for output or display in the range of 1000 different character representations.

Many problems (challenges) remain. Information processing requires more than a character set as the following hierarchy suggests:

Information
Knowledge
Language
Code
Alphabet
Character
Symbol
Sign

Meaning is a function of context. The above hierarchy suggests the context of a character. The data element/record/file/base formats and the precise meanings of the individual elements are an integral part of the code and language of information transfer. Without this context we do not know the meaning of our character sets.

"A " What is it? Without more context, I don't know. Do you? Is it one allograph of the first letter of the English (Latin) alphabet? Is it the 10th digit in hexadecimal notation? Is it a variable in a mathematical equation? Is it one allograph of the first letter in the Greek alphabet? Is it a student's grade? Is it the graphic symbol representation of the 8-bit code "11000001" in the code called EBCDIC?

Data Compression of Large Document Data Bases[†]

H. S. HEAPS

Department of Computer Science, Concordia University, Sir George Williams Campus, Montreal, Canada

Received November 8, 1974

Consideration is given to a document data base that is structured for information retrieval purposes by means of an inverted index and term dictionary. Vocabulary characteristics of various fields are described, and it is shown how the data base may be stored in a compressed form by use of restricted variable length codes that produce a compression not greatly in excess of the optimum that could be achieved through use of Huffman codes. The coding is word oriented. An alternative scheme of word fragment coding is described. It has the advantage that it allows the use of a small dictionary, but is less efficient with respect to compression of the data base.

INTRODUCTION

The subject of data compression includes many different aspects. The present paper is concerned with compression of textual data such as might appear in document titles, abstracts, author names, and so forth. It might include numerical data, but the methods to be described do not take advantage of relations that might exist between different sets of numerical data.

A data base for document retrieval may be envisaged as a set of records that are divided into fields, some of which may be searched for the presence of terms as specified in a question statement. Each searchable field is associated with a particular attribute such as "author names," "title," "abstract," "keywords," and so forth. There are many different ways in which such data bases may be structured for storage on computer accessible files.

The simplest method of structuring the data base is in a sequential manner as shown in Figure 1. An obvious disadvantage of sequential storage of a large data base is that any search for the records that contain particular terms involves search through the entire data base, and this is apt to be very time-consuming and therefore costly.

A different means of structuring a document data base is through use of an inverted file. A further alternative is through use of a hierarchically structured dictionary designed with reference to a hierarchical scheme of document classification. A detailed comparison of the two alternatives from the point of view of their effect on computing efficiency has been made by Ein-Dor.[1]

The form of inverted file shown in Figure 2 proves convenient for illustration of the concepts to be discussed in the present paper. It consists of a term dictionary in which title terms, author names, keywords, and so forth are arranged in alphabetic, or other, order together with pointers to sets of document numbers, or accession numbers, that indicate those documents that contain the particular term. A search for the documents that contain specified terms, or satisfy a particular question logic, involves application of logic operations to the appropriate sets of document numbers.

Once the pertinent document numbers have been determined, the document texts may be read from the sequential file. In order to avoid having to read the entire sequential file, a document directory may be referenced first. It consists of pointers that indicate the position at which a document is stored in the sequential file.

It is clear that, while an inverted file structure allows more rapid retrieval of information, it uses more storage than does a purely sequentially structured data base since, in fact, a sequential file is also required in order to retrieve the details of any stored document item. However, as indicated below, the sequential file shown in Figure 2 may be stored in a coded form which occupies less space than the sequential file shown in Figure 1.

This is possible because the file structure of Figure 2 includes a term dictionary of all the different terms present in the sequential data base, and so, instead of repeating these terms in the sequential file, it is sufficient to store instead a pointer to the position of the term dictionary. This

[†] Presented in the "Conference on Large Data Bases," sponsored by the NAS/NRC Committee on Chemical Information, National Academy of Sciences, May 22–23, 1974.

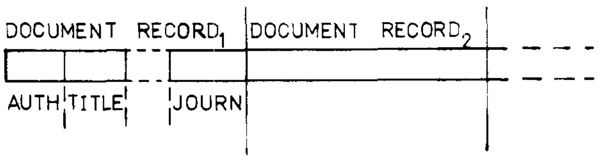

Figure 1. Records of a sequentially structured document data base.

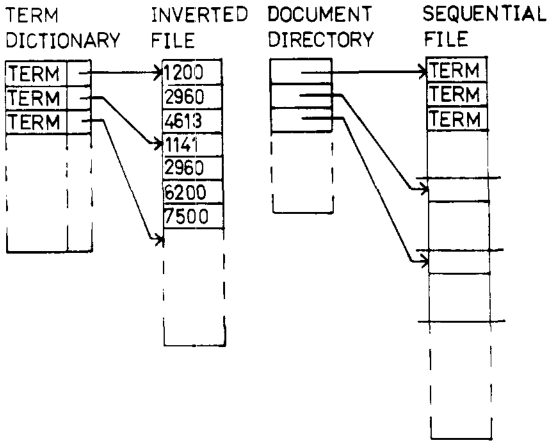

Figure 2. Document data base structured in terms of an inverted file and a directory to a sequential file.

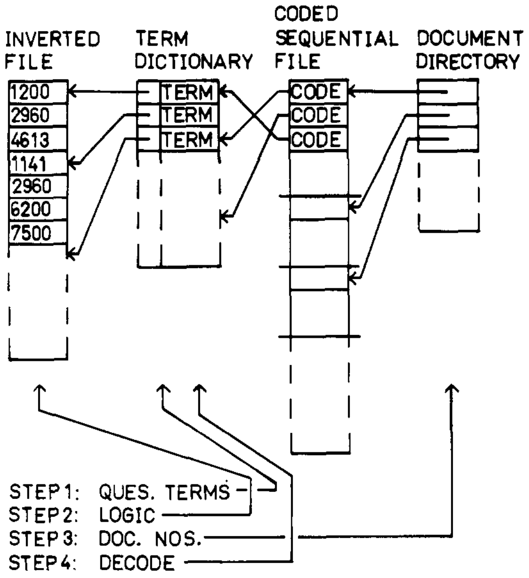

Figure 3. Data base in which terms in the sequential file are coded as pointers to the term dictionary.

is shown in Figure 3 in which the steps of the retrieval process are also shown. These steps will be discussed later.

If there are D different terms in the original data base, then each code stored in the coded sequential data base requires only $\log_2 D$ bits. A term of length L bits that appears n times in the original data base contributes to the inverted file storage through requiring L bits in the term directory and $n \log_2 D$ bits in the coded sequential file. Replacement of the sequential file by a term dictionary and coded sequential file therefore allows a space saving of

$$(n - 1)L - n \log_2 D \text{ bits} \qquad (1)$$

for each term that appears n times in the original data base. For frequently occurring terms such a space saving is significant. For terms that appear only once in the data base there is no saving, and, in fact, for such terms there is space wasted through storage of the code of length $\log_2 D$ bits.

For a data base structured as in Figure 3 the question processing proceeds as follows, and as illustrated by the four steps listed in Figure 3. The terms of each question are searched for in the term dictionary, and the corresponding pointers are used to access the inverted file of document numbers. The question logic is applied to these sets in order to find the accession numbers of the documents that satisfy the logic. Then the document directory and sequential file are accessed, and finally the codes in the relevant portions of the sequential file are decoded by reference to the term dictionary.

For a given sequential data base in the form shown in Figure 1, creation of the inverted file structure of Figure 3 is a relatively trivial operation that may be carried out essentially by a sequence of sort operations. Two important problems that arise are storage of the files to occupy a minimum of space, and further structuring of the files to allow the four steps of the search to proceed with a minimum number of file accesses. Methods for organizing and handling such files have been discussed by a number of authors.[2-5] Hash storage schemes offer some attractions. Use of a hash storage scheme, and the structuring of a file, for an on-line library catalog of one million volumes has been described in a previous paper.[6] A different "tree-like" structuring of the dictionary has been used by Benbow[7] for design of an on-line search system for a small UDC based library of the Boreal Institute for Northern Studies at the University of Alberta; in this system the structuring is designed to take advantage of the structure of the UDC classification numbers and the UDC subject words.

The above discussion has served to introduce the idea of coding the terms present in the sequential file, although such coding is of a simple nature and has been used many times. It also leads to the question as to how to best code the data in order to produce the most compressed form of data base. It is obvious that codes of fixed length are not the best to use for textual terms and that better compression should result from use of variable length codes in which the shortest codes are used to represent the most frequent terms. A term of very infrequent occurrence does not have much effect on the length of the data base regardless of how it is stored, whereas representation of a frequent term by a code of shorter length contributes significantly to the overall compression.

It is therefore appropriate to discuss some general properties of word occurrences in data bases. This provides some insight that is useful in connection with the general problem of coding for compression.

VOCABULARY CHARACTERISTICS OF DOCUMENT DATA BASES

If a sufficiently large sample of general English text is examined, and the different words are ranked in descending order of their frequencies of occurrence, it is found that the word of rank r occurs with a probability p_r that is given approximately by the equation

$$p_r = A/r \qquad (2)$$

where A is a constant that depends on the length and type of text. The equation is known as the Zipf law[8] and may be verified by examination of word frequency counts such as have been made by Kučera and Francis.[9] If there are D different words, the value of A may be computed from the formula

$$A = \frac{1}{1 + 1/2 + 1/3 + \ldots + 1/D} \qquad (3)$$

Table I. Values of rp_r for Words of General Text and for Title Terms in Three Different Data Bases

	Kučera and Francis, N = 1,000,000			*Chemical Titles*, 1965, N = 1,058,359			MARC 01-58, N = 317,581			*Gas Chromatography*, N = 220,000 (approx)		
Rank	Term	Freq	rp_r	Term	Freq	rp_r	Term	Freq	rp_r	Term	Freq	rp_r
1	THE	69,971	0.070	OF	107,687	0.102	THE	25,647	0.081	OF	23,569	0.107
2	OF	36,411	0.073	AND	37,578	0.071	OF	21,471	0.135	GAS	11,091	0.101
3	AND	28,852	0.086	THE	36,318	0.103	AND	12,975	0.123	THE	9,560	0.130
4	TO	26,149	0.104	IN	32,868	0.124	IN	8,987	0.113	AND	8,141	0.148
5	A	23,237	0.116	ON	10,984	0.052	A	5,149	0.081	CHROMATOGRAPHY	7,776	0.177
6	IN	21,341	0.128	BY	10,727	0.070	TO	3,741	0.071	IN	7,086	0.193
7	THAT	10,595	0.074	A	10,252	0.068	FOR	3,464	0.076	BY	4,031	0.128
8	IS	10,099	0.081	DI	8,419	0.064	ON	2,726	0.069	CHROMATOGRAPHIC	3,566	0.130
9	WAS	9,816	0.088	WITH	7,964	0.068	HISTORY	1,313	0.034	FOR	3,167	0.130
10	HE	9,543	0.095	FOR	6,509	0.062	NEW	1,203	0.038	ANALYSIS	3,061	0.139
20	I	5,173	0.103	METHYL	4,030	0.076	LAW	649	0.041	FROM	1,223	0.111
30	THEY	3,618	0.108	ACIDS	2,697	0.076	AMERICA	511	0.048	COMPOSITION	702	0.096
40	THEIR	2,670	0.107	5	2,236	0.085	INTRODUCTION	439	0.055	APPLICATION	576	0.105
50	IF	2,199	0.110	AN	1,890	0.089	HIS	371	0.058	OILS	492	0.112
100	WELL	897	0.090	PER	1,210	0.114	BUSINESS	302	0.095	MILK	256	0.116
200	ALMOST	432	0.086	SULFIDE	736	0.139	INFORMATION	162	0.102	AMINES	136	0.124
300	HELP	311	0.093	8	503	0.143	CONSTRUCTION	118	0.111	SPECTROSCOPY	89	0.121
400	TURN	233	0.093	METALLIC	395	0.149	MODEL	93	0.117	PEAKS	67	0.122
500	STARTED	194	0.097	FLUORESCENCE	316	0.149	CHARACTERISTICS	78	0.123	PRELIMINARY	53	0.120
1000	REACH	106	0.106	TOXIN	147	0.139	IMPLICATIONS	45	0.142	STERIC	24	0.109
2000	SOLDIERS	56	0.112	OXYTOCIN	67	0.127	DIPLOMACY	20	0.126	JUICES	9	0.082
3000	SURVEY	37	0.111	DECREASE	36	0.102	PSYCHIC	13	0.123	DILUENTS	5	0.068
4000	SOUTHERNERS	26	0.108	GERMINATING	20	0.076	FRICTION	9	0.113	ANCHIMERICALLY	3	0.055
5000	ATTRACT	19	0.095	RESONATOR	14	0.066	PEASANT	7	0.110	WORLD	3	0.068

Table II. Values of rp_r for Subject Descriptors on the *Gas Chromatography* Tapes

	Type 1, N = 59,613			Type 2, N = 10,155			Type 3, N = 17,322		
Rank	Term	Freq	rp_r	Term	Freq	rp_r	Term	Freq	rp_r
1	ACID	2850	0.048	ISOMERS	766	0.075	REVIEW	915	0.053
2	HYDROCARBONS	1127	0.038	ESSENTIAL OILS	658	0.130	COLUMN PACKING	769	0.089
3	ALCOHOLS	1123	0.056	FLAVORS	345	0.102	INSTRUMENTATION	721	0.125
4	PESTICIDES	678	0.045	LIPIDS	305	0.120	PYROLYSIS	662	0.153
5	NITROGEN COMPOUNDS	651	0.054	URINE	291	0.144	MASS SPECTROSCOPY	641	0.185
6	ALDEHYDES	613	0.062	BLOOD	272	0.160	DETECTORS	598	0.207
7	ESTERS	579	0.068	AIR	248	0.171	THEORY	554	0.224
8	STEROIDS	541	0.073	FOODS	217	0.171	DERIVATIVES TMS	485	0.224
9	KETONES	485	0.073	POLYMERS	184	0.162	DERIVATIVES	479	0.254
10	HALOGENATED COMPOUNDS	427	0.072	PETROLEUM	163	0.160	PREPARATIVE DETECTORS	393	0.227
20	AROMATIC HYDROCARBON	248	0.083	CIGARETTE SMOKE	66	0.130	IONIZATION	157	0.181
30	SUGARS	217	0.109	TRIGLYCERIDES	44	0.130	GAS SOLID	96	0.166
40	ETHYLENE	165	0.110	WINE	33	0.130	CARRIER GAS	65	0.150
50	DIELDRIN	123	0.103	COSMETRICS	25	0.123	ADSORPTION ISOTHERMS	47	0.135
100	ISOPROPANOL	60	0.100	PHOSPHOLIPIDS	14	0.138	MOLECULAR WEIGHT	17	0.098
200	BUTENE	31	0.104	ALFALFA	7	0.138	ARGON IONIZATION	6	0.069
300	HYDROCARBONS A E	21	0.105	PETROLEUM PRODUCTS	5	0.148	DIGITAL CONTROL	4	0.069
400	MONOTERPENES	15	0.100	AIR SPACECRAFT	3	0.118	FLASH EXCHANGE	3	0.069
500	PYRUVIC	12	0.100	OIL SHALE	3	0.148	ATTENUATORS	2	0.058
1000	SUGAR ALCOHOLS	6	0.100						
2000	ACIDS BRANCHED	2	0.067						
3000	PREGNANEDIONE	2	0.100						

which may be approximated in the form

$$A = \frac{1}{\log_e (2D + 1)} \qquad (4)$$

The Zipf law implies that the product rp_r is constant. It is found to be approximately true, not only for general English text but also for the distribution of terms present in many different types of fields within document data bases. The extent to which it is satisfied for different selections of title words, keywords, and publisher names is illustrated in Tables I–III.

In Table I some values of rp_r are listed for general text and for title terms of the *Chemical Titles* tapes issued by Chemical Abstracts Services during 1965. The corresponding values are also listed for an accumulation of 58 issues of the MARC library tapes that appeared between March 1969 and May 1970. The values shown in the final columns of Table I are for some of the *Gas Chromatography* tapes prepared by *Preston Abstracts*. In each instance N denotes the total number of terms present in the sample considered.

Of the four data bases represented in Table I, that of

Table III. Values of rp_r for Subject Headings and Publisher Names on the MARC Tapes

	Subject headings, N = 60,778			Publisher names, N = 53,387		
Rank	Term	Freq	rp_r	Term	Freq	rp_r
1	NEGROES	398	0.007	U.S. GOVT. PRINT. OFF	1829	0.034
2	GEOLOGY	351	0.011	BOOKS FOR LIBRARIES PRESS	1213	0.045
3	ENGLISH LANGUAGE	344	0.017	GREENWOOD PRESS	795	0.045
4	EDUCATION	287	0.019	MACMILLAN	569	0.043
5	SLAVERY IN THE UNITED STATES	279	0.023	DOUBLEDAY	547	0.051
6	AGRICULTURE	257	0.025	NEGRO UNIVERSITIES PRESS	547	0.061
7	ELECTRONIC DATA PROCESSING	288	0.026	MCGRAW-HILL	537	0.070
8	CITIES AND TOWNS	222	0.029	FOR SALE BY THE SUPT. OF DOCS., U.S....	524	0.079
9	SCIENCE	197	0.029	PRENTICE-HALL	495	0.073
10	WORLD WAR, 1939–1945	182	0.030	KENNIKAT PRESS	438	0.082
20	RAILROADS	125	0.041	PUTNAM	233	0.087
30	WATER	105	0.052	NATIONAL AERONAUTICS AND SPACE ADMINISTR	155	0.087
40	ANIMALS	96	0.063	METHUEN	137	0.103
50	ARCHITECTURE	86	0.071	F. WATTS	127	0.119
100	WOMAN	63	0.104	PATTERSON SMITH	76	0.142
200	TREES	39	0.128	BODLEY HEAD	41	0.153
300	COMMUNITY MENTAL HEALTH SERVICES	29	0.143	MURRAY	28	0.157
400	SOCIAL PROBLEMS	24	0.158	IOWA STATE UNIVERSITY PRESS	21	0.157
500	TELEVISION BROADCASTING	21	0.173	INSTITUTE OF CONTINUING LEGAL EDUCATION	16	0.150
1000	ADVENTURE AND ADVENTURES	11	0.181	AGRICULTURAL RESEARCH SERVICE, U.S....	6	0.112
2000	JUDO	6	0.197			
3000	GYMNASTICS FOR WOMEN	4	0.197			
4000	MEDICAL STATISTICS	3	0.197			
5000	CULVERTS	2	0.166			

Table IV. The Most Frequent of 1,058,359 Title Terms on the *Chemical Titles* Tapes for 1965

A	10252	OF	107687	AND	37578	WITH	7964	OXIDE	3564	EFFECT	6015	EFFECTS	2081	REACTION	2991
2	5779	IN	32868	THE	36318	ACID	6098	AMINO	2919	METHYL	4030	SPECTRA	1827	CHLORIDE	2439
1	4379	ON	10984	FOR	6509	FROM	5564	ACIDS	2697	CARBON	2542	BETWEEN	1782	MAGNETIC	2360
3	4135	BY	10727	TRI	3072	POLY	3294	AMINE	2081	METHOD	2299	STUDIES	1519	ACTIVITY	1973
4	3219	DI	8419	ITS	1527	SOME	2498	CYCLO	1856	PHENYL	2187	HYDROXY	1481	HYDROGEN	1970
N	2465	TO	4544	RAT	1514	IRON	1583	WATER	1619	SYSTEM	1847	NUCLEAR	1410	ELECTRON	1913
5	2236	AT	2236	GAS	1451	HIGH	1571	STUDY	1554	SODIUM	1813	THERMAL	1302	ANALYSIS	1901
B	2085	AN	1890	ION	1369	THIO	1297	ETHYL	1427	DURING	1656	ORGANIC	1213	CHEMICAL	1584
6	1679	AS	1763	OXY	1229	IONS	1136	METAL	1366	CHLORO	1653	CRYSTAL	1125	CRYSTALS	1300
X	1142	II	1165	PER	1210	MONO	1059	ALKYL	1343	OXYGEN	1562	SYSTEMS	1113	KINETICS	1215
D	1108	DE	858	ISO	1190	ANTI	1047	TETRA	1338	LIQUID	1407	PROTEIN	1071	ALUMINUM	1173
P	1060	CO	743	LOW	987	ZINC	910	THEIR	1320	COPPER	1310	SILICON	979	NITROGEN	1171
L	1015	14	656	USE	985	FREE	909	GAMMA	1146	FLUORO	1201	CALCIUM	975	INFRARED	1162
S	948	OR	519	III	917	RATS	885	PHASE	1118	ENERGY	1182	AQUEOUS	935	EXCHANGE	1135
C	830	12	440	RAY	831	SPIN	827	LIVER	1079	NICKEL	1151	INDUCED	928	ETHYLENE	1055
		10	433	RNA	827	THIN	779	PHOTO	988	ACTION	1076	BENZENE	911	SOLUTION	1055
	42332		185932		97514		37421		27415		27415		20652		26397

Kučera and Francis is the most general with respect to subject matter, although the titles on the MARC tapes also cover a broad range of subjects. The *Chemical Titles* subject matter is clearly more restrictive, while the *Gas Chromatography* titles relate to a still narrower area of specialization. For all four data bases relation 2 may be used with A = 0.1 to predict most of the frequencies of occurrence of the first 5,000 most frequent terms to within a factor of 2.

In addition to including titles, the *Gas Chromatography* tapes include three fields of subject descriptors that refer to chemical compounds, substances, and general descriptors. For the sample investigated the values of rp_r are as listed in Table II. Although the subject descriptors are different from title words and do not combine into English sentences, the values of rp_r may be predicted with about the same accuracy as for title words.

The MARC tapes also contain subject headings and publisher names. While it might be expected that their distributions would differ significantly from those of title words, it may be observed that for subject headings the value of rp_r remains between 0.1 and 0.2 as r ranges between 100 and 5000. Similarly, for publisher names the value of rp_r remains between 0.08 and 0.16 as r ranges between 10 and 1000.

As indicated by Booth,[10] one of the consequences of the Zipf law is that it implies that 50% of the different terms occur only once in the entire data base, 16% occur only twice, and 8% occur only three times.

The 127 terms listed in Table IV are the most frequent terms of given lengths present as title terms on the *Chemical Titles* tapes for 1965. They account for 44% of all occurrences of title terms. The 15 words that are underlined account for 26% of all title term occurrences. A similar

Table V. The Most Frequent of 100,000 Title Terms on a Sample of the CAIN Tapes

A	1331	OF	8680	THE	4965	WITH	562	PLANT	277	EFFECT	437	CONTROL	319	RESEARCH	152
L	250	IN	3914	AND	3693	FROM	506	WHEAT	203	CATTLE	307	STUDIES	299	INDUSTRY	142
I	168	ON	1790	FOR	979	SOME	326	STUDY	182	GROWTH	254	SPECIES	223	NITROGEN	132
S	116	TO	805	NEW	412	SOIL	277	WATER	178	PLANTS	226	EFFECTS	222	ANALYSIS	128
2	105	BY	537	ITS	166	MILK	209	DAIRY	164	DURING	144	QUALITY	142	BREEDING	124
U	66	AS	255	USE	159	ACID	201	SHEEP	159	FOREST	133	BETWEEN	124	DISEASES	119
N	65	AN	234	TWO	124	FOOD	186	SOILS	157	METHOD	126	DISEASE	120	CHEMICAL	108
1	60	AT	166	OIL	79	SEED	136	VIRUS	139	INSECT	109	CONTENT	111	ACTIVITY	106
3	58	II	144	III	70	FARM	123	YIELD	130	ANIMAL	105	METHODS	109	PROBLEMS	98
4	57	IS	59	HOW	48	1969	101	FRUIT	126	REPORT	93	PROTEIN	101	PRODUCTS	96
R	53	OR	58	DOG	47	RICE	98	THEIR	111	COTTON	91	FEEDING	97	TAXONOMY	95
D	50	IV	34	LOW	47	FEED	95	GRAIN	102	BARLEY	79	FACTORS	93	RELATION	89
C	41	BE	29	NON	40	CORN	91	SWINE	99	ANNUAL	70	STORAGE	86	ECONOMIC	84
X	35	SP	25	RED	40	1968	89	UNDER	94	CITRUS	64	CHANGES	83	MOISTURE	61
P	30	IT	21	DRY	32	LEAF	80	FIELD	90	MARKET	63	POULTRY	80	RESPONSE	57
F	29	NO	19	EGG	32	HIGH	77	CROPS	78	LEAVES	61	TOBACCO	78	BEHAVIOR	42
	2514		16770		10933		3157		2289		2362		2287		1633

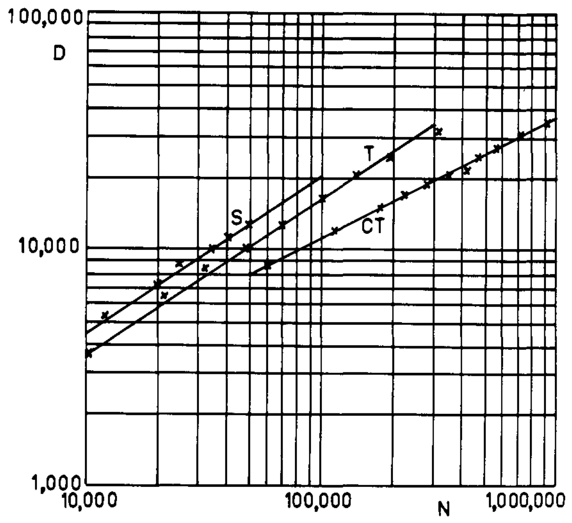

Figure 4. Logarithmic relation between the number D of different terms and the total number N of terms for *Chemical Titles* (CT), and for MARC titles (T) and subject headings (S).

table for the CAIN tapes issued by the U. S. Department of Agriculture is shown in Table V. The 127 most frequent title terms are different from those of Table IV, but they still account for 42% of all title term occurrences. The 15 words that are underlined account for 29% of all occurrences.

The statistics of the occurrences of data base terms, as illustrated in Tables I–V and predicted approximately by the Zipf law, indicate that within the inverted index there will be considerable variation in the number of entries required for the different terms. The inverted file usually resides on random access storage. In order to allow fast response to questions, it is desirable that the full set of entries for any term should be located and transferred to core by means of a single disk access. It is also desirable that the sets of inverted index entries be stored compactly with a minimum of unused storage. Discussion of the extent to which the two requirements of rapid access and compact storage may be satisfied has been given by Lowe.[11] In the design of the inverted index structure, and allocation of the required space, it is clearly of value to make predictions based on observed frequencies and use of approximation 2.

A further characteristic of many document data bases is their similarity with regard to growth of vocabulary. As data bases extend by addition of further document items, there are many fields in which the vocabulary grows in the following manner. If the data base contains a total of N terms associated with a certain attribute, such as title, abstract, or keywords, then as N increases the number D of different terms increases according to the formula

$$D = kN^{\beta} \tag{5}$$

where k and β are constants whose values depend on the particular data base. Thus vocabulary grows logarithmically as a function of data base size. This is illustrated in Figure 4 for title words of the *Chemical Titles* tapes and for both title words and subject headings of the MARC tapes. Relation 5 is also known to be valid for general English text of up to 20,000 words.[12]

It should be emphasized that, although the Zipf law for term frequencies and the logarithmic law for vocabulary growth are purely empirical, they have been found to hold approximately for the vocabularies in different types of fields within many document data bases. They are therefore extremely useful for prediction of data base characteristics and for estimating the efficiencies of various schemes for compression and coding of data bases.

COMPRESSION CODING OF WORDS AND SEARCH TERMS

In the file structure shown in Figure 3 the representation of terms in the sequential file is by means of a code that constitutes a pointer to the position at which the uncoded term is located in the term dictionary. If there are a total of D different terms, then each code may be of length $\log_2 D$ bits. However, the use of codes of fixed length does not produce the most compressed form of sequential file.

The most compact way to code terms that occur with known probabilities is through use of Huffman codes[13] as proposed by Schwartz[14] and others.[15,16] The average code length, in bits, of such codes is given by

$$H = -\sum_{r=1}^{D} p_r \log_2 p_r \tag{6}$$

which for the Zipf law becomes

$$H = -\sum_{r=1}^{D} (A/r) \log_2 (A/r) = A(\log_e D)^2/(2 \log_e 2) - \log_2 A \tag{7}$$

For vocabularies of 10,000, 100,000, and 1,000,000 different terms, the values of H are respectively 9.5, 11.4, and 13.4 bits. The average length of a title word on the *Chemical Titles* tapes is about 6.6 characters including a single delimiter such as a blank. Thus with an eight-bit character representation, the uncompressed words have an average length of 53 bits. No delimiters are required with Huffman codes, and so for a 100,000-word vocabulary the Huffman code produces a compression ratio of 11.4/53 = 21%. For subject headings or publisher names the compression is several times better.

In practice a Huffman coding scheme is not practical for coding of text because of the problems of coding and decoding. However, the Huffman coding scheme is useful in that it provides a standard against which to compare any other coding scheme.

A coding scheme that has been used for compression of text in document data bases has been described in previous papers[17,18] and analyzed in some detail.[19] The coding scheme may be summarized as follows. Some 127 frequent terms are represented by an eight-bit code whose left-hand bit is always equal to one. However, the code 10000000 is excluded so that it may be used for a different purpose as described later. The 127 frequent terms are chosen to comprise the 15 most frequent one-letter terms or symbols, the 15 most frequent two-letter terms, and so forth to include the 15 most frequent 8-letter terms. It might be noted that for a data base whose average word length is 6.5 eight-bit characters, and whose most frequent 127 terms are distributed as in Tables I and II, coding of only the 127 most frequent terms, while leaving the remaining words uncoded, results in a compression ratio of 63%. This is far above the 2 obtainable by use of the Huffman code, but it results from use of a very simple coding scheme. The code 10000000 could be used as a flag instead of a blank to precede uncoded terms.

After assignment of 8-bit codes to the 127 most frequent terms, the next most frequent 16,384 terms may be assigned 16-bit codes in which the first bit is always one and the ninth bit is always zero. Similarly, the next most frequent 2,097,152 terms may be given 24-bit codes whose first, ninth, and seventeenth bits are respectively 0, 0, and 1. Such a set of codes may be said to be of restricted variable length.

For a vocabulary of 100,000 different words, the restricted variable length coding scheme gives a compression ratio of 25% which compares quite favorably with the optimum of 20% obtainable through use of Huffman codes. In a previous paper[6] it was shown that this type of coding applied to titles in a library file of one million titles allows the monthly file storage cost of one of the files to be reduced from $4440 to $950.

It may be noted that assignment of codes, and storage of terms in the term dictionary, may be arranged automatically. The term dictionary may be created as a sequence of term strings, or buckets, each of which contains a fixed number, say 128, of terms of the same length. When assigning the 16-bit codes, each new term is stored in the first available bucket for terms of its length and is assigned the code *m, n* where *m* is an 8-bit bucket number and *n* is an 8-bit position number as shown in Figure 5. A similar structure is used for storage of terms that correspond to 8-bit and 24-bit codes. Analysis of the storage efficiency has been discussed in a previous paper.[19]

Compression coding of a data base in the manner described above was first tested by use with a search program for a data base of 80,000 titles. The program was subsequently redesigned as described in a previous paper[18] and is presently being tested experimentally by on-line searches of a sample of the INSPEC tapes in which there are 20 searchable fields.

It may be remarked that a number of methods of coding words through use of abbreviation and truncation codes have been described by various authors.[20-25] Such codes are often designed for use as search keys rather than for data base compression. The codes described in the present section have smaller average length and may be decoded uniquely to reconstruct the uncoded terms without ambiguity. Furthermore, since the codes are assigned sequentially in ascending numerical order, all possible codes of a given length may be used before the introduction of codes of greater length. Also, the codes may be assigned by purely automatic means.

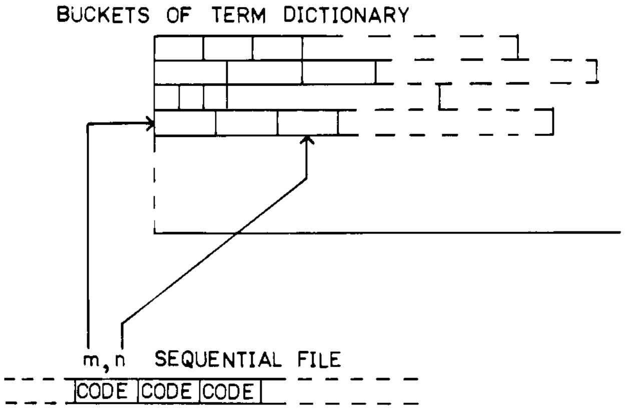

Figure 5. Relation between term codes and the position at which terms are stored in the buckets of a term dictionary.

Although the above coding scheme, and its many possible variations, allows good overall compression of data, it suffers from a number of disadvantages. One of the most serious arises from the fact that in many document data bases approximately 50% of all different terms occur only once. If the term dictionary and coded sequential file are regarded as together forming a compressed data base, then to code these infrequent terms produces negative compression since both the term and its code are stored. Furthermore, many terms that occur only once may never be used subsequently as question terms and so their storage enlarges the term dictionary without enhancing its usefulness. Yet, it is usually desirable to provide the capability to search on all terms regardless of how rarely they appear in the data base.

A further disadvantage is that as the data base grows the number of different terms increases logarithmically, and hence the term dictionary continues to grow. Unfortunately it grows by addition of terms that are likely to be less and less frequently used as search terms.

It is therefore of interest to consider the possibility of coding text by means of text fragments other than complete words or search terms. This is examined in the next section.

USE OF TEXT FRAGMENTS AS LANGUAGE ELEMENTS

In the discussion of the previous section, it was supposed that words constitute the codable elements from which text is formed. In the present section, consideration is given to the choice of language elements that consist of more general character strings that will be called text fragments. It will be supposed that the text within the document fields may be expressed in terms of the fragments which are then coded to derive a compressed data base.

The use of text fragments as the basic language elements of a data base used for purposes of information retrieval was proposed by Columbo and Rush.[26] The representation of names by compaction of variable length character strings called X-grams has been suggested by Walker.[27] Clare, *et al.*,[28] have proposed the representation of text by means of equifrequent text fragments which, in contrast to word fragments, may extend across word boundaries.

The advantage of using equifrequent text fragments is that each fragment then contains the same amount of information content in the sense of information theory as proposed by Shannon.[29,30] Therefore the fragments may be coded in the most compact manner by use of fixed length codes. Furthermore, the equifrequency property implies that the fragment oriented inverted file has the same number of entries for each item; as remarked by Salton[31] this is a desirable property, and it allows organization of the file into fixed length buckets, each of which contains all the document numbers of those documents that contain a particular fragment. A detailed discussion of the advantages of adopting fragments as codable language elements has been given by Schuegraf.[32]

Barton, *et al.*,[33] and Lynch, *et al.*,[34] have discussed the use of text fragments for title searches of the INSPEC tapes issued by the Institution of Electrical Engineers, London. Considerations affecting the choice of a suitable set of fragments are discussed below.

The inverted file indicates the occurrences of text fragments in the data base. If the average fragment length is less than the average word length, then the fragment-oriented inverted index will contain more entries than the word-oriented index. However, it will be supposed that the inverted file contains no entries that correspond to fragments of one letter only. Thus if the word CONTROVERSIAL were represented by the six fragments

CON/T/RO/V/ERS/IAL

there would be inverted file entries for CON, RO, ERS, and IAL, but not for T or V.

It should also be postulated that the set of chosen text fragments be complete in the sense that the text of any possible document item, whether or not already present in the data base, shall be representable through concatenation of members of the selected set of fragments. This latter criterion implies a fixed size of dictionary of text fragments regardless of the growth of the data base. This is an extremely attractive feature made possible by use of text fragments since it means that the dictionary may be structured without consideration of the need to update. Also, consideration may be given to the possibility of storing the entire dictionary in core without the danger of having to allocate more storage as the data base grows.

A method for selection of equifrequent fragments has been described by Schuegraf and Heaps.[35] It leads to a fragment dictionary that is a function of a certain threshold frequency that describes the desired frequency of the equifrequent fragments: the larger the threshold the smaller the number of fragments. The method is based on formulation of the problem of fragment selection as follows. For a given threshold frequency it is required to determine a set of fragments that:

i. By concatenation represent the entire data base.
ii. Are close to being equifrequent.
iii. Have maxim average fragment length.
iv. Are such that few words of the uncoded data base do not contain at least one fragment of more than one letter.
v. Are not redundant.

Condition iii is desirable in order that the coded data base exhibit good compression. Condition iv is imposed for the same reason and also so that few words of the data base do not contain at least one fragment for which there are inverted index entries. Condition v is clearly desirable in order to reduce the size of the fragment dictionary.

Determination of a set of fragments that approximately satisfy above conditions i–v has been illustrated by application to the author names, titles, and subject headings, on a single issue of the MARC tapes.[35] The uncompressed data base contained 51,047 characters, the dictionary of 2891 different words required 16,500 characters of storage, and the inverted index required storage for 68,000 bits.

The number of selected word fragments and the space required for storage of the resulting fragment dictionary, compressed data base, and inverted index is shown in Table VI. The similar quantities, when words are coded as language elements, are included in parentheses. It is apparent that coding of word fragments instead of words could allow a spectacular reduction in the size of the dictionary but would lead to less efficient compression and an increase in the size of the inverted index.

Table VI. Space Required for Storage of Word Fragments, Compressed Data Base, and Inverted Index[a]

Threshold	No. of dif fragments	Dictionary (8-bit chars)	Compressed data base (8-bit chars)	Inv index (bits)
5	1,387	5,500	14,000	97,000
10	824	2,900	14,900	110,000
15	601	1,900	15,000	120,000
20	490	1,500	15,400	126,000
	(2,891)	(16,500)	(8,300)	(68,000)

[a] Figures in parentheses are those in which words are the coded elements.

The total space required for storage of the data base, dictionary, and inverted index is equal to 76,000 characters if words are not coded, to 33,000 characters if words are coded, and to approximately 32,000 characters if word fragments are coded.

The sets of fragments referred to in Table VI were chosen to be approximately equifrequent, and it was assumed that the data base was coded in a manner that preserves the equifrequency property. While such a coding is possible, it may not be easy to achieve through use of an algorithm that is economical with respect to use of computer time. Some simple algorithms for coding a data base in terms of a predetermined set of equifrequent fragments have been described by Schuegraf and Heaps.[36] It was found that a consequence of using different algorithms for fragment selection and data base coding is that the resulting compressed data base may become double the size predicted in Table VI. Determination of an economical algorithm, suitable for both fragment selection and data base coding, thus remains a problem of considerable practical importance.

CONCLUSIONS

The storage requirements for large document data bases may be reduced significantly if title words, subject headings, and similar vocabulary terms are stored in coded form. The codes may be chosen compactly in a manner that allows unique decoding and is not an abbreviation with consequent loss in information. A reduction in dictionary size may be achieved through choice of word fragments rather than words, although use of equifrequent fragments as described in the present paper leads to less efficient compression and an increase in the size of the inverted file.

Both word and fragment coding are feasible for use in practical information retrieval systems. It is believed, however, that the studies to date also suggest the value of further work with a view to developing a coding scheme that would combine some of the advantages of word and fragment coding.

LITERATURE CITED

(1) Ein-Dor, P., "The Comparative Efficiency of Two Dictionary Structures for Document Retrieval," *INFOR J.*, **12**, 87–111 (1974).

(2) Cardenas, A. F., "Evaluation and Selection of File Organizations. A Model and System," *C. ACM*, **16**, 540–548 (1973).

(3) Collmeyer, A. J., and Shemer, J. E., "Analysis of Retrieval Performance for Selected File Organization Techniques," *Proc. Fall Joint Comput. Conf.*, 201–210 (1970).

(4) Hsiao, D., and Harary, F., "A Formal System for Information Retrieval

from Files," *C. ACM*, **13,** 67-73 (1970).

(5) Lefkovitz, D., "File Structures for On-Line Systems," Spartan Books, New York, N. Y., 1969.

(6) Dimsdale, J. J., and Heaps, H. S., "File Structure for an On-Line Catalog of One Million Titles," *J. Libr. Autom.*, **6,** 37-55 (1973).

(7) Benbow, J. A., "Design of an On-Line UDC Library Automation System," Thesis, University of Alberta, Edmonton, Canada, 1974.

(8) Zipf, G. K., "Human Behaviour and the Principle of Least Effort," Addison Wesley, Cambridge, Mass., 1949.

(9) Kučera, H., and Francis, W. N., "Computational Analysis of Present-day American English," Brown University Press, Providence, R. I., 1967.

(10) Booth, A. D., "A "Law" of Occurrences for Words of Low Frequency," *Inform. Control*, **10,** 386-393 (1967).

(11) Lowe, T. C., "The Influence of Data Base Characteristics and Usage on Direct Access File Organizations," *J. ACM*, **15,** 535-548 (1968).

(12) Herdan, G., "The Advanced Theory of Language as Choice and Chance," Springer-Verlag, New York, N. Y., 1966.

(13) Huffman, D. A., "A Method for the Construction of Minimum Redundancy Codes," *Proc. IRE*, **40,** 1098-1101 (1952).

(14) Schwartz, E. S., "A Dictionary for Minimum Redundancy Coding," *J. ACM*, **10,** 413-439 (1963).

(15) Ruth, S. R., and Dreutzer, P. J., "Data Compression for Large Business Files," *Datamation*, **18,** 62-66 (Sept 1972).

(16) Wells, M., "File Compression Using Variable Length Encodings," *Comput. J.*, **15,** 308-313 (1972).

(17) Heaps, H. S., and Thiel, L. H., "Optimization Procedures for Economic Information Retrieval," *Inform. Storage Retr.*, **6,** 137-153 (1970).

(18) Thiel, L. H., and Heaps, H. S., "Program Design for Retrospective Searches on Large Data Bases," *Inform. Storage Retr.*, **8,** 1-20 (1972).

(19) Heaps, H. S., "Storage Analysis of a Compression Coding for Document Data Bases," *INFOR J.*, **10,** 47-61 (1972).

(20) Bourne, C. P., and Ford, D. F., "A Study of Methods for Systematically Abbreviating English Words and Names," *J. ACM*, **8,** 538-552 (1961).

(21) Ruecking, F. H., "Bibliographic Retrieval from Bibliographic Input; the Hypothesis and Construction of a Test," *J. Libr. Autom.*, **1,** 227-238 (1968).

(22) Libetz, B. A., Stangl, P., and Taylor, K. F., "Performance of Ruecking's Word-Compression Method when Applied to Machine Retrieval from a Library Catalog," *J. Libr. Autom.*, **2,** 266-271 (1969).

(23) Nugent, W. R., "Compression Word Coding Techniques for Information Retrieval," *J. Libr. Autom.*, **1,** 250-260 (1968).

(24) Dolby, J. L., "An Algorithm for Variable-Length Proper-Name Compression," *J. Libr. Autom.*, **3,** 257-275 (1970).

(25) Treveaven, R. L., "Abbreviation of English Words to Standard Length for Computer Processing," Thesis, University of Alberta, Edmonton, Canada, 1970.

(26) Columbo, D. S., and Rush, J. E., "Use of Word Fragments in Computer Based Retrieval Systems," *J. Chem. Doc.*, **9,** 47-50 (1969).

(27) Walker, V. R., "Compaction of Names by X-grams," *Proc. Amer. Soc. Inform. Sci.*, **6,** 129-135 (1969).

(28) Clare, A. C., Cook, E. M., and Lynch, M. F., "The Identification of Variable Length Equifrequent Character Strings in a Natural Language Data Base," *Comput. J.* 15, 259-262 (1972).

(29) Shannon, C. E., "A Mathematical Theory of Communication," *Bell Syst. Tech. J.*, **27,** 379-423, 623-656 (1948).

(30) Shannon, C. E., "Prediction and Entropy of Printed English," *Bell Syst. Tech. J.*, **30,** 50-64 (1951).

(31) Salton, G. A., "Computer Evaluation of Indexing and Text Processing," *J. ACM*, **15,** 8-36 (1968).

(32) Schuegraf, E. J., "The Use of Equifrequent Fragments in Retrospective Retrieval Systems," Ph.D. Thesis, University of Alberta, Edmonton, Canada, 1974.

(33) Barton, I. J., Creasey, S. E., Lynch, M. F., and Snell, M. J., "An Information-Theoretic Approach to Text Searching in Direct Access Systems," *C. ACM*, **17,** 345-350 (1974).

(34) Lynch, M. F., Petrie, J. H., and Snell, M. J., "Analysis of the Microstructure of Titles in the INSPEC Data-Base," *Inform. Storage Retr.*, **9,** 331-337 (1973).

(35) Schuegraf, E. J., and Heaps, H. S., "Selection of Equifrequent Fragments for Information Retrieval," *Inform. Storage Retr.*, **9,** 697-711 (1973).

(36) Schuegraf, E. J., and Heaps, H. S., "A Comparison of Algorithms for Data Base Compression by Use of Fragments as Language Elements," *Inform. Storage Retr.*, in press.