Evaluation of Search Time for Two Computerized Information Retrieval Systems at the University of Georgia

GLENN O. WARE* and MARGARET K. PARK
Computer Center, University of Georgia, Athens, Ga. 30601

Received June 12, 1972

Two statistical models for estimating search time have been developed for the CA Condensates data base using the University of Georgia Text Search System. Graphs showing the effect of data base size and number of search terms on search time are presented. Comparative timings between the Chemical Abstracts Service search program and the University of Georgia search program are made for the CA Condensates data base.

Over the past five to ten years a number of organizations have begun to offer computer-based information retrieval services for their faculty or research staff. Initially, services were limited to one or, at the most, two relatively small data bases within any individual organization—usually using computer search programs supplied by the producer of the magnetic tape service with his data base. However, there are now some 30 organizations who are full members of the Association of Scientific Information Dissemination Centers (ASIDIC) and another nine members of EUSIDIC, its European counterpart, all of whom are offering bibliographic current awareness searches on at least two data bases. Most of these centers have either developed their own generalized computer software systems to support the search services or have adapted software packages available from some other source, such as computer hardware vendors, commercial software houses, government agencies, etc.

Grunstra and Johnson have reported on the implementation and evaluation of two computerized information retrieval systems at the University of Pittsburgh.[1] Other than this article, very little has been reported in the literature concerning the performance of these software systems, either individually or as comparisons. Nor have the techniques used been evaluated in a quantative manner such that the results can be used for comparison with alternative methods. However, several approaches to the evaluation of information systems have been reported. Bourne and Ford modeled a storage and retrieval system in terms of time and cost data for equipment, personnel, materials, and procedures.[2] The model was representative of a machine-based information system and provided an economic evaluation of alternative system configurations. Blount, Duquet, and Luckie considered factors such as processing time, linkages between different processing components in the system, service units and their reliability, and processing components in the system, service units and their reliability, and processing loads in the construction of a model for a machine-based information system.[3] Now, more and more work is being done, and a greater awareness has developed of the need for the evaluation of information systems and component subsystems.

The University of Georgia has had occasion to install and operate two computer software systems in support of its information dissemination activities. From mid 1968 until late 1970, the Center operated its search services using computer programs which were originally developed for the Chemical Abstracts Service (CAS) data bases. Other data bases were also run using these programs by converting these other files to the CAS tape format. A model which allowed the prediction of search times using this software was published in 1970, using CA Condensates as the data base.[4] As the volume of the Center's activities increased, several problems were encountered which reflected the fact that the CAS programs were being used for purposes far exceeding the capacities for which they had been designed. Consequently, a new search system was designed and programmed for use with the large volumes of search profiles and data bases being used in the Georgia Center.

Among the important design considerations was the generalization required to handle the 17 bibliographic data bases presently in use as well as new, as yet undefined, files of similar data in the future. A survey was made of some 20 different file formats and at least a dozen different text search systems as part of the preliminary design investigation. The result was a mixture of subjective opinions concerning the optimum file structures to be used, the desirability of various profile construction and search strategy features, and the efficiency of alternative programming techniques. No quantitative or comparative information was readily available for the file formats and systems investigated at this point. Consequently, the Computer Center staff proceeded to design the new system using a combination of best available information, intuition, and experienced programming skills. The result is another in the series of available computer software packages for bibliographic search and a detailed comparison study with respect to at least one other search system which has been in fairly wide use. It is the hope of the authors that other installations will apply similar quantitative measures to their own software systems in order to contribute to the general area of free-text information retrieval technology.

UGA TEXT SEARCH SYSTEM

The file structure which was chosen for implementation in the UGA Text Search System was the Standard File

[*] To whom inquiries should be addressed.

Format (SFF) developed by Chemical Abstracts Service.[5] The SFF structure consists of a data element directory which identifies each type of data contained in the logical record along with control information defining the location and length of the data itself. The directory is then followed by the variable length data strings, recorded one field immediately following another. The SFF structure is strongly oriented toward use on an IBM 360 or 370 or compatible computer hardware configuration because of its 8-bit byte representation of certain bit string portions of the record. The important feature of this generalized format to the problem of handling a large number of different data bases from several suppliers is the flexibility it allows in both programming and data base content. All application programs can be easily generalized to operate on this file structure, regardless of data base content. The search programs were written for the SFF file structure and are essentially independent of any specific data contained in the file.

Two types of data elements are recognized by the search system: left-anchored and free text. Left-anchored elements are those types of data for which the format and content are precisely defined. They are usually controlled by an authority list or thesaurus. The content and format of the data are known from their initial or left-most character so that left truncation is not required or allowed. Free-text elements, on the other hand, are free-vocabulary fields in which the content and format are uncontrolled. For retrieval purposes, it usually is necessary to have the capability of using both left and/or right truncation of the search terms.

For left-anchored terms, the first character of the term is used as an index into an associated vector for each data element type. This location points to a threaded list which contains all of the terms for that data element which start with this first character. In other words, the vector contains one word for each alphabetic letter, each number, and the character "blank." Each word contains the address of an alphabetically threaded list; the first word points to a list of all terms beginning with A, the second to all terms beginning with B, etc. Each word contains the length of the shortest string in the threaded list. The term is entered into the list in alphabetical sequence. If, when entering a term, a preceding term is encountered that is equal in length and content, then the string part of the term being entered is deleted and an equal string switch is set "true."

The same general technique is used for free-text terms except that the first two characters of the term are used as an index into a two-dimensional matrix, rather than simply using the first character. The matrix provides much higher discrimination between text terms which must be searched essentially character-by-character against the free-text data elements, and thus decreases the number of comparisons which must be made between the terms and the data base.

The equal string switch and corresponding routine are used to minimize the number of comparisons which must be made against the search terms themselves and the amount of core storage required. An analysis of several collections of profiles indicates high degrees of redundancy between terms from different profiles. The percentage of duplication varies according to the data base and the number of profiles considered (i.e., batch size), but it can run as high as 60% duplication in large batches. Thus, identifying the duplication can significantly decrease the number of character string matches performed in the search, consequently decreasing execution time considerably, and resulting in as high as 30% reduction in core storage requirements for very large batches of profiles.

The only major logic extension made in the UGA system over the CAS search program was the ability to use indefinitely nested Boolean logic expressions. The UGA system incorporated the technique of using a two-dimensional matrix approach of searching instead of the vector approach used by the CAS program. Also, the capability of searching for unique terms instead of total terms was not available with the CAS program.

METHODS

In this investigation, CA Condensates was used as the data base for development of a model for estimating search time using the UGA Text Search System. Throughout this study, search time refers to the time required to match profile terms against the data base file to determine which bibliographic citations are answers to the search logic. This includes inversion of the profiles in core, matching of the profile terms against appropriate data base elements, evaluation of the Boolean logic and weighting requirements, and outputing the selected bibliographic citations. The time required for the selection of the profiles as search input and for the printing of the search results is not included in search time.

Eight subsets from the even-numbered issues of volume 73 of CA Condensates and eleven sets of search profiles randomly selected from the current awareness profiles normally run in routine production were selected for investigation. The subsets of the data base ranged from 1,559 to 38,965 documents while the number of profiles within a given set ranged from 6 to 168 profiles. A total of 60 search runs were made for selected combinations of data base size and number of profiles. The central processing unit (CPU) time was recorded for each search run. All searches were run on an IBM 360/65, OS MVT using Release 20.1 with HASP 3.0, in high speed core, using 90 KB tape drives (800 bpi) for input and 2314 DASD for output files.

RESULTS AND DISCUSSION

The characteristics of the sets of profiles are shown in Table I. As can be seen, the total number of terms increased as the size of a set of profiles increased. The number of terms per profile ranged from 27.07 to 47.73 with an average of 37.43. Also, the number of equal or redundant terms were recorded for the profile sets. As the number of terms per profile set increases so does the per cent of equal or redundant terms. The per cent of redundancy in the terms ranged from 2.17 for 11 profiles to 17.50 for 168 profiles.

Table I. Profile Characteristics

Run No.	No. Profiles	No. Terms	No. Equal Terms	Terms/ Profile	Equal Terms/ Profile
1	6	236	12	39.33	2.00
2	11	525	14	47.73	1.27
3	15	406	18	27.07	1.20
4	21	774	33	36.86	1.57
5	26	1,022	63	39.31	2.42
6	30	956	63	31.87	2.10
7	39	1,439	149	36.90	3.82
8	51	1,977	168	38.76	3.29
9	72	3,245	419	45.07	5.82
10	101	3,480	489	34.46	4.84
11	168	5,789	1013	34.46	6.03

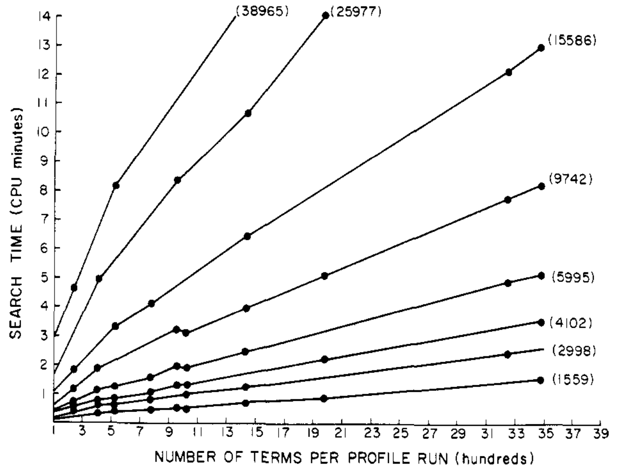

Figure 1. Search time vs. number of profile terms

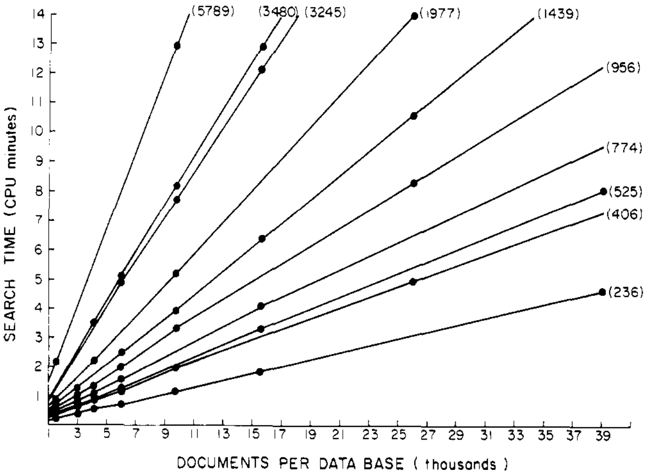

Figure 2. Search time vs. number of documents in the data base

The data base contained 77,930 total documents. Document record size ranged from 204 characters per document to 1004 with an average of 615.

A plot of search time (CPU) against the number of terms per profile run for each data base size is shown in Figure 1. For a constant number of documents per data base, a linear function exists with the number of terms explaining approximately 99% of the variability in search time. Figure 2 shows a plot of search time against documents per data base for the various number of terms. Also, for a constant number of terms, a linear function exists with documents per data base explaining approximately 99% of the variability in search time.

As mentioned above, search time is a linear function of data base documents when the number of terms is held constant and is a linear function of the number of terms when data base documents are held constant. A model for estimating search time for any combination of terms (T) and data base documents (D) is given by the following equation:

$$\text{Search Time (CPU min.)} = [753.4 - 0.134T + 0.8922D + 0.002177TD] \times 10^{-4} \quad (1)$$

This model explained 99.86% of the variability in search time with a standard error of the estimate of 0.2094 minute.

Since the search program minimizes the number of comparisons which must be made against the profile terms by searching against unique terms (U) only (i.e., total terms minus equal terms), a second model for estimating search time is given by the following equation:

$$\text{Search Time (CPU min.)} = [278.0 + 0.432U + 0.7587D + 0.002521UD] \times 10^{-4} \quad (2)$$

This model explained 99.93% of the variability in search time with a standard error of the estimate of 0.1475 minute.

Both models do an excellent job in predicting search time. Prior knowledge of the way the search program operated indicated that using unique terms instead of total terms in conjunction with the number of data base documents should be a much better model. Comparing Equations 1 and 2, one can see a sign difference associated with the coefficients of total terms and unique terms. In Equation 1, the coefficient associated with total terms is negative while the coefficient associated with unique terms in Equation 2 is positive. In both equations, the interaction between the number of terms and data base documents has the greatest influence in predicting search time. Therefore, the sign change based on whether total or unique terms are used is simply an adjustment to the total search time. Since total terms are used in the interaction with data base documents in Equation 1, the effect owing to total terms alone is a downward adjustment in search time. In equation 2, unique terms have an additive effect in predicting search time since only unique terms are included in the interaction effect with data base documents.

Park *et al.* developed a similar model for predicting CPU time for the CAS search program.[4] Search time (CPU) is given by:

$$\text{Search Time (CPU min.)} = [2031.552 + 3.46839T + 1.20747D + 0.01724D] \times 10^{-4} \quad (3)$$

where T = total terms, and
D = documents in data base

Comparison of search runs were made between the CAS search program and the UGA text search system based on Equations 3 and 1, respectively. CPU time for selected combinations of total number of terms and number of documents in data base are shown below:

Terms	Documents	CAS Search CPU Minutes	UGA Search CPU Minutes
100	1000	0.5310	0.1850
100	5000	1.7036	0.6290
100	25000	7.5665	2.8488
500	1000	1.3593	0.2667
500	5000	5.2903	1.0590
500	25000	24.9452	5.0204
2000	5000	18.7406	2.6716
2000	25000	90.1155	13.1640
2000	50000	179.3341	26.2795
5000	5000	45.6411	5.8969
5000	25000	220.4560	29.4513
5000	50000	438.9747	58.8943

For a small number of terms and data base documents, the UGA text search program indicates a factor of 3 to 4 times the improvement in over-all performance. For a larger number of terms and data base documents, an overall improvement factor of 7 to 8 times is achieved with the UGA text search program. Comparative timings for other issues of CA Condensates which were run on both systems have shown a similar pattern.

Basically the improvements came in two areas; the first was in the searching of unique terms. As mentioned earlier, the percentage of duplication varies according to the data base and the number of profiles being searched. Thus, using unique terms only can significantly decrease the number of matches performed in searching. The second area of improvement was the use of the two-dimensional matrix approach in searching instead of the vector string approach used in the CAS search system.

CONCLUSIONS

The emphasis of this study has been two-fold: to develop statistical models for estimating computer search time for CA Condensates using the UGA text search system, and to make comparative timings between the CAS search program and the UGA search program using the CA Condensates data base.

The two statistical models accounted for over 99% of the variation in search time. The models are currently being used for estimating CPU run times in the University of Georgia Information Science Center. Similar models are being constructed for other data bases run in the information center.

The UGA search program showed a significant improvement in over-all performance over the CAS programs in searching CA Condensates. Other preliminary studies at the University of Georgia Information Science Center have shown even a greater improvement for other data bases searched. However, this is due to the fact that the CAS programs were not specifically designed for searching these data bases. Therefore, the comparative timings were not presented in this study.

ACKNOWLEDGMENT

This work was partially supported by the National Science Foundation under Grants GN-851 and GN-32214.

LITERATURE CITED

(1) Grunstra, Neale S., and Johnson, K. Jeffrey, "Implementation and Evaluation of Two Computerized Information Retrieval Systems at the University of Pittsburgh," *J. Chem. Doc.* **10**, 272-7 (1970).

(2) Bourne, C. P., and Ford, D. F., "Cost Analysis and Simulation Procedures for the Evaluation of Large Information Systems," *Amer. Doc.* **15**, 142-9 (1964).

(3) Blount, C. R., Duquet, R. T., and Luckie, P. T., "A General Model for Simulating Information Storage and Retrieval Systems," HRB-Singer, Inc., Science Park, State College, Pa., April 1966.

(4) Park, M. K., Carmon, J. L., and Stearns, R. E., "The Development of a General Model for Estimating Computer Search Time for CA Condensates," *J. Chem. Doc.* **10**, 282-4 (1970).

(5) "Standard Distribution Format Technical Specifications," American Chemical Society, Washington, D.C., Std. Book 8412-0106-4.

Development and Use of a Termatrex Chemical File System*

THEODORE LEGATT,** ELIZABETH A. BELLAMY, SAMUEL X. deLORENZO
Department of Technical Documentation, Corporate Laboratories,
Research Division, Schering Corp., 60 Orange St., Bloomfield, N.J. 07003

Received June 26, 1972

A storage and retrieval system for chemical structures, using optical coincidence, is in use at Schering Corp. The system rapidly provides the research laboratories with structural information at any generic level. The design, maintenance, costs, and applications are discussed.

One of the responsibilities of a Technical Documentation center in a pharmaceutical organization is to provide information on the chemical structure of compounds. Basically, each compound being evaluated must be recorded and systematized so that at any time, information on its whole or partial structure can be supplied on request.

The internal file of chemical structures at Schering Corp. was reorganized about seven years ago. At that time, a Beilstein-type classification code was used; the structures were incorporated on McBee Keysort cards. Several factors necessitated a reorganization of the chemical coding system. The number of compounds had grown to about 12,000, making the needle-sorting of McBee cards a slow and cumbersome task. As the file size grew, compounds were lost and false drops increased owing to the inherent inadequacies of a Beilstein classification code. In addition, a system was needed to provide the department of analytical chemistry with correlations between structural features and data from nuclear magnetic resonance or infrared analysis. To meet these requirements, a new chemical code and new search equipment were needed. The code had to be relatively easy to learn and capable of searching for generic and specific structures. The search equipment had to be compact, provide rapid output, browsability, and personal control.

DESCRIPTION OF SYSTEM

A modified Ringdoc code and a Termatrex optical-coincidence system[3] were chosen, since they seemed to fulfill our requirements. Optical coincidence was desirable

*Presented before the Seventh Middle-Atlantic Regional Meeting, ACS, Philadelphia, Pa., February 1972.

**To whom questions should be addressed.