XVII. EVALUATION OF RETRIEVAL RESULTS IN THE EXTENDED SMART SYSTEM

Michael E. Lesk

1. Introduction

Efficient comparison of retrieval procedures requires a method of summarizing the results of test computer runs. In the SMART system, this evaluation problem is handled by a series of programs which aim both to compute such evaluation measures for actual retrieval runs, and also to aid in the development of better measures for the evaluation of retrieval systems.

Two evaluation programs are incorporated into the new SMART system. The first is a single run evaluation system that can be executed at the end of a given production run, which shows immediately the approximate degree of success of this run. The second set of programs maintains and uses a file of results of a whole series of runs on a given collection, With this second program, results from different retrieval procedures are readily compared, and changes in the evaluation algorithms can be extended to all runs simultaneously.

The basic information required for the evaluation of a run is the correlation list and the relevant document list for each request. The correlation list gives the correlation coefficient of each document with the request. This list is generated by link 8 of the SMART system and written on tape. The correlations are numbers between zero and one which

reflect the degree of similarity between each request and each document, in terms of the concept vectors formed during the runs. The relevant document list is a manually constructed list of documents which are assumed to be relevant in some sense, to the request. Degrees of relevance are not considered in the present SMART system, but provision is made for their use if this should become useful in the future. The relevant documents may not actually contain a specific answer to a given request, but may include documents which would be considered useful by a real user asking a real question. Ideally, the user himself should in fact make the relevance judgements; unfortunately, it is rare to find a group of users willing to spend the time required to make relevance judgments.

The object of a retrieval operation is to produce for the user all the documents graded "relevant," and none of those graded "irrelevant." Evaluation measures are coefficient reflecting the success the system has had in achieving this aim with a given document collection and processing method.

All current retrieval algorithms incorporated into SMART begin by considering the correlation list for a given request in descending size of the correlation coefficient. This produces a ranked list of documents, beginning with documents that had a high correlation with the request, and ending with documents that had a zero correlation with the request. This list is compared against a hypothetically perfect list, which has the manually judged "relevant" documents at the top and the irrelevant documents at the bottom. The numerical magnitude of the correlation is not

currently considered in this procedure, although it is hoped to include it in more sophisticated algorithms later. The advantage of using the complete correlation list is that effects caused by the introduction of a cutoff are eliminated.

Two basic properties are measured by most evaluation algorithms. These are called "recall" and "precision." Recall measures the completeness of the retrieval operation, that is the extent to which all relevant documents have been placed near the top of the ranked document list. Poor recall implies that the system is performing errors of omission; relevant material is being missed. Precision, on the other hand, is the accuracy of the retrieval operation; that is, the extent to which irrelevant documents have been kept away from the top of the ranked document list. Poor precision implies that the system is performing errors of commission; irrelevant material is being found. The distinction may be made clear by considering some hypothetical examples of a retrieval process. Suppose that five relevant documents are being sought in a collection of 100 documents. If, in the ranked document list prepared by the computer, the five relevant documents were to appear in positions 1,2,3,4, and 5, this would be perfect recall and perfect precision. If they appeared in the rank positions 1,2,3,4 and 100, this would be excellent precision but less satisfactory recall, since one relevant document is completely missed. If the rank orders were 2,3,4,5, and 6, the recall would be excellent but the precision somewhat deficient. 96,97,98,99 and 100 represents total failure, of course.

Both recall and precision measures should be computed, since they are each useful for different types of users. A user asking questions of the type "what precedents apply to the problem of railroad tax rebates?" is obviously more interested in finding all the precedents (good recall) than in avoiding questionably useful material. Questions such as "what is the boiling point of sulphur?" are clearly asked by people who want high precision, since one document will satisfy the questioner, and he will not care about any other documents that might also contain this information.

An elementary criterion for the recall measure might be the rank of the lowest relevant document in the correlation list. Similarly, precision could be judged by the rank of the highest irrelevant document in the correlation list. These criteria would have many disadvantages, however. They are highly sensitive to the behavior of individual documents; furthermore, they are not comparable for requests with different numbers of relevant documents. A good retrieval measure should have the following properties:

(a) accurate measurement of the property in question;

(b) insensitivity to fluctuations of individual documents, as far as possible;

(c) insensitivity to individual relevance judgments, as far as possible (many relevance judgments raise borderline problems which are difficult to decide);

(d) insensitivity to different numbers of relevant documents;

(e) insensitivity to different sizes of document collections;

(f) ease of understanding;

(g) ease of computation.

Since the development of measures with these properties is reviewed by J. Rocchio (ISR-8-IV, ISR-9-XXI), the final measurements only are given here. They are:

1. Rank recall = $$\frac{n(n+1)}{2\sum_{i=1}^{n} r_i}$$

2. Log precision = $$\frac{\sum_{i=1}^{n} \log i}{\sum_{i=1}^{n} \log r_i}$$

3. Normalized recall = $$1 - \frac{\sum_{i=1}^{n} r_i - \frac{n(n+1)}{2}}{n(N-n)}$$

4. Normalized precision = $$1 - \frac{\sum_{i=1}^{n} \ln r_i - \sum_{i=1}^{n} \ln i}{\ln \binom{N}{n}}$$

where r_i = rank of ith relevant document,

n = number of relevant documents,

and N = number of documents in the collection.

The next part of this section describes the computer programs used to compute these measures for a single run. The multi-run program is described in Part 3.

2. Single-run Evaluation

The correlations required are written on tape by subroutine OUTCOR of link 8. They appear as five word items, arranged as follows:

Word 1: request name, first six characters,

2: request name, second six characters,

3: correlation, in floating point, complemented magnitude,

4: document name, first six characters,

5: document name, second six characters.

These are blocked into physical records of 250 words. The last record is filled with words of 7474747474_8. This tape is written on the unit specified in CORTAP. It may be augmented by the results of document-document correlation. When control reaches chain link 12, the following operations are performed:

(1) A 250 word record is written on CORTAP which contains the top 100 words of COMMON storage, arranged so that they will be placed at the front of the correlation tape by the tape sort routine.

(2) Any relevance judgments appearing on A2 are copied onto the correlation tape in such a way that they will sort in front of the request they refer to. These judgments are written as five word items: the first two words constitute the request name, the second word is the number 2.0 complemented (this can be distinguished from a correlation by the fact that correlations are always between 0 and 1), and the next two words are the relevant document.

(3) The tape is now sorted into the following order: first, the 25-word record containing the top of common storage; then the requests in order; for each request, the relevant documents appear first, and then the correlation list in order of decreasing correlation.

This operation is performed by a straightforward sort on the first 108 bits of each five-word item. Flowchart 1 shows the operation of link 12.

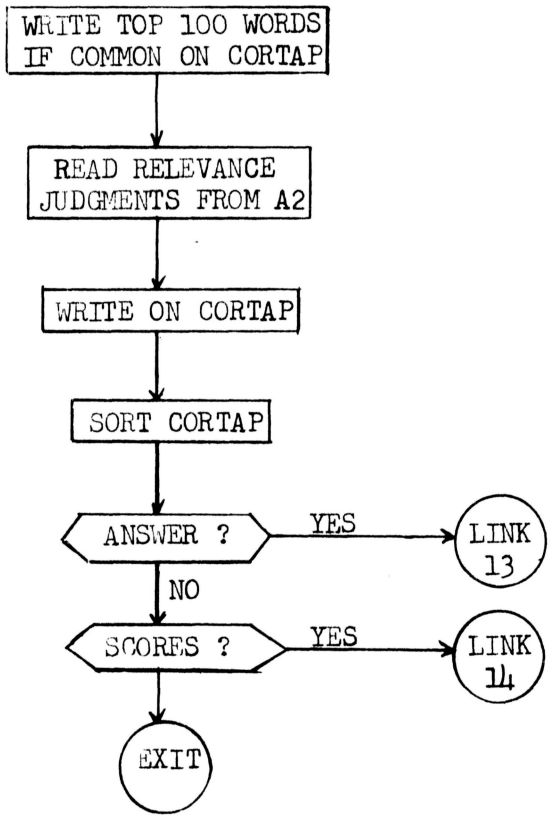
If the specification ANSWER has been requested, link 13 is now called. This link, which writes the answers to each request on the print tape, is described in ISR-9-XVI (by G. Hochgesang).

Finally, if SCORES has been requested, control goes to link 14, the actual evaluation link. This link reads in the correlations and writes out the following data for each request:

1. The fifteen documents which had the highest correlation coefficient with this request. With each document are printed its rank, its correlation, and a special mark if it is relevant.

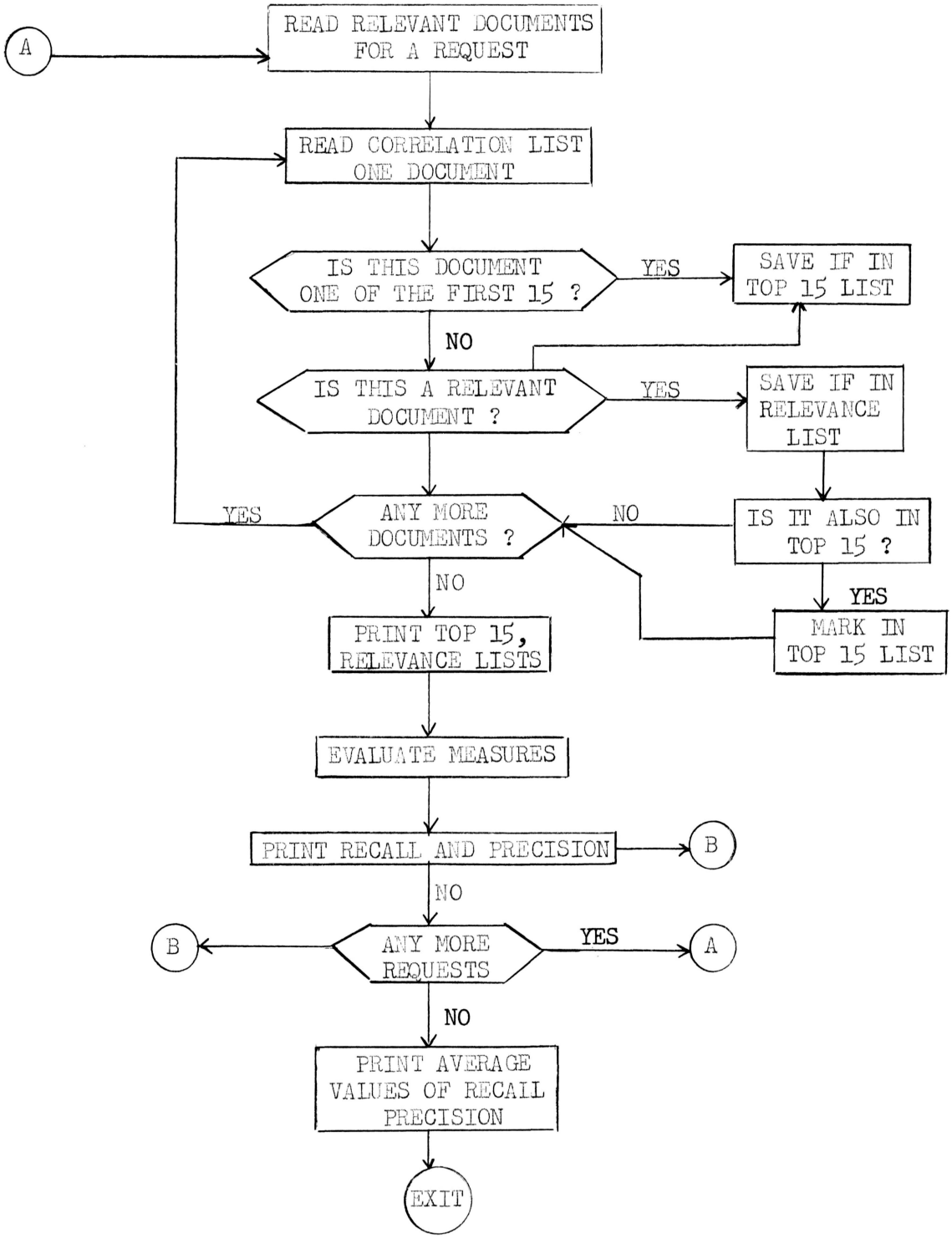
2. Each relevant document, its rank, and its correlation.

3. The four evaluation measures given above, and the two "overall" measures obtained by adding the measures (1) and (2), and by adding the measures (3) and (4). Link 14 is shown in Flowchart 2.



Link 12 Operations

Flowchart 1



Link 14 Operations

Flowchart 2

3. Multiple-run Evaluation

The system for evaluating many runs at once divides into two parts: a tape editor that maintains the collected file of evaluation data, and the evaluation routines to compute the measures. The tape editor combines an old collected file with new correlation tapes and control cards to produce a new set of tapes containing the complete correlation data for a given document collection. The collected correlation file is then processed by the evaluation routines. The evaluation system does not contain the fifty request limitation present in the retrieval system, and therefore may be used to obtain simultaneous evaluation of larger request sets.

The tape maintenance program reads a set of control cards from tape A2 that tell it what retrieval runs are available on what tapes, and what changes are to be made in relevance judgments in the file. It then writes an updated file with corrections, and may be told to evaluate this new file. The control cards have a six-character code in columns 1-6, followed by additional information in columns 7-72. The control cards and their functions are as follows:

Code word	Cols.	Information	Function
OLDTAP	7-11	NTAPES	This card is used when a collected file already exists, which is to be updated or evaluated. NTAPES is the number of tapes in the file, punched as a decimal integer right adjusted in columns 7-11. This card is followed by NTAPES cards, each having eighteen characters of identification (to be printed for the operator) in columns 1-18. This identification is followed by the tape unit on which the tape is to be mounted (either a right-adjusted logical tape number or a physical unit address) punched in columns 18 and 19.

Code word	Cols.	Information	Function
NEWRUN	8-19	RUNNAM	Name of new run being added to master file, twelve BCD characters.
	20-37	TAPE	Eighteen characters identifying tape containing correlations, to be printed for the operator.
	38-39	UNIT	Tape unit (either logical or physical).
	41-42	PLACE	Place where to insert this run in the collected file. PLACE should be the name of another run already in the file or being inserted. This run is placed immediately after it. If PLACE is omitted, the new run is placed at the end of the file. If a run named RUNNAM is already in the collected file, with the same requests, the old data are removed and replaced by data from the new tape.
APPEND			Other data are same as for NEWRUN. The only difference between NEWRUN and APPEND is that if a run with the same name already exists in the collected file, for the same requests, APPEND causes the data from the old and new runs to be merged together.
DELETE	8-19	RUNNAM	The run RUNNAM is deleted from the collected file.
NUMDOC	7-13	NUMBER	The number of documents in the collection is set to NUMBER. If this is different from what it used to be, or if it is not what the program finds in counting the tape items, a message is printed, but the command remains valid.
COLLEC	8-19	COLLNAM	The collection name is COLLNAM (any twelve BCD characters). This is used for identifying printouts only.
CHANGE	8-19	REQUEST	Twelve character request name.

Code word	Cols.	Information	Function
	21-32	DOCUMENT	Twelve character document name. If DOCUMENT appears in the list of relevant documents to request REQUEST, it is removed; if it is not now in the list of relevant documents for REQUEST, it is inserted.
USEREL	8-19	RUNNAM	Twelve character run name.
	21-32	REQUEST	Twelve character request name. The relevance judgments for request REQUEST in run RUNNAM (RUNNAM should be a run name mentioned in an APPEND or a NEWRUN) the previous relevance judgments are replaced.
UPDATE			Write a merged, revised collected run tape file. Do not perform any evaluation. The new file will be written on tape drives B1 and B2 (alternately), and the operator will be instructed to save the tapes. Since the size of the collected file increases as the product of the number of requests, number of runs, and number of documents, the collected file should be written on full length maximum density tapes.
SCORES			Update collected tape, if any other control cards previously appeared; then evaluate every run in the file. If SCORES is the only control card in the data deck, no new collected file is written. In order to avoid the remounting of a new reel, the program will write the collected file on tapes B3, B5, and B6 as well as B1 and B2 if evaluation is requested, provided that these additional tapes are available. Note that the last card in a correct data deck is either UPDATE or SCORES.

Although the programmer may specify the tape units, the following assignments are assumed if not otherwise specified: Old collected file: alternately A5, then A6.

New tapes: first B5, then B6, then A4, then A7, then A8. Of course, all tapes specified by the programmer must be available on the machine; they must not conflict with the system tapes A1, A2, A3 or B4; and no two tapes may be the same (except that nonconsecutive tapes of the old collected file may go on the same unit).

The format of the collected file is as follows: First record: twelve words

1. SMART_(_signifies blank)
2. _EVAL_
3. _LESK_
4. Serial no. (number of tape in the file)
5. Time (six BCD characters, e.g. 10_AM_, _3_PM_)
6. 7. 8. Date (18 BCD characters, e.g. DECEMBER_7,_1941_ _)
9. Collection name, first six characters
10. Collection name, second six characters
11. Number of documents in the collection
12. Number of runs

All further records are 300 words long. The data are packed without regard to record boundaries in the following format:

1. The list of runs, two 6-character words per run.
2. The data describing each individual run. This consists of a number telling how many different computer runs were made to get a full set of request-document correlations for this run

(since one retrieval run can process only fifty requests), the time and date of each computer run, and the parameter words giving the weights and processing options used.

3. The data for each request. This consists of the number of runs made with that request, the type of runs, and the way they were run. In addition, the number of relevant documents and the list of relevant documents are given. All this information is packed into three word items. The correlation list follows for each run, in three word items: two words for the document name, one word for the correlation. The following are used as special sentinels to indicate unusual three word items:

   (a) at the beginning of each request, the first item contains the number of runs and the number of relevant documents, plus one blank word;

   (b) after this word, each pair of three-word items contains two words of run name, one word for the time, and three words for the date (a pair of items for each run);

   (c) correlations above 1.0 indicate relevant document lists;

   (d) if the first six characters of the document name are (728), then the next six characters have the following meanings:
   0 = end of correlation list

   ENDREQ = end of request

   ENDTAP = end of tape

   ////// = end of collected correlation file.

After the complete collected correlation tape file is written, the evaluation section of the program is called to compute the evaluation measures for each request and each run. This program behaves in exactly the same way as link 14 of the main system, except that it evaluates a great many more correlation lists. It also computes average measures for each run as well as for each request. The numerical procedures used are the same as those described in Part 1 of this section.