system and are listed for completeness only.

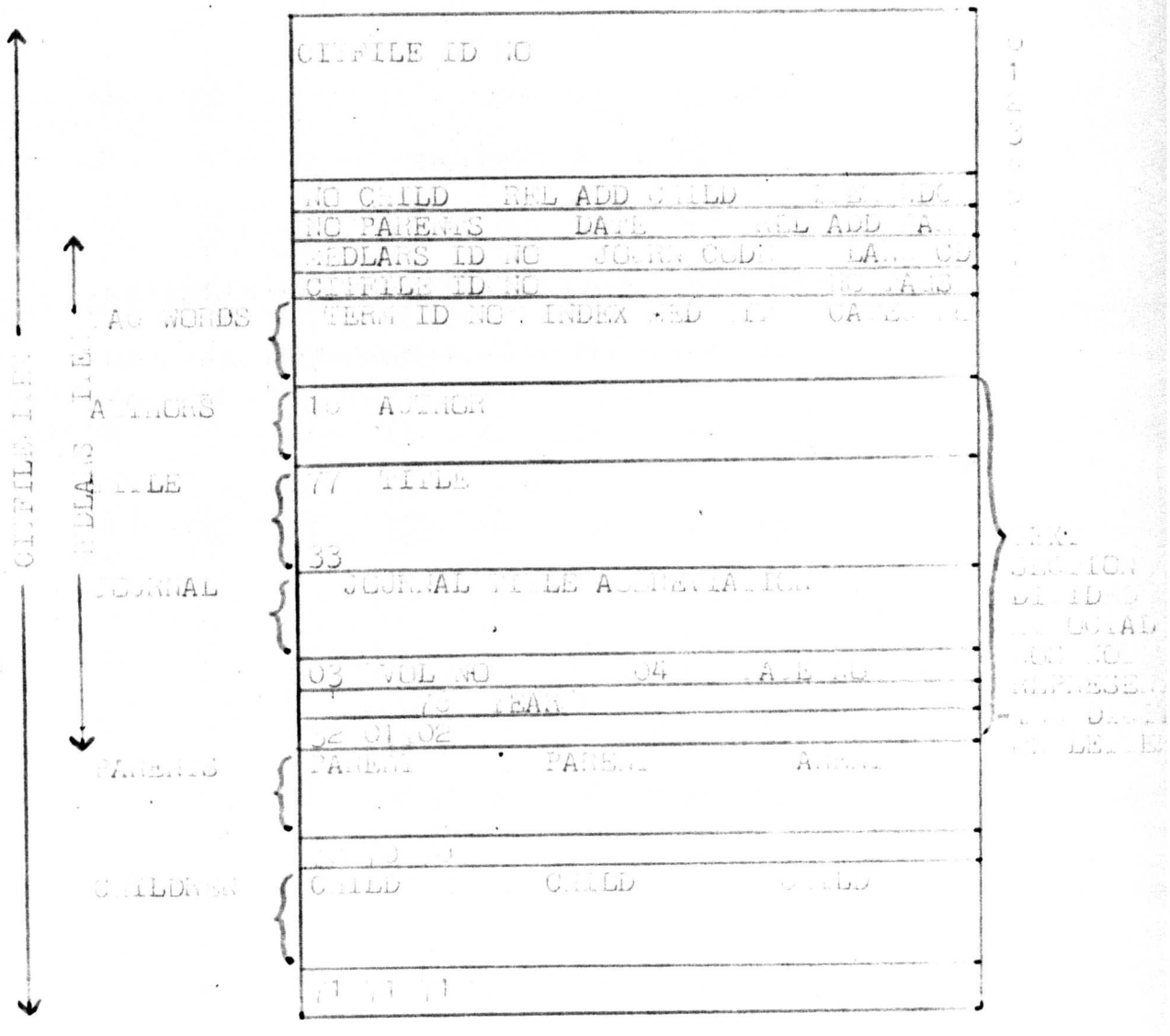
APPENDIX I

FORMAT OF THE MEDLARS FILE AND THE CITATION FILE

APPENDIX I

Format of the MEDLARS file and of the Citation File

The items on the Citation file consist of MEDLARS items with five computer words added on at the beginning (for purely technical reasons) and lists of cited references ('parents') and citing references ('children') added on at the end. Where it was not possible to find a MEDLARS item for a reference the same format was used but the number of index terms ('tags') is zero.



APPENDIX II

EXAMPLES OF SEARCH STATEMENTS FOR CITATION, BOOLEAN, PROBABILISTIC, AND TITLE SEARCHES.

BOOLEAN SEARCH STATEMENT

```
NO0100
M1=MILK
M2=COW
M3=GOAT
C1=H.32.7
Ra:=M1andC1and(M2orM3)
```

PROBABILISTIC SEARCH STATEMENT

```
00100  0  30                    REQ NO     SUB NO      MAX
50  M3478                      WTS AS PERCENTAGES
50  M12777                     MESH TERM ID NOS

00100  1  30
100  M47656

00100  2  30
100  CH.32.7
```

TITLE SEARCH STATEMENT

```
00100  0  30                    REQ NO     SUB NO      MAX
1   ; GOAT;
1   ; COW;

00100  1  30
1   ; MILK;

00100  2  30
1   ;CHROMATOG;
```

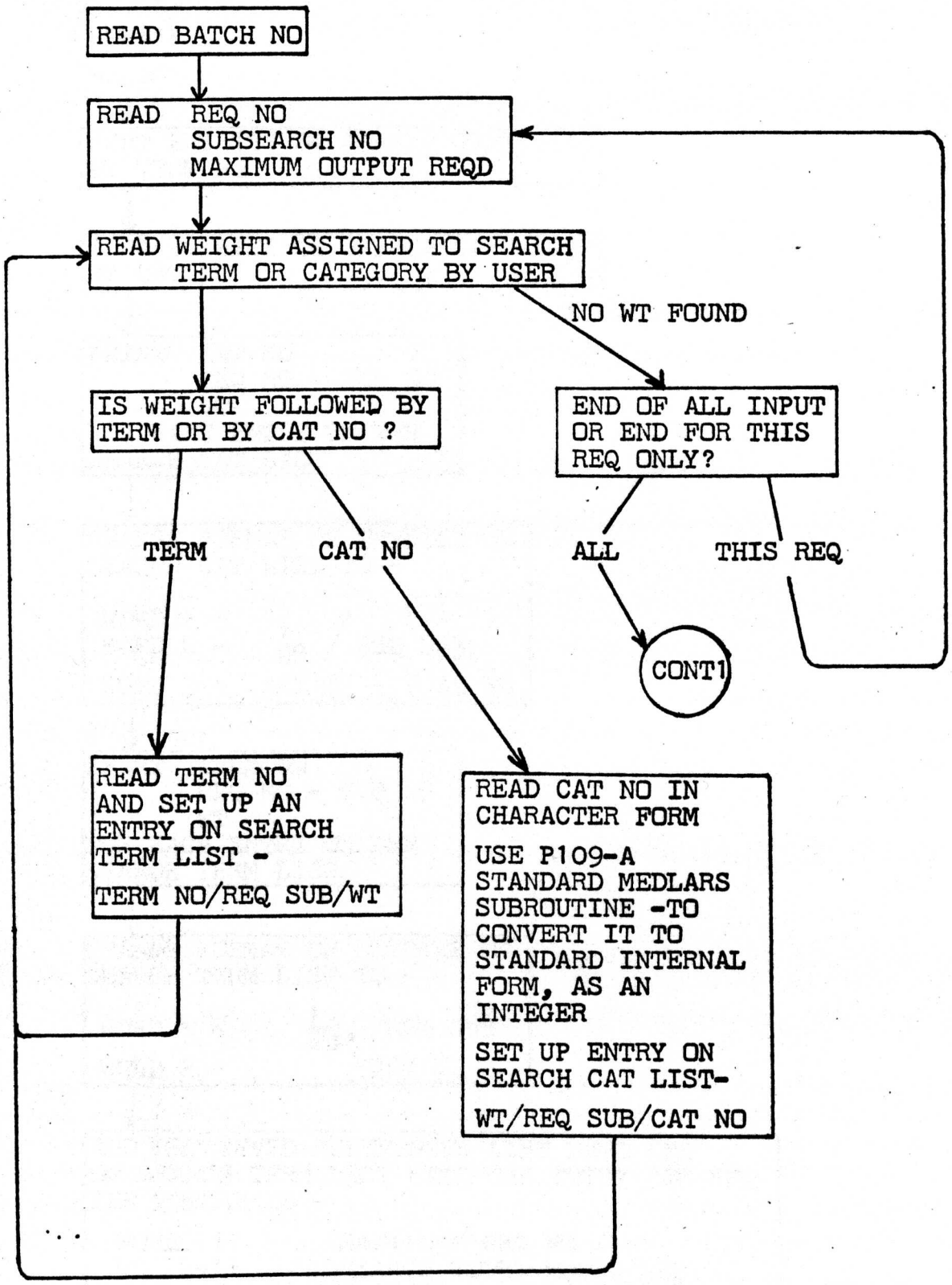
CITATION SEARCH STATEMENT

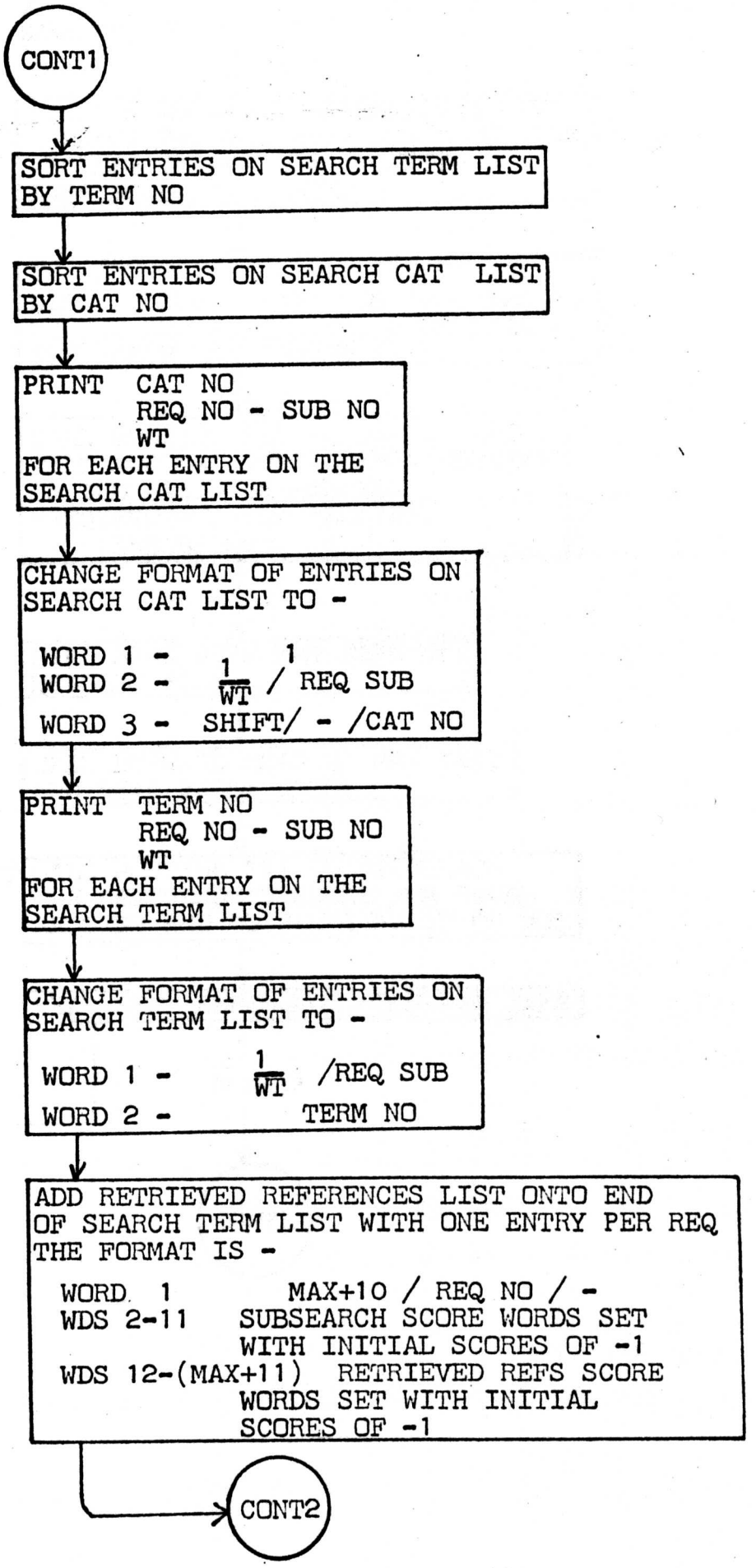
```
00100                          REQ NO
2  1414
1  1762                        WTS     REF ID NOS.
2  503
```

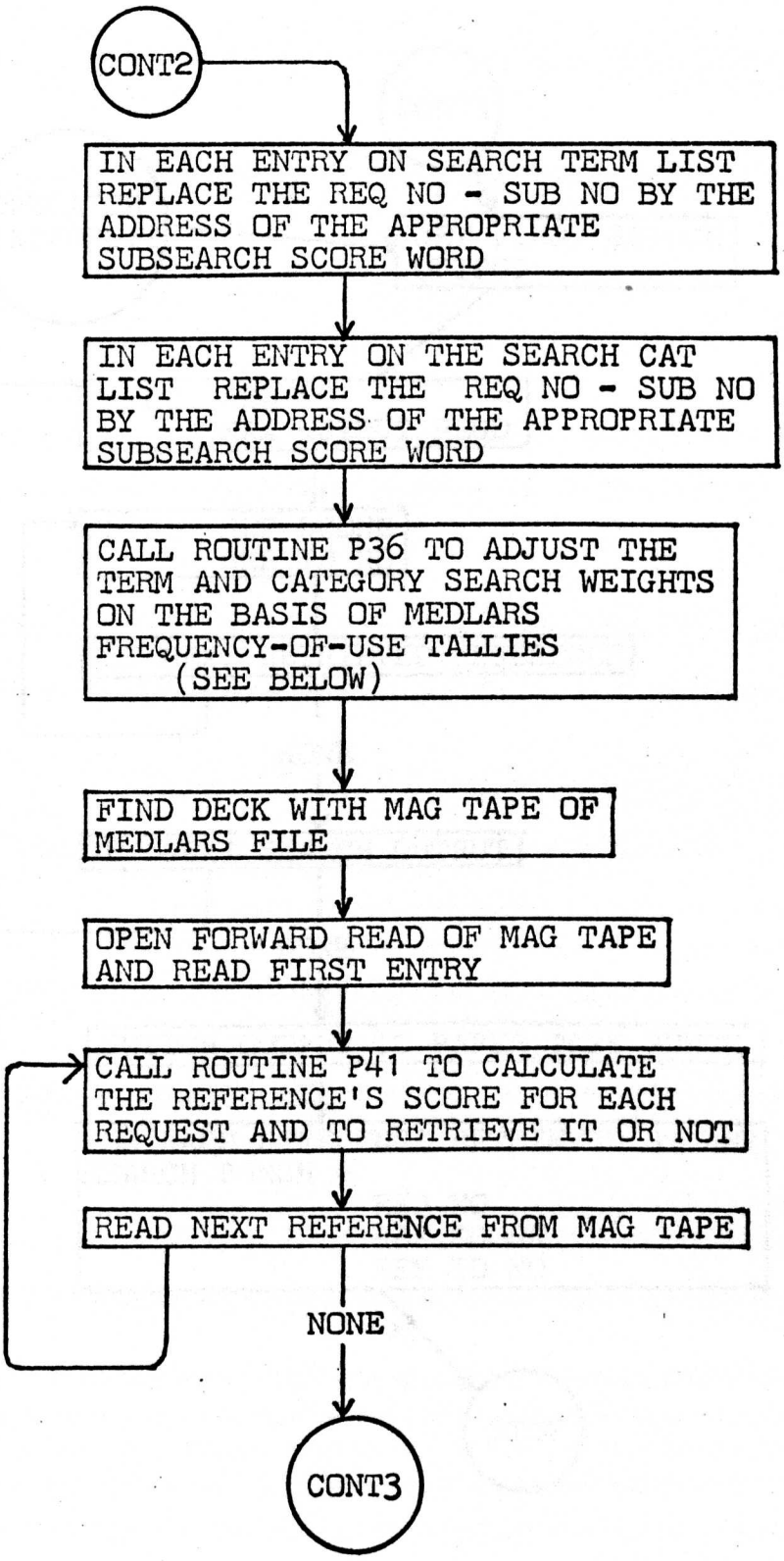
APPENDIX III

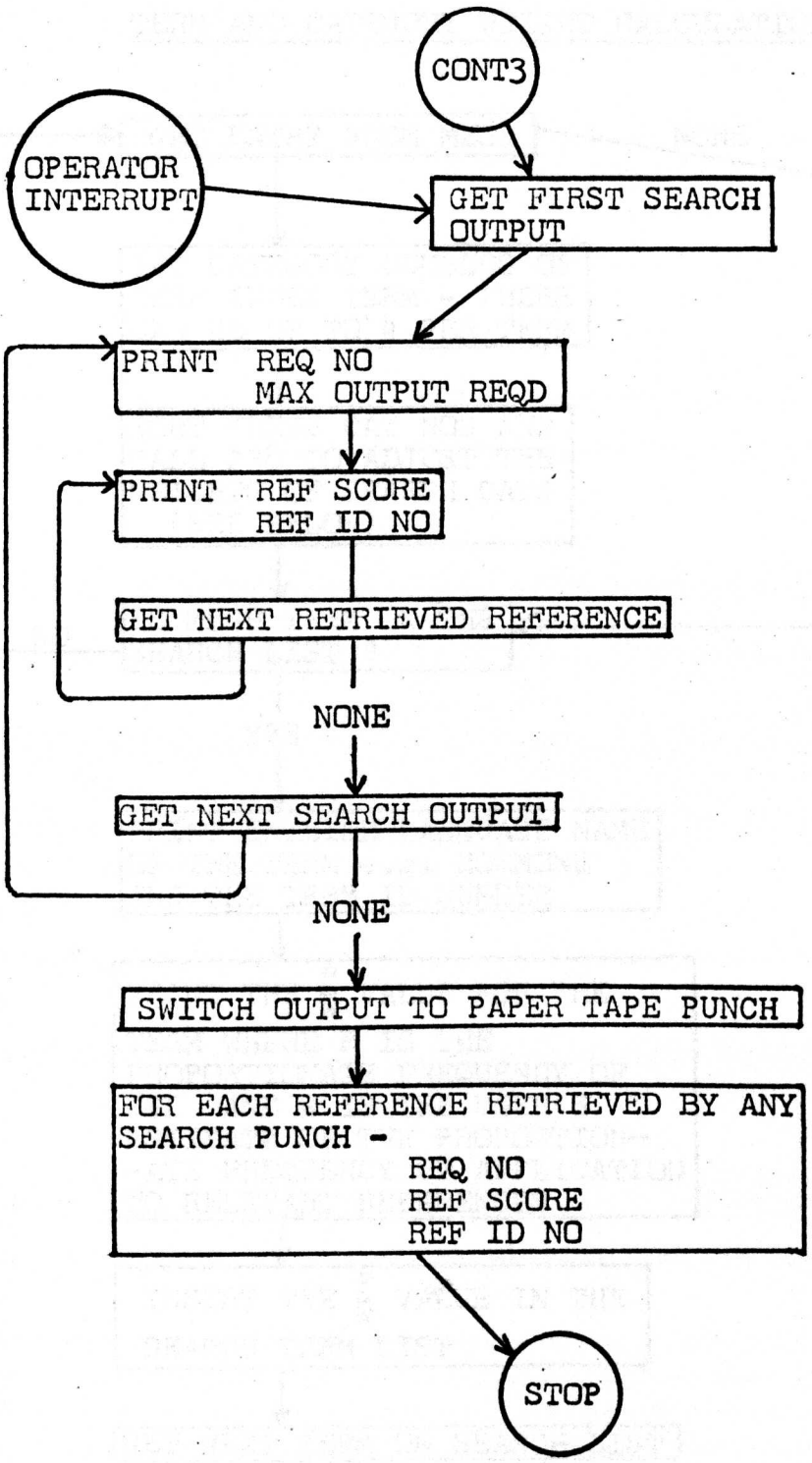
THE PROBABILISTIC SEARCH PROGRAM

PROBABILISTIC SEARCH PROGRAM

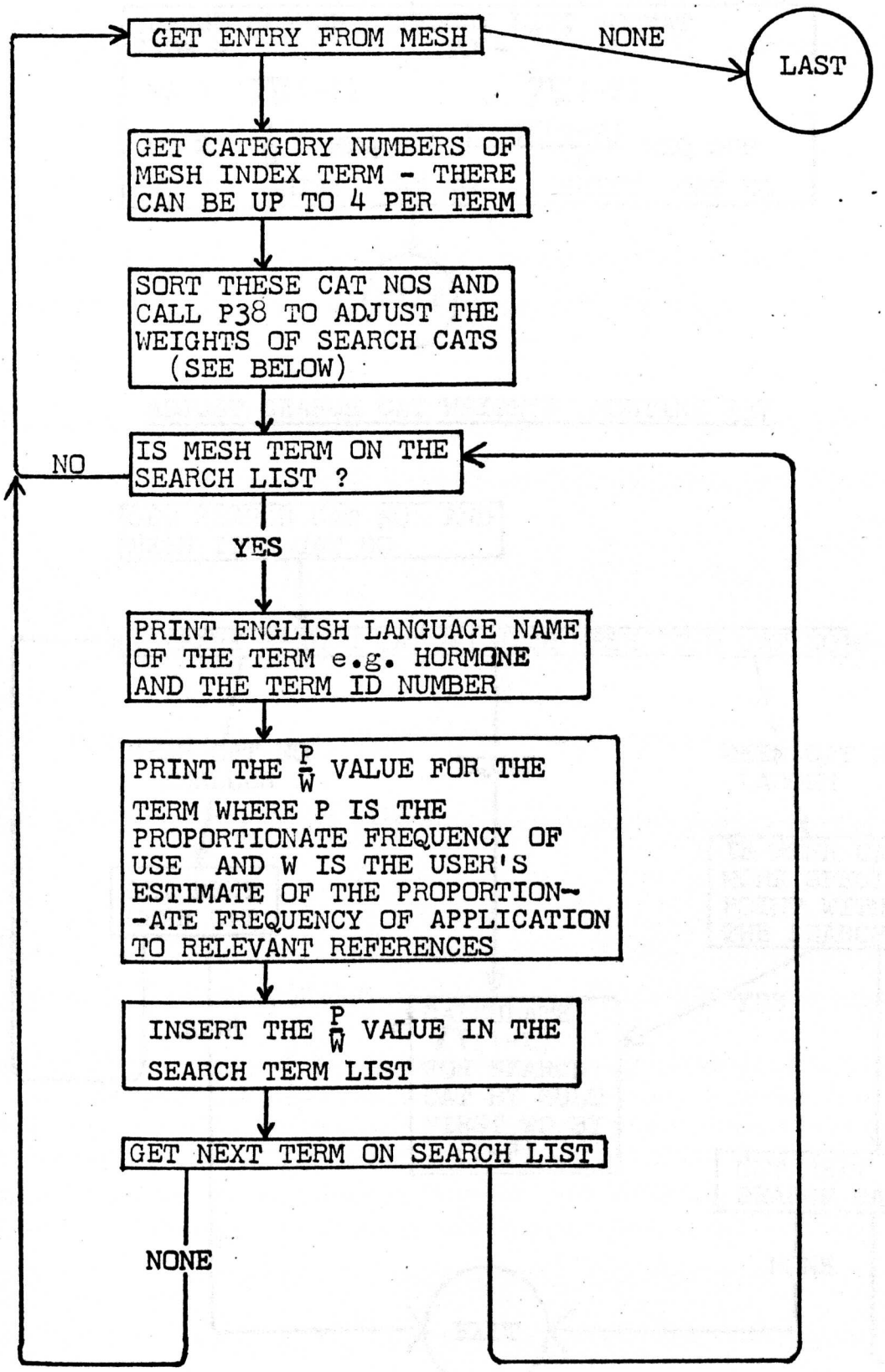


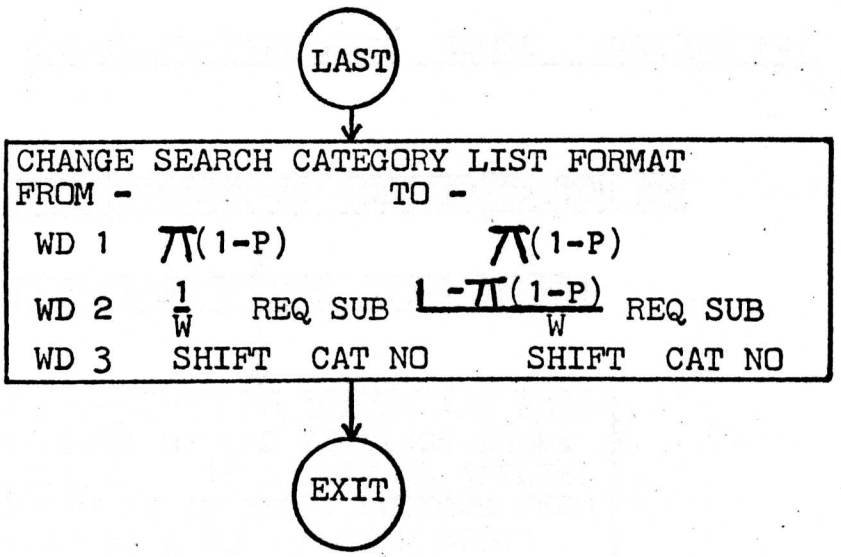

CONT1
SORT ENTRIES ON SEARCH TERM LIST BY TERM NO
SORT ENTRIES ON SEARCH CAT LIST BY CAT NO
PRINT CAT NO
REQ NO - SUB NO
WT
FOR EACH ENTRY ON THE SEARCH CAT LIST
CHANGE FORMAT OF ENTRIES ON SEARCH CAT LIST TO -
WORD 1 -
WORD 2 - 1/WT / 1 REQ SUB
WORD 3 - SHIFT/ - /CAT NO
PRINT TERM NO
REQ NO - SUB NO
WT
FOR EACH ENTRY ON THE SEARCH TERM LIST
CHANGE FORMAT OF ENTRIES ON SEARCH TERM LIST TO -
WORD 1 - 1/WT /REQ SUB
WORD 2 - TERM NO
ADD RETRIEVED REFERENCES LIST ONTO END OF SEARCH TERM LIST WITH ONE ENTRY PER REQ
THE FORMAT IS -
WORD 1 MAX+10 / REQ NO / -
WDS 2-11 SUBSEARCH SCORE WORDS SET WITH INITIAL SCORES OF -1
WDS 12-(MAX+11) RETRIEVED REFS SCORE WORDS SET WITH INITIAL SCORES OF -1
CONT2


CONT2
IN EACH ENTRY ON SEARCH TERM LIST REPLACE THE REQ NO - SUB NO BY THE ADDRESS OF THE APPROPRIATE SUBSEARCH SCORE WORD
IN EACH ENTRY ON THE SEARCH CAT LIST REPLACE THE REQ NO - SUB NO BY THE ADDRESS OF THE APPROPRIATE SUBSEARCH SCORE WORD
CALL ROUTINE P36 TO ADJUST THE TERM AND CATEGORY SEARCH WEIGHTS ON THE BASIS OF MEDLARS FREQUENCY-OF-USE TALLIES (SEE BELOW)
FIND DECK WITH MAG TAPE OF MEDLARS FILE
OPEN FORWARD READ OF MAG TAPE AND READ FIRST ENTRY
CALL ROUTINE P41 TO CALCULATE THE REFERENCE'S SCORE FOR EACH REQUEST AND TO RETRIEVE IT OR NOT
READ NEXT REFERENCE FROM MAG TAPE
NONE
CONT3

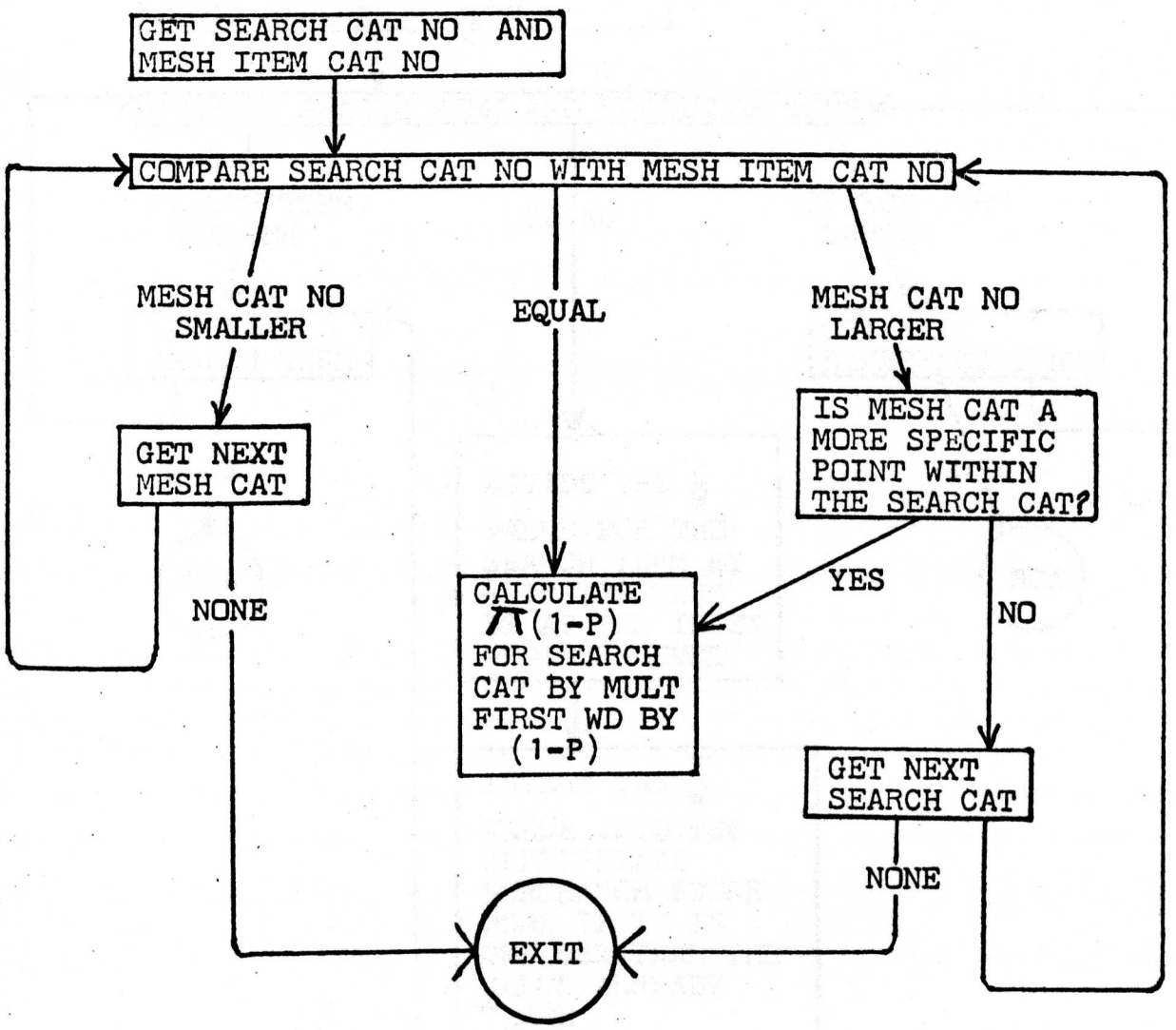

CONT3
OPERATOR INTERRUPT
GET FIRST SEARCH OUTPUT
PRINT REQ NO
MAX OUTPUT REQD
PRINT REF SCORE
REF ID NO
GET NEXT RETRIEVED REFERENCE
NONE
GET NEXT SEARCH OUTPUT
NONE
SWITCH OUTPUT TO PAPER TAPE PUNCH
FOR EACH REFERENCE RETRIEVED BY ANY SEARCH PUNCH -
REQ NO
REF SCORE
REF ID NO
STOP

TERM AND CATEGORY WEIGHT CALCULATION ROUTINE P36

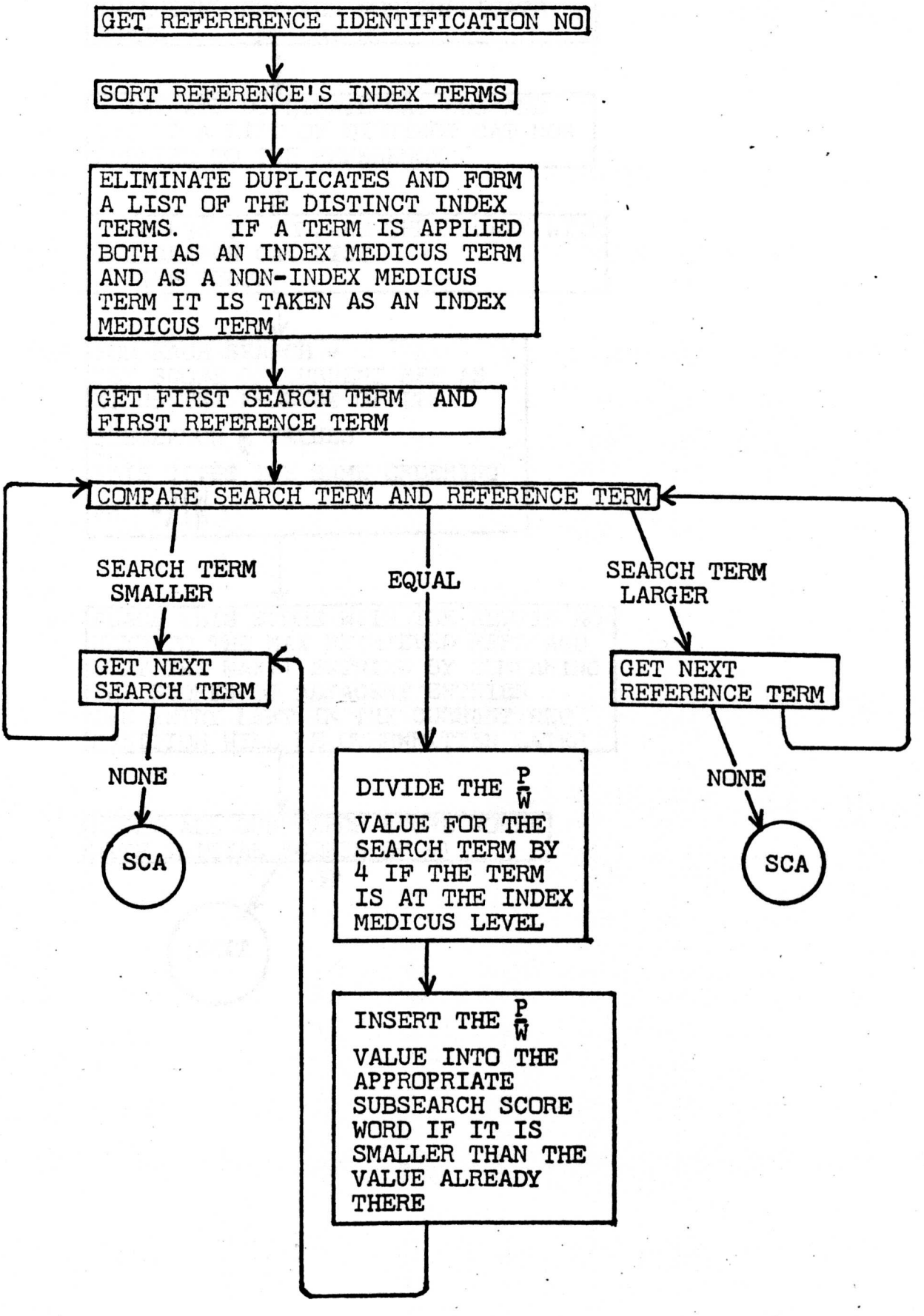


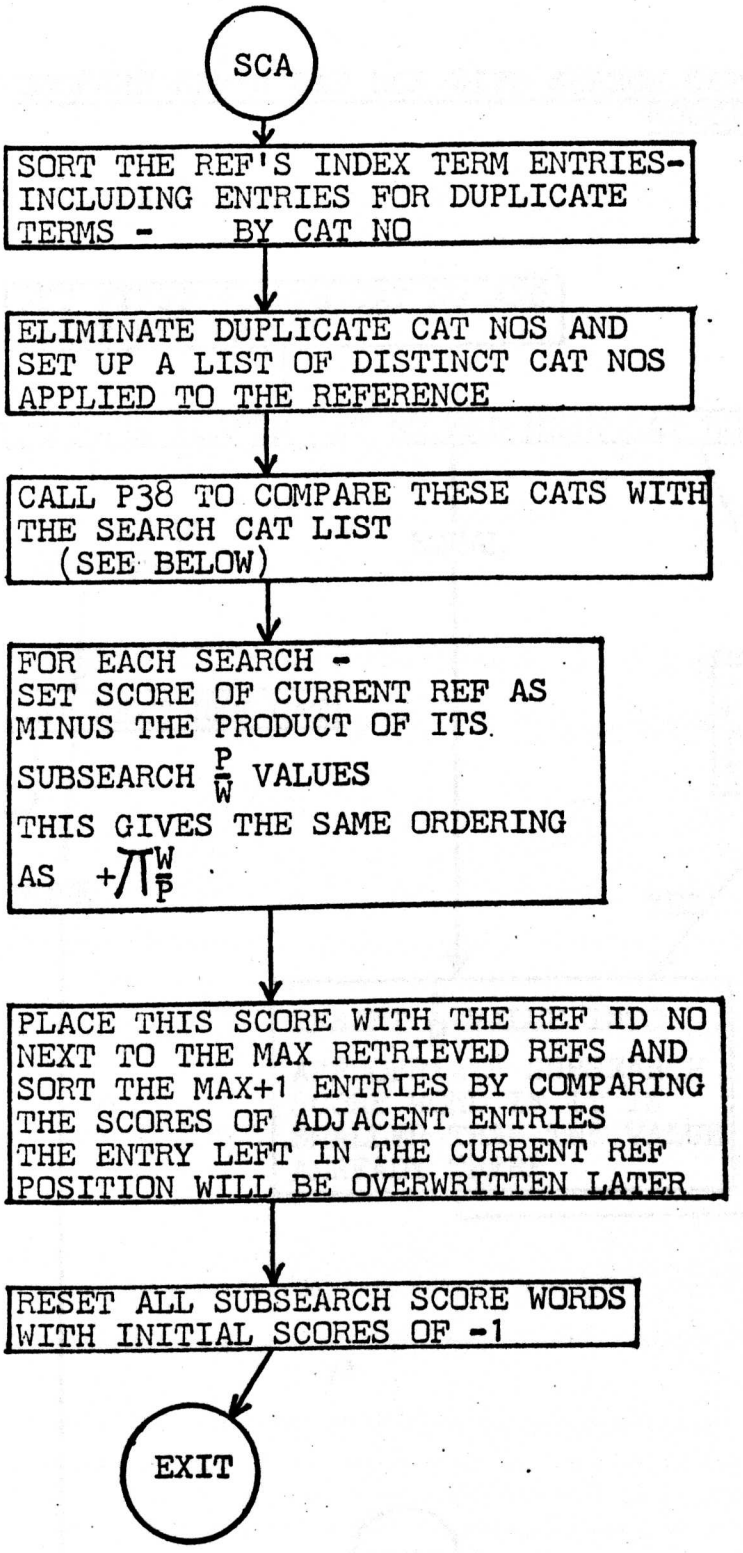


ADJUST SEARCH CAT WEIGHTS ROUTINE P37

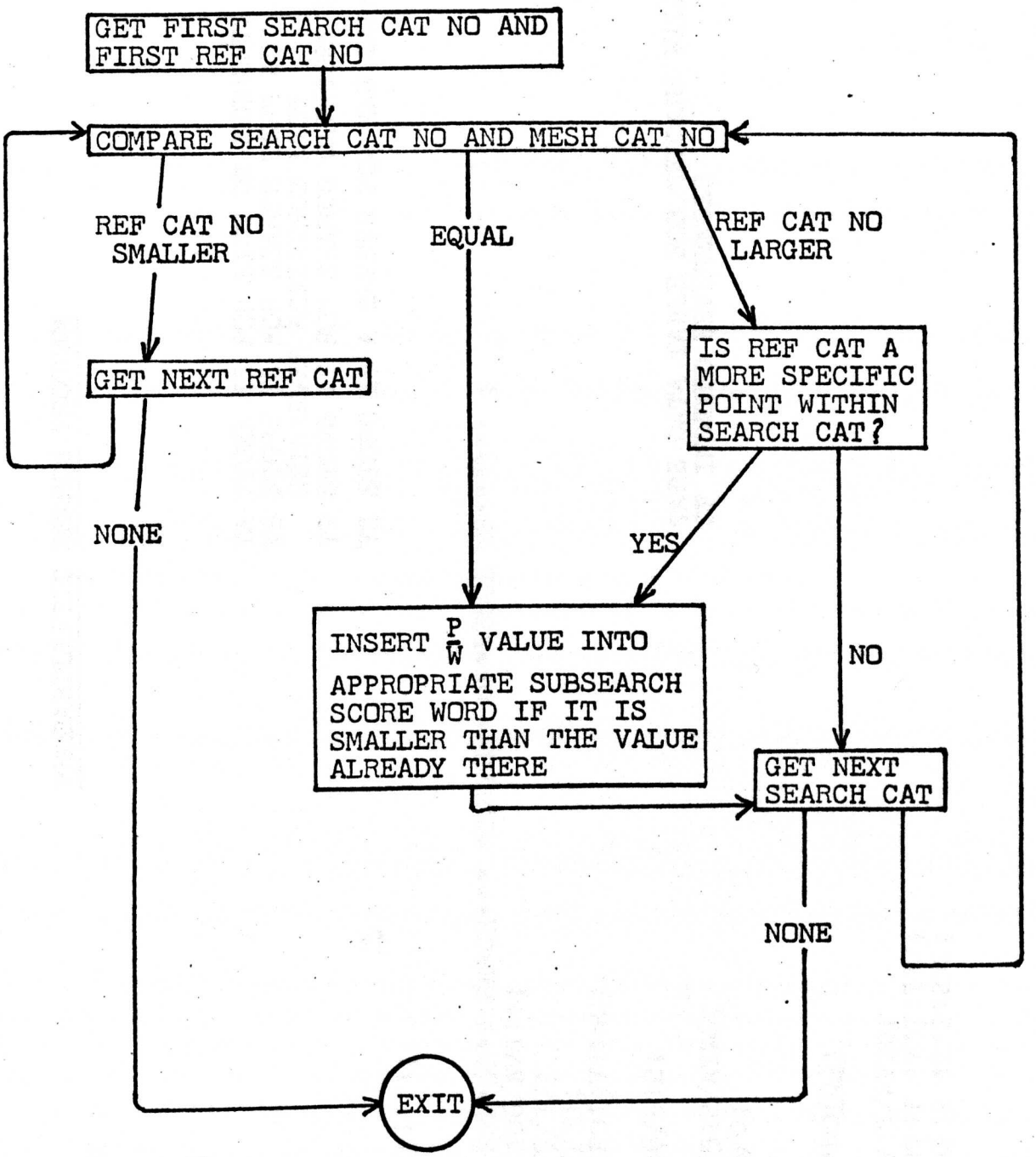


CALCULATE REFERENCE SCORE ROUTINE P41




SCA
SORT THE REF'S INDEX TERM ENTRIES- INCLUDING ENTRIES FOR DUPLICATE TERMS - BY CAT NO
ELIMINATE DUPLICATE CAT NOS AND SET UP A LIST OF DISTINCT CAT NOS APPLIED TO THE REFERENCE
CALL P38 TO COMPARE THESE CATS WITH THE SEARCH CAT LIST (SEE BELOW)
FOR EACH SEARCH - SET SCORE OF CURRENT REF AS MINUS THE PRODUCT OF ITS SUBSEARCH P/W VALUES
THIS GIVES THE SAME ORDERING AS +∏ W/P
PLACE THIS SCORE WITH THE REF ID NO NEXT TO THE MAX RETRIEVED REFS AND SORT THE MAX+1 ENTRIES BY COMPARING THE SCORES OF ADJACENT ENTRIES THE ENTRY LEFT IN THE CURRENT REF POSITION WILL BE OVERWRITTEN LATER
RESET ALL SUBSEARCH SCORE WORDS WITH INITIAL SCORES OF -1
EXIT

COMPARE REF'S CAT NOS WITH SEARCH CAT LIST
ROUTINE P38



```
P                                          PROBABILISTIC SEARCH PROGRAM
DHCL41CATSCR
→
ST14000;
TL12000;
V114;
W30;                                                    YA STORE IS FOR LIST OF REQUESTS
YA200;                                                  YB STORE IS FOR SEARCH TERM AND
YB9000;                                                    RETRIEVED REFERENCE LIST
YC200;                                                  YT STORE IS FOR SEARCH CATEGORY LIST
YT1000;
YZ1023;                                                 YZ STORE IS A BUFFER FOR MAG TAPES
RESTART;J82;J111;
PROG;
V106=0;
V107/111=P END*OF*S EARCH*** *OUTPUT* OF*RESUL TS****** ;
V112=Q 0/AV106/AV111;
V4=B 377 777/17;
V5=B 77 777/31;
V10=P MEDCCF01;V11=0;
V12=0;
V13=Q 0/4096/512;
V14=0;                                                  TAPE DATA TABLE FOR MEDLARS FILE
V15=Q 0/AYZ512/AYZ0;                                    OF INDEXED REFERENCES
V16=Q 32/16/0;
V17=0;
V18=0;
V19=P REEL*NO[Q];
V20=0;
V21=Q -1/AV18/AV20;
V22=B 17171717 17171717;
V23=B 12121212 12121212;
```

```
V30=P SEARCH**;
V31/32=P MAX*OUTPUT*REQD*;
V33/36=P ****DOC*SCORE*******CITATION *NO*;
V37/38=P END*OF*SEARCH***;
V39=B 00000000 02020202;
V40=Q 10/0/2;
V41=Q 10/0/0;
V42=B 377 7777;
V43=B 177777;
V44=B 37 777/31;
V45=B 177 777/31;
V90/93=P ******TAG***SEARCH*NO*****WEIGHT;
V94/97=P ******CAT***SEARCH*NO*****WEIGHT;
V113=0;
```

OUTPUT HEADINGS

COMPARISON MASKS

```
JSP2801;J112;JS1;=V114;
ZERO;=RM1;ZERO;=RM2;ZERO;=RM3;
3;JSP2801;J112;JS1;JSP2801;J112;JS1;+;
  DUP; JSP2801;J112;JS1; SET10;+;
  SHL32;REV;SHL16;OR;=YAOM1Q;SHL16;DUP;
4;JSP2801;J2;JS1;
  JSP2800;J112;DUP;SETB43;-;J41=Z;SETB55;-;J112≠Z;
  JSP2801;J112;JS1;SHL30;OR;OR;=YBOM2Q;DUP;J4;
41; ERASE;(read ytm3     WT11  REQ16  CLASS21);
```

READS BATCH NO

READ REQ NO, SUBSEARCH NO
READS MAX OUTPUT REQUIRED
YA FORMAT IS: MAX+10/REQSUB/-

READS INTEGER WEIGHT
JUMP TO 41 IF CATEGORY NO
READ INDEX TERM (AS INTEGER ID NO)
YB FORMAT IS: TERM/REQ/WT

```
   SHL37;REV;SHL5;OR;J42;
43;ERASE;
42;JSP2800;J111;DUP;J43=Z;SET7;=C4;ZERO;=C5;DC5;
44;JSP2800;J111;DUP;J45=Z;DUP;SET2;-;J45=Z;
   SHL42;REV;SHLD6;REV;ERASE;DC4;DC5;J44C4NZ;
   =V102;JSP2800;J111;DUP;J45=Z;DUP;SET2;-;J45=Z;
   SET7;=C4;DC5;J44;
45;ERASE;NC5;C5;SET8;-;DUP;J46=Z;J47<Z;
   C4;SET6;xD;CONT;=C4;SHLC4;=V103;J48;
46;ERASE;J48;
47;C4;SET6;xD;CONT;=C4;SHLC4;=V102;
48;Q2;Q3;C5; SETAV102;=M3;
   SETAV103;=M2; JSP109;J113;=Q3;=Q2;
   V104;SHL2;SHL-27;OR;=YTOM3Q; DUP;J4;
2;CAB;ERASE;REV;ERASE;DUP;=Q15;J3=Z;Q15;SET1;J112≠;ERASE;

M1;=V1;M2;=V2;M3;=V101;V2;J53=Z;
SETAYBO;V2;SHL32;OR;V4;REV;JSL82;

53;  V101;J54=Z;SETAYTO;V101;=RC15;=M15;
     Q15;DUP;ZERO;NOT;SHL-27;REV;JSL82;
     =Q4;V94;JS1P29;V95;JSP29;V96;JSP29;V97;JS2P29;
65;  MOM4Q;DUP;DUP;
     SHL27;SHL-27;SET47;V41;JSP2803;
     SHL11;SHL-32;SET47;V41;JSP2803;
     SHL-37;SET47;V40;JSP2803;J65C4NZ;
```

READ IN CATEGORY NO IN CHARACTER FORM

CALL SUBROUTINE P109 TO CONVERT CATEGORY NUMBER TO INTERNAL FORM (AS INTEGER)

YT FORMAT: WT/REQ/CAT
END OF INPUT OF DATA

SORT INDEX TERMS

MERGE SORT AND PRINT CATEGORY NUMBERS USED IN BATCH

PRINT CAT
PRINT REQ
PRINT WT

```
     SETAYTO;=M2;SET-1;DUP;=I3;=I4;
     V101;=M3;SET-1;=+M3;V101;DUP;SHL1;+;
     =M4;SET-1;=+M4;V101;=C3;SET1;SET47;FLOAT;STAND;DUP;
51;  M2M3Q;DUP;DUP;SHL27;SET14;=C4;ZERO;SHLD7;REV;DUP;J52=Z;
     REV;SHLD7;SET-7;=+C4;REV;DUP;J52=Z;
     REV;SHLD6;SET-6;=+C4;REV;DUP;J52=Z;
     REV;SHLD1;REV;
     SET-1;=+C4;
52;  ERASE;SHLC4;NC4;C4;SHA32;OR;=M2M4Q;SHL-37;
     SET47;FLOAT;SET100;SET47;FLOAT;÷F;CAB;REV;÷F;
     SHL-16;SHL16;REV;SHL11;SHL-32;OR;=M2M4Q;
     DUP;=M2M4Q;DUP;J51C3NZ;ERASE;ERASE;

54;  V2;J55=Z;
     V90;JS1P29;V91;JSP29;V92;JSP29;V93;JS2P29;
     V2;=RC2;
77;  YBOM2Q;ZERO;SHLD18;SET47;V41;JSP2803;
     ZERO;SHLD14;SET47;V41;JSP2803;
     SHL-32;SET47;V40;JSP2803;J77C2NZ;
     SETAYBO;=M2;SET-1;DUP;=I3;=I4;
     V2;=M3;SET-1;=+M3;V2;SHL1;=M4;SET-1;=+M4;
V2;= V2;=C3;SET1;SET47;FLOAT;STAND;DUP;
5;   M2M3Q;DUP;DUP;SHL-30;=M2M4Q;
     V43;AND;SET47;FLOAT;SET100;SET47;FLOAT;÷F;
     CAB;REV;÷F;SHL-16;SHL16;REV;V44;AND;SHL-16;
     OR;=M2M4Q;DUP;J5C3NZ;ERASE;ERASE;
```

CHANGE FORMAT OF
CAT LIST TO:

WORD 1:
WORD 2: $\frac{1}{WT}$ / ^{1} REQ NO
WORD 3: SHIFT/ - /CAT NO

PRINT INDEX TERM

PRINT REQ NO

PRINT WT

CHANGE SEARCH TERM LIST

FORMAT TO:
WORD 1: $\frac{1}{WT}$ / REQ NO
WORD 2: INDEX TERM NO

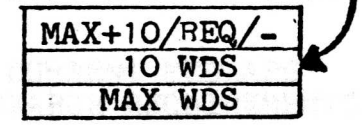
```
55;SETAYAO;=RM1;V1;=C1;ZERO;NOT;SHL16;SHL-32;
   SHL16;Q1;JSL82; ZERO;=RM1;V1;=C1;
   V2;SHL1; SET5;+; =RM2;
6;YAOM1;DUP;SHL16;SHL-32;SET100;÷I;ERASE;
  V113;REV;DUP;=V113;
  -;J62≠Z;=YAOM1Q;J6C1NZ;J63;
62;=YBOM2Q;SETAYBO;M2;+;YAOM1;OR;=YAOM1;
   YAOM1Q;SHL-32;=C5;SET-1;SET47;FLOAT;STAND;DUP;
61;=YBOM2Q;DUP;DC5;J61C5NZ;ERASE;ERASE;J6C1NZ;
63;M2;V2;SHL1;-;=V9;


   V2;J56=Z;ZERO;=RM2;SET2;=I2;V2;=C2;
72;YBOM2;SHL32;SHL-32;SET100;÷I;REV;SET100;
   ×D;CONT;V114;+;ZERO;=RM1;V1;=C1;
71;YAOM1;SHL16;SHL-32;J7=;
   SET1;=+M1;SET-1;=+C1;J71C1NZ;J112;
7;ERASE;YAOM1Q;DUP;SHL32;J112=Z;+;V114;-;
  SHL32;YBOM2;SHL-16;
  SHLD16;=YBOM2Q;ERASE;J72C2NZ;

56; V101;J57=Z;
    SET1;=M2;SET3;=I2;V101;=C2;
73; YTOM2;SHL32;SHL-32;SET100;÷I;REV;SET100;
    ×D;CONT;V114;+;ZERO;=RM1;V1;=C1;
74; YAOM1;SHL16;SHL-32;J75=;
    SET1;=+M1;SET-1;=+C1;J74C1NZ;J112;
75; ERASE;YAOM1Q;DUP;SHL32;J112=Z;+;V114;-;
    SHL32;YTOM2;SHL-16;
    SHLD16;=YTOM2Q;ERASE;J73C2NZ;
```

SORT YA ON <u>REQ SUB</u>

CHANGE FORMAT OF YA LIST TO:
YA: MAX+10/REQ/ ADDRESS

AND ADD TO THE YB LIST THE RETRIEVED REFERENCES LISTS OF FORMAT:

YB:



IN SEARCH TERM LIST ON YB,
REPLACE REQ NOS BY ADDRESSES
EACH SUBS HAS A DISTINCT ADD

IN SEARCH CAT LIST ON YT
REPLACE REQ NOS BY ADDRESSES
EACH SUBS HAS A DISTINCT ADD

```
57;(no cats jump);
SETAYB1;V2;SHL32;OR;
SET2;SHL16;OR;DUP;=V6;JSP36;

V10;SET4;OUT;SHL16;=V14;
V21;SET8;OUT;V23;V20;SHL-36;V22;AND;TOB;SHL16;=V11;
SETAV10;JS12L34;J113;=V17;
V1;=RC1;SETAYA0;=M1;Q1;=V46;
81;V46;V6;V17;JSP41;
V17;SETAV10;JSL34;J8;=V17;J81;
8;SET2;-;J113≠Z;
  ZERO;83;ERASE;82;J83NEN;  V112;SET8;OUT;
SETAV10;JS10L34;DUMMY;DUMMY;DUMMY;
SETAYA0;V1;SHL32;OR;V45;REV;JSL82;

ZERO;=RM1;V1;=C1;
91;SET3;JSP2802;YAOM1Q;DUP;=RM2;=C2;J93C2Z;
SET-1;=+M2;EOM2Q; V30;JSP29;
ZERO;SHLD16;=C2;  SET-10;=+C2; SET10;=+M2;
SHL-32;DUP;=V8;SET47;V40;JSP2803;
V31;JS1P29;V32;JSP29;
C2;SET47;V40;JSP2803;
V39;JSP29;V33;JSP29;V34;JSP29;V35;JSP29;V36;JS2P29;
9;EOM2Q;DUP;V42;AND;J92=Z;
ZERO;SHLD28;SHL21;SHL-40;
SET128;-;SET47;V41;JSP2803;
SHL-28;SET47;V40;JSP2803;
93;J9C2NZ;J91C1NZ;J94;
92;ERASE;J93;
```

CALL P36 TO ADJUST WEIGHTS OF SEARCH TERMS AND CATEGORIES USING FREQUENCY-OF-USE OF TERMS

OPEN MAG TAPE CONTAINING MEDLARS FILE AND READ REFERENCE

CALL P41 TO COMPARE CURRENT MEDLARS REFERENCE WITH THE SEARCH REQUIREMENTS

END OF SEARCH

PRINT SECTION STARTS

PRINT SEARCH NUMBER

PRINT MAX OUTPUT REQUIRED

PRINTS REFERENCE SCORE
PRINTS REFERENCE IDENTIFICATION NO

END OF PRINT SECTION

```
94;   JS1P2802;SET2;JSP2802;                          START OF OUTPUT TO PUNCHED PAPER TAPE,
   SET-1;=RC15;SET10;=I15;Q15;=V41;                   FOR INPUT TO MERGE SORT AND PRINT OF
    ZERO;=RM1;V1;=C1;                                 FULL MEDLARS ITEMS
101;YAOM1Q;DUP;=RM2;=C2;J103C2Z;  SET-1;=+M2;
    EOM2Q;ZERO;SHLD16;=C2; SET-10;=+C2;SET10;=+M2;SHL-32;=V8;
100;EOM2Q;
    DUP;V42;AND;J102=Z;
    ZERO;SHLD28;SHL20;FIX;
    V8;SET47;V41;JSP2803;                             PUNCHES REQ NO
    V41;JSP2803;                                      PUNCHES REF SCORE
    SHL-28;SET47;V40;JSP2803;                         PUNCHES REF IDENTIFICATION  NO
103;J100C2NZ;J101C1NZ;J104;
102;ERASE;J103;
104;JS2P2802;JS1P2802;SET2;JSP2802;JS2P2802;ZERO;OUT;     END OF PROGRAM
1;SET47;-;=C15;SHLC15;EXIT1;                          INTEGER INPUT CONVERSION

111;JS114;SETB21;J115;
112;JS114;SETB22;J115;
113;JS114;SETB23;J115;
114;J116EN;SET47;V40;JSP2803;J114;                    FAILURE EXITS
116;V37;JS1P29;V38;JSP29;EXIT1;
115;JSP2802;JS2P2802;ZERO;OUT;

P29;
3;SET8;=C13;
4;ZERO;SHLD6;JSP2802;DC13;J4C13NZ;                    THIS SUBROUTINE PRINTS
ERASE;EXIT1;                                          A KDF9 WORD AS EIGHT
1;SET2;JSP2802;J3;                                    CHARS OF TEXT
2;JS3;SET2;JSP2802;EXIT1;
```

```
P36V15;V0=F 400 000;                    THIS SUBROUTINE USES THE FREQUENCY-OF-USE OF
V1/2=P WRONG*MEDDICTA**;                SEARCH TERMS AND CATS TO CALCULATE THEIR WTS
V3=Q 10/0/0;
V4=Q 0/10/2;
V5=P TAG*LIST;
V6=P END*OF**;
=Q2;SET-3;=M4;SET3;JSP2802;J5C2Z;  MOM2Q;DUP;     GETS NEXT SEARCH TERM
2;JSP118;J11;=RM15;EOM15Q;SET1;SHL7;+;=V7;       CALLS P118 TO GET NEXT ENTRY ON
  EOM15Q;=V8;EOM15Q;=V9;EOM15Q;=V10;             THE MEDLARS DICTIONARY TAPE
  V101P0;J25=Z; ZERO;=RM14;

  EOM15Q;SHL27;SHL-27;=V11M14Q;                  GETS CATEGORY NOS OF TAPE TERM
  SET3;=C13;M14;=V15;
23;EOM15Q;SHL27;SHL-27;V15;=RC14;
21;V11M14Q;J22=;J21C14NZ;=V11M14Q;M14;=V15;ZERO;
22;ERASE;DC13;J23C13NZ;

   SETAV11;=RM15;V15;=C15;ZERO;NOT;SHL-27;       SORTS CAT NOS AND CALLS P37 TO
 Q15;DUP;PERM;JSL82; =Q15;V11;J31≠Z;MOM15Q;ERASE;  ADJUST WEIGHTS OF SEARCH CATS
31;Q15;JSP37;
25;C2;SET1;+;J2=Z;
3;V7;SHL1;SHL-31;SIGN;DUP;J10<Z;J4=Z;DUP;J2;    JUMP TO 4 IF TAPE TERM IS A
                                                SEARCH TERM
4;V8;JSP29;V9;JSP29;V10;JSP29;SET47;V3;JSP2803;  PRINTS ENGLISH TEXT OF THE
  V7;SHL18;SHL-25;SHL24;ZERO;FLOAT;STAND;        INDEX TERM, AND THE TERM NO
41;M4M2;ZERO;SHLD32;SHL16;                      PRINTS THE P/W VALUE FOR THE
  CAB;×F;DUP;FIX;V4;JSP2803;SHL-16;SHLD-32;ERASE;  TERM AND INSERTS IT ON THE
  =M4M2;J5C2Z;MOM2Q;DUP;J3;                     SEARCH LIST

5;SET-1;=C2;SET3;JSP2802;V101P0;J26=Z;J2;
10;ERASE;ZERO;JSP29;SET47;V3;JSP2803;SET1;SET46;FLOAT;J41;   MISSING TERM LOOP
```

```
11;DUP;DUP;SET2;-;J10≠Z;ERASE;ERASE;                    NO MORE TERMS IN DICTIONARY
   V101P0;J26=Z;  V101P0;=RC15;C15;DUP;+;=+C15;
   SET1;SET47;FLOAT;DUP;
24;YTOM15Q;-F; ZERO;YTOM15;SHLD-16;                      CHANGES YT LIST FORMAT
   SHL16;CAB;×F;SHL-16;SHLD16;=YTOM15Q;                 FROM:WD1 π(1-P)      TO:   π(1-P)
   ERASE;YTOM15Q;ERASE;                                      WD2  1/W  /REQ        [1-π(1-P)]/W  /REQ
   DUP;J24C15NZ;ERASE;ERASE;                                 WD3  SHIFT/CAT          SHIFT/CAT
26;JS2P118;EXIT1;(needs meddicta in numeric order);


P37V5;                                                  THIS SUBROUTINE CALCULATES THE WT OF
=Q15;SET2;=RM14;V101P0;=C14;SET3;=I14;                  SEARCH CATEGORIES
     YTOM14Q;MOM15Q;J4;
5;   REV;ERASE;YTOM14Q;REV;
4;   DUPD;SHL27;SHL-27;REV;                             JUMP TO 2 IF SEARCH CAT = TERM CAT
     SHL27;SHL-27;SIGN;DUP;J1<Z;J2=Z;
     DUPD;REV;DUP;=Q13;
     SHL16;SHL-16;SHLC13;REV;SHLC13;SIGN;J3=Z;J5C14NZ;   JUMP TO 3 IF S CAT ⊃ TERM CAT
     ERASE;ERASE;EXIT1;
1;   ERASE;ERASE;J6C15Z;MOM15Q;J4;
6;   ERASE;EXIT1;
2;   JS40;J5C14NZ;ERASE;ERASE;EXIT1;                    CALL WEIGHT CALCULATION SEQUENCE
3;   JS40;J5C14NZ;ERASE;ERASE;EXIT1;

40;  V7P36;SHL-7;SHL25;SHL-1;ZERO;FLOAT;                 CALCULATES π(1-P)
     SET1;SET47;FLOAT;REV;-F;
     M14TOQ13;SET-5;=+M13;YTOM13;×F;=YTOM13Q;EXIT1;
```

```
P38;
=Q15;SET2;=RM14;V101PO;=C14;SET3;=I14;
   YTOM14Q;MOM15Q;J4;
5;  REV;ERASE;YTOM14Q;REV;
4;  DUPD;SHL27;SHL-27;REV;
    SHL27;SHL-27;SIGN;DUP;J1<Z;J2=Z;
    DUPD;REV;DUP;=Q13;SHL16;SHL-16;
    SHLC13;REV;SHLC13;SIGN;J3=Z;J5C14NZ;
    ERASE;ERASE;EXIT1;
1;  ERASE;ERASE;J6C15Z;MOM15Q;J4;
6;  ERASE;EXIT1;
2;  JS40;J5C14NZ;ERASE;ERASE;EXIT1;
3;  JS40;J5C14NZ;ERASE;ERASE;EXIT1;

40; M14TOQ13;SET-4;=+M13;YTOM13;
    SHC-16;ZERO;SHLD16;=M13;
    MOM13;SHL-20;SHL20;
    NEGF;MAXF;ERASE;NEGF;
    SHL-20;SHL20;
    V8P41;OR;=MOM13;EXIT1;
(end p38);
```

THIS SUBROUTINE COMPARES THE CURRENT REFERENCES CATS WITH THE SEARCH CATS

JUMP TO 2 IF SEARCH CAT = TERM CAT

JUMP TO 3 IF SEARCH CAT $\supset$ TERM CAT

CALL HIT ACTION

INSERT $\frac{P}{W}$ VALUE FOR CURRENT CAT BUT ONLY IF IT IS SMALLER THAN THE VALUE ALREADY THERE

```
P41V119;
V5=B 377 777;
V6=B 1/6;
V119=F.25
=V3;=V2;=V1;
V3;=M4;E1M4;SHL-28; =V8;
E2M4;SETB77;AND;M4;SET3;+;=RM4;=C4;Q4;=V4;J10C4Z;
```

THIS SUBROUTINE CALCULATES THE SCORE OF A REFERENCE FROM THE TERMS AND CATEGORIES USED TO INDEX IT

GET REFERENCE IDENTIFICATION NO

```
1;EOM4;SHC17;=EOM4Q;  J1C4NZ;
  V2;SHL-32;J100=Z;
  V5;V4;JSL82;
V4;=Q4;ZERO;=RM3;EOM4Q;DUP;=V7;V5;AND;=V9;J4C4Z;
3;EOM4Q;DUP;V5;AND;V9;-;J2=Z;
V9;=YCOM3Q;V7;V6;AND;=YCOM3Q;
DUP;=V7;V5;AND;=V9;J3C4NZ;
4;V9;=YCOM3Q;V7;V6;AND;=YCOM3Q;J5;
2;V7;OR;=V7;J3C4NZ;J4;

5;M3;SHL-1;=RC3;SET2;=I3;SETAYCO;=M3;  V2;=Q2;
SET-3;=M4;SET-1;=M5;SET-2;=M6;
MOM2Q;MOM3Q;
51;DUPD;SIGN;DUP;J6=Z;J52<Z;
ERASE;J101C3Z;MOM3Q;J51;
52;REV;ERASE;J101C2Z;MOM2Q;REV;J51;

6;ERASE;ERASE;ERASE;
ZERO;M4M2;SHLD-16;SHL16;M5M3;J61=Z;V119;xF;
61;REV;SHL-32;=M15;MOM15;SHL-20;SHL20;
   NEGF;MAXF;ERASE;NEGF;
SHL-20;SHL20;
V8;OR;=MOM15;
J100C2Z;MOM2Q;M6M3;J51;
101;ERASE;

100;V101P0;J25=Z;V4;=Q4;
102;EOM4;SHC-17;=EOM4Q;J102C4NZ;
    ZERO;NOT;SHL-27;V4;JSL82;
```

SORT INDEX TERMS OF REFERENCE

ASSEMBLES IN YC A LIST OF DISTINCT INDEX TERMS WITH INDEX MEDICUS BITS SET IF ANY OF THE APPLICATIONS OF THE TERM WERE INDEX MEDICUS

COMPARISON OF SEARCH TERM AND REF INDEX TERM

REFERENCE INDEXED WITH SEARCH TERM. DIVIDE P/W VALUE BY 4 IF TERM APPLIED AT INDEX MEDICUS LEVEL

INSERT THE P/W VALUE FOR TERM, BUT ONLY IF IT IS SMALLER THAN THE VALUE ALREADY THERE

NO MORE SEARCH TERMS

SORT SEARCH CATS

```
    V4;=Q15;SETAV10;=RM14;ZERO;=V10;                    SET UP A LIST OF DISTINCT CATEGORIES
23; MOM15Q;SHL27;SHL-27;MOM14;J22≠;
     ERASE;J23C15NZ;J24;
22;SET1;=+M14; =MOM14;J23C15NZ;
24;  SET1;=+M14;M14;=RC14;SETAV10;=M14;Q14;JSP38;     CALL P38 TO COMPARE THE SEARCH CATS
                                                      WITH THE REF CATS
25;  V1;=Q1;SET-2;=M14;
99;J10C1Z;EOM1Q;DUP;=RM2;=C2;J99C2Z;
   SET-1;=+M2;EOM2Q;SHL-32;=C2;M2;=M15;
EOM2Q;SHL-20;SHL20;
EOM2Q;SHL-20;SHL20;XF;
EOM2Q;SHL-20;SHL20;XF;                                FOR EACH SEARCH,  CALCULATE THE
EOM2Q;SHL-20;SHL20;XF;                                REFERENCE'S SCORE AS THE PRODUCT
EOM2Q;SHL-20;SHL20;XF;                                OF ITS SUBSEARCH SCORES
EOM2Q;SHL-20;SHL20;XF;
EOM2Q;SHL-20;SHL20;XF;
EOM2Q;SHL-20;SHL20;XF;
EOM2Q;SHL-20;SHL20;XF;
EOM2;SHL-20;SHL20;XF;NEGF;SHL-20;SHL20;V8;OR;=EOM2;

EOM2;SHL-20;SHL20;  SET-1;SET47;FLOAT;
SHL-20;SHL20;-F;J99=Z; EOM2Q;EOM2Q;                   SORT THE CURRENT REFERENCE INTO THE
98;DUPD;SIGNF;J97<Z;=M14M2;J96C2Z;EOM2Q;J98;          SET OF RETRIEVED REFERENCES. IF IT
96;SET1;=+M2;=M14M2;J95;                              IS NOT MOVED INTO THIS SET IT IS
97;REV;=M14M2Q;=M14M2;                                EFFECTIVELY DELETED
```

```
95;SET-1;SET47;FLOAT;STAND;
DUP;=E0M15;DUP;=E1M15;DUP;=E2M15;
DUP;=E3M15;DUP;=E4M15;DUP;=E5M15;
DUP;=E6M15;DUP;=E7M15;DUP;=E8M15;
=E9M15;J99;
10;EXIT1;

P118V7; (read dict tape written by E.D.B.);
V0=B5545444451436441;
V1=Q0/1/1;
V2=0;
V3=Q0/4096/512;
V4=Q0/0/0;
V5=Q0/AYZ512/AYZ0;
V6=+11;
V7=0;
      V7; DUP; J1≠Z;
      ERASE; V0; SET4; OUT;
      V4; =Q15; =I15; Q15; =V4; SETAV0;
      JS12L34; EXIT1; J3;
1;    SETAV0; JSL34; EXIT1;
3;    DUP; =V7; EXIT2;( exit with add of item in n1);
2;      SETAV0; JS10L34; ERASE; DUMMY; DUMMY; EXIT1;
      (end of p118);
```

RESET THE SUBSEARCH SCORES FOR THE NEXT REFERENCE

THIS SUBROUTINE IS FROM THE STANDARD MEDLARS SYSTEM

```
P109V17; (Category number assembler written by E.D.B.);
V0U=0.0077/0; V0L=0;
V1U=0.0096/0; V1L=B22;
V2U=0.0142/0; V2L=B36;
V3U=0.0210/0; V3L=B63;
V4U=0.0302/0; V4L=B100;
V5U=0.0192/0; V5L=B113;
V6U=0.0590/0; V6L=B116;
V7U=0.0328/0; V7L=B127;
V8U=0.0237/0; V8L=B133;
V9U=0.0125/0; V9L=B137;
V10U=0.0048/0; V10L=B143;
V11U=0.0055/0; V11L=B147;
V12U=0.0214/0; V12L=B153;
V13U=0;       V13L=B157;
V14U=0;       V14L=B163;
V15U=0;       V15L=B167;
V16U=0.10;    V16L=B172;
V17=B010607;

        VR; =RC15; (count of chars); SET 8; =C3; ZERO; =RM13;
        MOM3; (1st word of chars); DUP;
1;      ERASE; J3C15Z; (J if end of chars);
        JS2; (J to read 1 char); DUP; SETB20; -; J1<Z; (ignore);
        SET B41; -; DUP; (L-41); J4<Z; (Fail not letter);
        DUP; SET 16; -; J5<Z; (J if A-P); DUP; SET 25; -; J4≠Z; (J not Z);
        SET 9; -;
5;      =M15; VOM15; (read code and tally); DUP; SET B177; AND;
        REV; SHL-24; =Q14; (tally to Q14); V1M15; SHL+24;
        SHL-24; (next code); V17; =M15; J8C15Z; (J if end of chars);
9;      REV; CAB; (N1=string, N2=L, N3=L+1);
        JS6; (read number); REV; REVD; CAB;
        +; MAX; ERASE; J4NV; (Fail if category exceeded);
```

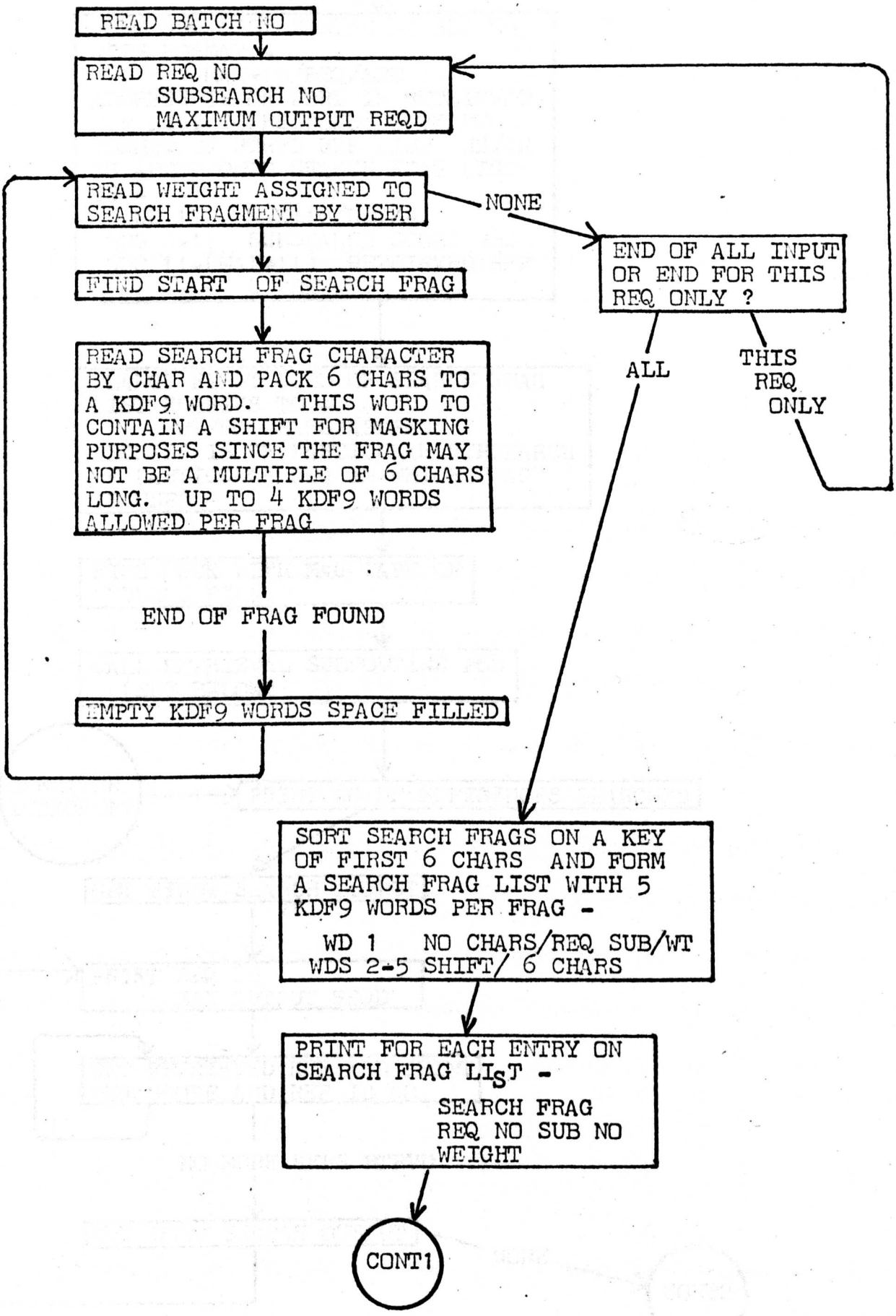
THIS SUBROUTINE IS FROM THE STANDARD MEDLARS SYSTEM

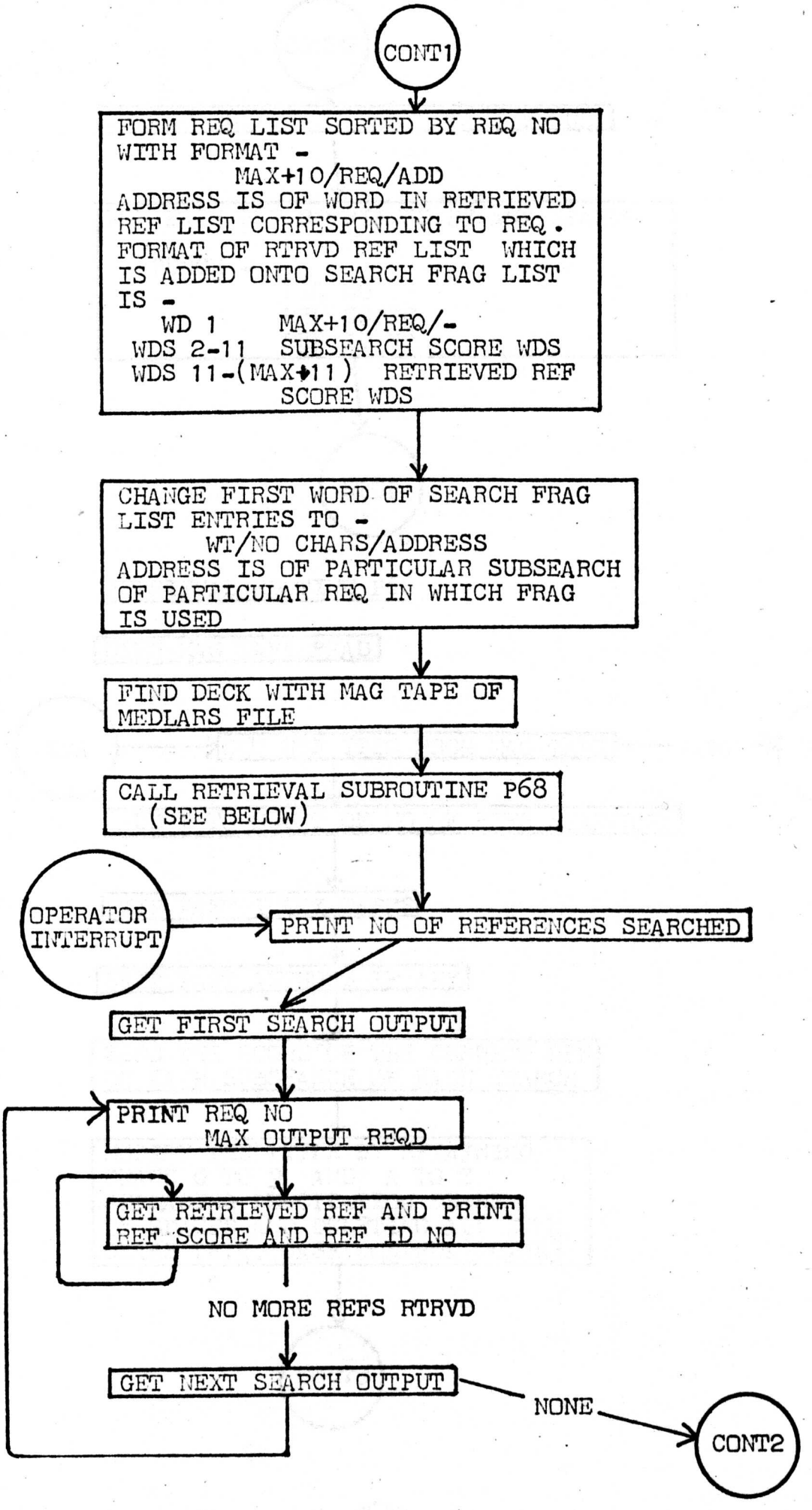
```
        DUP; NOT; NEG; J8C15Z; M+I13;
        M15; DUP; SHL-6; =M15; SETB77; AND;
        DUP; (shift); =C13; =+C14; SHLDC13; J9;
8;      SET 39; NC14; =+C14; ERASE;
        SHLC14; (Cat No bits 3-24); COTOQ14; Q14;
        MOM2; M13; SHL+16; OR; =MOM2Q;
        OR; =MOM2Q; ERASE; EXIT2;
4;      ERASE;
3;      ERASE; EXIT1;
2;      (Read char); ZERO; SHLD+6; DC3; J11C3NZ;
        M+I3; SET 8; =C3; REV; ERASE; MOM3; REV;
11;      DC15; EXIT1;
6;      ZERO; (Read Number);
13;     REV; JS2; SETB37; J12=; (J if .); SET B20; -; DUP; J12<Z;
        CAB; DUP; SHLD+1; SHL+2; +; +; J13C15NZ; (J not end);
        REV; DUP;
12;     ERASE; EXIT1;
        (end of P109);
```

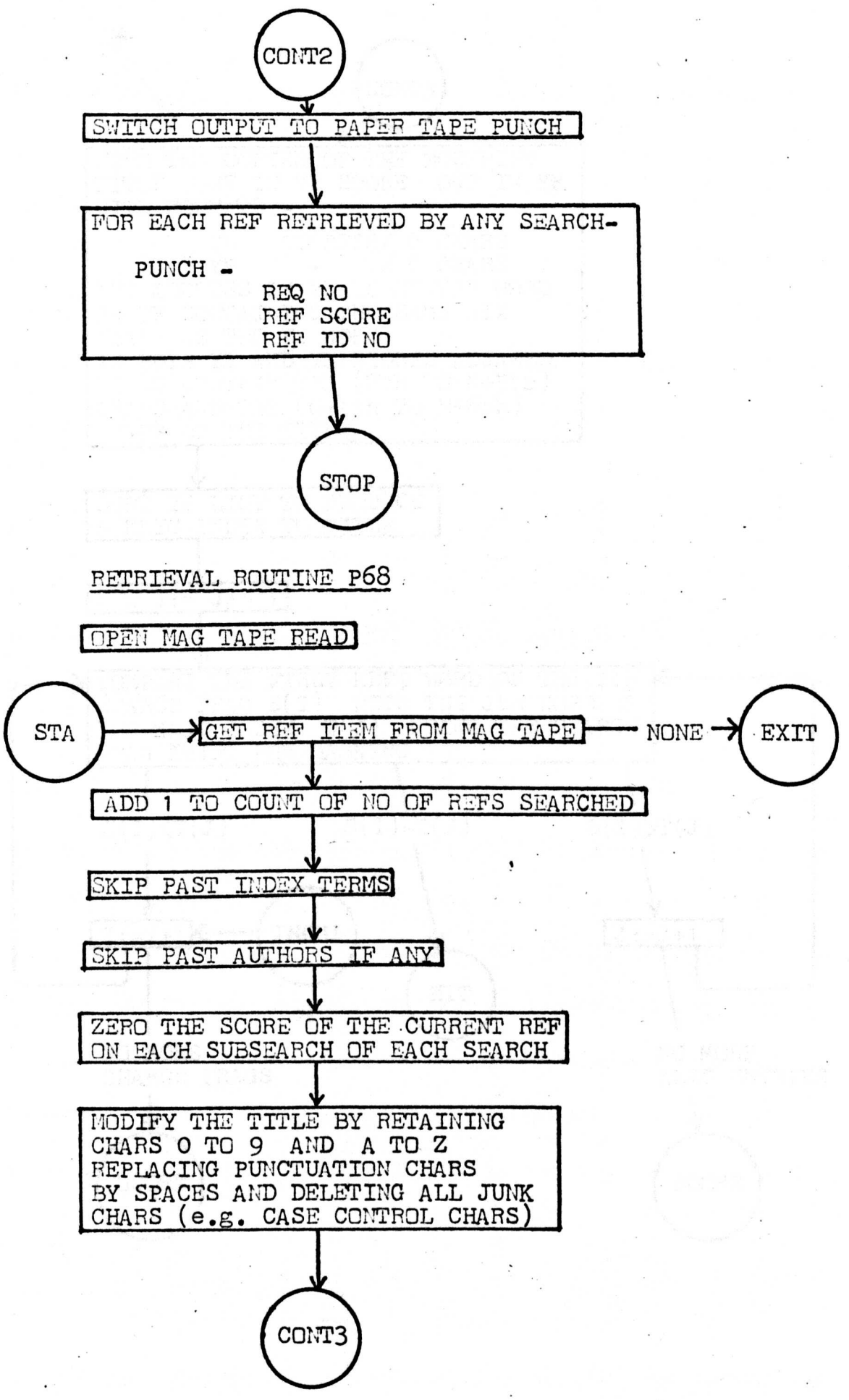
APPENDIX IV

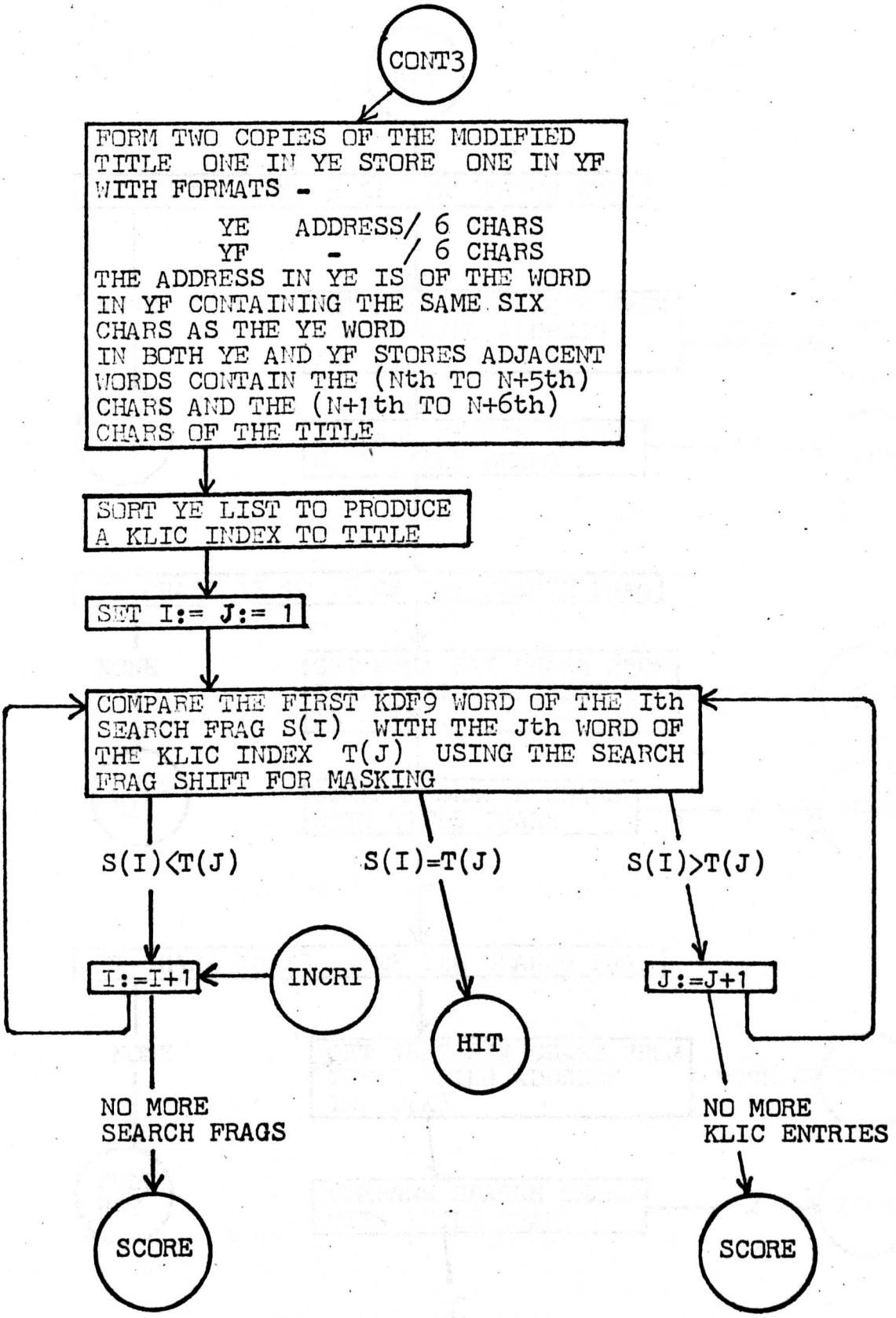
THE TITLE SEARCH PROGRAM

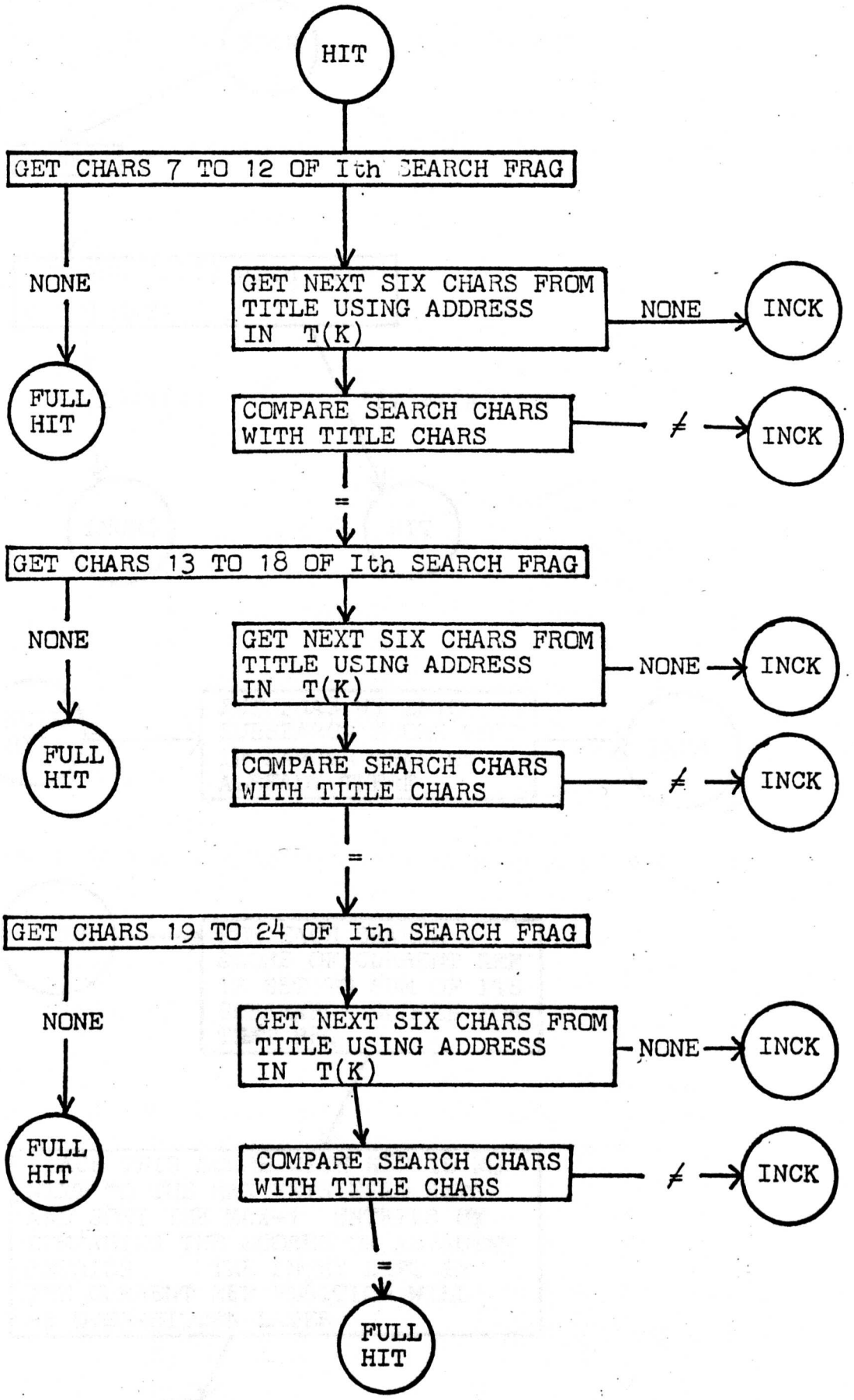
TITLE SEARCH PROGRAM

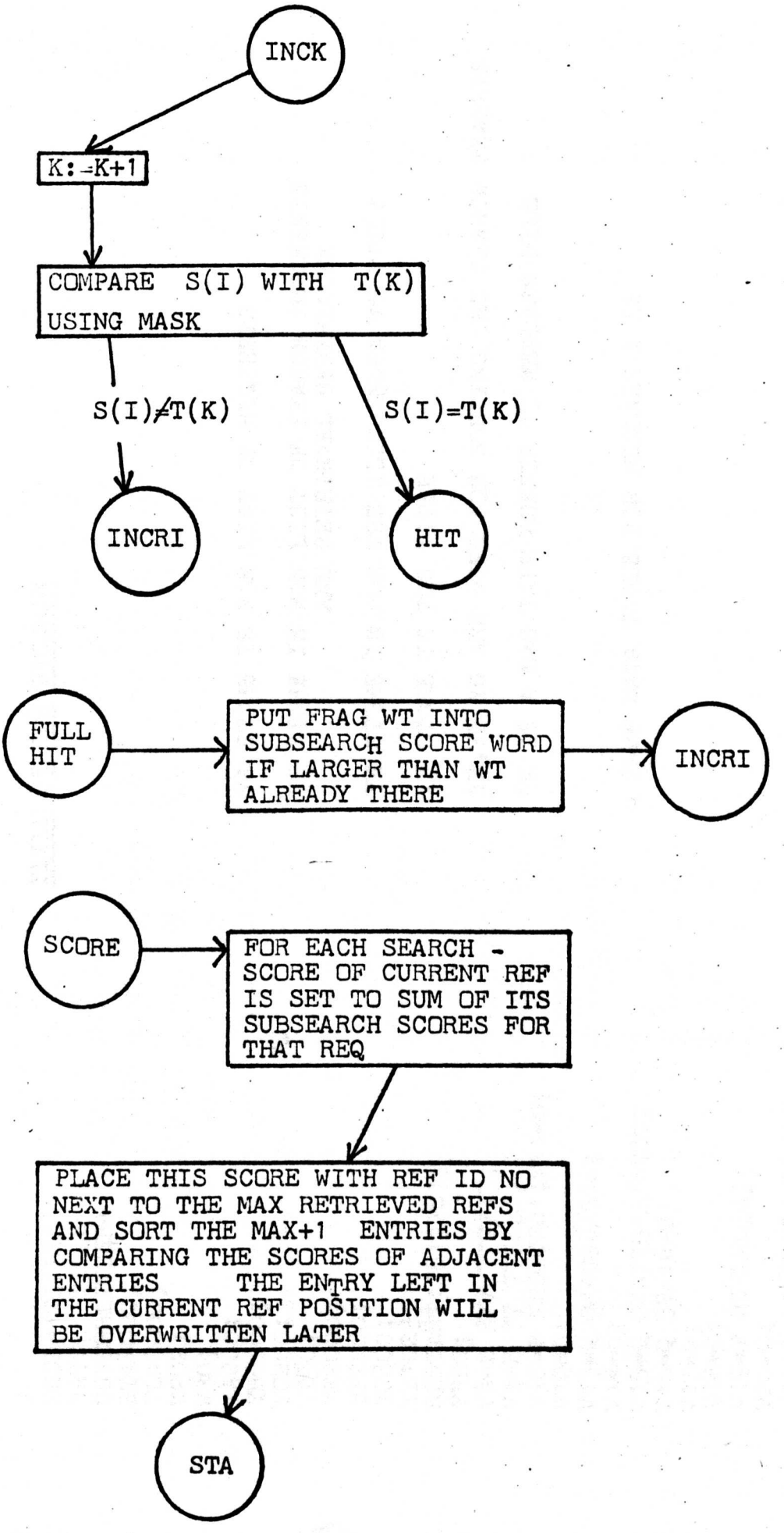



CONT1
FORM REQ LIST SORTED BY REQ NO WITH FORMAT - MAX+10/REQ/ADD ADDRESS IS OF WORD IN RETRIEVED REF LIST CORRESPONDING TO REQ. FORMAT OF RTRVD REF LIST WHICH IS ADDED ONTO SEARCH FRAG LIST IS - WD 1 MAX+10/REQ/- WDS 2-11 SUBSEARCH SCORE WDS WDS 11-(MAX+11) RETRIEVED REF SCORE WDS
CHANGE FIRST WORD OF SEARCH FRAG LIST ENTRIES TO - WT/NO CHARS/ADDRESS ADDRESS IS OF PARTICULAR SUBSEARCH OF PARTICULAR REQ IN WHICH FRAG IS USED
FIND DECK WITH MAG TAPE OF MEDLARS FILE
CALL RETRIEVAL SUBROUTINE P68 (SEE BELOW)
OPERATOR INTERRUPT
PRINT NO OF REFERENCES SEARCHED
GET FIRST SEARCH OUTPUT
PRINT REQ NO MAX OUTPUT REQD
GET RETRIEVED REF AND PRINT REF SCORE AND REF ID NO
NO MORE REFS RTRVD
GET NEXT SEARCH OUTPUT
NONE
CONT2


CONT2
SWITCH OUTPUT TO PAPER TAPE PUNCH
FOR EACH REF RETRIEVED BY ANY SEARCH-
PUNCH -
REQ NO
REF SCORE
REF ID NO
STOP
RETRIEVAL ROUTINE P68
OPEN MAG TAPE READ
STA
GET REF ITEM FROM MAG TAPE
NONE
EXIT
ADD 1 TO COUNT OF NO OF REFS SEARCHED
SKIP PAST INDEX TERMS
SKIP PAST AUTHORS IF ANY
ZERO THE SCORE OF THE CURRENT REF ON EACH SUBSEARCH OF EACH SEARCH
MODIFY THE TITLE BY RETAINING CHARS 0 TO 9 AND A TO Z REPLACING PUNCTUATION CHARS BY SPACES AND DELETING ALL JUNK CHARS (e.g. CASE CONTROL CHARS)
CONT3


CONT3
FORM TWO COPIES OF THE MODIFIED TITLE ONE IN YE STORE ONE IN YF WITH FORMATS -
YE ADDRESS/ 6 CHARS
YF - / 6 CHARS
THE ADDRESS IN YE IS OF THE WORD IN YF CONTAINING THE SAME SIX CHARS AS THE YE WORD
IN BOTH YE AND YF STORES ADJACENT WORDS CONTAIN THE (Nth TO N+5th) CHARS AND THE (N+1th TO N+6th) CHARS OF THE TITLE
SORT YE LIST TO PRODUCE A KLIC INDEX TO TITLE
SET I:= J:= 1
COMPARE THE FIRST KDF9 WORD OF THE Ith SEARCH FRAG S(I) WITH THE Jth WORD OF THE KLIC INDEX T(J) USING THE SEARCH FRAG SHIFT FOR MASKING
S(I)<T(J)
S(I)=T(J)
S(I)>T(J)
I:=I+1
INCRI
HIT
J:=J+1
NO MORE SEARCH FRAGS
NO MORE KLIC ENTRIES
SCORE
SCORE


HIT
GET CHARS 7 TO 12 OF Ith SEARCH FRAG
NONE
FULL HIT
GET NEXT SIX CHARS FROM TITLE USING ADDRESS IN T(K)
NONE
INCK
COMPARE SEARCH CHARS WITH TITLE CHARS
≠
INCK
=
GET CHARS 13 TO 18 OF Ith SEARCH FRAG
NONE
FULL HIT
GET NEXT SIX CHARS FROM TITLE USING ADDRESS IN T(K)
NONE
INCK
COMPARE SEARCH CHARS WITH TITLE CHARS
≠
INCK
=
GET CHARS 19 TO 24 OF Ith SEARCH FRAG
NONE
FULL HIT
GET NEXT SIX CHARS FROM TITLE USING ADDRESS IN T(K)
NONE
INCK
COMPARE SEARCH CHARS WITH TITLE CHARS
≠
INCK
=
FULL HIT


INCK
K:=K+1
COMPARE S(I) WITH T(K)
USING MASK
S(I)≠T(K)
S(I)=T(K)
INCRI
HIT
FULL HIT
PUT FRAG WT INTO SUBSEARCH SCORE WORD IF LARGER THAN WT ALREADY THERE
INCRI
SCORE
FOR EACH SEARCH - SCORE OF CURRENT REF IS SET TO SUM OF ITS SUBSEARCH SCORES FOR THAT REQ
PLACE THIS SCORE WITH REF ID NO NEXT TO THE MAX RETRIEVED REFS AND SORT THE MAX+1 ENTRIES BY COMPARING THE SCORES OF ADJACENT ENTRIES THE ENTRY LEFT IN THE CURRENT REF POSITION WILL BE OVERWRITTEN LATER
STA

```
P
DHCL41TITLES
BATCH  TITLE
SUPERBLITZER→
ST14000;
TL12000;
V202;
W30;
YA200;
YB6000;
YC200;
YD50;
YE2000;
YF2000;
YG800;
YZ1023;
RESTART;J81;J81;
PROG;
V4=B 377 777/17;
V5=B 77 777/31;
V10=P MEDCCF01;V11=0;
V12=0;
V13=Q 0/4096/512;
V14=0;
V15=Q 0/AYZ512/AYZ0;
V16=Q 32/16/0;
V17=0;
V18=0;
V19=P REELNO[Q];
V20=0;
V21=Q -1/AV18/AV20;
```

TITLE SEARCH PROGRAM

YA STORE IS FOR LIST OF REQUESTS

YB STORE IS FOR LIST OF SEARCH FRAGMENTS
AND RETRIEVED REFERENCES

YE STORE IS FOR THE KLIC INDEX TO TITLE

YF STORE IS FOR TITLE

YE,YF,YG ARE USED FOR SORTING THE SEARCH FRAGMTS

YZ IS A MAG TAPE BUFFER TO MEDLARS FILE

TAPE DATA TABLE FOR MEDLARS FILE

```
V22=B 17171717 17171717;
V23=B 12121212 12121212;
V30=P SEARCH**;
V31/32=P MAX*OUTPUT*REQD*;
V33/36=P ****DOC*SCORE********CITATION *NO*;                    OUTPUT HEADINGS
V37/38=P END*OF*SEARCH***;
V39=B 00000000 02020202;
V40=Q 10/0/2;
V41=Q 10/0/0;
V42=B 377 7777;
V43=B 177777;                                                   MASKS FOR TESTS ON PARTS OF KDF9 WORDS
V44=B 37 777/31;
V45=B 177 777/31;
V202=0;
V90/93=P ******TAG***SEARCH*NO*****WEIGHT;


JSP2801;J112;JS1;=V201;                                         READ RUN NUMBER
ZERO;=RM1;ZERO;=RM2;ZERO;=Q5;ZERO;=RM7;
3;JSP2801;J112;JS1;  JSP2801;J112;JS1;+;DUP;                    READ REQ NO, SUBSEARCH NO
 JSP2801;J112;JS1;SET10;+;                                      READ MAX OUTPUT REQUIRED
SHL32;REV;SHL16;OR;=YAOM1Q;
SHL16;DUP;=YD2;                                                 YA FORMAT:  MAX+10/REQ+SUB/-
4;JSP2801;J2;JS1;DC5;                                           READ INTEGER WEIGHT

    M2;=YD1;OR;=YEOM2Q;SET6;=C3;SET-12;=YDO;ZERO;=Q4;SET4;=C6;
 41;   JSP2800;J112;SETB34;-;J41≠Z;
 42;   JSP2800;J112;SETB34;J43=;                                DETECT START OF SEARCH FRAGMENT
```

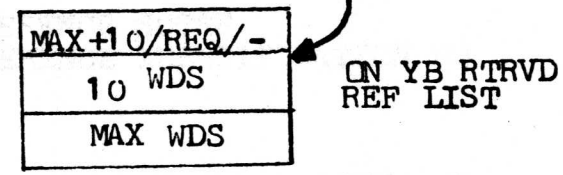
```
44;    SHL42;YDO;SHLD6;=YDO;ERASE;DC3;DC4;
       J42C3NZ;YDO;=YEOM2Q;SET-12;=YDO;DC6;
       SET6;=C3;J42;
43;ERASE;   NC4;YD1;=M3;YEOM3;C4;SHL32;OR;=YEOM3;
       C3;SET6;-;J45=Z;C3;YDO;
46;    J47C3Z;SHL6;DC3;J46;
47;    REV;SET-6;xD;CONT;SHA36;+;=YEOM2Q;DC6;
45;    SET1;=+M3;YEOM3;SHL12;
       M3;SET1;-;SHLD-12;ERASE;=YGOM7Q;
49;    J48C6Z;ZERO;=YEOM2Q;DC6;J49;
48;    YD2;J4;
2;     REV;ERASE;J3=Z;
       M1;=V1;M2;=V2;NC5;C5;=V200;

V90;JS1P29;V91;JSP29;V92;JSP29;V93;JS2P29;
       ZERO;NOT;SHL-12;V200;SHL32;SETAYGO;OR;JSL82;
       V200;=RC5;
       ZERO;=RC4;
51;    YGOM5Q;SHL-36;=RM2;SET5;=C2;
50;    YEOM2Q;=YBOM4Q;J50C2NZ;       J51C5NZ;

   ZERO;=RC2;V200;=C5;
77;J78C5Z;  DC5;YBOM2Q;DUP;SHL-32;=C4;  SET4;=C6;
   SETB34;JSP2802;   ZERO;DUP;=C3;
79;J76C4Z;DC4;J75C3NZ;ERASE;SET6;=C3;YBOM2Q;SHL12;DC6;
75;ZERO;SHLD6;JSP2802;DC3;J79;
76;ERASE;SETB34;JSP2802;
   SHL16;ZERO;SHLD16;SET47;V41;JSP2803;
   SHL-32;SET47;V40;JSP2803;
70;   J77C6Z;YBOM2Q;ERASE;DC6;J70;
```

```
SETS UP SEARCH FRAGMENT IN YE/F AS-

    WORD1      NO CHARS/REQSUB/WT
    WORD2    SHIFT/SIX CHARS
    WORDS3-5   AS WORD 2

 FRAG IS PADDED WITH ZEROS
SHIFT=-6x(2+X) WHEN THERE ARE
       6-X CHARS IN THE WORD

ALSO SETS UP A SORT KEY IN YG
CORRESPONDING TO THE YE ITEM
ITS FORMAT IS: ADD IN YE/FIRST SIX
                              CHARS
END OF INPUT
SORT YG LIST AND USE IT TO TRANSFER
YE TO YB NOW IN ORDER OF FIRST
SIX CHARS



PRINT --   SEARCH FRAG/REQ/WT
```

```
78; SETAYAO;=RM1;V1;=C1; ZERO;NOT;SHL16;SHL-32;SHL16; Q1;JSL82;
    ZERO;=RM1;V1;=C1;        V2;=RM2;
6;YAOM1;  DUP;SHL16;SHL-32;SET100;÷I;ERASE;
          V202;REV;DUP;=V202;-;J62≠Z;
          =YAOM1Q;J6C1NZ;J63;
62;=YBOM2Q;SETAYBO;M2;+;YAOM1;OR;=YAOM1;
   YAOM1Q;SHL-32;=C5;ZERO;DUP;
61;=YBOM2Q;DUP;DC5;J61C5NZ;ERASE;ERASE;J6C1NZ;
63;M2;V2;-;=V9;

V200;=RC2;SET5;=I2;
72;YBOM2;SHL16;SHL-32;SET100;÷I;REV;SET100;xD;
   CONT;V201;+;ZERO;=RM1;V1;=C1;
71;YAOM1;SHL16;SHL-32;J7=;
   SET1;=+M1;SET-1;=+C1;J71C1NZ;J1 2;
7;ERASE;YAOM1Q;DUP;SHL32;J112=Z;+;V201;-;  SHL32;YBOM2;
  SHC-16;SHL-16;SHLD16;=YBOM2Q;ERASE;J72C2NZ;

    V200;=V2P68;V1;=V1P68;
    SETAYBO;=VOP68;SETAYAO;=V10P68;
V10;SET4;OUT;SHL16;=V14;
V21;SET8;OUT;V23;V20;SHL-36;V22;AND;TOB;SHL16;=V11;
83;J84EN;ERASE;J83;
84;JSP68;
82;ERASE;
81;J82NEN;SET1;

8;SET3;JSP2802;V98P68;SET47;V40;JSP2803;V30;JS2P29;
  SET1;-;J113≠Z;SETAV10;JS10L34;DUMMY;DUMMY;DUMMY;
  SETAYAO;V1;SHL32;OR;V45;REV;JSL82;
```

SORT YA LIST BY REQUEST AND ADD IN ADDRESS FOR SCORES OF CURRENT AND RETRIEVED REFER--ENCES. YA FORMAT IS NOW---

YA: MAX+10/REQ/ADD



IN YB SEARCH FRAG LIST, REPLACE REQ BY ADD. EACH SUBSEARCH HAS A DIFFERENT ADDRESS.

OPEN MAG TAPE AND PASS OVER LIST ADDRESSES TO P68

CALL P68 WHICH COMPARES THE MEDLARS FILE WITH THE SEARCH FRAGMENT LIST

PRINTS NUMBER OF REFERENCES SEARCHED BY END OF TAPE OR AT OPERATOR INTERUPT

```
ZERO;=RM1;V1;=C1;
91;SET3;JSP2802;YAOM1Q;DUP;=RM2;=C2;J93C2Z;
SET-1;=+M2;EOM2Q;V30;JSP29;
ZERO;SHLD16;=C2;SET-10;=+C2;SET10;=+M2;
SHL-32;DUP;=V8;SET47;V40;JSP2803;
V31;JS1P29;V32;JSP29;
C2;SET47;V40;JSP2803;
V39;JSP29;V33;JSP29;V34;JSP29;V35;JSP29;V36;JS2P29;
9;EOM2Q;DUP;V42;AND;J92=Z;
ZERO;SHLD28;SET47;V41;JSP2803;
SHL-28;SET47;V40;JSP2803;
93;J9C2NZ;J91C1NZ;J94;
92;ERASE;J93;

94;   JS1P2802;SET2;JSP2802;
   SET-1;=RC15;SET10;=I15;Q15;=V41;
    ZERO;=RM1;V1;=C1;
101;YAOM1Q;DUP;=RM2;=C2;J103C2Z;  SET-1;=+M2;
    EOM2Q;ZERO;SHLD16;=C2;SET-10;
    =+C2;SET10;=+M2;SHL-32;=V8;
100;EOM2Q;DUP;V42;AND;J102=Z;
    ZERO;SHLD28;V8;SET47;V41;JSP2803;
    SET47;V41;JSP2803;SHL-28;SET47;V40;JSP2803;
103;J100C2NZ;J101C1NZ;J104;
102;ERASE;J103;
104;JS2P2802;
JS1P2802;SET2;JSP2802;JS2P2802;ZERO;OUT;
1;SET47;-;=C15;SHLC15;EXIT1;
```

PRINT OF REFERENCES RETRIEVED

PRINTS REQUEST NUMBER
PRINTS MAX OUTPUT REQUIRED

PRINTS REFERENCE SCORE
PRINTS REFERENCE IDENTIFICATION NO

END OF PRINT , START OF PUNCHING OF PAPER TAPE FOR INPUT TO MERGE,SORT, AND PRINT OF FULL MEDLARS ITEMS

PUNCHES REQUEST NO
PUNCHES REFERENCE SCORE
PUNCHES REFERENCE IDENTIFICATION NO

END OF PROGRAM

INTEGER CONVERSION

```
111;JS114;SETB21;J115;
112;JS114;SETB22;J115;
113;JS114;SETB23;J115;                                  FAILURE EXITS
114;J116EN;SET47;V40;JSP2803;J114;
116;V37;JS1P29;V38;JSP29;EXIT1;
115;JSP2802;JS2P2802;ZERO;OUT;


P29;
3;SET8;=C13;
4;ZERO;SHLD6;JSP2802;DC13;J4C13NZ;                      PRINT A KDF9 WORD
ERASE;EXIT1;                                            AS EIGHT CHARS
1;SET2;JSP2802;J3;                                      OF TEXT
2;JS3;SET2;JSP2802;EXIT1;


P68V200;                                                P68 IS THE RETRIEVAL ROUTINE
V98=0;
  SETAV10P0;JS12L34;J113P0;DUP;=V17P0;J35;
136;ERASE;
36;V17P0;SETAV10P0;JSL34;J8P0;DUP;=V17P0;               READ REFERENCE FROM MAG TAPE
35;   V98;SET1;+;=V98;                                  COUNT NO OF REFS SEARCHED.
   =RM7;MOM7Q;=C7;DC7;MOM7Q;SHL-28;=V200;               REF IDENTIFICATION NUMBER.
   MOM7Q;SETB77;AND;DUP;NEG;=+C7;=+M7;                  SKIP PAST INDEX TERMS
   SETB77; C7;J71<Z;
6;  J136C7Z;MOM7Q;SHL-42;J6≠;ERASE;SET-1;=+M7;SET1;=+C7;   SKIP PAST AUTHORS
                                                        IF ANY
  V10;=RM1;V1;=C1;V200;                                 SET THE INITIAL SCORE IN EACH
5;J51C1Z;MOM1Q;DUP;=RC8;=M8;J5C8Z;SET10;=C8;            SUBSEARCH OF EACH SEARCH TO
52;DUP;=MOM8Q;J52C8NZ;J5;                               ZERO, AND INSERT THE CURRENT
51;ERASE;                                               REF IDENTIFICATION NO
```

```
            ZERO;=RM9;ZERO;=C11;
    MOM7;SHL6;SHL-6;=MOM7;ZERO;ZERO;=YEOM9;ZERO;=YFOM9;
7;  ERASE;J73C7Z;MOM7Q;SET8;=C10;
8;   J7C10Z;DC10;ZERO;SHLD6;
     ZERO;J85=;SETB33;J81=;
     DUP;SETB72;-;J83>Z;
     DUP;SETB20;-;J83<Z;
     DUP;SETB31;-;J84≤Z;
     DUP;SETB41;-;J83<Z;
84;  SHL42;YFOM9Q;SHLD6;DUP;=YFOM9;SHL12;
M9;SHLD-12;ERASE;=YEOM9;ERASE;J8;
83;SETB10;-;J8<Z;ZERO;
85;  ERASE;YFOM9Q;SHL6;DUP;=YFOM9;SHL12;
     M9;SHLD-12;ERASE;=YEOM9;J8;

71;SETB21;J74;
72;SETB22;J74;
73;SETB23;
74;SET2;DUP;JSP2802;JSP2802;
   SETB27;JSP2802;JSP2802;SET2;JSP2802;ZERO;OUT;

81;ERASE;ERASE;M9;J36=Z;SET6;=C9;
82;YFOM9Q;SHL6;DUP;=YFOM9;SHL12;
   M9;SHLD-12;ERASE;=YEOM9;J82C9NZ;
    M9;=V3;ZERO;NOT;SHL-12;V3;SET4;-;SHL32;
    SETAYE5;OR;JSL82;  ZERO;=RM1;ZERO;=RM5;
    V2;=C1;SET5;=I1;V3;=C5;SET-4;=+C5;
    YB1M1Q;DUP;SHL-36;SHL36;SHA-36;=C15;
    SHL12;SHLC15;YE5M5Q;SHL12;
```

IN THE TITLE, REPLACE ALL CHARS OTHER THAN 0 TO 9, AND A TO Z BY SPACE CHARS, UNLESS THEY ARE JUNK CHARS WHICH ARE DELETED

SET UP THE TITLE IN YE AND YF WITH FORMATS:

YE: ADD/SIX CHARS

YF: -/SIX CHARS

FAILURE EXITS

SORT THE YE LIST TO FORM A KLIC INDEX TO THE TITLE

```
J1;
116;ERASE;                                          COMPARISON OF FIRST SIX CHARS OF THE
12;REV;ERASE;J13C1Z;YB1M1Q;                         SEARCH FRAGMENT WITH SIX CHARS
   DUP;SHA-4;=Q15;SHL12;SHLC15;REV;                 FROM THE KLIC INDEX
1; DUPD;SHLC15;-;DUP;J11=Z;
   J12<Z;ERASE;J13C5Z;YE5M5Q;SHL12;J1;

11;ERASE;Q5TOQ6;YE4M5;                                   FIRST SIX CHARS IN AGREEMENT
123;SHL-36;=M8;     M1;SET5;-;=M2;  V3;M8;-;=C3;
     YB2M2;DUP;J14=Z;C3;SET6;-;J122<Z;                   HIT IF NO MORE CHARS IN S FRAG
     DUP;SHA-4;=Q14;SHL12;SHLC14;                        CHECK TITLE END NOT EXCEEDED
     YF6M8;SHL12;SHLC14;-;J121≠Z;                GET NEXT SIX CHARS TITLE,J TO 121 IF NOT=
     YB3M2;DUP;J14=Z;C3;SET12;-;J122<Z;
     DUP;SHA-4;=Q14;SHL12;SHLC14;                        REPEAT FOR CHARS 13-18
     YF12M8;SHL12;SHLC14;-;J121≠Z;
     YB4M2;DUP;J14=Z;C3;SET18;-;J122<Z;                  REPEAT FOR CHARS 19-24
     DUP;SHA-4;=Q14;SHL12;SHLC14;
     YF18M8;SHL12;SHLC14;-;J121≠Z;J114;                  IF NEXT ITEM ON KLIC LIST HAS SAME
                                                         FIRST CHARS AS CURRENT KLIC ENTRY
122;ERASE;                                               THEN COMPARE THIS WITH SEARCH
121;DUPD;ERASE;YE5M6Q;SHL12;SHLC15;-;J12≠Z;YE4M6;J123;    FRAGMENT  ALSO

14;(hit);ERASE;                                          ENTER WEIGHT AS SUBSEARCH SCORE
114;YBOM2;ZERO;SHLD16;SHL20;V200;OR;                     BUT ONLY IF IT IS GREATER THAN
    REV;SHL-16;=M9;MOM9;MAX;REV;ERASE;=MOM9;J12;         THE SCORE ALREADY THERE

 13;ERASE;                                               TITLE OR SEARCH FRAG LIST
 15;V10;=RM1;V1;=C1;SET-2;=M14;                          EXHAUSTED
 159;J42NEN;J36C1Z;EOM1Q;DUP;=RM2;=C2;J159C2Z;
     M2;=M15;SET-1;=+M2;EOM2Q;SHL-32;=C2;
```

```
EOM2Q;
EOM2Q;SHL-20;SHL20;+;
EOM2Q;SHL-20;SHL20;+;
EOM2Q;SHL-20;SHL20;+;
EOM2Q;SHL-20;SHL20;+;
EOM2Q;SHL-20;SHL20;+;
EOM2Q;SHL-20;SHL20;+;
EOM2Q;SHL-20;SHL20;+;
EOM2Q;SHL-20;SHL20;+;
EOM2;SHL-20;SHL20;+;=EOM2;
EOM2;SHL-20;J159=Z;
```

FOR EACH REQUEST, FORM THE SCORE OF THE CURRENT REF BY ADDING ITS SUBSEARCH SCORES

```
    EOM2Q;EOM2Q;
158; DUPD;SIGN;J157<Z;=M14M2;J156C2Z;EOM2Q;J158;
156;SET1;=+M2;=M14M2;J159;
157; REV;=M14M2Q;=M14M2;J159;
```

SORT THE CURRENT REFERENCE INTO THE RETRIEVED LIST IF IT DOES NOT MOVE INTO THIS LIST IT IS EFFECTIVELY DELETED

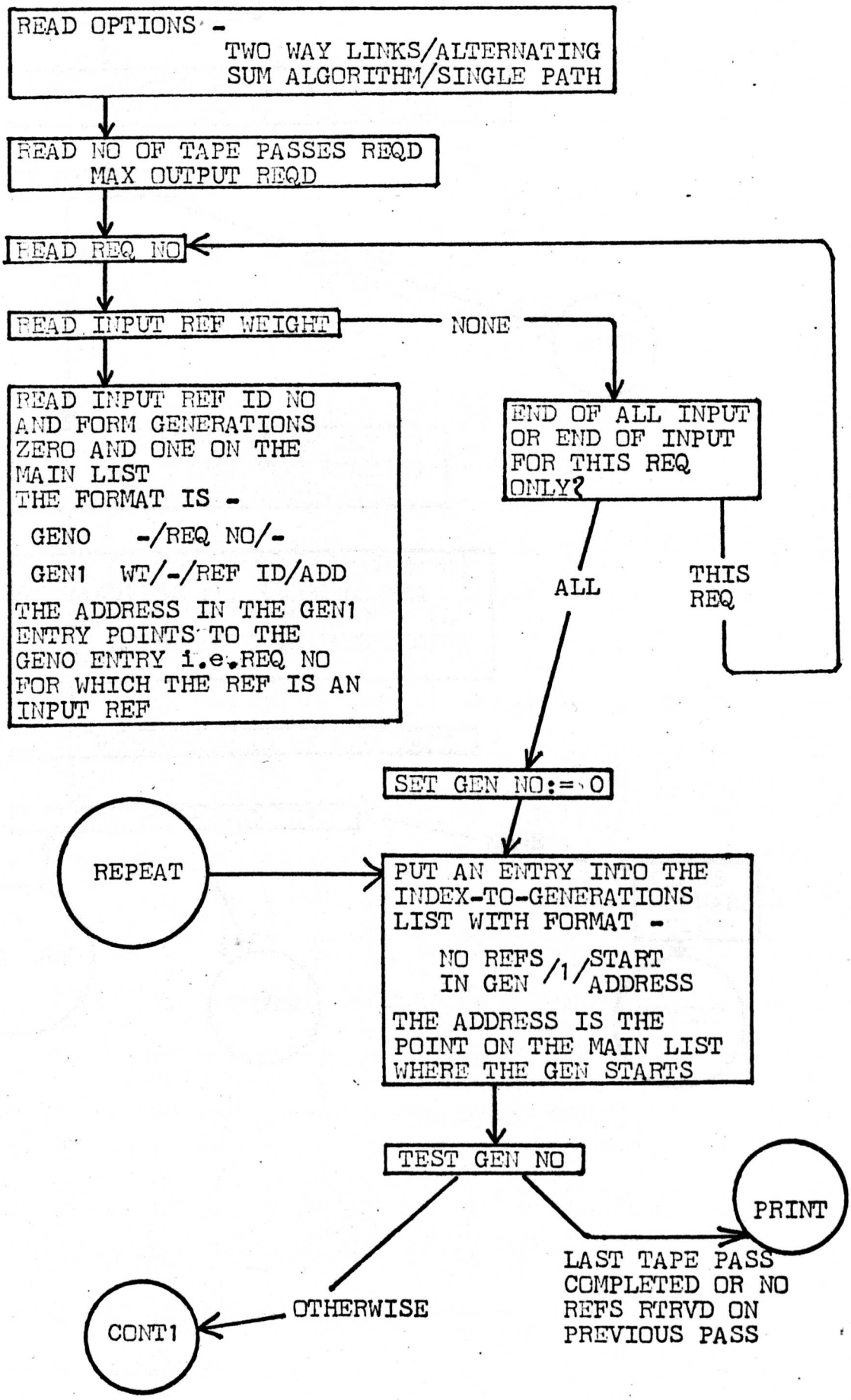
```
37;JS43;SETB27;J44;
38;JS43;SETB30;J44;
39;JS43;SETB31;J44;
40;JS45;SETB20;J44;
41;JS45;SETB21;J44;
42;JS45;SETB22;J44;
43;SETB23;JSP2802;EXIT1;
44;JSP2802;SET2;JSP2802;ZERO;OUT;
45;SETB24;JSP2802;EXIT1;
```

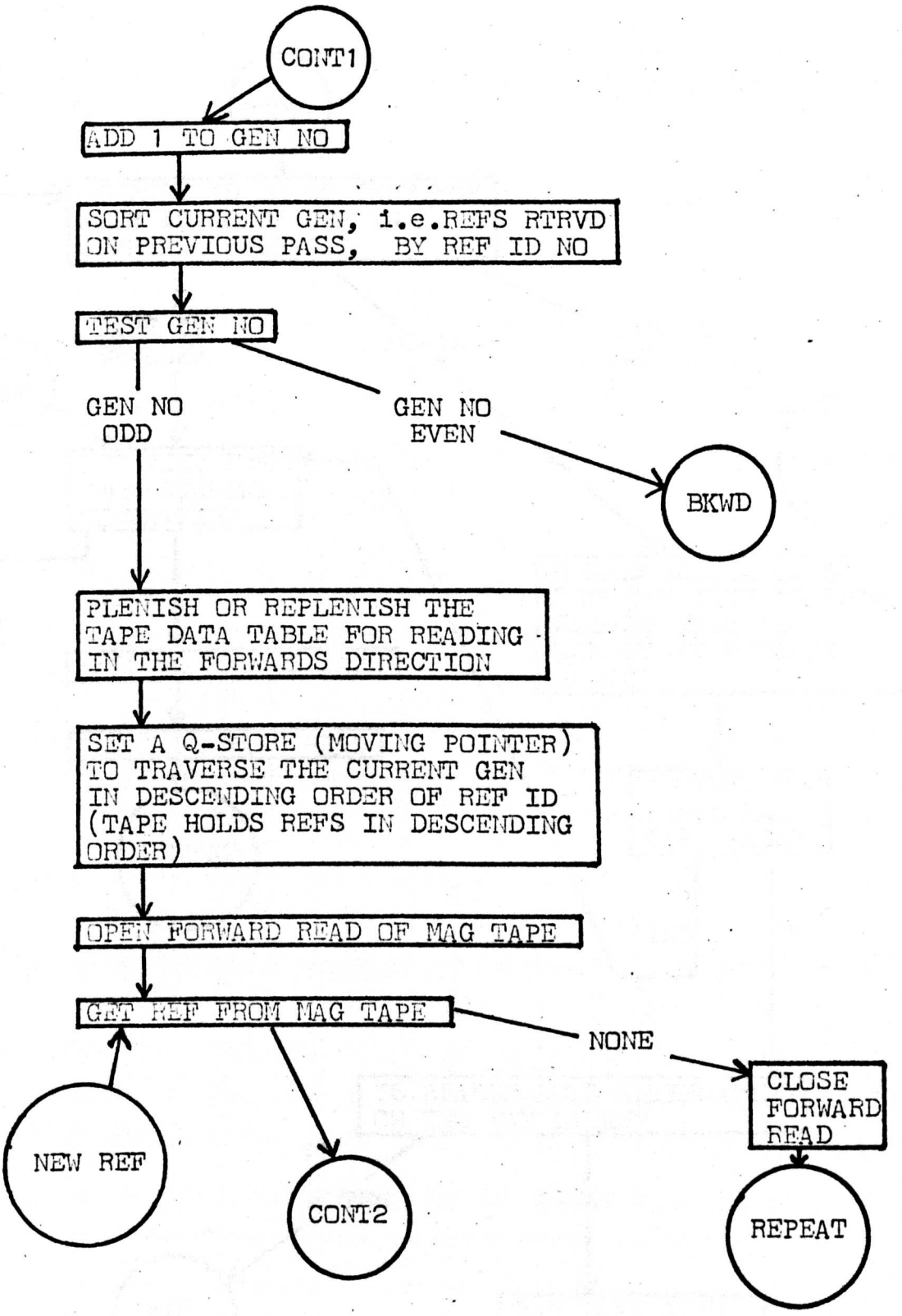
FAILURE EXITS

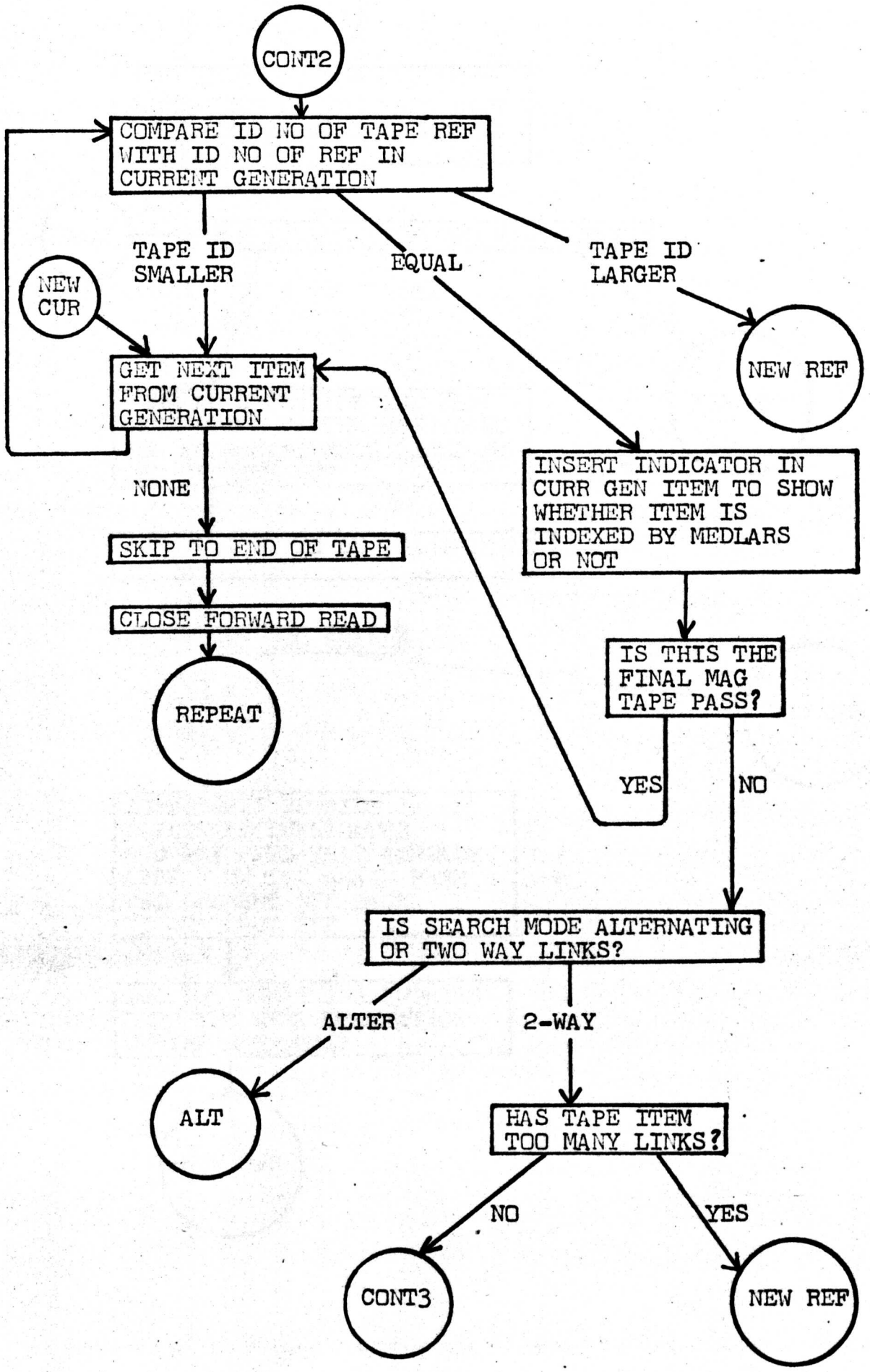
APPENDIX V

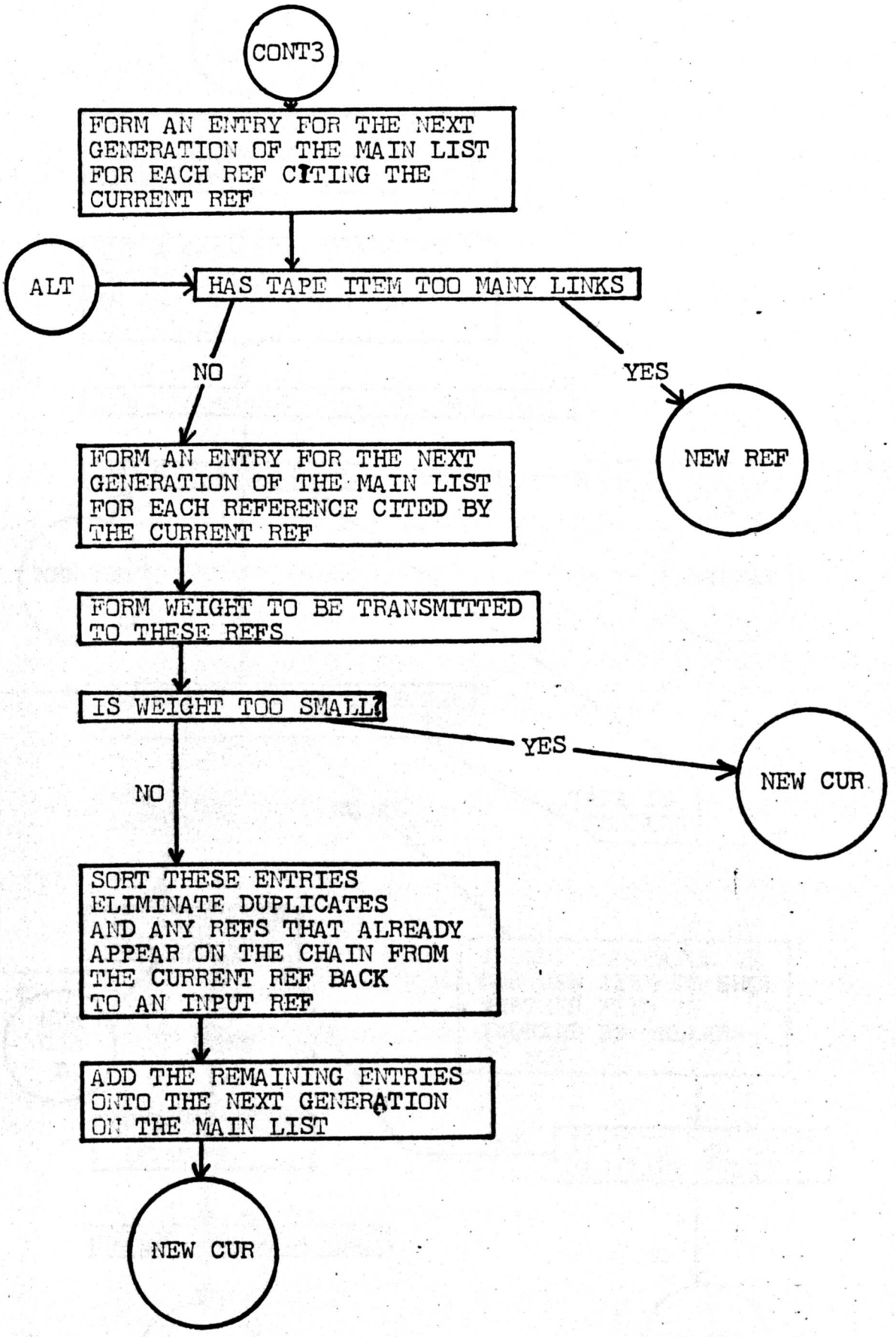
THE CITATION SEARCH PROGRAM

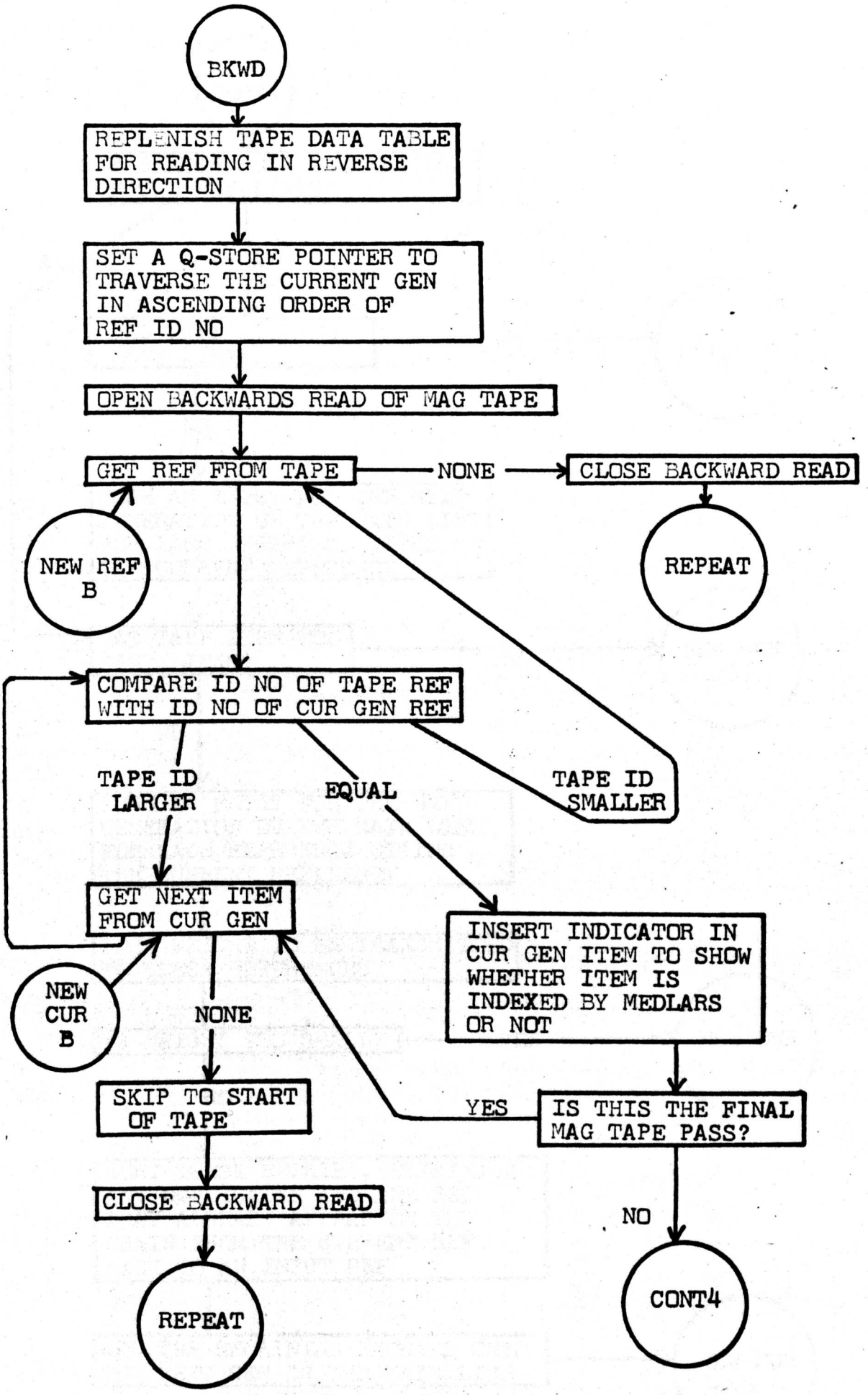
CITATION SEARCH PROGRAM

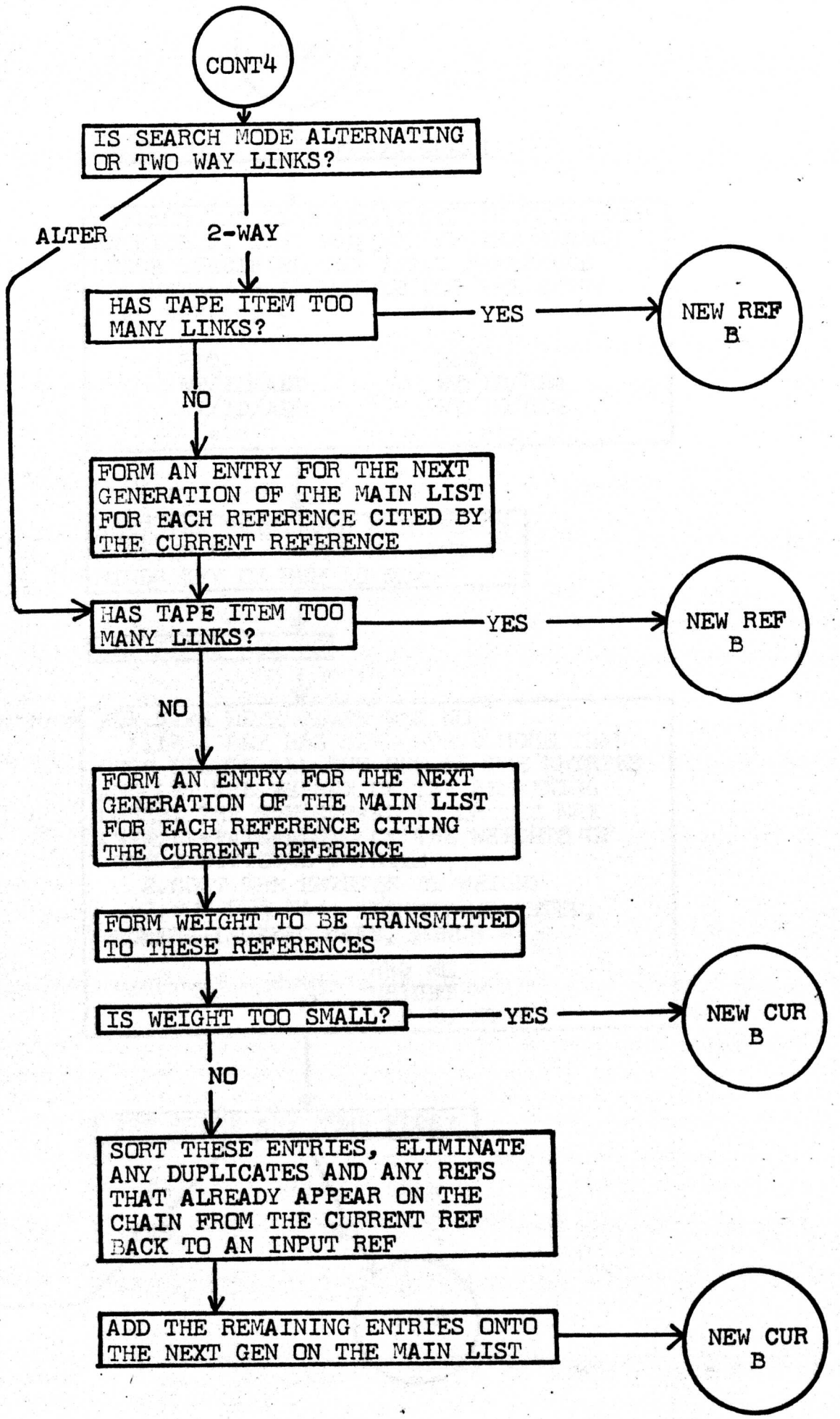


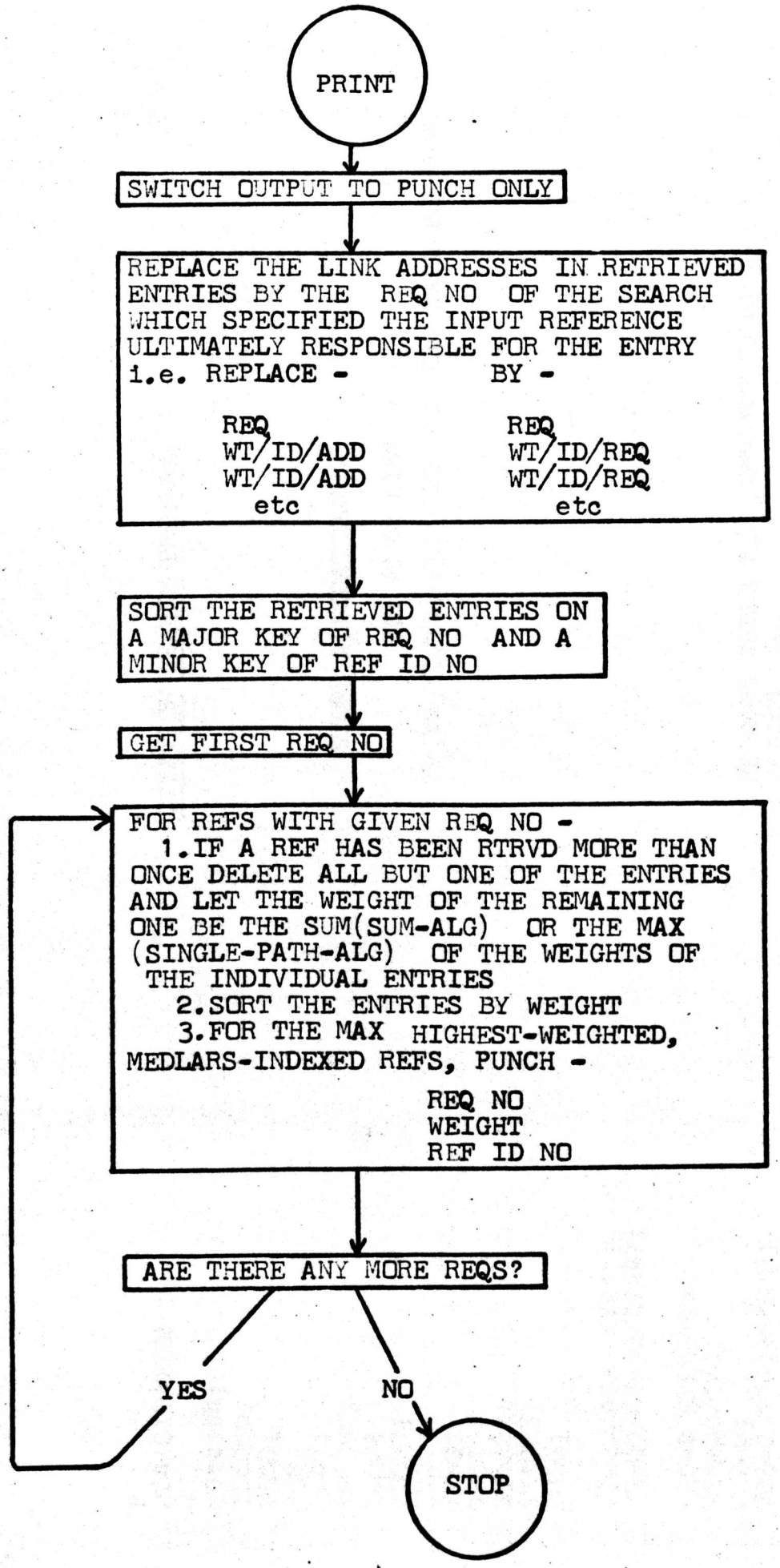

CONT1
ADD 1 TO GEN NO
SORT CURRENT GEN, i.e.REFS RTRVD ON PREVIOUS PASS, BY REF ID NO
TEST GEN NO
GEN NO ODD
GEN NO EVEN
BKWD
PLENISH OR REPLENISH THE TAPE DATA TABLE FOR READING IN THE FORWARDS DIRECTION
SET A Q-STORE (MOVING POINTER) TO TRAVERSE THE CURRENT GEN IN DESCENDING ORDER OF REF ID (TAPE HOLDS REFS IN DESCENDING ORDER)
OPEN FORWARD READ OF MAG TAPE
GET REF FROM MAG TAPE
NONE
CLOSE FORWARD READ
REPEAT
NEW REF
CONT2


CONT2
COMPARE ID NO OF TAPE REF WITH ID NO OF REF IN CURRENT GENERATION
TAPE ID SMALLER
EQUAL
TAPE ID LARGER
NEW CUR
NEW REF
GET NEXT ITEM FROM CURRENT GENERATION
INSERT INDICATOR IN CURR GEN ITEM TO SHOW WHETHER ITEM IS INDEXED BY MEDLARS OR NOT
NONE
SKIP TO END OF TAPE
CLOSE FORWARD READ
REPEAT
IS THIS THE FINAL MAG TAPE PASS?
YES
NO
IS SEARCH MODE ALTERNATING OR TWO WAY LINKS?
ALTER
2-WAY
ALT
HAS TAPE ITEM TOO MANY LINKS?
NO
YES
CONT3
NEW REF


CONT3
FORM AN ENTRY FOR THE NEXT GENERATION OF THE MAIN LIST FOR EACH REF CITING THE CURRENT REF
ALT
HAS TAPE ITEM TOO MANY LINKS
NO
YES
NEW REF
FORM AN ENTRY FOR THE NEXT GENERATION OF THE MAIN LIST FOR EACH REFERENCE CITED BY THE CURRENT REF
FORM WEIGHT TO BE TRANSMITTED TO THESE REFS
IS WEIGHT TOO SMALL?
YES
NEW CUR
NO
SORT THESE ENTRIES ELIMINATE DUPLICATES AND ANY REFS THAT ALREADY APPEAR ON THE CHAIN FROM THE CURRENT REF BACK TO AN INPUT REF
ADD THE REMAINING ENTRIES ONTO THE NEXT GENERATION ON THE MAIN LIST
NEW CUR


BKWD
REPLENISH TAPE DATA TABLE FOR READING IN REVERSE DIRECTION
SET A Q-STORE POINTER TO TRAVERSE THE CURRENT GEN IN ASCENDING ORDER OF REF ID NO
OPEN BACKWARDS READ OF MAG TAPE
GET REF FROM TAPE
NONE
CLOSE BACKWARD READ
NEW REF B
REPEAT
COMPARE ID NO OF TAPE REF WITH ID NO OF CUR GEN REF
TAPE ID LARGER
EQUAL
TAPE ID SMALLER
GET NEXT ITEM FROM CUR GEN
NEW CUR B
NONE
INSERT INDICATOR IN CUR GEN ITEM TO SHOW WHETHER ITEM IS INDEXED BY MEDLARS OR NOT
SKIP TO START OF TAPE
YES
IS THIS THE FINAL MAG TAPE PASS?
CLOSE BACKWARD READ
NO
REPEAT
CONT4


CONT4
IS SEARCH MODE ALTERNATING OR TWO WAY LINKS?
ALTER
2-WAY
HAS TAPE ITEM TOO MANY LINKS?
YES
NEW REF B
NO
FORM AN ENTRY FOR THE NEXT GENERATION OF THE MAIN LIST FOR EACH REFERENCE CITED BY THE CURRENT REFERENCE
HAS TAPE ITEM TOO MANY LINKS?
YES
NEW REF B
NO
FORM AN ENTRY FOR THE NEXT GENERATION OF THE MAIN LIST FOR EACH REFERENCE CITING THE CURRENT REFERENCE
FORM WEIGHT TO BE TRANSMITTED TO THESE REFERENCES
IS WEIGHT TOO SMALL?
YES
NEW CUR B
NO
SORT THESE ENTRIES, ELIMINATE ANY DUPLICATES AND ANY REFS THAT ALREADY APPEAR ON THE CHAIN FROM THE CURRENT REF BACK TO AN INPUT REF
ADD THE REMAINING ENTRIES ONTO THE NEXT GEN ON THE MAIN LIST
NEW CUR B


PRINT
SWITCH OUTPUT TO PUNCH ONLY
REPLACE THE LINK ADDRESSES IN RETRIEVED ENTRIES BY THE REQ NO OF THE SEARCH WHICH SPECIFIED THE INPUT REFERENCE ULTIMATELY RESPONSIBLE FOR THE ENTRY
i.e. REPLACE - BY -
REQ WT/ID/ADD WT/ID/ADD etc
REQ WT/ID/REQ WT/ID/REQ etc
SORT THE RETRIEVED ENTRIES ON A MAJOR KEY OF REQ NO AND A MINOR KEY OF REF ID NO
GET FIRST REQ NO
FOR REFS WITH GIVEN REQ NO -
1. IF A REF HAS BEEN RTRVD MORE THAN ONCE DELETE ALL BUT ONE OF THE ENTRIES AND LET THE WEIGHT OF THE REMAINING ONE BE THE SUM(SUM-ALG) OR THE MAX (SINGLE-PATH-ALG) OF THE WEIGHTS OF THE INDIVIDUAL ENTRIES
2. SORT THE ENTRIES BY WEIGHT
3. FOR THE MAX HIGHEST-WEIGHTED, MEDLARS-INDEXED REFS, PUNCH -
REQ NO
WEIGHT
REF ID NO
ARE THERE ANY MORE REQS?
YES
NO
STOP

```
P
CL41CITATION
→
ST14144;
TL6000;
V500;
W30;
YA47;
YB8000;
YG50;
YL500;
YM20;
YV49;
YW511;
YY1023;
YZ1023;
PROG;

V2=Q0/AYA0/AYA15;
V3=Q0/AYA32/AYA47;

V20=P MILLCITA;
V21=Q0/1/0;
V22=0;
V23=Q 1/4096/512;
V24=0;
V25=Q AV10/AYY512/AYY0;
V26=Q32/16/5;
V27=Q 0/0/AYW0;
```

CITATION SEARCH PROGRAM

STORE ALLOCATIONS:-

YB: MAIN LIST

YV,YW,YY,YZ, ALL USED FOR MAG TAPE BUFFERS

TAPE DATA TABLE FOR FORWARDS READING OF CITATION FILE

```
V30=P MILLCITA;
V31=Q 0/1/0;
V32=0;
V33=Q 1/4096/512;
V34=0;
V35=Q AV0/AYZ512/AYZ0;
V36=Q 32/16/5;
V37=Q 100/0/AYV0;
V44=Q 10/10/0;

V45=256;
V150=F 0.005;
V2;=V2P2802;V3;=V3P2802;
J2;
1;JSP2801;EXIT1;SET47;-;=C15;SHLC15;EXIT2;(read integer);
2;(start of prog);
JS1;J111;=V40;(option on two way links or alternating);
JSP2801;J111;FLOAT;=V500;(option on sum or single path alg);
JS1;J111;=V41;(no of tape passes);
JS1;J111;=V42;(max output);
SETAYB1;=RM3;(Q3  0/1/AYB1);(req  nos);
SETAYB101;=RM4;(Q4  0/1/AYB101);(generat one);
3;JS1;J111;SHL16;=MOM3Q;(YB1  -/REQ/-);

5;JS1;J6;SET47;FLOAT;SHL-33;SHL33;
JS1;J111;SHL16;OR;
 M3;SET1;-;OR;=MOM4Q;J5;
(YB101on  wt  med  item  absadd);
```

TAPE DATA TABLE FOR BACKWARDS READING
OF CITATION FILE

READS
DATA
FROM
PAPER
TAPE

YB LIST HAS FORMAT:-

GEN ZERO: - /REQ/ -

GEN ONE: WT/MED/REF ID/ADDRESS TO
BIT GEN ZERO

```
6;ZERO;J7=;SET1;-;J8=Z;J111;(input fail);
7;ERASE;J3;
8;NC3;C3;=YBO;(no of req);
(end of paper tp input);
V20;SET4;OUT;SHL16;DUP;=V24;=V34;(dv no);
SETAYGO;=RM5;ZERO;=YGO;(YGO is no of passes completed);
Q4TOQ2;
40;V45;SHL-1;=V45;Q2TOQ4;M4;=RM2;(Q2 set for next gen 0/1/stadd);
C4;=+M4;NC4;Q4;
SET1;=+M5;=MOM5;(M5 list is len/1/stadd  of currgen);
```

YG LIST IS INDEX TO GENERATIONS AND HAS FORMAT:

YG: NO OF REFS IN GENERAT / 1 / START ADD

```
V41;YGO;-;J30=Z;(j  if last pass completed);
J30C4Z;(j  if last gen empty);
YGO;SET1;+;=YGO;(incr no of passes);
9;ZERO;NOT;SHL16;SHL-32;SHL16;MOM5;JSL82;(sort curgen by item no);
YGO;SHL47;SHL-47;J20=Z;
```

SORT CURRENT GEN BY REF ID NO, AND GO TO 10 OR 20 FOR FORWARD OR BKWD PASS UP MAG TAPE

```
10;V20;=V0;
V21;=V1;
V22;=V2;
V23;=V3;
V24;=V4;
V25;YGO;SET1;-;J11≠Z;SHL16;SHL-16;
11;=V5;(omit rerf to last TDT if on first pass);
V26;=V6;
V27;=V7;
MOM5;=Q1;
C1;SET1;-;=+M1;SET-1;=I1;(Q1 set for  desc len/-1/lastadd);
```

RESET TAPE DATA TABLE FOR FORWARD READ

```
65;SETAV0;JS12L35;J111;=V8;(open fwd read);64;

16;MOM1Q;=V9;
14;V8;=RM8;E8M8;SHL-32;(tape item no);
V9;SHL16;SHL-32;(YB item no);

SIGN;(+1b<t   0equal   -1b>t);
DUP;J12=Z;J13<Z;

18;(b<t  get new t);V8;SETAV0;JSL35;J15;=V8;J14;
13;(b>t  get new b);J31C1Z;MOM1Q;=V9;J14;
12;(b=t); ERASE;

E8M8;SETB77;AND;J81=Z;SET1;=M15;M1M15;SET1;SHL32;OR;=M1M15;
81;  V41;YG0;-;J13=Z;

ZERO;=YL0;
SET1;=RM9;
V40;J71≠Z;(j  if alternating searching);
E5M8;SHL-32;DUP;=YL0;(no of children);
V45;-;J18>Z;(omit if too many links);
YL0;J71=Z;
E5M8;SHL16;SHL-32;M8;+;SET5;+;=RM10;E5M8;SHL-32;=C10;
(Q10  no ch/1/stadd);
19;MOM10;
ZERO;SHLD16;=YLOM9Q;DC10;J17C10Z;
ZERO;SHLD16;=YLOM9Q;DC10;J17C10Z;
ZERO;SHLD16;=YLOM9Q;DC10;J17C10Z;
ERASE;SET1;=+M10;J19;
```

OPEN FORWARD READ

COMPARE ITEM ID NO ON TAPE
WITH ITEM ID NO IN CURRENT
GENERATION

ITEM ID NOS EQUAL
INSERT INDICATOR OF WHETHER
ITEM IS A MEDLARS ITEM OR NOT.
IF LAST PASS RETURN TO
COMPARISON

PUT ID NOS OF CITING REFS
INTO YL LIST

OMIT THIS IF SEARCHING IN
ALTERNATING MODE

```
17;ERASE;
71;(get pars);
E6M8;SHL-32;YLO;+;DUP;=YLO;J18=Z;YLO;
V45;-;J18>Z;(omit if too many links);
E6M8;SHL32;SHL-32;M8;+;SET6;+;=RM10;E6M8;SHL-32;=C10;
(Q10  no par/1/stadd);                                  ADD ID NOS OF CITED REFS
91;MOM10;J92C10Z;
ZERO;SHLD16;=YLOM9Q;DC10;J92C10Z;
ZERO;SHLD16;=YLOM9Q;DC10;J92C10Z;
ZERO;SHLD16;=YLOM9Q;DC10;J92C10Z;
ERASE;SET1;=+M10;J91;

92;ERASE;
(prune starts here);
YLO;=RC9;
SETAYL1;=M9;Q9;ZERO;NOT;SHL-32;REV;JSL82;               SORT RELATIVES
SET1;=M9;(Q9 set for yl);

ZERO;=RM11;
M1;I1;-;=M10;
M10;=V152;
MOM10;(end of chain);
ZERO;SHLD15;SHL33;(old wt);
YLO;SET47;FLOAT;+F;SHL-33;SHL33;=V50;(new wt);          CALCULATE WEIGHT TO BE
=V151;V50;V150;-F;J13<Z;V151;                           TRANSMITTED TO RELATIVES.
SHL1;ZERO;SHLD16;=YMOM11Q;(item no);                    OMIT IF THIS LESS THAN THRESHHOLD
SHL-32;=M10;
98;MOM10;SHL16;ZERO;SHLD16;REV;SHL-32;DUP;              ASSEMBLE IN YM    CHAIN  OF REFS
J93=Z;=M10;=YMOM11Q;J98;                                LEADING FROM THE REQUEST SET TO
                                                        THE CURRENT REFERENCE (INCL BOTH)
```

```
93;ERASE;ERASE;NC11;ZERO;=V51;                    ELIMINATE ANY RELATIVE THAT ALREADY APPEARS
94;C11;=RC12;                                     ON THIS CHAIN
YLOM9Q;V51;J95=;DUP;=V51;
96;YMOM12Q;J95=;J96C12NZ;
SHL16;V152;OR;V50;OR;97;=MOM2Q;J94C9NZ;J13;       ADD OTHERS INTO NEXT GEN IN YB.
95;ERASE;J94C9NZ;J13;                             RETURN TO COMPARISON SEQUENCE

31;V8;SETAV0;JSL35;J15;=V8;J31;
15;SET2;-;J109≠Z;                                   END OF FORWARD PASS
51;V8;SETAV0;JS11L35;ERASE;DUMMY;DUMMY;J40;
(end of fwd read loop);


20;V30;=V10;V31;=V11;V32;=V12;                    START OF BACKWARDS PASS UP MAG TAPE
V33;=V13;V34;=V14;V35;=V15;
V36;=V16;                                         REPLENISH TAPE DATA TABLE FOR
V37;=V17;                                               BACKWARDS READ
MOM5;=Q1;(Q1 set for ascend  len/1/STADD);
SETAV10;JS12L36;J108;=V8;(open rev read);         OPEN BACKWARDS READ


116;MOM1Q;=V9;                                       COMPARE ITEM ID ON TAPE WITH ITEM ID
114;V8;=RM8;E8M8;SHL-32;(tp item);                   IN CURRENT GENERATION
V9;SHL16;SHL-32;(YB item);
SIGN;NEG;(+1b>t  0=  -1b<t);
DUP;J112=Z;J113<Z;
118;(b>t  get nw t);V8;SETAV10;JSL36;J115;=V8;J114;
113;(b<t get new b);J131C1Z;MOM1Q;=V9;J114;
112;(b=t);ERASE;
```

```
                                                                                    ID NOS EQUAL
E8M8;SETB77;AND;J181=Z;SET-1;=M15;M1M15;SET1;SHL32;OR;=M1M15;                        INSERT MEDLARS INDICATOR
181;  V41;YGO;-;J113=Z;                                                             IF REF IS A MEDLARS REF

                                                                                    IF LAST PASS RETURN TO
ZERO;=YLO;                                                                          COMPARISON
SET1;=RM9;
V40;J171≠Z;(j  if alternating searching);
E6M8;SHL-32;DUP;=YLO;(no of pars);
V45;-;J118>Z;(omit if too many links);
YLO;J171=Z;(omit if no links);                                                      PUT ID NOS OF CITED REFS
E6M8;SHL32;SHL-32;M8;+;SET6;+;=RM10;E6M8;SHL-32;=C10;                               INTO YL LIST
(Q10 nopar/1/staddpar);                                                             OMIT THIS IF SEARCHING IN
119;MOM10;                                                                          ALTERNATING MODE
ZERO;SHLD16;=YLOM9Q;DC10;J117C10Z;
ZERO;SHLD16;=YLOM9Q;DC10;J117C10Z;
ZERO;SHLD16;=YLOM9Q;DC10;J117C10Z;
ERASE;SET1;=+M10;J119;

117;ERASE;
171;(get child);
E5M8;SHL-32;YLO;+;DUP;=YLO;J118=Z;YLO;
V45;-;J118>Z;(omit if too many links);
E5M8;SHL16;SHL-32;M8;+;SET5;+;=RM10;E5M8;SHL-32;=C10;                               ADD ID NOS OF CITING REFS
(Q10 noch/1/staddch);                                                               INTO YL LIST
191;MOM10;J192C10Z;
ZERO;SHLD16;=YLOM9Q;DC10;J192C10Z;
ZERO;SHLD16;=YLOM9Q;DC10;J192C10Z;
ZERO;SHLD16;=YLOM9Q;DC10;J192C10Z;
ERASE;SET1;=+M10;J191;
```

```
192;ERASE;(prune starts here);
YLO;=RC9;
SETAYL1;=M9;Q9;ZERO;NOT;SHL-32;REV;JSL82;
SET1;=M9;(Q9  set for y1);
ZERO;=RM11;(Q11 set for YM list);

M1;I1;-;=M10;
M10;=V152;
MOM10;(end of chain);
ZERO;SHLD15;SHL33;(old wt);
YLO;SET47;FLOAT;+F;SHL-33;SHL33;=V50;(new wt);
=V151;V50;V150;-F;J113<Z;V151;

SHL1;ZERO;SHLD16;=YMOM11Q;(item no);
SHL-32;=M10;
198;MOM10;SHL16;ZERO;SHLD16;REV;SHL-32;DUP;
J193=Z;=M10;=YMOM11Q;J198;
193;ERASE;ERASE;NC11;ZERO;=V51;
194;C11;=RC12;
YLOM9Q;V51;J195=;DUP;=V51;
196;YMOM12Q;J195=;J196C12NZ;
SHL16;V152;OR;V50;OR;197;=MOM2Q;J194C9NZ;J113;
195;ERASE;J194C9NZ;J113;

131;V8;SETAV10;JSL36;J115;=V8;J131;
115;SET2;-;J107≠Z;
151;V8;SETAV10;JS11L36;ERASE;DUMMY;DUMMY;J40;
(end of rev read loop);
```

SORT RELATIVES

CALCULATE WEIGHT TO BE
TRANSMITTED TO RELATIVES.
OMIT IF THIS LESS THAN THRESHHD

ASSEMBLE IN YM THE CHAIN OF REFS
LEADING FROM THE REQUEST SET TO
THE CURRENT REFERENCE (INCL BOTH)

ELIMINATE ANY RELATIVE ON THIS CHAIN

ADD OTHERS INTO NEXT GENERATION IN YB

RETURN TO COMPARISON SEQUENCE

END OF BACKWARDS PASS UP MAG TAPE

```
111;SETB41;J106;
109;SETB42;J106;
108;SETB43;J106;
107;SETB44;J106;
106;J105NEN;JSP2802;SET2;JSP2802;ZERO;OUT;
105;SET47;V44;JSP2803;SET2;JSP2802;J106;
(end failures);

30;(print section);
SET2;JSP2802;JS1P2802;SET2;JSP2802;
YGO;=RC1;SET1;=M1;SET1;=+C1;
ZERO;
32;YGOM1Q;SHL-32;+;J32C1NZ;=V43;
(total no of refs rtrvd);

YBO;=RC1;SET1;=M1;
33;YBOM1;SHL-16;=YBOM1Q;J33C1NZ;(shift req nos);
V43;=RC1;SET101;=M1;
34;YBOM1;=Q2;MOM2;=M2;Q2;=YBOM1Q;J34C1NZ;
(put req in place of add);
V43;=RC1;SET101;=M1;
35;YBOM1;=Q2;I2;M2;=I2;=M2;Q2;=YBOM1Q;J35C1NZ;
(wt m req item);
V43;=RC1;SETAYB101;=M1;ZERO;NOT;SHL-16;Q1;JSL82;
(sort on -reqitem);
```

FAILURE EXITS

START OF OUTPUT SECTION.
RESULTS ARE OUTPUT TO PUNCH ONLY

CALCULATE TOTAL NO OF REFS RTRVD

REPLACE THE LINK ADDRESSES IN CHAINS
BY THE REQ NO FROM WHICH THE CHAIN
ORIGINATES

SORT RTRVD REFS ON KEY:- REQ NO REF ID NO

```
V43;=RC1;SET101;=M1;
36;M1;=RM6;(C6 counts no of refs in req);
M1;=M2;YBOM1Q;=Q3;(item); DC6;
37;J38C1Z;M1;=M4;YBOM1Q;=Q5;( Nitem );
I3;I5;-;J38≠Z;DC6;
M3;M5;-;J39≠Z;
C3;SHL-1;SHL33;C5;SHL-1;SHL33;
DUPD;MAXF;ERASE;V500;xF;-F;+F;SHL-33;SHL1;
C3;C5;OR;SHL47;SHL-47;OR;=C3;
(wts maxed or summed according to algo);
ZERO;=YBOM4;J37;
39;Q3;=YBOM2;Q4TOQ2;Q5TOQ3;J37;
```

WHERE A REFERENCE HAS BEEN RETRIEVED MORE THAN ONCE FOR THE SAME REQUEST, DELETE ALL BUT ONE OF THE DUPLICATES AND LET THE REMAINING ONE HAVE A WEIGHT EQUAL TO THE SUM(SUM ALG) OR THE MAXIMUM(SINGLE PATH ALG) OF THE INDIVIDUAL WEIGHTS

```
38;Q3;=YBOM2;
NC6;SETAYBO;=+M6;Q6TOQ15;C6;=+M6;SET-1;=+M6;SET-1;=I6;
ZERO;NOT;SHL-33;SHL33;Q15;JSL82;
```

SORT THE RTRVD REFS FOR A SINGLE REQUEST BY WEIGHT

```
V42;=C7;(max);
41;MOM6Q;DUP;SHL15;SHL-47;J42=Z;(j  if non medlars);
ZERO;SHLD15;SHL33;REV;SHL1;ZERO;SHLD16;
REV;SHL-32;PERM;
SET47;V44;JSP2803;FIX;V44;JSP2803;
SET47;V44;JSP2803;SET2;JSP2802;DC7;ZERO;
42;ERASE;J43C7Z;J41C6NZ;
43;J44C1Z;
SET1;=+C1;SET-1;=+M1;J36;
```

FOR THE MAX HIGHEST SCORING, MEDLARS INDEXED REFERENCES, PUNCH:-

REQ NO WEIGHT REF ID NO

REPEAT PUNCHING ROUTINE FOR NEXT REQUEST

```
44;JS2P2802;ZERO;OUT;
```

END OF PROGRAM