Application of the Variety-Generator Approach to Searches of Personal Names in Bibliographic Data Bases—Part 2. Optimization of Key-Sets, and Evaluation of Their Retrieval Efficiency

Dirk W. FOKKER and Michael F. LYNCH: Postgraduate School of Librarianship and Information Science, University of Sheffield, England.

Keys consisting of variable-length character strings from the front and rear of surnames, derived by analysis of author names in a particular data base, are used to provide approximate representations of author names. When combined in appropriate ratios, and used together with keys for each of the first two initials of personal names, they provide a high degree of discrimination in search.

Methods for optimization of key-sets are described, and the performance of key-sets varying in size between 150 and 300 is determined at file sizes of up to 50,000 name entries. The effects of varying the proportions of the queries present in the file are also examined. The results obtained with fixed-length keys are compared with those for variable-length keys, showing the latter to be greatly superior.

Implications of the work for a variety of types of information systems are discussed.

INTRODUCTION

In Part 1 of this series the development of variety generators, or sets of variable-length keys with high relative entropies of occurrence, from the initial and terminal character strings of authors' surnames was described.[1] Their purpose, used singly or in combination, is to provide a high and constant degree of discrimination among personal names so as to facilitate searches for them. In this paper the selection of optimal combinations of the keys and evaluation of their efficiency in search are described. The performance of combined key-sets of various compositions is determined at a range of file sizes and compared with fixed-length keys. In addition,

the extent of statistical associations among keys from different positions in the names is determined.

BALANCING OF KEY-SETS

The relative entropies of distribution of the first and last letters of the surnames of authors in the file of 100,000 entries from the INSPEC data base differ significantly, the former being 0.92 and the latter 0.86. As a result, a larger key-set has to be produced from the back of the surnames to reach the same value of the relative entropy as that of a key-set of given size from the front of the surname. For instance, the value of 0.954 is reached by a key-set comprising 41 keys from the front of the name, but a set of 101 keys from the back is needed to attain this value. It seemed reasonable to assume that keys from the front and rear should be combined in different proportions in order to maximize the relative entropy of the combined system, and that their proportions should reflect the redundancies of each distribution (redundancy = $1 - H_r$). In order to test this, a series of combined key-sets of different total sizes was produced, in which the proportions of keys were varied around the ratio of the redundancies of the first and last character positions, i.e., $(1 - 0.92):(1 - 0.86)$, or 8:14. The relative entropies of the name representations provided by combining these key-sets with keys for the first and second initials were determined by applying them to the 50,000 name file, and the entropy value used to determine the optimal ratio of keys. In one case, the correlation between the value of the relative entropy and retrieval efficiency, as measured by the precision ratio, was also studied, and shown to be high.

The sizes of the combined key-sets studied were 148 and 296, with an intermediate set of 254 keys. The values of 148 and 296 were chosen in view of the projected implementation in the serial-parallel file organization.[2] This relates the size of the key-set to the number of blocks on one cylinder of a disc. (The 30Mbyte disc cartridges available to us have 296 blocks per cylinder.) Otherwise the choice of key-set is arbitrary, and can be varied at will.

The minimum key-set size is 106, consisting of 26 letters each for the first and last letter of the surname, and 27 (26 letters and the space symbol) each for the first and second initials. The numbers of n-gram keys ($n \geqq 2$) required for the key-sets numbering 148, 254, and 296 in size are thus 42, 148, and 190. Full details are given of the composition of the first and third of these sets.

A slight refinement to key-set generation was employed to ensure as close an approximation to equifrequency as possible, especially with the smallest key-sets. Precise application of a threshold frequency may occasionally result in arbitrary inclusion of either very high or very low frequency keys. Thus, if almost all the occurrences of a longer key are accounted for by a shorter key (as with -MANN and -ANN), only the shorter n-gram is included.

OPTIMAL SET OF 148 KEYS

The number of n-gram keys ($n \geqq 2$) to be added to the minimum set of 106 keys is 42, the presumed optimum proportion being 8:14, which implies about 16 keys from the front of the name and 26 from the back. In order to examine the relationship between the ratio of keys from the front and rear of the surname and the relative entropy of the combined sets, the ratios were varied at intervals between 1:1 and 1:3 so that the numbers of n-grams varied from 21 and 21 to 11 and 31 respectively. For each ratio the keys were applied to the 50,000 name entries, and the distribution of the resultant descriptions determined. The ratios, the number of n-gram keys, and the relative entropies of the distributions are shown in Table 1. The maximum value of the entropy is taken to be $\log_2 50{,}000$. In this case the balancing point, with the key-set including 16 n-gram keys

Table 1. Relation between Ratio of n-grams from Front and Rear of Surname, Entropy of Combined Key-Sets, and Retrieval Efficiency for a Series of Sets of 148 Keys

Ratio of n-gram Keys	*Number of n-gram Keys*		*Number of Different Representations in 50,000 Entries*	*Relative Entropy of System*	*Precision (%) (File Size = 25,000)*
	Front	*Back*			
1:1	21	21	33,485	0.9450	71.5
3:4	18	24	33,501	0.9450	71.3
17:25	17	25	33,434	0.9447	70.9
8:13	16	26*	33,454	0.9453	72.2
5:9	15	27	33,402	0.9450	72.0
1:2	14	28	33,378	0.9449	72.1
1:3	11	31	33,126	0.9437	71.5

Total number of different name entries = 41,469.

* Key-set with highest relative entropy.

from the front and 26 from the back, corresponds with the ratio of the redundancies of the first and last letters of the surnames. Table 2 shows the composition of the optimal key-set of 148 keys, while Table 3 gives the distribution of the name representations compiled from the combined key-set, and its corresponding relative entropy.

OPTIMAL SET OF 296 KEYS

A similar procedure to that used for the optimal 148-key key-set was also applied in this instance. Here the ratios of front and rear n-gram keys varied from 57 and 133 to 69 and 121 respectively. For each of the sets chosen, the distributions of the entries resulting from application of the combined key-sets to the file of 50,000 names were determined. These showed virtually no difference in terms of the relative entropy alone, although the total number of different entries differed slightly between key-sets, and the highest value was used to choose the optimal set, detailed in Table 4. The range of combinations studied is shown in Table 5, and the distribution of the entries for the optimal set is given in Table 6.

Table 2. Composition of Balanced Key-Set of 148 Keys

Keys from front of surname (42):

Key	p_i	*Key*	p_i	*Key*	p_i	*Key*	p_i
A	.035	G	.055	MA	.030	SH	.016
B	.020	H	.035	N	.025	ST	.016
BA	.020	HA	.021	O	.017	T	.040
BE	.017	I	.013	P	.038	U	.005
BO	.014	J	.017	PA	.014	V	.025
BR	.014	K	.041	Q	.001	W	.040
C	.036	KA	.017	R	.032	X	—
CH	.016	KO	.017	RO	.017	Y	.011
D	.044	L	.033	S	.049	Z	.013
E	.018	LE	.014	SA	.016		
F	.034	M	.050	SC	.015		

Keys from rear of surname (52):

A	.060	II	.015	NN	.010	IS	.012
RA	.010	KI	.015	ON	.018	T	.042
VA	.015	J	.001	SON	.027	U	.013
B	.003	K	.033	O	.028	V	.001
C	.005	L	.013	KO	.013	EV	.018
D	.030	EL	.012	P	.004	OV	.026
E	.068	LL	.016	Q	.001	KOV	.012
F	.006	M	.013	R	.016	NOV	.011
G	.012	N	.009	ER	.064	W	.005
NG	.014	AN	.020	LER	.013	X	.003
H	.020	MAN	.017	NER	.010	Y	.031
CH	.017	EN	.025	S	.055	EY	.012
I	.044	IN	.039	ES	.015	Z	.013

Keys from first initial: 27 characters
Keys from second initial: 27 characters

Table 3. Frequencies of Entries Represented by Optimal 148-Key Key-Set in a File of 50,000 Names

Frequency f	*Number of Entries with Frequency f*
1	24,363
2	5,622
3	1,850
4	757
5	372
6	193
7	103
8	68
9	32
10	24
11–15	54
16–20	11
21–30	4
33	1

Total number of different entries = 33,454

Maximum number of possible combinations = 1,592,136 (i.e., $42 \times 52 \times 27^2$)

$H = 14.7553$ $H_{max} = 15.6096(\log_2 50{,}000)$ $H_r = 0.9453$

Table 4. Composition of Balanced Key-Set of 296 Keys

Keys from front of surname (87):

A	BU	E	HA	KI	MA	NI	RA	SI	WA
AL	C	F	HE	KO	MAR	O	RE	SO	WE
AN	CA	FR	HO	KR	MC	P	RI	ST	WI
B	CH	G	HU	KU	ME	PA	RO	T	X
BA	CO	GA	I	L	MI	PE	S	TA	Y
BAR	D	GO	J	LA	MO	PO	SA	U	Z
BE	DA	GR	JO	LE	MU	PR	SC	V	
BO	DE	GU	K	LI	N	Q	SE	VA	
BR	DO	H	KA	M	NA	R	SH	W	

Keys from rear of surname (155):

A	LD	NG	VSKII	EL	LIN	R	OR	NT	SOV
CA	ND	ANG	KI	LL	TIN	AR	S	RT	W
DA	RD	ING	SKI	ALL	NN	ER	AS	ERT	X
KA	E	RG	WSKI	ELL	ON	BER	ES	ST	Y
MA	DE	H	LI	M	SON	DER	NES	TT	AY
NA	EE	CH	NI	AM	LSON	GER	IS	ETT	EY
INA	GE	ICH	RI	N	NSON	NGER	NS	U	LEY
RA	KE	VICH	TI	AN	RSON	HER	INS	V	KY
TA	LE	GH	J	MAN	TON	IER	OS	EV	RY
VA	NE	SH	K	RMAN	O	KER	RS	OV	Z
OVA	RE	TH	AK	YAN	KO	LER	SS	KOV	TZ
WA	SE	ITH	CK	EN	NKO	LLER	TS	IKOV	
YA	TE	I	EK	SEN	NO	MER	US	LOV	
B	F	AI	IK	IN	TO	NER	T	NOV	
C	FF	HI	L	EIN	P	SER	DT	ANOV	
D	G	II	AL	KIN	Q	TER	ET	ROV	

Keys from first initial: 27 characters
Keys from second initial: 27 characters

Table 5. Relation between Ratio of n-grams from Front and Rear of Surname and Entropy of Combined Key-Sets for a Series of Sets of 296 Keys (File Size = 50,000)

Ratio of n-gram Keys	*Number of n-gram Keys*		*Number of Different Representations*	*Relative Entropy of System*
	Front	*Back*		
3:7	57	133	39,182	0.9679
61:129	61	129*	39,191	0.9679
13:25	65	125	39,186	0.9679
69:121	69	121	39,179	0.9679

* Key-set with highest number of different entries.

In this instance, the ratio of *n*-gram keys from the front and back of the surnames has been displaced from the ratio of the redundancies of the first and last characters of the surnames, i.e., 8:14 (1:1.7). Here the ratio is roughly 1:2. This is undoubtedly due to the fact that the relative entropies of key-sets from the back of the surname increase less rapidly than those of key-sets from the front, and hence larger sets must be employed.

EVALUATION OF RETRIEVAL EFFECTIVENESS

The keys in the optimized key-sets represent name entries in an approxi-

Table 6. Frequencies of Entries Represented by Optimal Key-Set of 296 Keys in a File of 50,000 Names

Frequency f	*Number of Entries with Frequency f*
1	31,705
2	5,394
3	1,371
4	442
5	164
6	63
7	27
8	12
9	4
10	3
11	2
12	2
13	—
14	—
15	1
16	1

Total number of different entries = 39,191

Maximum number of possible combinations = 9,830,565 (i.e., $87 \times 155 \times 27^2$)

$H = 15.108$ $H_{max} = 15.6096(\log_2 50{,}000)$ $H_r = 0.9679$

mate manner only, so that when a search for a name is performed, additional entries represented by the same combination of keys are identified. While these may be eliminated in a subsequent character-by-character match of the candidate hits, the proportion of unwanted items should remain low if the method is to offer advantages.

In evaluating the effectiveness of the key-sets in the retrieval, the names in the search file were represented by concatenating the codes for the keys from the front and back of the surnames and the initials, and subjecting the query names to the same procedure. The matching procedure produced lists of candidate entries, of which the desired entries were a subset. The final determination was carried out manually.

The tests were performed first with names sampled from the search file, so that correct items were retrieved for each query. Since searches for name entries may be performed with varying probabilities that the authors' names are present in the file (especially in current-awareness searches), varying proportions of names of the same provenance, but known not to be present in the search file, were also added. In these cases candidate items were selected which included none of the desired entries. Recall tests were also performed and recall shown to be complete.

The measure used in determining the performance of the variety-generator search method is the precision ratio, defined as the ratio of correctly identified names to all names retrieved. It is presented both as the ratio of averages (i.e., the summation of items retrieved in the search and calculation of the average) and as the average of ratios (i.e., averaging the

figures for individual searches). The latter gives higher figures, since many of the individual searches give 100 percent precision ratios.

The precision ratio was found to be dependent on file size and to fall somewhat as the size of file increases. This is due to the fact that the key-sets provided only a limited, if very high, total number of possible combinations, while the total possible variety of personal names is virtually unlimited.

The evaluation was performed with a sample of 700 names, selected by interval sampling. This number ensured a 99 percent confidence limit in the results. A comparison of the interval sampled query names with randomly sampled names showed that no bias was introduced by interval sampling.

A test to confirm that the retrieval effectiveness reached a peak at the maximum value of the relative entropy of a balanced key-set was performed first. This was carried out on a file of 25,000 names, using as queries names selected from the file and the optimal 148-key key-set. As shown in Table 1, the values of the precision ratio (ratio of averages) and of the relative entropy both peak at the same ratio of *n*-gram keys from the front and back of the surnames.

The performance of the optimal key-sets of 148, 254, and 296 keys with files of 10,000, 25,000, and 50,000 names is shown in Table 7. Calculated as the ratio of averages, the smallest key-set (148 keys) shows a precision ratio of 64 percent with a file of 50,000 names, which means that of every three names identified in the variety-generator search, two are those desired. With the largest key-set (296 keys), this rises to nine correctly identified names in every ten retrieved at this stage. On the other hand, calculated as the average of ratios, the precision ratios rise to 81 percent and 94 percent respectively. For smaller file sizes—typical, for instance, of current-awareness searches—the figures for all of these are correspondingly higher.

Table 7. Precision Ratios Obtained in Variety-Generator Searches of Personal Names—Queries Sampled from Search File (Confidence Level = 99 Percent)

Precision as ratio of averages (%):

File Size	*Key-Set Size*		
	148	*254*	*296*
50,000	64	87	90
25,000	71	90	91
10,000	84	93	94

Precision as average of ratios (%):

File Size	*Key-Set Size*		
	148	*254*	*296*
50,000	81	91	94
25,000	87	95	96
10,000	93	97	97

The effect of sampling from a larger file, so that increasing proportions of the names searched for are not present in the search file, is shown in Table 8 for a file of 25,000 names. In this case, the proportion of correctly identified names in the total falls, so that overall performance is somewhat reduced. Thus, depending both on file size and on the expected proportion of queries identifying hits, the key-set size can be adjusted to reach a desired level of performance. In addition, tests to determine the

Table 8. Effect of Varying Proportion of Query Names Not Present in Search File of 25,000 Names, Using 296 Keys (Ratio of Averages)

% of Names Not in Search File	*Precision % (Ratio of Averages)*	*Number of Names Retrieved*	*Number of Names Correctly Retrieved*
21	90	766	691
42	85	595	505
61	83	449	371
74	76	319	242
84	68	228	154

applicability of a key-set optimized for one file of 50,000 names to another file of the same provenance and size were carried out. The three key-sets derived from the first file were applied to the second, query names sampled from the latter, and the precision ratios determined. Some reduction in performance was observed; expressed as ratio of averages, the precision with the 296-key key-set fell from 90 to 83 percent, with the 254-key key-set from 87 to 82 percent, and with the 148-key key-set from 64 to 56 percent, figures which seem unlikely to prejudice the net performance in any marked way. Nonetheless, monitoring of performance and of data base name characteristics over a period of operation might well be advisable.

DISTRIBUTION CHARACTERISTICS OF OTHER TYPES OF KEYS

It is particularly instructive to examine the distribution characteristics of other types of keys, including those of fixed length, generated from various positions in the names, and to compare them with those of the optimal key-sets employed in the variety-generator approach. To this end, the file of 50,000 names was processed to produce the following keys or key-sets:

1. Initial digram of surname.
2. Initial trigram of surname.
3. Key-set of ninety-four *n*-grams from the front of the surname, with first and second initials.
4. Key-set consisting of first and last character of surname, with first and second initials.

The figures (Table 9) show clearly that all have distributions which leave no doubt as to their relative inadequacy in resolving power, where this is defined as the ratio of distinct name representations provided by the key-set used to the number of different name entries (41,469) in the file. At the digram level, the value of the resolving power is 0.009, i.e., each

digram represents, on average, 110 different name entries, while no fewer than thirty-two specific digrams each represent between 500 and 1,000 different names. At the trigram level, the value of the resolving power rises to 0.08, a tenfold increase; however, one trigram still represents between 500 and 1,000 different names.

Use of the first and last letters of the surname plus the initials again increases the value of the resolving power to 0.627, or 1.6 distinct names per entry; eight of the representations now account for between thirty-one

Table 9. Distributions of a Variety of Other Representations of Personal Names in a File of 50,000 Entries

Frequency f	*Initial Digram of Surname*	*Initial Trigram of Surname*	*94 n-grams from Front of Surname Plus 2 Initials*	*First and Last Letter of Surname Plus 2 Initials*
1	40	735	8,964	16,346
2	22	428	3,929	4,919
3	16	249	1,884	2,025
4	11	197	1,006	973
5	7	170	646	581
6	7	110	397	340
7	10	112	234	224
8	4	98	186	146
9	7	81	144	92
10	5	66	108	72
11	6	61	70	49
12	2	56	88	36
13	5	51	74	33
14	1	48	50	24
15	2	35	51	23
16	3	37	36	25
17	2	35	29	15
18	3	33	29	11
19	8	35	28	6
20	8	40	23	5
21–30	21	207	127	49
31–40	23	109	47	8
41–50	13	88	13	
51–100	36	142	3	
101–200	24	62		
201–500	57	15		
501–1000	32	1		
Total	375	3,301	18,166	26,002
Resolving power	.009	.080	.438	.627

and forty distinct entries. In contrast, however, the key-set of 148 keys comprising ninety-four *n*-gram keys from the front of the name and the first and second initials, although almost 50 percent larger than the four-character representation, has a resolving power of only 0.438 (or 2.28 entries per representation). This contrast provides particularly strong evidence for the superiority of keys from the front and rear of the surnames over those from the front alone, even when the latter are variable in

length. As expected, the precision ratio of the four-character representation is low, at 37 percent (ratio of averages), compared with 64 percent for the optimal 148-key key-set.

EXTENT OF STATISTICAL ASSOCIATION AMONG KEYS

Thus far, the frequency of occurrence of variable-length character strings from the front and back of the surnames is the only factor considered in their selection as keys. It is well known in other areas that statistical associations among keys can influence the effectiveness of their combinations.[3] Where a strong positive association between two keys exists, their intersection results in only a small reduction of the number of items retrieved over that obtained by using each independently. When the association is strongly negative, the result of intersection may be much greater than that predicted on the basis of the product of the individual probabilities of the keys.

To assess the extent of associations among keys from the front and rear of surnames and initials, sets of both fixed- and variable-length keys from each of these positions were examined. The Kendall correlation coefficient V was calculated for each of the twenty most frequent combinations of these. This is related to the chi-square value by the expression

$$X^2 = m V^2$$

where m is the file size, or 50,000. Table 10 shows the values of the association coefficient for certain of the characters in the full name. Those above .012 are significant at a 99 percent confidence level. Positive associations are

Table 10. Association Coefficients for Sets of the Most Frequent Digrams from Various Positions in Personal Names

First and Last Letters of Surname		*First Letter of Surname and First Initial*		*First and Second Initials*	
Digram	*V*	*Digram*	*V*	*Digram*	*V*
KV	.064	KV	.054	H▽	.078
WR	.050	HJ	.027	M▽	.069
KA	.038	BR	−.024	K▽	.069
HN	.028	SJ	−.023	R▽	−.055
SA	.024	DJ	.022	D▽	−.053
SN	.024	BG	.018	T▽	.053
CN	.022	KA	.018	J▽	−.045
KN	−.020	CJ	.018	S▽	.034
MA	.014	SD	.015	F▽	.033
KR	−.011	SV	.013	N▽	−.029
SV	.010	MM	.011	G▽	.022
RN	.010	MJ	.007	L▽	−.022
BN	−.008	BJ	.005	I▽	−.019
BR	.008	SG	−.004	A▽	−.019
MN	−.007	SR	.004	C▽	−.018
SR	.007	BA	.004	P▽	.017
MR	.004	MA	.004	W▽	−.014
SI	−.002	SM	−.003	Y▽	.010
GN	.001	MR	.002	B▽	.005
LN	.001	SA	−.000	E▽	−.002

more frequent than negative. The figures indicate that intersection of certain of these characters as keys in search would result in some slight diminution in performance against that expected.

The figures for the association coefficients among the twenty most frequent combinations of keys from the front and back of surnames in the 148- and 296-key key-sets show magnitudes (mostly positive) which are substantially greater than those for single characters (see Table 11). The reasons for these values are obvious; in certain instances, e.g., MILLER, JONES, and MARTIN, common complete names are apparent, while in one case, LEE, an overlap between keys from the front and rear exists. In others, linguistic variations on common names can be discerned, as with BR N—BROWN or BRAUN.

Table 11. Association Coefficients in the Twenty Most Frequent Key Combinations from Front and Back of Surnames in Two Key-Sets

Key-Set Size 148			*Key-Set Size 296*		
Keys		*V*	*Keys*		*V*
S	H	.146	S	ITH	.343
J	SON	.127	JO	NSON	.297
SC	ER	.104	JO	NES	.278
W	S	.043	AN	RSON	.274
T	A	.038	SI	GH	.249
T	I	.038	LE	EE	.221
W	ER	.038	MU	LLER	.214
C	E	.034	TA	OR	.195
F	ER	.033	GU	TA	.168
P	S	.025	BR	N	.160
D	E	.023	MI	LLER	.151
L	E	.022	MAR	TIN	.145
W	E	.022	WI	S	.137
G	IN	.020	F	HER	.133
M	E	.009	SC	DER	.121
S	A	.008	SA	TO	.110
G	E	.006	T	AS	.084
M	A	.005	SC	ER	.069
M	ER	−.004	CH	EN	.055
G	ER	−.000	T	SON	.050

Such associations are inevitable. When the selection of keys is based solely on frequency, some deviation from the ideal of independence must result, becoming larger as the size of the key-sets increases, and as the length of certain of the keys increases. However, since its effect in the most extreme cases is merely to lead to virtually exact definition of the most frequent surnames, no particular disadvantage results.

POSSIBLE IMPLEMENTATIONS OF THE VARIETY-GENERATOR NAME SEARCH APPROACH

The variety-generator approach permits a number of possible implementations of searches for personal names to be considered, if only in outline

at this stage, using a variety of file organization methods. The most widely known methods (apart from purely sequential files) are direct access (utilizing hash-addressing), chained, and index sequential files.

Direct application of the concatenated key-numbers as the basis for hash-address computation appears attractive in instances where the personal name is used alone or in combination (as, for instance, with a part of the document title). The almost random distribution of the bits in this code should result in a general diminution of the collision and overflow problems commonly encountered with fixed-length keys.

Since only four keys are used to represent each name, and the four sets of keys from which these are selected are limited in number and of approximately equal probability, the keys can be used to construct chained indexes, to which, however, the usual constraints still apply.

Index sequential storage again offers opportunities, in particular since the low variety of key types means that the sorting operations which this entails can be eliminated. In effect, each name entry would be represented by an entry in each of four lists of document numbers or addresses, and documents retrieved by intersection of the lists. While four such numbers are stored for each name, in contrast to a single entry for the more conventional name list, the removal of the name list itself would more than compensate for the additional storage required for the lists.

In the index sequential mode, the lists of document addresses or numbers stored with each key are more or less equally long. They may thus be replaced by bit-vectors in which the position of a bit corresponds to a name or document number. If the number of keys bears a simple relation to the number of blocks on a disc cylinder, the vectors can be stored in predetermined positions within a cylinder, resulting in the serial-parallel file.

The usefulness of this file organization has yet to be fully evaluated; however, it also promises substantial economies in storage. On average, only four of the bits are set at the positions in the vectors corresponding to the name or document entry. On average, then, the density of 1-bits is very low, and long runs of zeros occur in the vectors. They can, therefore, be compressed using run-length coding, for instance as applied by Bradley.[3, 4] Preliminary work with the 296-key key-set has indicated already that a gross compression ratio of nine to one is attainable, so that the explicit storage requirements to identify the association between a name and a document number would be just over thirty bits.

CONCLUSIONS

The work described here relates solely to searches for individual occurrences of personal names. Clearly, in operational systems in which one or more author names are associated with a particular bibliographical item, it will be necessary to provide for description of each of these for access. If this is provided solely on the basis of a document number, some false coordination will occur—for instance, when the initials of one entry are

combined with the surname of another. A number of strategies can be envisaged to overcome this problem.

The performance figures show clearly that a small number of characteristics—between 100 and 300 in this study—are sufficient to characterize the entries in large files of personal names and to provide a high degree of resolution in searches for them. While performance in much larger files, involving the extension of key-set sizes to larger numbers, has yet to be studied, the logical application of the concept of variety generation would appear to open the way to novel approaches to searches for documents associated with particular personal names, which seem likely to offer advantages in terms of the overall economic performance of search systems, not only in bibliographic but also in more general computer-based information systems.

ACKNOWLEDGMENTS

We thank M. D. Martin of the Institution of Electrical Engineers for provision of a part of the INSPEC data base and of file-handling software, and the Potchefstroom University for C.H.E. (South Africa) for awarding a National Grant to D. Fokker to pursue this work. We also thank Dr. I. J. Barton and Dr. G. W. Adamson for valuable discussions, and the former for *n*-gram generation programs.

REFERENCES

1. D. W. Fokker and M. F. Lynch, "Application of the Variety-Generator Approach to Searches of Personal Names in Bibliographic Data Bases—Part 1. Microstructure of Personal Authors' Names," *Journal of Library Automation* 7:105–18 (June 1974).
2. I. J. Barton, S. E. Creasey, M. F. Lynch, and M. J. Snell, "An Information-Theoretic Approach to Text Searching in Direct-Access Systems," *Communications of the ACM* (in press).
3. S. D. Bradley, "Optimizing a Scheme for Run-Length Encoding," *Proceedings of the IEEE* 57:108–9 (1969).
4. M. F. Lynch, "Compression of Bibliographic Files Using an Adaptation of Run-Length Coding," *Information Storage and Retrieval* 9:207–14 (1973).