Comparison of Five Methods for Searching Chemical Fragments

JOHN F. TINKER
Research Laboratories, Eastman Kodak Co., Rochester, N. Y. 14650

Received April 29, 1968

A sorter-collator card deck, a scan column index, a peek-a-boo set, a permuted alphabetical list, and a dual dictionary on microfilm, all based on fragment codes, were compared. The latter two were found to be faster to use than the first three. Digit coding was used for the peek-a-boo sets. Satisfactory digit coding can decrease substantially the number of cards needed for a large number of infrequently applied descriptors. The false answers introduced by the digit coding procedure are decreased when (a) the number of fragments per chemical in the index decreases, (b) the number of fragments used in searching increases, and (c) the fraction of the digit coding numbers that are unused increases. Surprisingly, the integrity of a sorter-collator deck was found to be poor. The ease of adding a new descriptor to the searching system is not related to the type of searching device employed but depends solely on the updating procedures.

Comparison studies of retrieval systems are relatively rare. Yet without such comparisons, it is difficult to improve a system or to design a new system. This paper reports a study of five retrieval systems, each containing identical information. These were a sorter-collator card deck,[1,2] a scan-column index,[3,4,5] a peek-a-boo set,[6] a list of permuted codes resembling a KWIC index,[7,8,9,10] and a dual dictionary on microfilm. These methods are illustrated for a particular descriptor, azo, applied to a chemical with the accession number 908480 (Figure 1). For use with the peek-a-boo method, to each of the chemicals in the file was assigned a three-digit sequence number, which appears in the lower left-hand corner of Figure 1. The terms consisted of fragment codes[1,2] applied to chemicals chosen randomly from an existing index. To compare the utility, a series of searches were compiled randomly. All searches were made using each of the searching methods and the retrieval time and number of retrieval errors were recorded. In addition, estimates were made of the delays involved in updating the searching methods and of the ease of introducing a new fragment term or descriptor.

The various methods differ in two other important aspects that influence their utility in some applications: the ease of publishing the index and the care that must be taken to maintain the index intact—i.e., the file integrity.

SEARCHES

The number of fragments in a search was varied from 1 to 15, and the search logic for each pair of terms could be an intersect or a join—e.g., Boolean AND or OR. The terms were chosen at random from the dictionary of codes but these search queries, for the most part, defined reasonable chemical classes. However, if the intersect of two terms was required, no answer was found among the 600 chemicals forming the sample set. This is a result

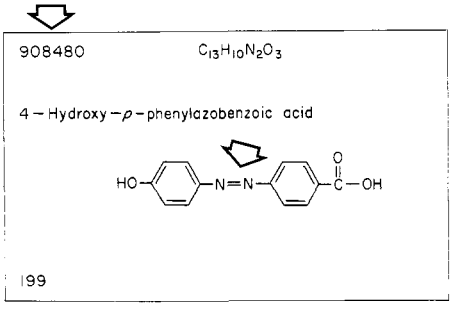


Figure 1. The original document of finding keys. The accession number (908480), the sequence number (199), and a chemical attribute (azo) are shown

of applying highly specific fragment descriptors to a small file. Figure 2 illustrates this point. It shows that one descriptor was used 174 times, two were used 56 times, but the majority were used infrequently. The average fragment is applied to 1.1% of the file; two randomly chosen fragments apply to about 0.1% of the chemicals in the index. Consequently, more searches were made by randomly choosing chemicals and fragments from the index information, instead of from the descriptor list. Of the total of 25 searches conducted by each of the five methods, 12 yielded *in toto* no answer, and 13 yielded one to several answers, an average of about two answers per search.

This experiment was intended to reflect on a small scale a much larger collection of index information. The larger collection might be, for example, an index to chemicals appearing in company reports. Such an index might accommodate a total of 100,000 different chemical individuals, of which 1 to 100 are index terms for each of 10,000 reports. A collection of this size is not readily suited to a peek-a-boo system, yet we wished to include a peek-

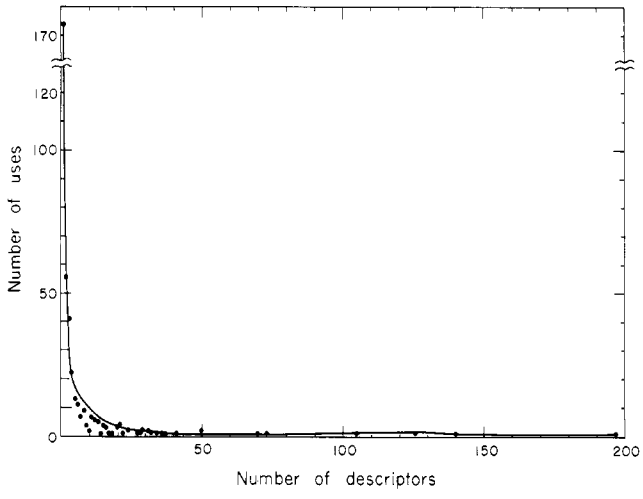


Figure 2. Frequency of use of descriptors for chemical fragments. Total number of chemicals is 600

a-boo index in the comparison—one which would be a miniature version of the one that would be required for the large collection.

In making a peek-a-boo deck for this large collection, one has considerable latitude in choosing the index terms. Each chemical, referred to by name or by accession number, might be an index term; but now 100,000 term cards are needed. Such an index would give good selectivity, but is cumbersome. At the other extreme, a single descriptor, "a chemical," might be used, resulting in fewer cards (only one, in fact) but less selectivity. Intermediate decisions can be made by choosing descriptors of the form "A chemical, the accession number of which begins with 9" (see Figure 3). This procedure is known as digit coding;[11] a smaller number of cards is needed, but some loss of selectivity results. For instance, in searching for all reports with the index term "chemical 000014," we would superimpose the six appropriate digit-coded cards,

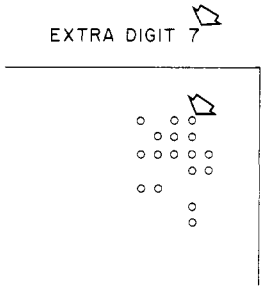


Figure 3. The peek-a-boo card *Extra Digit 7*, one of the four used to search for azo. The hole for sequence number 199 is indicated. The other three cards used to search for azo were called "First Digit 9, Second Digit 1, Third Digit 1"

locating one report, say, number 36; however, this might have, for example, the index terms "chemical 000010" and "chemical 000094," but not "chemical 000014," the actual search term. This false drop, or noise, has been introduced by the digit-coding procedure, in the same way it is introduced by superimposed coding.

A random 4-digit number was assigned to each fragment descriptor. For instance, the fragment descriptor "azo group connected to a carbon atom in a ring" was assigned 7911 (see Figure 3). The numbers were taken from a large random number table and checked to eliminate duplicates. Numbers that matched when the thousands digit was ignored were also eliminated, so that the 3-digit numbers uniquely refer to the fragment descriptors. These numbers are included in the dictionary, a part of which is shown in Figure 4.

False answers are introduced by this procedure, in addition to the false answers characteristic of any system. The 600 chemicals forming the sample had 403 descriptors given 4-digit numbers at random, between 0000 and 9999. Therefore most of the 4-digit numbers were not used. For instance, if chemical number 199 has an azo group (coded 7911) and another group (coded 2345), a hole is drilled in eight cards: thousands digits 2 and 7, hundreds digits 3 and 9, tens digits 1 and 4, and units digits 1 and 5. When combinations of four cards are superimposed, the hole at number 199 remains for the combination 2311, 2315, and the other six 4-digit numbers. Six of the eight are incorrect fragment descriptors of chemical 199. If all eight of these numbers were assigned to useful descriptors, the noise would be much too large. Hence, as the number of fragments per chemical is increased, the number of cards drilled increases and the noise becomes worse. In this experiment, as we have already pointed out, only 4% of the possible 4-digit numbers are assigned to terms; 96% of them are unused. When a single fragment descriptor is applied—e.g., in answering the request, "What compounds contain a single azo group connected to a carbon in a ring?"—the number of false answers is high but tolerable (about six wrong answers per correct answer). When the search involves the intersect of two descriptors, the noise is reduced (about one wrong answer per one correct answer).

	Number of Groups	Digit Codes	For Peek-a-boo	Codes for Other Keys
Arsonate or arsono	1	(4)	006	QJ51
Azide				
Connected to a carbon atom not in ring	1	(6)	253	JW51
Azo				
Connected to a carbon atom in a ring	1	(7)	911	**JT21**
	2	(3)	578	JT22
Azepine		(1)	728	0355
Azetidine		(8)	165	0043
Benzene				
See Aromatic Ring				
Bromide ions				
Uncertain connector	1	(4)	707	LB91
Bromine				
Connected to a carbon atom in a ring	1	(3)	118	LC21
	2	(8)	256	LC22
	3	(6)	618	LC23
	4	(6)	782	LC24

Figure 4. A portion of the dictionary of chemical attributes, showing codes used for azo

If the thousands digits are ignored, then 40% of the 3-digit code numbers have meaning. As expected, the noise increases sharply. When a single descriptor is applied, 20 to 50 wrong answers appear for each correct one; with two descriptors intersecting, there are still 3 to 6 wrong answers that appear for each correct one.

This is an intolerable search result. It does not demonstrate failure of the peek-a-boo system, nor failure of the fragment code—it merely emphasizes a principle that should be better appreciated. A searching method that gives facile Boolean intersect, of which peek-a-boo is an example, best fits a set of descriptors that are fairly broad, to which a constant subdivision can be applied. Many traditional classification schemes illustrate such a descriptor set and subdivision. However, if a peek-a-boo deck must be prepared from existing index information that consists of a large number of highly specific descriptors, a satisfactory deck may be prepared by using digit coding, if the procedure followed is this: Assuming that the descriptors have been assigned consecutive sequence numbers and that about 6 descriptors on the average have been applied to an item or document, add to each descriptor number a randomly chosen letter. Digit code the resulting string of alphanumeric characters. For instance, chemical fragment number 7069 becomes 7069Q, and in searching requires five digit-coded cards—first digit 7, second digit 0, third digit 6, fourth digit 9, and final letter Q. Note that the numbers 7069A through 7069P and 7069R through 7069Z are not used in the system. This procedure allows a peek-a-boo deck to consist of 26 + log N cards (where N is the number of descriptors) without introducing excessive noise. If the number of descriptors per item exceeds 6, or if lower noise is desirable, add two randomly chosen letters to the descriptor code.

None of the searching methods allowed an easy realization of a logical join, but the digit-coded peek-a-boo system was the most difficult to use in this way. A logical join is more likely to be needed when the descriptors are highly specific. Because the peek-a-boo deck leads to searching difficulties when directly coded and to false drops when digit-coded, it is not an optimum finding key for systems that involve deep indexing by a large collection of highly specific descriptors.

The figures show how the fragment "a20" in chemical 908480 is expressed in the various searching methods. The sorter-collator card deck[1] input card is shown in Figure 5, and the output card in Figure 6.

SEARCH TIME

In searching, the time is used in three ways: one way that depends on the total number of items in the file; another that depends on the number of alternate terms (Boolean OR) in the question; and a third that increases as the total number of indicated answers increases. Equation 1 is an expression for the total time, T, these three demands for time being taken into account.

$$T = k_1NO + k_2B \quad (1)$$

The first term accounts for the first two ways of taking time; k_1 is a constant that is characteristic of each system and reflects the effort required for setup of the search and for the scanning and matching during the search regardless of the file size.

The variable O in the formula is the number of "OR" terms in the Boolean expression describing the search procedure. This expression of the dependence was selected because it reflected exactly the way in which searching was actually done. That is, the expression $AB\ (C + D)$ could be carried out in a single searching operation, whereas $(F + G)\ (H + I)$ had to be carried out as two searches, $F\ (H + I)$ and $G\ (H + I)$. (This is true of all the methods except the sorter-collator deck.) The value of O is 1 for the former expression and 2 for the latter.

The second term accounts for the last way of taking time, and the constant k_2 reflects the effort required for each indicated answer. The variable B is the number of answers given by the searching method, counting both correct and incorrect results. For the peek-a-boo deck, k_2 represents the time taken to read and record an accession number, but not the time taken to look up the structure and verify the correctness. All operations, such as look-up and verifications, that are common to all finding keys have been ignored in calculating the constants, which illustrate only the differences in searching time.

In the equation, N is the number of items in the total store, in this case, 600. Obviously, in all cases except the peek-a-boo deck, adding another 600 items at least doubles the searching time. The dependence on N of the searching time with the peek-a-boo method is more complicated: adding 9400 more items does not change the time, assuming that one set can accommodate 10,000 cards; but one beyond—i.e., 9401—doubles it. Consequently, this dependence is ignored when the comparisons involve peek-a-boo and only the search time for 600 compounds, k_1N, was compared.

The efficiency of the simple equation in relating the search time to only three kinds of time demand is clear from Table I. The constant k_2 is determined with poor accuracy, but is of little importance. The average search time, k_1N, for three of the keys is determined fairly accurately. The probability that no difference in search time exists among any of the finding keys with the exception of the dual dictionary is less than 10%.

ERRORS

The error rates were recorded and were calculated as the average number of errors per search and as the average number of errors per correct answer. The rather high error rate for the peek-a-boo deck is a result of the digit coding, as already discussed. The number of errors resulting from use of the sorter-collator deck was 10 times as great as in the two manual searching systems. This rather surprising result shows that the completeness or integrity of a set of punched cards is not easily guaranteed.

Difficulty in maintaining a file that consists of many separate précis has often been noted, particularly when many people have access to it and when a relatively complicated filing procedure is needed (as in filing cards by molecular formula). The poor file integrity of the sorter-collator deck is probably caused by inadequate control of filing procedures if the filing is not done entirely by machine.

When comparing the search times shown in the table, it is important to remember that the times were measured on a sample of 600 chemicals. The value of k_1 for three of the methods should not change substantially with

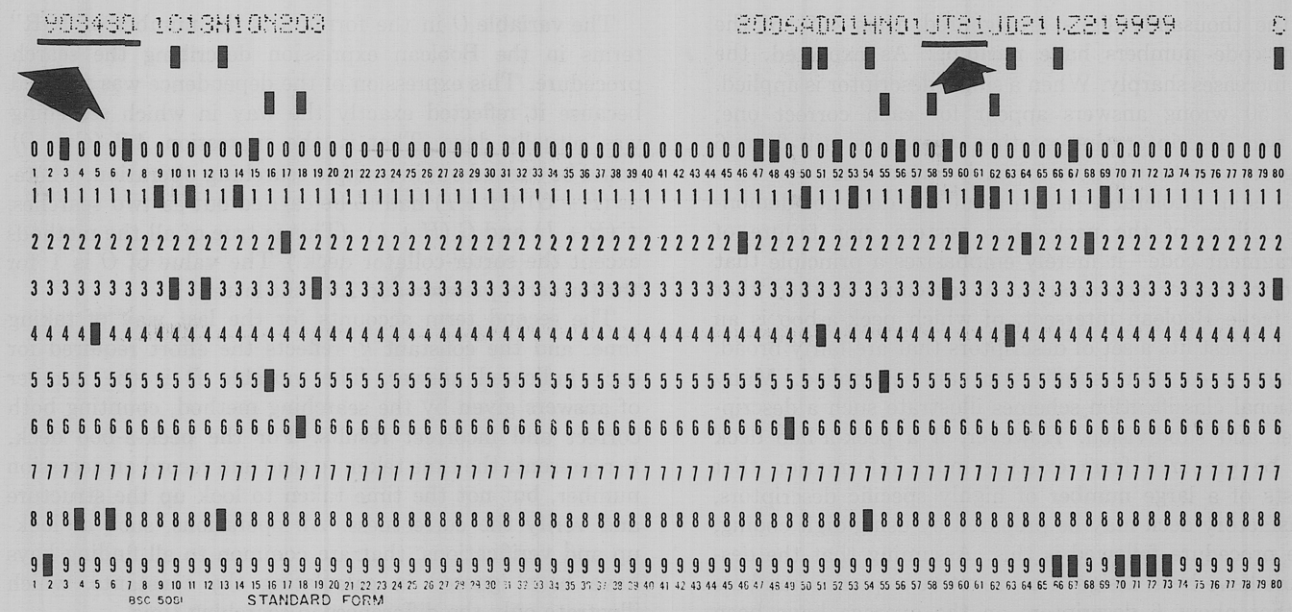


Figure 5. Input card for chemical fragments, showing azo code for chemical 908480

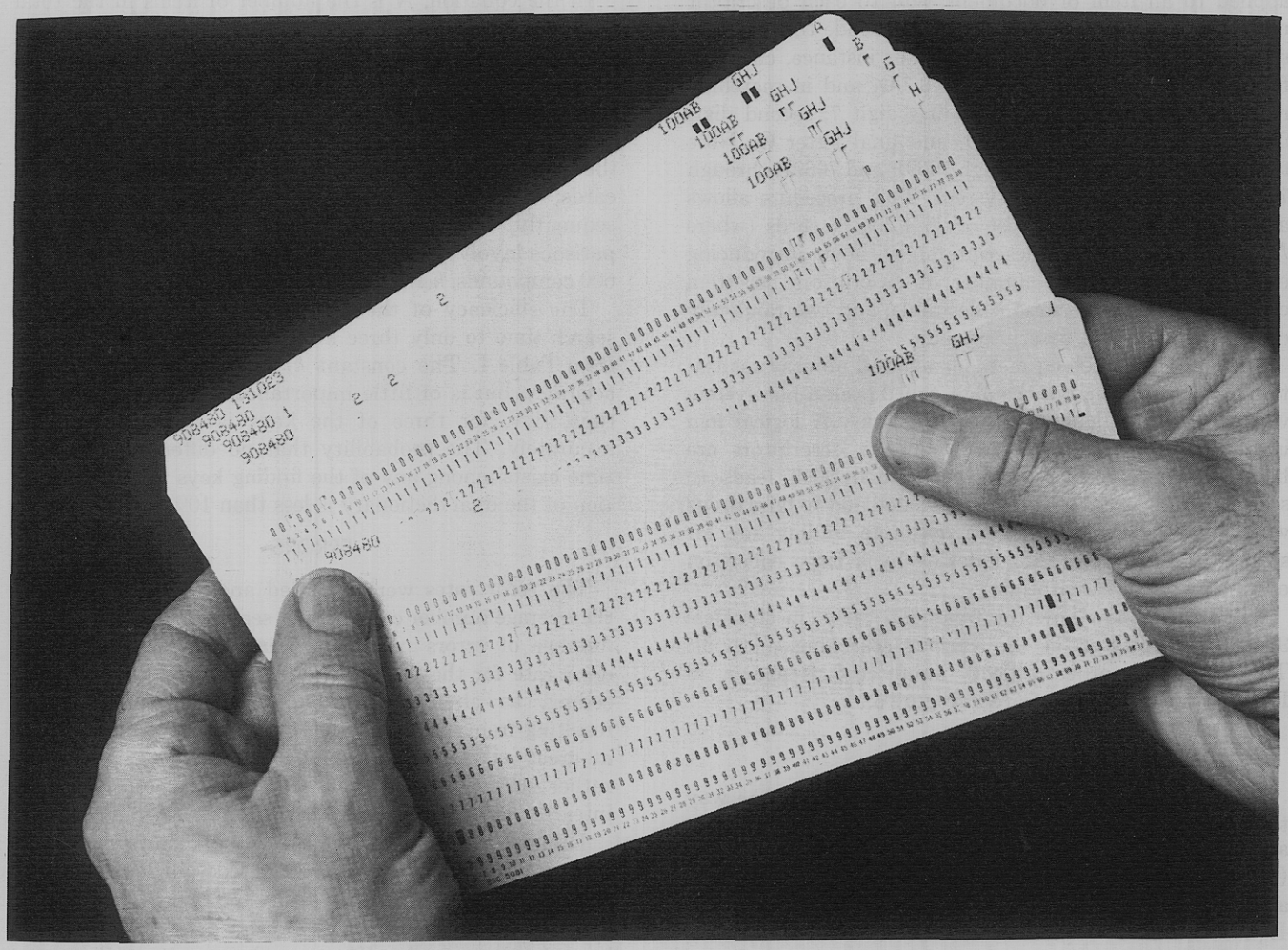


Figure 6. The sorter-collator deck for chemical 908480, showing the punch for JT21, the code for azo

change in N. (The same value of k_1 has been found for one method with N = 50,000.) However, the rate k_1 for the peek-a-boo deck does change. For instance, when n is increased from 600 to 6000, the rate k_1 for the 4-digit coding (the deck of 40 cards) goes from 0.15 second per compound to 0.015 second. However, the error rate is multiplied by 10, going to about 83 errors per search, or 29 per correct answer. To reduce the error rate, more digits can be used: for 6-digit coding (a deck of 60 cards), the values estimated for N = 6000 are k_1, 0.03 second per compound, an error rate of 20 per search, or 8 per correct answer.

Table I. Search Times and Error Rates for Five Finding Keys

Finding Key	Search Time, Seconds[a] k_1N	k_1, per compound	k_2	Errors Per search	Per correct answer
Permuted index	33 ± 12	0.055 ± 0.02	26 ± 20	0.12	0.041
Scan-column index	130 ± 16	0.220 ± 0.03	35 ± 31	0.19	0.068
Peek-a-boo deck					
3-digit coding	75 ± 10		29 ± 30	19	6.7
4-digit coding	90 ± 15		35 ± 28	8.3	2.9
Sorter-collator	250 ± 140	0.420 ± 0.18	113 ± 130	1.2	0.41
Dual dictionary[b]	3	Not comparable	1	0.5	0.05

[a] The probable error is given for each of the times. [b] Estimated from measurements on a sample of 50,000.

Acc. No.	AD	A	B C D	E F G	HN	H	I	J	K	L	M N O	P Q R	S T U	V W X	Digits
908473	AD01							JD26							
908474	AD02				HN02	HP01	IW21	JF51							
908480	AD02				HN01		IZ21	JD21							
								JT21							
908485					HN02	HP03	IV51	JC51							
908501			AF01	EI01	HN01	HP01	IU21								0142

Figure 7. A portion of the scan-column index, showing number 908480 found during search for azo

ACCESSION NUMBER	FIRST PERMUTED TERM	OTHER TERMS						
907642	JS21	AF01	ET01	HN01	HP01	IU21	QE01	0127
907261	JT21	AD01	AF01	EJ01	HN01	HP02	JN21	0277
901505	JT21	AD02	HN01	HP01	JD21			
908480	JT21	AD02	HN01	IZ21	JD21			
908277	JT21	AD02	HN01	JD21	KF21	QX21		
907492	JT21	AD02	HN02	IW21	LC51			

Figure 8. A portion of the permuted listing, showing 908480 found during search for azo

NEW DESCRIPTORS

An information system with a fixed vocabulary tends to become moribund. The addition of new terms and the modification of existing terms must be easy to accomplish, or the system will fall into disuse.

It is difficult to add descriptors and new indexing information for documents that have entered the system at an earlier time, and the difficulty increases as the document collection grows. It has often been noticed that, in certain operating systems, the rate of growth of the descriptor list falls off. This phenomenon, ordinarily explained by alleging that an adequate descriptor list has accumulated, may, instead, reflect the increasing difficulty of adding descriptors.

Two types of problems are encountered in making additions: procedural and logical. The first can be alleviated by good planning of work procedures. The logical problem is this: How can one be sure that the new descriptor has been applied as required to the entire collection? The initial design for an information system must include specific and detailed procedures for adding descriptors and index information, to maintain high recall.

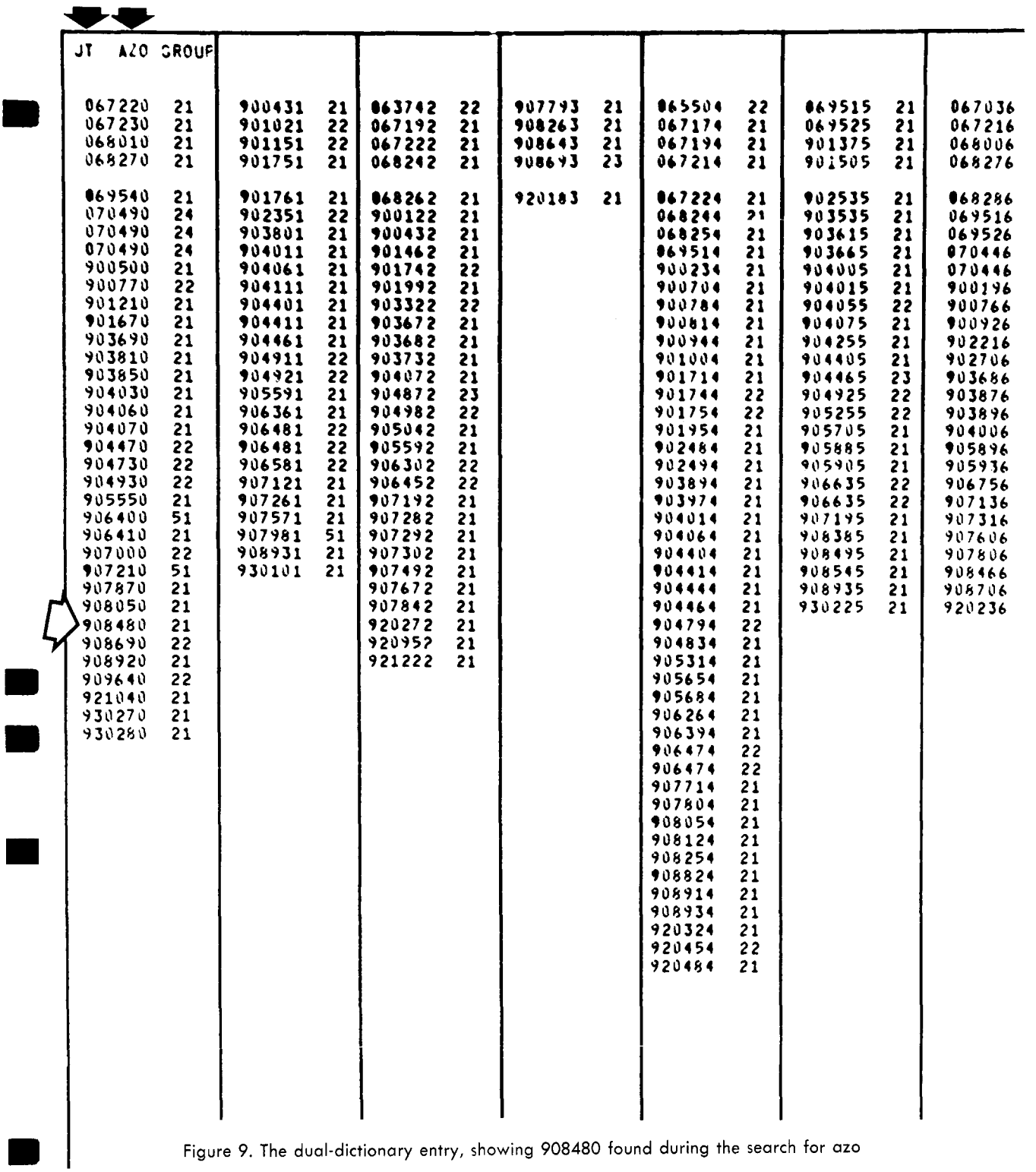
One case can be adequately handled in the following way: Suppose that an expert is looking over a fraction of the document collection and decides that part of this fraction deserves a new descriptor, and part does not. However, he has the answer he wants and is unwilling to examine the rest of the corpus to decide whether or not the new descriptor applies; he thinks this unnecessary and unrewarding. Yet this intellectual effort, a most expensive and valuable commodity, should be captured in a form useful to others. How can this be done?

The steps needed are these: (1) The new descriptor is entered in the thesaurus, along with notes, other terms, and relations, so that its meaning will be clear to anyone with training in the area of knowledge. The new descriptor is tagged to make it obvious that it applies only to a part of the file. (2) Its logical obverse is also entered, with appropriate notes. (3) The index information for the new term and its obverse for that part of the corpus are added.

The searcher can use these new descriptors to subdivide the library into any of the three parts: (1) to which the descriptor has been applied, (2) to which the obverse has been applied, and (3) to which neither has been applied.

For instance, if the searcher wants high relevance, but does not require high recall, he should select only those documents retrieved in part (1). If, however, he must have high recall, he will reject part (2) and retain parts (1) and (3).

Each of the five search methods could be readily modified by adding or changing a fragment descriptor. However, serious difficulties arose when such a change was attempted. Depending on the way the index material was handled, there might be a long lapse between the time the indexer decided to apply the term and the time it was incorporated into the finding key. During this time, the material was difficult to identify and to change. Consequently, the ease of making these changes depends not on the searching method, but on the procedures used for generating and transferring the index information. If these procedures are slow or ineffective, updating the



JT AZO GROUP													
067220	21	900431	21	063742	22	907793	21	065504	22	069515	21	067036	
067230	21	901021	22	067192	21	908263	21	067174	21	069525	21	067216	
068010	21	901151	22	067222	21	908643	21	067194	21	901375	21	068006	
068270	21	901751	21	068242	21	908693	23	067214	21	901505	21	068276	
069540	21	901761	21	068262	21	920183	21	067224	21	902535	21	068286	
070490	24	902351	22	900122	21			068244	21	903535	21	069516	
070490	24	903801	21	900432	21			068254	21	903615	21	069526	
070490	24	904011	21	901462	21			069514	21	903665	21	070446	
900500	21	904061	21	901742	22			900234	21	904005	21	070446	
900770	22	904111	21	901992	21			900704	21	904015	21	900196	
901210	21	904401	21	903322	22			900784	21	904055	22	900766	
901670	21	904411	21	903672	21			900814	21	904075	21	900926	
903690	21	904461	21	903682	21			900944	21	904255	21	902216	
903810	21	904911	22	903732	21			901004	21	904405	21	902706	
903850	21	904921	22	904072	21			901714	21	904465	23	903686	
904030	21	905591	21	904872	23			901744	22	904925	22	903876	
904060	21	906361	21	904982	22			901754	22	905255	22	903896	
904070	21	906481	22	905042	21			901954	21	905705	21	904006	
904470	22	906481	22	905592	21			902484	21	905885	21	905896	
904730	22	906581	22	906302	22			902494	21	905905	21	905936	
904930	22	907121	21	906452	22			903894	21	906635	22	906756	
905550	21	907261	21	907192	21			903974	21	906635	22	907136	
906400	51	907571	21	907282	21			904014	21	907195	21	907316	
906410	21	907981	51	907292	21			904064	21	908385	21	907606	
907000	22	908931	21	907302	21			904404	21	908495	21	907806	
907210	51	930101	21	907492	21			904414	21	908545	21	908466	
907870	21			907672	21			904444	21	908935	21	908706	
908050	21			907842	21			904464	21	930225	21	920236	
908480	21			920272	21			904794	22				
908690	22			920952	21			904834	21				
908920	21			921222	21			905314	21				
909640	22							905654	21				
921040	21							905684	21				
930270	21							906264	21				
930280	21							906394	21				
								906474	22				
								906474	22				
								907714	21				
								907804	21				
								908054	21				
								908124	21				
								908254	21				
								908824	21				
								908914	21				
								908934	21				
								920324	21				
								920454	22				
								920484	21				

Figure 9. The dual-dictionary entry, showing 908480 found during the search for azo

finding key is slow, the descriptors tend to be inflexible, and the system will lose its usefulness. The procedures used for transferring index information must be designed with careful consideration to the need for flexibility of the descriptors.

SOME ADVANTAGES AND DISADVANTAGES OF THE FIVE FINDING KEYS

I. Sorter-Collator Deck. Table I reveals the rather poor performance of this method. It is a convenient and effective system for a small collection, but it was expected to become less efficient as the size of the file increased. We realized when we adopted this method that it would become increasingly inefficient, and it became almost useless when the number of chemicals reached about 80,000.

Rather unexpectedly, the file integrity is poor; cards are lost or misfiled, or duplicated or key-punched incorrectly. Errors by machine occur; for instance, duplicating worn or damaged cards occasionally leads to dropped punches. Possibly machine verification of these duplicated

cards would help. The greater number of losses occurs during clerical manipulation. Related studies of filing procedures show abundantly that complicated refiling procedures (resembling those necessary in using the card deck) can be accompanied by errors in 5 to 10% of the filed items unless procedures are carefully established and followed. Using the card sorter as part of the filing routine reduces the error rate, but not to zero.

The update time of the methods that involve machines depends to a considerable degree on the schedules of the machine operations. The times quoted are approximations based on our experiences in 1965–66. For the sorter-collator deck, three steps are involved: key-punch, pass through a computer program, and finally sort and file. The delay is about 15 days, and the range of delay times extends to several months.

II. Scan-Column Index. Figure 7 shows a sample page from a scan-column index. Rapid and easy updating, of either descriptors or new items, is possible; a new entry can be added within a few seconds of the time the index terms have been applied. However, the searching procedure (flipping pages) becomes tiring and inaccurate. A microfilm version would help, but only by sacrificing the easy updating. A more detailed discussion of the advantages is given by O'Connor,[3] Wiswesser,[4] and the J. T. Baker Chemical Co.[5]

III. Permuted Alphabetical Listing. The list of fragment codes was considered to be a title. It was permuted as described by Luhn[7] and exemplified by Figure 8. The accession number 908480 appears under AD, HN, IZ, and JD, as well as under JT, as shown in the figure. A frequency table precedes the listing, allowing the user to locate the smallest section that applies to his question in those cases in which this procedure is appropriate. If a properly conceived thesaurus is employed, the frequency table is of limited usefulness. In bulk, the listing is about twice the size of the scan-column index, equal to the peek-a-boo set, and a third the size of the sorter-collator deck. The complete listing of 100,000 chemicals and 1400 codes can be printed from magnetic tape directly on 16-mm. microfilm and stored in two cartridges. (The reference file for 100,000 chemicals occupies 52 cartridges, 4 for alphabetical trivial names, 22 for accession number sequence, and 26 for molecular formula sequence.)

One disadvantage of the listing is that the answers are not accumulated in order as they are in the other three methods. If a query has many answers, the checking is slowed if successive answers are not on the same reel of microfilm of the reference file. This disadvantage can be overcome by use of a work sheet on which the accession numbers for the compounds on the first frame of each microfilm reel have been printed along the top of each column. As the search is made, each answer can be recorded in the appropriate column. The answers in each column compared to the contents of a single reel and look-up can be carried out efficiently.

The delay expected in updating the permuted list is from 5 to 10 days and is caused principally by scheduling and delivery problems.

IV. Peek-a-boo Deck. Maintaining multiple copies of a peek-a-boo deck presents dificulties. Photographic reproduction systems have been tested, but no completely satisfactory one has been found.

Several peek-a-boo decks are maintained by a central service group. The update delay is about two days, extending to several more days. A considerable part of this time is taken by delivery.

V. Dual Dictionary on Microfilm. Figure 9 illustrates the display. The descriptor code, JT, is given at the top of the page, followed by the code translation, "Azo group." The connector and multiple, 21 for chemical number 908480, appear after the accession number. The numbers are arranged in columns, with all numbers having terminal digit 0 in one column, those with terminal digit 1 in a second column, and so on. As many frames of microfilm will be generated as are needed to accommodate the numbers. (This system is described in more detail in a forthcoming paper.)[12]

The update delay is about the same as with the permuted alphabetical listing—i.e., 5 to 10 days—and is a result of the same types of problems.

Table I summarizes the results of time and error studies. It is not intended as proof of the intrinsic superiority of one or another method, but as a guide to designers of information systems in choosing a manual searching method. The table illustrates the trade-off in digit-coded peek-a-boo systems between search time and error rate. Adding more digits increases the search time but reduces the error rate.

The greatest differences among the five keys lie in ease of publication and in file integrity. An index must have these attributes if it is to be used by many searchers, to index a large file, and to have good accuracy.

The various searching methods have individual strong points and disadvantages. Once the designer has assessed the importance of certain attributes of the system—rapidity of update is most important—he can choose a searching method accordingly.

LITERATURE CITED

(1) Haefele, C. R., and J. F. Tinker, J. CHEM. DOC. **4**, 112 (1964).
(2) Gelberg, A., W. Nelson, G. S. Yee, and E. A. Metcalf, *Ibid.*, **2**, 7 (1962).
(3) O'Connor, John, "The Scan-Column Index, a Book-Form Coordinate Information Retrieval System," AD-236 466, Remington-Rand, Univac Div., Sperry Rand Corp., Philadelphia, Pa., February 1960.
(4) Wiswesser, W. J., *Adv. Chem. Ser.* **16**, 64 (1956).
(5) BATCH Directory, J. T. Baker Chemical Co., Phillipsburg, N.J., 1965.
(6) Jonker, F., *Am. Doc.* **11**, 305 (1960).
(7) Luhn, H. P., *Am. Doc.* **11**, 288 (1960).
(8) Granito, C. E., A. Gelberg, J. E. Schultz, G. W. Gibson, and E. A. Metcalf, J. CHEM. DOC. **5**, 52 (1965).
(9) Granito, C. E., J. E. Schultz, G. W. Gibson, A. Gelberg, R. J. Williams, and E. A. Metcalf, *Ibid.*, **5**, 229 (1965).
(10) Gelberg, A., *Ibid.*, **6**, 60 (1966).
(11) "Systems Manual," Section IV, pp. 6,7, Jonker Corp., Gaithersburg, Md. 1967.
(12) Tinker, J. F., *Am. Doc.*, in press, 1969.