Automatic classification and indexing: a supplement

Hoyle, W. G.

Report (National Research Council of Canada. Radio and Electrical Engineering Division : ERB), 1968-11

ERB-793

UNCLASSIFIED

NATIONAL RESEARCH COUNCIL OF CANADA

RADIO AND ELECTRICAL ENGINEERING DIVISION

AUTOMATIC CLASSIFICATION AND INDEXING - A SUPPLEMENT

- W. G. HOYLE -

OTTAWA

NOVEMBER 1968

ABSTRACT

The occurrence of a word, one or more times, in a document is taken as an attribute of that document. Using a simple formula from Bayes probability, a probability is derived, based on that word, that the document belongs in a certain category. The procedure is applied to all the words of a document and the words are then ordered by probability to form a list. The procedure is also used to form category lists from existing categories although original categories could be formed. Document lists are compared to category lists and probability sums formed for indexing. Two sample category lists, derived from abstracts are given. Simple modifications show the ease of modifying list characteristics – two occurrences of a word, or occurrence in two documents being substituted for a single simple occurrence.

PROLOGUE

The substance of this report has had a rather varied history. On 15 May 1968 it was accepted by the 34th Conference of the International Conference for Documentation (F.I.D.) for presentation at the scheduled meeting of that body in Moscow, September 9–18, 1968. When the political situation caused that conference to be postponed, I was told by the Canadian F.I.D. authorities that I was then free to publish elsewhere. In October it was submitted to American Documentation for publication and subsequently withdrawn when a telegram, from the All Union Institute for Scientific and Technical Information (VINITI), announced that they were publishing the paper. Because of the need for interim copies of the paper, because certain additional work has been done (the word lists are not those in the original paper), and also because a report of the present nature can be far more complete (the original paper contained no program nor flow-chart information) it was decided to issue this supplementary report.

AUTOMATIC CLASSIFICATION AND INDEXING
A Supplement

– W.G. Hoyle –

We use the word automatic to mean done in a purely mechanical way, as by rule or rote, regardless of whether such work is actually done by people or machines. An automatic procedure for developing categories and assigning items to them offers a potentially large saving, hence our interest. Statistical methods without semantics or syntax seem most promising and our work is in that area. An excellent summary of statistical methods and a bibliography are given by Stevens [1], and more recent work is described by Borko [2]. The statistical method not only avoids arguments about meaning, but is probably independent of the language. In fact, the method need not employ a spoken language at all but could apply to any set of symbols, as for maps, etc.

In this area of text statistics, the work of Doyle [3] seemed particularly hopeful. He recognized that, despite its marvels, the computer could not do document–to–document comparisons (i.e., retrieval without classification) in large collections because of the cost factor in the square law inherent in such a procedure. He realized, also, that the classical idea of classification was still valid, though not necessarily the classical method. He generates 'profiles' or 'word lists', each representing a group of documents or a category. The idea of a word list representing a group of documents is fundamental to our paper but our lists are generated on totally different principles from Doyle's. The work of Trachenberg [4] and Williams [5] is close in character to that described here.

All work in the area of text statistics, as far as we are aware, depends in some way on word counting, with high counts being required for significance. The trouble is, of course, that many insignificant words also have counts of high frequency. (Such words may have significance in other contexts; see, for example, Wallace [1]). In practice, such words are eliminated by the exercise of human intelligence. Little intelligence is required when words like 'the' and 'and' are discarded, but as the list lengthens more and more intelligence is needed and eventually arguments and differences in judgment arise. For a good example, see Miller [7] et al., page 377. Eliminating such words as 'the' and 'and' mechanically; i.e., by rule, seemed at first a trivial problem. In practice, it offered surprising difficulty and only gradually did it dawn on us that it was fundamentally the same problem as preparing a list of significant words, only starting at the other end so to speak. Thus we backed into our problem. The preparation of an ordered list of significant words is the core of the problem of classification and of retrieval – at least by single terms. Word counting, then, has its limitations for such a purpose.

By accident (while attempting to reduce costs by substituting word length for frequency) we came across Miller's paper [7]. Wrongly or no, we decided, after reading his paper, to abandon word counting. The decision forced us to look for some other statistic having to do with word occurrence, and the thought grew that the extent to which a word occurred, that is, the field across which a word was spread, might have significance.

We proceeded as follows, using an existing body of documents that had already been classified in the usual way. First, we counted the number of documents in which a word appeared (one or more times) in the whole population of documents. Then we did the same for the documents in a category. We also counted the total documents in each case and then formed the ratio:

$$\frac{\text{no docs with word in cat}}{\text{no docs in cat}} \Big/ \frac{\text{no docs with word in pop}}{\text{no docs in population}} \qquad (1)$$

(no = number, cat = category, docs = documents)

In explanation, it might be said that we are taking the occurrence of a word in a document to indicate an attribute of that document. This decision is yes or no, regardless of how many times the word appears. We could have used two occurrences, or three or more to indicate the presence of the attribute; we could even weight the attribute, but these are not fundamental questions. Misspellings and sample sizes bear on the problem.

Expression (1) relates the frequency of occurrence of a word among the documents of a category to its frequency among the documents of the set; i.e., the whole population, or the sum of all categories. As it stands, the expression is an indicator of word significance. Words with frequency independent of the category (such as 'the' and 'and') should give a value of unity while significant words would be those with a greater value. A word which occurs in one document only will give for expression (1) the value:

$$\frac{\text{no docs in pop}}{\text{no docs in cat}} \qquad (2)$$

An expression such as (2), dependent on category size, is undesirable but we can normalize expression (1) by multiplying by the inverse of expression (2) giving:

$$\left[\frac{\text{no docs with word in cat}}{\text{no docs in cat}} \Big/ \frac{\text{no docs with word in pop}}{\text{no docs in pop}}\right]\left[\frac{\text{no docs in cat}}{\text{no docs in pop}}\right] \qquad (3)$$

which reduces readily to

$$\frac{\text{no docs with word in cat}}{\text{no docs with word in pop}} \qquad (4)$$

For those who like things mathematical, equation (3) is a simple formula in Bayes probability. Birnbaum and Maxwell [8] page 157 give an identical formula (in their notation) for classifying patients in a mental hospital.

We shall call expression (4) the probability, based on the word (measurement) *w*, that the document containing the word *w* belongs in the category. If we calculate (4) for all words in the category, then order them by magnitude, we have a list of keywords, in order of importance, for the category. Similar lists can be prepared for all categories. There are immediate obvious applications for such lists.

To classify a new document, we try words from the document one by one, against the various category lists, and sum the probabilities for each match in a category till one category shows specified numerically greater probability than the others. We can go down one category list and find how many words are needed to reach a given probability, or alternatively, we can search the document for the first word on the category lists, then the second, etc., and choose the category first attaining a specified margin.

We have tacitly assumed that the total lists would be used. In practice it is expected that very much less than the whole list would need to be consulted to reach a decision. Costs might be the ultimate consideration. Words which occur with equal frequency in the category and outside (words such as 'the' and 'and') offer little help in classification. Note that words below this point on the list are contra-indicated. In fact they indicate that documents with these words belong *not* in this category. Of course, if no decision is reached, the document would be rejected for examination. It should be fairly obvious that these initial category lists can be treated like document lists and classified into groups of higher categories.

If the collection is large, it might be too expensive to use the total collection for population statistics and a sample could be selected. Rather than use a random sample, it is proposed to eliminate documents from the population statistics (not from the actual categories of course) chronologically, oldest first, perhaps maintaining a specified number in the category, or else covering a specified time period – perhaps some years. As new documents are constantly entering and being included in the statistics the lists of significant words would change with time, and the lists would in fact update themselves. The actual nature of the categories could change, and older documents would then be indicated by word lists that did not fit them too well. Such a situation is preferable to the present one, where the categories suit the old documents but not the new ones. See for example Borko's [9] comments concerning angels and tunnel diodes in the Dewey System. The basic idea that classification systems are dynamic is expressed by Cherenin [10] in the opening of his paper.

We have assumed that our lists were formed from existing categories of documents and our experimental lists are in fact so derived. We must accept that a mechanized system will, at least initially, have to use existing material and traditional classifications

and our experimental work reflects this fact. We have of course some thoughts for the future. With time the category boundaries may well change, as mechanized selection of additions alters the nature of the category list. There is nothing to prevent us hastening this process by regeneration. We take a document from the collection and reclassify it – it will not necessarily return to its original category as the category lists have changed in the interim.

There is also no need to start with existing categories. Suppose, as an example, we simply divide 100 documents randomly into 10 piles of 10 documents each. (The basis for such a choice is given in another paper [12]). We form category lists for each group and then remove a document (altering statistics accordingly) and reassign it to that category giving the best match. If the process converges its continuation should lead to a stable condition which is in some sense an optimum categorization as Doyle [3] has indicated. Any document removed from a category will now be found to return there, as the optimum match, and in this sense the system is stable. Whether such categories (investigation would be needed to derive even the subject title) are intrinsically better for library use on a long-term basis is difficult to say. (They have great interest applied to maps and photographs, but there the interest is in the search for the reason that the procedure lumped certain items together.) We have already indicated that our categories may drift with time. They could be regenerated but I doubt if users could stand it.

When a document is to be added to an existing collection it is assigned to a category on the basis of a closer resemblance than to the collection as a whole – but a 'wild' or irrelevant document could give trouble. What is required is some sort of minimum match to the total set. The opening paragraph of Cherenin's paper [10] gives an excellent discussion of this problem. For example, he says, 'Proceeding from the defined set of information the scope of the questions asked is also determined. . .'. He goes on, 'this does not mean that all the possible questions are previously known, but that for each question it is usually known whether or not it can be asked'. In using our category lists, of course, a question is treated as a document and its words are tried against the various category lists, then, if necessary, against each document in a chosen category. The document word list, of course, consists of those words which matched the category list or classification.

We have used abstracts rather than full text merely to save keypunching costs and time in an experimental situation. Full text is completely suitable and, as our first step is to eliminate word repetitions in a document, compression would be much greater and computer storage requirements should be comparable. We expect to repeat some work now done with abstracts with full text data. It could not be done in time for this paper. Operationally, full text would be used. We do not expect a significant difference in performance using full text. Strong support for this belief is found in Salton [11], page III-31. We quote; 'document abstracts are more effective for content analysis purposes than document titles alone; further improvements appear possible when abstracts are replaced by large text portions; however, the increase in effectiveness in not large enough to reach the unequivocal conclusion that full text processing is always superior

to abstract processing'. We also expect to repeat some work with material published several years later and see if any noticeable vocabulary changes occur in the list.

For our experimental work we selected abstracts from the I.E.E.E. Transactions on Electronic Computers (see Appendix) choosing categories 3,5,6, and 8.

We decided to form keyword lists for these categories by slightly different methods:

A. All words, regardless of their number of occurrences, were included in the statistical preparation if they occurred in at least two documents within a category.

B. A word had to occur two or more times in at least two documents before it was included in the procedure.

C. A word had to occur once or oftener in at least three different documents before it was used.

The resulting category lists are shown in the figures. We apologize for the inverted order – blame our programmer. Words having probability one are unique in that category. Procedure *A*, requiring occurrence of a word in at least two documents before being included, is, we feel, a promising procedure for preparation of category lists. The procedure gives, in a sense, words which not only represent the characteristics of documents, but which also indicate couplings between documents.

The requirement of procedure *B*, that a word occur twice in a document, has value in eliminating misspellings and odd expressions which creep in during keypunching. Our keypunching is not verified. Ultimately, of course, input material will be by character reader, not keypunching, and this elimination feature will then assume greater importance. It has other effects on the final list.

In any case, examination of the category lists shows that slight modifications to the procedure offer ready means of adjusting the size of the lists. We regret that time forced us to use such a small sample of material. We feel that the word lists in the figures are already surprisingly good for such a small sample – and they were chosen without human intervention.

References

1. Wallace, E.M. Rank order patterns of common words as discriminators of subject content in scientific and technical prose. *In* Statistical Association Methods for Mechanized Documentation, *Edited by* M.E. Stevens, V.E. Guidiano, and L. Heilprin, Miscellaneous Publication 269: 225–229; 1964.

2. Borko, H. (Ed.) Automated language processing, the state of the art. John Wiley and Sons Inc., New York, 1967.

3. Doyle, L.B. Breaking the cost barrier in automatic classification. System Development Corporation, Pub. No. SP-2516, Santa Monica, California, July 1966.

4. Trachenberg, A. Automatic document classification using information theoretical methods. 'Automation and Scientific Communication, Part 2'. H.P. Lukn (Ed.) American Documentation Institute, 349–350; 1962.

5. Williams, J.H. A discrimination method for automatically classifying documents. Proc. Fall Joint Computer Conference, 24: 161; 1963.

6. Miller, G.A., Newman, E.B. and Friedman, E.A. Length-frequency statistics for written English. Information and Control, 1: 370–389; 1958.

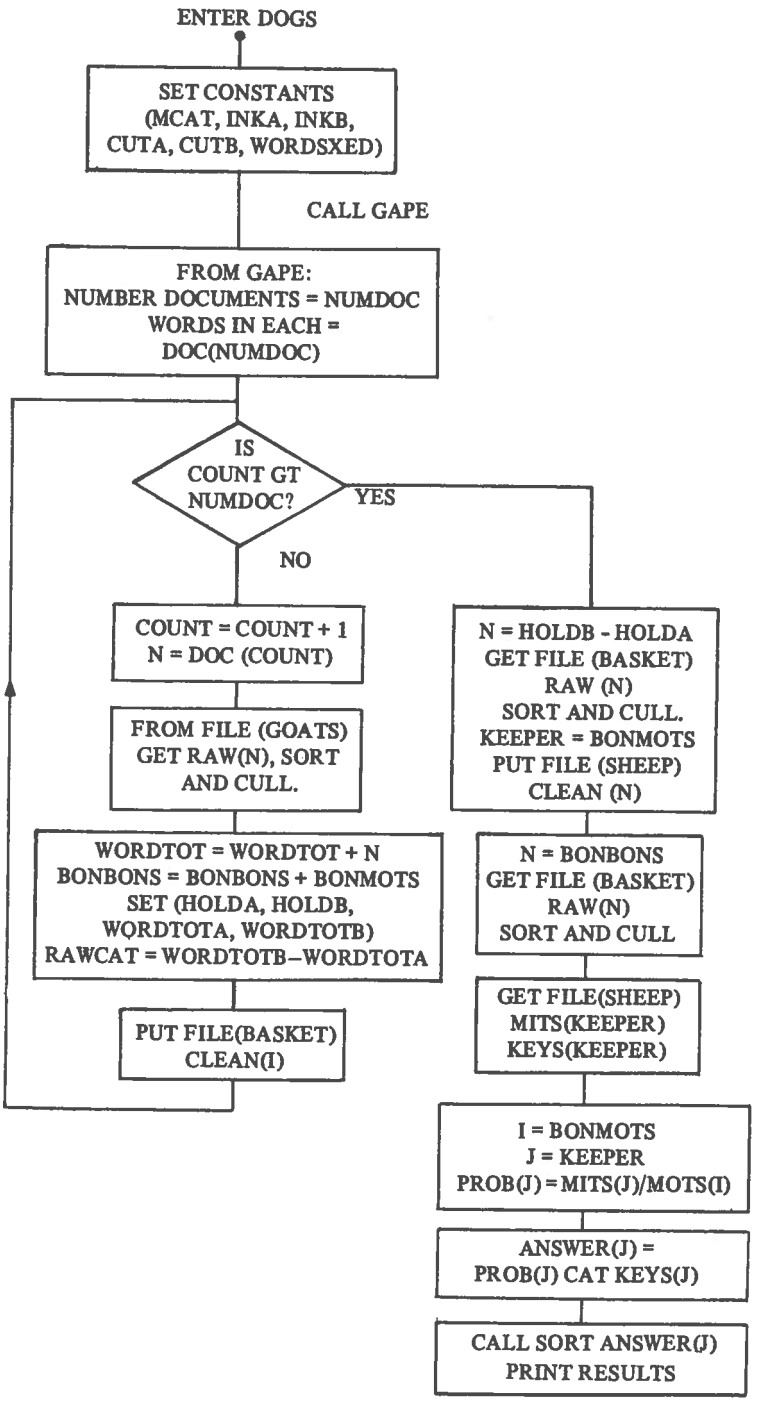
7. Birnbaum, A. and Maxwell, A.E. Classification procedures based on Bayes' formula. Applied Statistics, 9: 152–169; 1960.

8. Borko, H. The construction of an empirically based mathematically derived classification system. Proc. Spring Joint Computer Conference, American Federation of Information Processing Societies, 1962.

9. Cherenin, V.P. The basic types of information tasks and some methods of their solution. Proc. International Congress on Scientific Information, Washington, D.C., 2: 823–853; 1958.

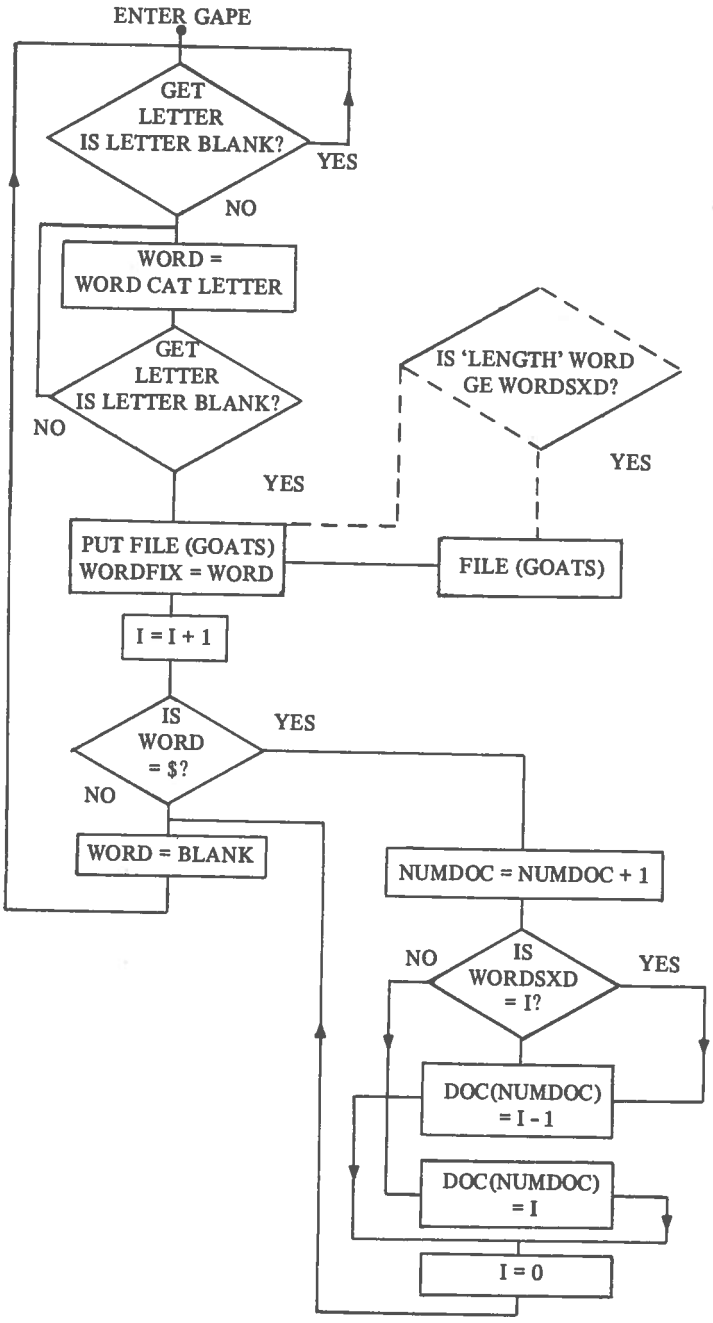
10. Salton, G. Information Storage and Retrieval Scientific Report No. ISR-12 to the National Science Foundation, Reports on Evaluation, Clustering and Feedback. PB176536. Ithaca, New York, June 1967.

11. Hoyle, W.G. On the number of categories for classification. (To be published)



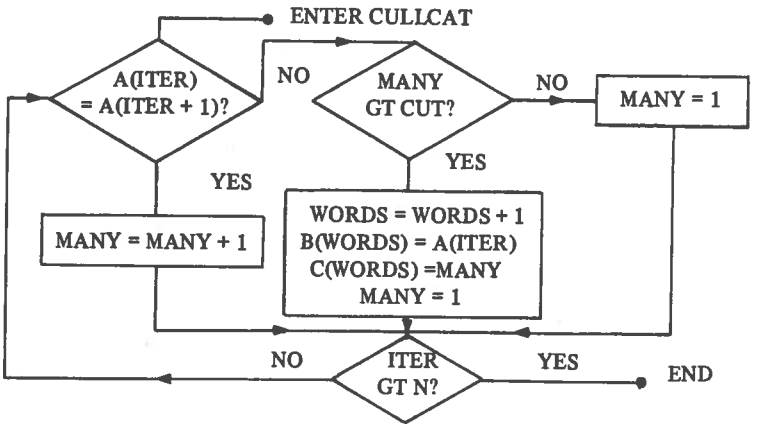
Main Program

DOGS



Subroutine

GAPE



Subroutine

CULLCAT

APPENDIX

Abstracts Used in the List Preparation

Cat. 3		Cat. 5		Cat. 6		Cat. 8	
5200	5305	5208	5330	5143	5353	5236	5364
5203	5306	5211	5331	5154	5355	5237	5365
5205	5307	5213	5332	5155	5356	5240	5366
5112	5308	5143	5333	5156	5630	5165	5367
5114	5309	5144	5622	5352	5631	5167	5642
5116	5611	5146	5624			5169	5643
5304	5612	5329				5363	

WORDS OCCUR 1 TIMES IN EACH DOC 2 TIMES IN CATEGORY

0.1200	BE	0.2045	IS	0.4000	FOUR	0.6666	ABILITY	1.0000 EDITING
0.1200	WHICH	0.2051	ARE	0.4000	MACHINE	0.6666	CAPABILIT	1.0000 EDITOR
0.1250	CAN	0.2105	IT	0.4000	POSSIBLE	0.6666	CLASSIFIC	1.0000 EMPHASIS
0.1538	PROGRAM	0.2127	A	0.4000	PRESENTED	0.6666	EACH	1.0000 ENGLISH
0.1538	USE	0.2142	IN	0.4000	PROCESS	0.6666	EXPERIMEN	1.0000 ENTRIES
0.1666	DESCRIBED	0.2162	FOR	0.4000	PRODUCTIO	0.6666	FIRST	1.0000 EXTENSIVE
0.1666	SOME	0.2222	ON	0.4000	SYSTEMS	0.6666	ITSELF	1.0000 INTERACTI
0.1764	AS	0.2380	BY	0.4000	THREE	0.6666	LATTER	1.0000 JUSTIFICA
0.1764	GIVEN	0.2380	COMPUTER	0.4285	ITS	0.6666	MATRIX	1.0000 LINGUISTI
0.1764	THIS	0.2500	AN	0.4285	RELATIONS	0.6666	NORMAL	1.0000 MAIN
0.1818	CHARACTER	0.2500	DEVELOPME	0.5000	COMPUTER-	0.6666	REVIEWED	1.0000 MATHEMATI
0.1818	PROGRAMS	0.2500	THERE	0.5000	CORRESPON	0.6666	UNDER	1.0000 NON-NUMER
0.1875	USED	0.2727	DATA	0.5000	FORM	0.6666	WORD	1.0000 OUT
0.1956	TO	0.2857	INTO	0.5000	FORMS	0.7500	LANGUAGES	1.0000 PIECE
0.2000	ANALYSIS	0.2857	THEN	0.5000	OPERATION	0.7500	ORGANIZAT	1.0000 PUBLICATI
0.2000	AND	0.3333	ALSO	0.5000	PART	0.7500	RESEARCH	1.0000 REGULAR
0.2000	BEEN	0.3333	HAS	0.5000	REQUIRED	1.0000	ARTICLES	1.0000 SELECTED
0.2000	FROM	0.3333	INPUT	0.5000	SCHEME	1.0000	BROAD	1.0000 TEXT
0.2000	OF	0.3333	PROGRAMMI	0.5000	STATE	1.0000	CLOSURE	1.0000 THEORETIC
0.2000	SYSTEM	0.3333	PROPERTIE	0.5000	THEIR	1.0000	CONCLUDED	1.0000 TRANSLATI
0.2000	THAT	0.3750	PROCEDURE	0.5000	USER	1.0000	CONTEXT-F	1.0000 VOCABULAR
0.2000	THE	0.3750	THAN	0.5000	WILL	1.0000	DICTIONAR	* * *
0.2000	WITH	0.4000	COMPUTATI	0.5714	LANGUAGE	1.0000	DIVIDED	* * *

TOTAL WORD TOKENS	5808
TOKENS IN CATEGORY	1150
TOTAL WORD TYPES	519
TYPES IN CATEGORY	113

Figure 1 List of keywords (in inverse order) selected on the basis of one or more occurrences in a document of a category and occurrence in two or more documents of the category. Actual category 'human communication, documentation, and humanities'.

WORDS OCCUR 2 TIMES IN EACH DOC 2 TIMES IN CATEGORY

0.1304	IN	0.2000	WHICH	0.2500	ON	1.0000	CONTEXT-F	1.0000 PROCEDURE
0.1500	A	0.2127	THE	0.2608	FOR	1.0000	DICTIONAR	1.0000 RESEARCH
0.1666	BE	0.2173	ARE	0.4000	OPERATION	1.0000	EACH	1.0000 TEXT
0.1714	TO	0.2222	SYSTEM	0.5000	PRODUCTIO	1.0000	ENGLISH	1.0000 TRANSLATI
0.1875	OF	0.2250	AND	0.6666	LANGUAGE	1.0000	ITS	* * *
0.2000	AN	0.2500	IS	1.0000	COMPUTATI	1.0000	LINGUISTI	* * *

TOTAL WORD TOKENS	5808
TOKENS IN CATEGORY	1150
TOTAL WORD TYPES	115
TYPES IN CATEGORY	28

WORDS OCCUR 1 TIMES IN EACH DOC 3 TIMES IN CATEGORY

0.1200	BE	0.2000	OF	0.2142	IN	0.3750	PROCEDURE	0.6666 EACH
0.1200	WHICH	0.2000	SYSTEM	0.2162	FOR	0.3750	THAN	0.6666 EXPERIMEN
0.1764	AS	0.2000	THAT	0.2222	ON	0.4000	PRESENTED	0.7500 LANGUAGES
0.1764	GIVEN	0.2000	THE	0.2380	BY	0.4000	SYSTEMS	0.7500 ORGANIZAT
0.1764	THIS	0.2000	WITH	0.2380	COMPUTER	0.4285	ITS	0.7500 RESEARCH
0.1875	USED	0.2045	IS	0.2500	AN	0.4285	RELATIONS	1.0000 DICTIONAR
0.1956	TO	0.2051	ARE	0.2727	DATA	0.5000	FORM	1.0000 INTERACTI
0.2000	AND	0.2105	IT	0.3333	ALSO	0.5000	OPERATION	* * *
0.2000	BEEN	0.2127	A	0.3333	HAS	0.5714	LANGUAGE	* * *

TOTAL WORD TOKENS	5808
TOKENS IN CATEGORY	1150
TOTAL WORD TYPES	279
TYPES IN CATEGORY	43

Figure 2

(A) Same data as Figure 1 but keyword selection basis requires that a word occur twice or oftener in at least two documents

(B) Same as Figure 1 except that requirement is that a word must now occur in at least three documents before being selected

WORDS OCCUR 1 TIMES IN EACH DOC 2 TIMES IN CATEGORY

0.1176	AS	0.2391	TO	0.3750	RESULTS	0.5000	SOLUTION	1.0000 APPROXIMA
0.1333	BEEN	0.2400	THAT	0.3750	SHOWN	0.5000	SOME	1.0000 CONNECTIO
0.1428	BY	0.2400	WHICH	0.3750	THAN	0.5000	STUDIED	1.0000 EQUATION
0.1538	USE	0.2432	FOR	0.4000	CONSIDERE	0.5714	PROBLEMS	1.0000 ERRORS
0.1666	DESCRIBED	0.2500	IS	0.4000	DEFINED	0.5714	THEN	1.0000 EVALUATE
0.1666	HAS	0.2500	THERE	0.4000	DIGITAL	0.6000	DO	1.0000 FORMULA
0.1818	CHARACTER	0.2564	ARE	0.4000	EQUATIONS	0.6000	EXAMPLE	1.0000 INDEPENDE
0.1875	CAN	0.2600	AND	0.4000	GENERAL	0.6666	APPLIED	1.0000 INITIAL
0.1904	COMPUTER	0.2600	OF	0.4000	LINEAR	0.6666	CALCULATI	1.0000 INTERPOLA
0.2000	ANALYSIS	0.2600	THE	0.4000	MACHINE	0.6666	DIFFERENT	1.0000 METHOD
0.2000	FROM	0.2631	IT	0.4000	NO	0.6666	ERROR	1.0000 NUMERICAL
0.2000	PRESENTED	0.2727	OR	0.4000	SEVERAL	0.6666	LESS	1.0000 POLYNOMIA
0.2000	SUCH	0.2800	BE	0.4000	THREE	0.6666	OBTAIN	1.0000 PROVED
0.2000	SYSTEM	0.2857	CONVENTIO	0.4285	MAY	0.6666	PROPERTY	1.0000 SOLVING
0.2000	WITH	0.2857	DISCUSSED	0.4285	PAPER	0.6666	RESULT	1.0000 SQUARES
0.2142	AN	0.2857	THESE	0.4375	USED	0.6666	S	1.0000 TAKES
0.2142	IN	0.2857	TYPE	0.4444	NOT	0.6666	STARTING	1.0000 VALUES
0.2222	ALSO	0.3333	DEVELOPED	0.4705	GIVEN	0.6666	VALUE	1.0000 X
0.2222	AT	0.3333	METHODS	0.5000	FUNCTION	0.7142	PROBLEM	* * *
0.2222	ON	0.3333	OBTAINED	0.5000	MANY	0.7500	COMPUTED	* * *
0.2340	A	0.3333	PARTICULA	0.5000	NEW	0.7500	IF	* * *
0.2352	THIS	0.3333	PROPERTIE	0.5000	PRACTICAL	0.7500	LEAST	* * *

TOTAL WORD TOKENS	5808
TOKENS IN CATEGORY	1150
TOTAL WORD TYPES	519
TYPES IN CATEGORY	106

Figure 3 Keywords selected as in Figure 1 except now for the category 'mathematics'

COMPUTER PROGRAM

```
      DOGS..PROC OPTIONS (MAIN),.DCL

 1              DOGS..PROC OPTIONS (MAIN),.DCL
                   PROB(KEEPER)FIXED DEC(5,4)CONTROLLED,
                   ANSWER(N)CHAR(18)CONTROLLED,
                   KEYS(KEEPER)CHAR(9)CONTROLLED,
                   (RAW(N),CLEAN(N))CHAR(9)CONTROLLED,
                   TEMPB CHAR(18),
                    (TEMPA,SCRAP) CHAR (9),
                  (NUMDOC,COUNT,WORDTOT,BONBONS,BONMOTS,RAWCAT,WORDTOTA,WORDTOTB,HOLDA)
                   FIXED DEC (4) INIT(0),
                   (WORDSXD,
                    HOLD,KEEPER,I,II,JJ,N,CUTA,CUTB,MCAT, INKA,INKB,HOLDB)FIXED DEC(4),
                   (DOC(MCAT),MITS(KEEPER),MOTS(N))FIXED DEC(4)CONTROLLED,.
                  /* * * * * * */
 3                 GET LIST(MCAT,INKA,INKB,CUTA,CUTB,WORDSXD),.
                   /*MCAT IS ESTIMATED MAX NO. OF DOCS,CUTA IS NO OF REQUIRED
                   PUT EDIT(MCAT,INKA,INKB,CUTA,CUTB,WORDSXD)  (SKIP,6(F(6),X(2))),.
                   REPETITIONS WITHIN A DOC., CUTB IS NO OF DOCS EXCEEDING ONE, IN
                   WHICH A WORD MUST OCCUR*/
                   /*THE CATEGORY INCLUDES DOCS INKA TO INKB INC.*/
                   /* WORDS OF LENGTH LESS THAN WORDSXD ARE NOT COUNTED */
 4              CALL GAPE,.
 5                 PUT EDIT('NO OF DOCS=',NUMDOC)(SKIP,A(15),F(3)),.
 6                 DO I= 1 TO NUMDOC,.
 7                 PUT EDIT('WORDS IN DOC',I,'=',DOC(I))
                   (SKIP,A(12),F(3),X(2),A(2),F(5)),.
 8                 END,.
 9              START..
                   COUNT=COUNT+1,.
10                 IF COUNT GT NUMDOC THEN
11                 GO TO SHRINK,.
12                 N=DOC(COUNT),.
13                 ALLOCATE RAW,.
14                  GET FILE(GOATS)EDIT((RAW(I)DO I=1 TO N))(A(9)),.
15                 IF WORDSXD = 1 THEN GET FILE (GOATS) EDIT (SCRAP)(A(9)),.
17                 CALL SORT(RAW,TEMPA),.
18                 ALLOCATE CLEAN(N),MOTS(N),.
19                 CALL CULLCAT(RAW,CLEAN,MOTS,BONMOTS,CUTA),.
20                 FREE RAW,.
21                 WORDTOT=WORDTOT+N,.
22                 BONBONS=BONBONS + BONMOTS,.
23                 IF COUNT =INKA-1 THEN HOLDA =BONBONS,.
25                 IF COUNT=INKB THEN HOLDB = BONBONS,.
27                 IF COUNT = INKA-1 THEN WORDTOTA = WORDTOT,.
29                 IF COUNT = INKB THEN DO,. WORDTOTB=WORDTOT,.
32                                          RAWCAT= WORDTOTB-WORDTOTA,.END,.
34              EXHALE..
                PUT FILE(BASKET)EDIT((CLEAN(I) DO I=1 TO BONMOTS))(A(9)),.
35                 FREE CLEAN,MOTS,.
36                 GO TO START,.
37              SHRINK.. CLOSE FILE (BASKET),.
38                 N=HOLDB-HOLDA,.
39                 ALLOCATE RAW(HOLDB),.
```

```
DOGS..PROC OPTIONS (MAIN),.DCL

40                GET FILE(BASKET)EDIT((RAW(I) DO I=1 TO HOLDA))(A(9)),.
41                GET FILE(BASKET)EDIT((RAW(I) DO I=1 TO N))(A(9)),.
42                CALL SORT(RAW,TEMPA),.
43                ALLOCATE CLEAN(N),MOTS(N),.
44                CALL CULLCAT(RAW,CLEAN,MOTS,BONMOTS,CUTB),.
45                FREE RAW,.
46                KEEPER=BONMOTS,.
47              SHEPHERD..
                  PUT FILE(SHEEP)EDIT((MOTS(I),CLEAN(I)DO I=1 TO KEEPER))
                  (F(7),A(9)),.
48                FREE CLEAN,MOTS,.
49                CLOSE FILE(BASKET),.
50                CLOSE FILE(SHEEP),.
51                N=BONBONS,.
52                ALLOCATE RAW(N),.
53                GET FILE(BASKET)EDIT((RAW(I)DO I=1 TO N))(A(9)),.
54                CALL SORT(RAW,TEMPA),.
55                ALLOCATE CLEAN(N),MOTS(N),.
56                CALL CULLCAT(RAW,CLEAN,MOTS,BONMOTS,CUTB),.
57                FREE RAW,.
58                     ALLOCATE MITS(KEEPER),KEYS(KEEPER),.
59                GET FILE(SHEEP)EDIT((MITS(I),KEYS(I)DO I=1 TO KEEPER))
                  (F(7),A(9)),.
60                ALLOCATE PROB(KEEPER),.
61                HOME..DO I=1 TO BONMOTS,.
62                DO J=1 TO KEEPER,.
63                IF KEYS(J)=CLEAN(I)THEN
64                PROB(J)=MITS(J)/MOTS(I),.
65                END HOME,.
67                N=KEEPER,.
68                ALLOCATE ANSWER(N+10),.
                   /*THE EXTRA 10 POSITIONS ARE TO ALLOW ROOM FOR THE
                     ASTERISKS FOR THE COMPLETIION OF THE LINE ON PRINTOUT */
69                DO J=1 TO N,.
70                ANSWER(J)=PROB(J)CAT' 'CAT KEYS(J),.
71                END,.
72                CALL SORT(ANSWER,TEMPB),.
73              PUTTER..
                   PUT PAGE,.
74                PUT EDIT ('WORDS OCCUR',CUTA+1,'TIMES IN EACH DOC',CUTB+1,
                  'TIMES IN CATEGORY')(A(13),F(3),X(1),A(21),F(3),X(1),A(22)),.
75                PUT SKIP(2),.
76                 II=1,.
77                JJ=CEIL(N/5),.
78                DO J=(N+1)TO(5*JJ),.
79                ANSWER(J)='     * * *',.
80                END,.
81                 DO WHILE ((II+4*JJ) NG (5*JJ)),.
82                PUT EDIT((ANSWER(J)DO J=II TO(II+4*JJ)BY JJ))(SKIP,5A(18)),.
83                II=1+II,.END,.
85                 PUT SKIP(2),.
86                PUT EDIT('TOTAL WORD TOKENS',WORDTOT,'TOKENS IN CATEGORY',
```

```
DOGS..PROC OPTIONS (MAIN),.DCL

                  RAWCAT,'TOTAL WORD TYPES',BONMOTS,'TYPES IN CATEGORY',KEEPER,
               'INKA=',INKA,'INKB=',INKB)
                  (SKIP,A(30),F(6)),.
 87               PUT SKIP LIST('DATE=',DATE),.
 88               FREE CLEAN,MOTS,PROB,ANSWER,.
                 /* * * * * * */
 89            GAPE..PROC,.
 90               DCL WORD CHAR(20)VAR INIT(''),
                  I FIXED INIT(0),
                  (WORDFIX,SCRAP,TEMPA)CHAR(9),
                  LETTER CHAR(1),.
 91               ON ENDFILE(SYSIN)GO TO GAFIN,.
               /* MCAT IS ESTIMATED MAXIMUM NUMBER OF DOCUMENTS */
 93               ALLOCATE DOC(MCAT),.
 94            GASTART..GET EDIT(LETTER)(A(1)),.
 95               IF LETTER=' ' THEN
 96               GO TO GASTART,.
 97            BUILD..WORD=WORD CAT LETTER,.
 98               GET EDIT(LETTER)(A(1)),.
 99               IF LETTER NE ' ' THEN
100               GO TO BUILD,.
101               WORDFIX=WORD,.
                   /*WORDFIX IS USED TO PUT THE WORDS IN FIXED FORMAT FOR
                   MORE EFFICIENT HANDLING */
102               IF LENGTH (WORD) GE WORDSXD THEN DO,.
104               PUT FILE(GOATS)EDIT(WORDFIX)(A(9)),.
105               I = I+1,. END,.
                   /* THE SYMBOL $ SEPARATES DOCUMENTS */
107               IF WORD='$'THEN DO,.
109               NUMDOC=NUMDOC+1,.
110               IF WORDSXD = 1 THEN /*DO NOT COUNT $ */
111               DOC(NUMDOC) = I-1,. ELSE
112               DOC(NUMDOC) = I,.
113               I=0,.
114               END,.
115               WORD='',.
116               GO TO GASTART,.
117            GAFIN..
                  CLOSE FILE(GOATS),.
118                                    END GAPE,.
119            SORT..PROC(RAN,TEMP),.DCL
                  TEMP CHAR(*),
                  RAN(*) CHAR(*),
                  (I,J,K,M,NO)FIXED DEC(4),.
121               M=N,.
122               LABELA.. M=FLOOR(M/2),.
123            IF M=0 THEN GO TO FINISH,.
125            K=N-M,.J=1,.
127               LABELB..I=J,.
128               LABELC..IF RAN(I) GT RAN(I+M)
129            THEN DO,.
130            TEMP=RAN(I),.
```

```
          DOGS..PROC OPTIONS (MAIN),.DCL

131                 RAN(I)=RAN(I+M),.
132                 RAN(I+M)=TEMP,.END,.
134                 ELSE GO TO LABELD,.
135                 I=I-M,.
136                 IF I LT 1 THEN
137                    LABELD.. J=J+1,.
138                 ELSE GO TO LABELC,.
139                 IF J GT K THEN
140                 GO TO LABELA,.
141                 ELSE GO TO LABELB,.
142                  FINISH..                                  END SORT,.
143                 CULLCAT..PROC(A,B,C,WORDS,CUT),.DCL CUT FIXED DEC(4),
                       (A(*),B(*))CHAR(*),
                       (WORDS,C(*))FIXED DEC(4),
                       MANY FIXED DEC(4)INIT(1),.
145                    WORDS=0,.
146                    DO ITER=1 TO N,.
147                    IF A(ITER)=A(ITER+1)THEN
148                    MANY=MANY+1,.ELSE
149                     IF MANY GT CUT THEN DO,.
151                          WORDS=WORDS+1,.
152                          B(WORDS)= A(ITER),.
153                          C(WORDS) = MANY,.
154                          MANY = 1,.END,.
156                          ELSE MANY = 1,.
157                    END,.
158                                          END CULLCAT,.
159                 TAIL..END DOGS,.

  NO ERROR DETECTED, ANY WARNINGS ARE NOT PRINTED.

COMPILE TIME          .99 MINS
```