pounds are distinguished by means of number(s)—e.g., MEPYR2 and MEPYR3 for 2-methylpyridine and 3-methylpyridine, respectively.

Of course, there are complex organic names yielding a more-than-six-character code. In these cases, a reasonable omission of several characters reduces the original abbreviated form—e.g., DEGDPE instead of DEYGDPRE = diethylene glycol dipropyl ether. Fortunately, such compounds do not occur among those often found in vapor-liquid equilibrium data. As far as inorganic compounds are concerned, the codes usually consist of formula.

Based on these mnemonic codes, reference input cards were prepared. A set of programs was prepared to check the punched cards to eliminate possible formal errors. Then, the cards were recorded on magnetic tape. The computer TESLA 200 (compatible with General Electric System) was applied for the automatic processing of the survey from 1900 through January 1972, listing 4500 literature citations on vapor-liquid equilibrium data for 1450 compounds. Examples of the cross- and author-indexes are shown in Figures 1 and 2, respectively.

In addition to the two indexes, the files offer alternative classifications, including data retrieval. The whole system can be supplemented easily and kept up-to-date.

LITERATURE CITED

(1) Chu, J. C., Wang, S. L., Levy, S. L., Paul, R., "Vapor-Liquid Equilibrium Data," Ann Arbor, Mich., 1966.
(2) Kogan, V. B., Fridman, V. M., Kafarov, V. V., "Ravnovesie mezhdu zhidkostiyu i parom" (Vapor-Liquid Equilibrium), Nauka, Moscow, 1966.
(3) Stage, H., Faldix, P., "Gleichgewichtsverhalten von Ein- und Mehrstoffsystemen," *Fortschr. Verfahrenstech.*, Verlag Chemie, 1952–1967.
(4) "Bulletin of Thermodynamics and Thermochemistry," Ann Arbor, Mich.
(5) Hála, E., Pick, J., Fried, V., Vilím, O., "Vapour-Liquid Equilibrium," Pergamon Press, New York, N. Y., 1967.

A Chemical Search System for a Small Computer*

DANIEL U. WILDE** and ALBERT C. STARKE
New England Research Application Center,
University of Connecticut, Storrs, Conn.

Received November 30, 1973

The mechanization of chemical information retrieval systems until now has been limited to those organizations that have access to extensive computer facilities. Now, small, low-cost computers, such as IBM's 1130 or DEC's PDP-11, are available with input/output capacities that make them suitable for SDI and retrospective searching on any of the many commercially available data bases. Such a machine, located at the New England Research Application Center (NERAC), is described and the problems of using it for chemical information retrieval are discussed. NERAC's SDI Chemical Search System is described, and an example profile is used to illustrate its capabilities.

During the past six years, the New England Research Application Center (NERAC) has operated as a NASA Research Dissemination Center at the University of Connecticut. The purpose of the Center is to aid and promote technology transfer in the Eastern United States by helping industry locate appropriate technical information. During this period, NERAC has performed some 5000 retrospective searches, while its data base has grown to over 2,000,000 documents. Presently, this data base includes six files: National Aeronautics and Space Administration (NASA), Department of Defense (DDC), U. S. Government Reports Abstracts (GRA), Education Resources Information Center (ERIC), American Society for Metals (METADEX), and World Aluminum Abstracts (WAA). Recently, searching of *CA Condensates* tapes of *Chemical Abstracts* (CA) has been added as a current awareness or Selective Dissemination of Information (SDI) service, thereby augmenting SDI services performed on each update of the previously cited data bases.

When NERAC began, there were two choices for computer power. First, time could be purchased on a large, extensive computer system; or second, NERAC could rent its own small machine. In the first case, although the amount of machine time used would be small, the organization would still be dependent upon the services and priorities of someone else. In addition, NERAC's computer cost would be directly proportional to use. In the second case, a small machine might be slower, but NERAC would be able to schedule it so as to meet its own requirements. Here, computer costs would be fixed, and increased use would result in decreased unit costs.

NERAC decided to rent its own machine. Currently, NERAC's computer system consists of an IBM 1130 with an 8k, 16-bit word memory with a 2.2 microsecond cycle time. Search questions are read via a 1442 card/read punch, and results are printed on a 1403 line printer. Data files are read from and intermediate results are saved on two 2401 tape drives. Here, the often used input drive has a transmission rate of 60,000 characters/second, and the less used output drive transmits at 30,000. This configuration rents for approximately $3700 per month and includes a small disk for storage of a monitor, user programs, and temporary data sets.

Random access searching on inverted files is out of the

* This research was sponsored in part by the National Aeronautics and Space Administration Contract NASW-2307 and 2516

** To whom correspondence should be addressed.

question with such a small machine. This is particularly true for large data bases, such as the NASA file of 750,000 documents or the Metadex file of 150,000. Consequently, data files are organized serially requiring that each and every document be examined during each search.

Techniques available for chemical file searching are well known;[1-5] and NERAC's basic linear search system has also been described.[6,7] NERAC's computer system successfully allowed us to schedule retrieval runs to meet client demand and also to reduce our cost per search as demand increased. At the same time, using a small computer to process large data files posed many problems. This paper discusses these problems as they relate to a chemical SDI system and then describes our solution to them.

PROBLEMS OF CHEMICAL SEARCHING

To date, all of NERAC's data bases have been indexed from a fixed thesaurus; or if not, it has been possible to generate a subject authority list showing frequency of use for all index terms (keywords, Uniterms, descriptors, etc.). Since all permissible terms are known in advance, lengthy index terms can be compressed into compact, but still unique, bit strings via hash coding. Doing so greatly reduces search time but requires that retrieval strategies enumerate each and every term including all possible spellings and variations. For example, a search on certain compounds of phosphorus might include each of the following:

PHOSPHATE
PHOSPHATES
PHOSPHONIC
PHOSPHORIC
PHOSPHOROUS
PHOSPHORUS

Unfortunately, complete enumeration of all index terms did not appear practical for chemical bibliographic information in the *Chemical Abstracts Condensates* data file. First, there is no official thesaurus, and index terms are generated freely. Second, the CA file grows so rapidly (approximately 6000 documents/week) that producing a subject authority list or index term frequency count would be very time consuming. Third, SDI profiles or strategies would need to be constantly revised and rerun as new index terms were discovered. Consequently, even though NERAC was using a small computer, it was necessary to develop a retrieval system that would allow for complete text searching as in IBM's TEXT-PAC system for large IBM 360's and 370's.[8]

NERAC's CHEMICAL SEARCH SYSTEM

Input to NERAC's chemical search system is a Boolean strategy composed of terms representing the user's requirements connected by logic symbols that specify the logic relationship between terms. Here, a term is a character string which may be an index term, author name, company name, journal name, ASTM Coden, CA section number, etc. These terms may be connected in any Boolean combination using *or's* (+), *and's* (&), and *not's* (⌐) with any number of parentheses.

Since the input to the Chemical Abstracts Service data base is uncontrolled—i.e., titles are produced by authors, and descriptors are generated freely—it is necessary to include all variations of a desired term in a retrieval strategy. This task is greatly simplified if all forms of truncation are allowed—i.e., none, left, right, and both. Here, right truncation would be used to specify singular and plural in one term—e.g., PHOSPH* would retrieve the six terms listed above. The various types of truncations are illustrated in Figure 1.

Mode	Input	Action	Examples
None	HALO	Retrieves only the term HALO	HALO
Right	HALO*	Retrieves any term beginning with HALO	HALO, HALOgen HALOgenation
Left	*HALO	Retrieves any term ending with HALO	HALO, diHALO triHALO
Both	*HALO*	Retrieves any term that contains HALO	HALO, HALOgen, diHALO, polyHALOgenated

Figure 1. Truncation modes

Sometimes it is desirable to be able to insist that character strings appear immediately adjacent to each other. For example, if references to products or processes of catalytic aluminum-type polymerizations are desired, it might be appropriate to use the term, ORGANOALUMINUM POLYMERIZATION. However, this term would miss variations, such as

alkylALUMINUM	POLYmer
ethylALUMINUM	coPOLYmer
diethylALUMINUM	homoPOLYmer
dimethylALUMINUM	interPOLYmer
propylALUMINUM	heteroPOLYmer
.	.
.	.
organoALUMINUM	coPOLYmerization
.	
.	
butylALUMINUM	POLYmerization
.	.
.	.
methylALUMINUM	POLYmerizing

Consequently, any combination of truncations are permitted in a single term, such as *ALUMINUM *POLY*. Here, a word ending in ALUMINUM must be followed *directly* by a word containing POLY. Note that ORGANOALUMINUM HALIDE CATALYZED POLYMERIZATIONS would not be retrieved unless one searched for *ALUMINUM & *POLY*.

Straight text searching with truncation can produce unexpected and sometimes humorous results. For example, *SILVER* might be used to retrieve all documents mentioning the subject SILVER, as in SILVER HALIDE. Unfortunately, it would also select QUICKSILVER. In addition, *SILVER* would retrieve articles written by R. F. SILVERMAN, work done at the TOULE SILVER CO., and documents written in the city of SILVER SPRINGS. Consequently, in the NERAC search system, each strategy term is accompanied by a field code that limits its scan to a specific field or fields. Figure 2 specifies these fields and their codes.

Here, blank specifies both title and index terms, the most commonly searched fields.

PROFILE EXAMPLE

A hypothetical user's question concerns photography. He would like information relating to the uses of dyes and

Field	Code
Title and Index Terms	blank
Title	T
Index Terms	I
Author	A
Citation	C
ASTM Coden	N
CA Section	S

Figure 2. Fields and field codes

A & (B + C + D + E) & F & −G

A	PHOTO*
B	EMULSION*
C	SEC 074
D	*DYE*
E	DANIZ, T*
F	KODAK CO*
G	INFRARED

Figure 3. Example profile

emulsions in photographic work at the Eastman Kodak Co. References to works by T. Daniz are of particular interest, but allusions to infrared are undesirable. Finally, relevant references are likely to be found in only the macromolecular and physical/analytical section groups of *Chemical Abstracts*. Terms used to express this search are given in Figure 3. In the Boolean logic expression, terms are represented by letters. In this case, A represents the character string, PHOTO*, C specifies one of the 80 sections of *Chemical Abstracts* of particular interest, and E specifies the desired author. Here, the initial is followed by an asterisk so as to overcome spelling and punctuation inconsistencies. The same is true for F which represents the company name.

This strategy is prepared for computer processing in four segments. The first and second segments are title and company cards which enable the computer to produce a cover sheet identifying the results of this search. The third segment is a group card which specifies which groups of CA sections are to be searched for this question. Here, only groups 3 and 5 need be scanned where the CA groups are arbitrarily numbered:

Group No.	Group Title
1	Biochemistry (Sec. 1-20)
2	Organic Chemistry (Sec. 21-34)
3	Macromolecular Chemistry (Sec. 35-46)
4	Applied Chemistry and Chemical Engineering (Sec. 47-64)
5	Physical and Analytical Chemistry (Sec. 65-80)

The final segment must list the strategy terms and specify their logic relationship. This can be done in two ways depending upon the characteristics of the computer that is used for searching. For a large machine where core size and cycle time permit batching of questions, the strategy terms and their logic relationship can be given separately. Here, large tables can be used to reduce greatly the number of computations required for term comparisons by processing all terms before logic relationships are determined.[9,10]

For a small machine with limited core, it is impossible to apply techniques that utilize large tables. Nevertheless, it is necessary that term comparisons be minimized. This can be done by decoding the Boolean equation as each term is processed to determine which term is logically next. For example, assume the equation: A & B. Here, B is said to follow A because if A is true, the equation can still be satisfied and B must then be checked. On the other hand, if A is not true, the equation cannot be satisfied, and B need not be processed. In a typical document, A will be present only a small percentage of the time, and term comparisons are reduced by nearly half. Savings are even greater for equations of the form: A & (B + C + D + . . .) & . . .

If a computer is to decode a logic equation, it must be told how to find its *next* term. This can be done by attaching two numbers to each term. The first number is referred to as the TRUE entry while the second is known as the FALSE one. These values are coded onto the strategy keypunch form as follows. If a term is present in the proper field, its TRUE entry tells the computer the relative location of the next term. In contrast, if the term is absent from that field, its FALSE entry is used. Here, an entry of 99 has been arbitrarily selected to indicate that the Boolean equation has been satisfied and that the document is to be retrieved. Similarly, 98 signals that the equation cannot be satisfied and that the document has been processed.

Figure 4 illustrates how this procedure is used to code the sample strategy shown in Figure 3.

It consists of seven lines which are keypunched, one line per card. Here, the first card contains the term, PHOTO*, with a blank field code specifying a scan of both term and title fields. If PHOTO* is present, then its TRUE entry is used to locate the next term. In this case, the 01 tells the computer to count down one term to EMULSION*. If PHOTO* is absent, its FALSE entry is used. Here, the 98 indicates that the Boolean equation cannot be satisfied, and processing of the document is complete. The computer follows this process using TRUE and FALSE entries depending upon the presence or absence of terms until it reaches a 99 which signals a desired document or a 98 which indicates the opposite. This example illustrates the enormous flexibility of using a *Chemical Abstracts* section as a keyword, and of searching various fields via truncation.

USING THE SYSTEM

After a weekly update tape has been converted into NERAC's search format, all Chemical Abstracts Service SDI strategies are read into the machine and saved on its small disk. Console sense switches are set telling the machine which CA section groups are on the input tape. The use of sense switches in conjunction with a group card in each strategy permits searching of any combinations of section groups with only one SDI card check. (An odd issue update tape contains groups 1 and 2 while an even issue contains groups 3, 4, and 5.)

Once all strategies are on disk, the computer reads the first strategy back from disk and checks to see if any of its group numbers agree with any console switches. If there are no matches, this strategy is not to be searched for this

Card No.	True Entry	False Entry	Term	Field Code
1	01	98	PHOTO*	
2	04	01	EMULSION*	
3	03	01	074	S
4	02	01	*DYE*	
5	01	98	DANIZ, T*	A
6	01	98	KODAK CO*	C
7	98	99	INFRARED	

Figure 4

tape, and the computer skips to the next one. If there is a match, the machine searches only the desired groups from the update tape. When an input document satisfies the strategy, it is saved on the output tape for later printing. When the end of the update tape is reached, it is rewound and the process is repeated for the next strategy.

SUMMARY

If an information center is to be successful, it must be responsive to the demands of its users and clients. If a center has its own computer system, it can schedule batched runs, special runs, or evening runs to satisfy client demands, to meet higher priorities, or to overcome equipment failures. When a center has its own machine, it is paying a flat rental fee or fixed monthly amortization charge. Thus, additional computer use results in a lower per unit cost. Until recently, computer systems with good input/output were too expensive for most centers. Now, small, low-cost machines are available that permit a center to consider acquiring its own dedicated computer system. This paper has described a chemical information retrieval system currently being used on such a machine at the New England Research Application Center. Future work will report on the operational characteristics of this system.

ACKNOWLEDGMENT

We thank Stuart Harris for his contributions to programming of the system.

LITERATURE CITED

(1) Swid, R. E., "Linear *vs.* Inverted File Searching on Serial Access Machines," 26th Annual Meeting of the American Documentation Institute, Chicago, Ill. October 1963.
(2) Prentice, D., deGraw, G., Smith, A., and Warheit, I., "1401 Information Storage and Retrieval System (The Combined File System) 1401 IBM General Program Library 10.3.047."
(3) Starke, A. C., Whaley, F. R., Carson, E. C., and Thompson, W. B., "GAF Document Storage and Retrieval System," *Amer. Doc.* **19** (2), 173–80 (1968).
(4) Williams, Martha E., and Schipma, Peter B., "Design and Operation of a Computer Search Center for Chemical Information," *J. Chem. Doc.*, **10**, 158–62 (1970).
(5) Roberts, Anita B., Hartwell, Ieva O., Counts, Richard W., and Davila, Roberta A., "Development of a Computerized Current Awareness Service Using Chemical Abstracts Condensates," *Ibid.*, **12**, 221–3 (1972).
(6) Wilde, D. U., "Iterative Strategy Design," *Amer. Doc.* **20**, 90–91 (1969).
(7) Wilde, D. U., "Using a Small/Low Cost Computer in an Information Center," Proc. A.S.I.S. Mid-Year Regional Conference, Dayton, Ohio, May 1972.
(8) IBM Corporation, "TEXT-PAC, S/360 Normal Text Information Processing, Retrieval, and Current Information Selection System," (360D-06.7.020).
(9) Williams, M. E., *et al.* "Educational and Commercial Utilization of a Chemical Information Center," IITRI Rep. No. **C6156-18**, July 30, 1972, Chicago, Ill.
(10) Onderisin, E. M., "The Least Common Bigram: A Dictionary Arrangement Technique for Computerized Natural-Language Text Searching," IITRI, Chicago, Ill.

An Evaluation of a Substructure Search Screen System Based on Bond-centered Fragments

GEORGE W. ADAMSON, JUDITH A. BUSH, ALICE H. W. McLURE, and MICHAEL F. LYNCH
Postgraduate School of Librarianship and Information Science, University of Sheffield, Sheffield S10 2TN, England

Received October 8, 1973

A substructure search screening system based on bond-centered fragments has been evaluated using 108 queries derived from user SDI profiles. The average screenout value obtained was 98.42%. Simple, augmented, and bonded pairs are used as a hierarchy of structural descriptors giving easy coding and good performance for both general and specific queries.

This paper reports on an evaluation of the Sheffield screen search system at a stage in its development and forms one of a series reporting on the Sheffield substructure search system. The basic philosophy of the system and a description of the screen search system have already been presented.[1] The evaluation has been carried out using queries obtained from user profiles supplied by the Experimental Information Unit at Oxford. A quantitative investigation of the gross structural characteristics of queries has also been made. The evaluation so far has only involved measurements of screenout. Precision will be determined at a later stage when iterative search programs become available locally. The queries were run against a bit screen file generated from the Chemical Abstract Service sample file of 28,963 compounds. The bit screen layout has already been described[1] and uses simple, augmented, and bonded pairs[2] as a hierarchy of structural descriptors. Thus, the screen generation program is designed so that at whatever level a fragment is initially defined in a structure from the file, the lower levels of description are also automatically included. In the description of queries on the other hand, fragments are described in the query bit string at the level specified in the search question. If a description is not available at this level in the screen set, then the search program automatically describes the fragment at the next less specific level. With