Information Processing & Management Vol. 21, No. 3, pp. 225–232, 1985
Printed in Great Britain

0306-4753/85 $3.00+.00
1985 Pergamon Press Ltd.

AN ALGORITHM FOR THE CALCULATION OF EXACT TERM DISCRIMINATION VALUES

PETER WILLETT
Department of Information Studies, University of Sheffield, Western Bank, Sheffield S10 2TN, U.K.

(*Received for publication* 28 *September* 1984)

Abstract—Term discrimination values have been suggested as an effective means for the selection and weighting of index terms in automatic document retrieval systems. This paper reports an algorithm for the calculation of term discrimination values that is sufficiently fast in operation to permit the use of exact values, rather than the approximate values studied in previous work. Evidence is presented to show that the relationship between term discrimination and term frequency is crucially dependent upon the type of inter-document similarity measure that is used for the calculation of the discrimination values.

1. INTRODUCTION

Many strategies have been suggested for the automatic selection of indexing terms to represent the content of documents in information retrieval systems[1, 2]. One of the most elegant approaches to automatic indexing is the term discrimination model that has been developed by Salton and his co-workers over several years[3–8]. The model suggests that the ideal retrieval environment would be an index term space in which all of the documents are as far apart as possible since it should then be possible to retrieve the (presumably relevant) documents close to a query when it is represented as a point in the space. Such an environment provides an obvious means for the evaluation of the worth of individual terms, since a good term will be one that helps to increase the separation of all of the documents, while the assignment of a poor term will tend to make the space contract so that the inter-document separations decrease.

A collection of N documents may be represented by a set of document vectors $\mathbf{d}_j$, $1 \le j \le N$. Each such vector contains M elements where M is the number of distinct terms that have been used for the indexing of that collection: the ith element, $1 \le i \le M$, contains the number of occurrences of the ith term in the jth document. A measure of the similarity between a pair of documents $\mathbf{d}_j$ and $\mathbf{d}_k$ may then be calculated using a function such as the cosine coefficient

$$\cos(\mathbf{d}_j, \mathbf{d}_k) = \sum d_{ji}d_{ki}/(\sum d_{ji}^2 \sum d_{ki}^2)^{1/2}$$

where the summations are from i equals 1 to M. A space in which the documents are as far apart as possible will be one that corresponds to minimizing the sum, Q, defined by

$$\sum \cos(\mathbf{d}_j, \mathbf{d}_k) \qquad 1 \le j, k \le N, j <> k.$$

The effect of an individual term, i, upon the inter-document similarities may be determined by calculating Q and then recalculating it when i has been deleted from each of the documents to which it has been assigned: the difference between Q and this new sum, Q_i, then gives a measure of the extent to which the presence of i in the indexing vocabulary affects the separation of the set of documents. The *term discrimination value* of i, DV_i, is defined to be

$$DV_i = (Q_i - Q)/Q \tag{1}$$

and thus if $DV_i > 0$, the document space is more compact when i is deleted. In such a case, i is said to be a discriminating term since its presence helps to increase the separation of the documents in the space; conversely, if $DV_i \leq 0$, i should be regarded as a poor discriminator.

An effective automatic indexing strategy would then be to calculate the DV_i values for each of the potential indexing terms for some collection: terms with positive discrimination values may be used directly for indexing purposes, while word phrase and classification strategies[7] may be adopted to modify the frequency characteristics of non-discriminating terms so as to increase their discrimination values. Additionally, the discrimination value may be used as the basis for the weighting of index terms at search time[2].

An outline algorithm, using a PASCAL-like notation, for the calculation of term discrimination values is shown in algorithm A (Fig. 1). This algorithm involves the calculation of $N(N - 1)/2$ similarity coefficients for each of the M indexing terms, as well as an initial set of $N(N - 1)/2$ coefficients for the evaluation of Q. The complexity of the algorithm is hence of order $O(MN^2)$ and as the magnitude of M is comparable with, or may even exceed, that of N in a free text retrieval system, the calculation of term discrimination values by this algorithm is infeasible for all but collections of quite trivial size. For this reason, previous work on term discrimination has used a quite different, but approximate, method for the calculation of the values[2–6]. The procedure that has been adopted involves the calculation of the *centroid*, **c**, of a collection, this being the arithmetic average of all of the document vectors. The elements of **c**, c_i, are defined by

$$c_i = \sum d_{ji}/N \qquad 1 \leq j \leq N$$

and instead of summing all of the $N(N - 1)/2 \cos(\mathbf{d}_j, \mathbf{d}_k)$ values to obtain Q it is obtained from

$$Q = \sum \cos(\mathbf{c}, \mathbf{d}_j) \qquad 1 \leq j \leq N.$$

Similarly, the Q_i values are obtained from

$$Q_i = \sum \cos(\mathbf{c}^i, \mathbf{d}_j^i) \qquad 1 \leq j \leq N.$$

where $\mathbf{c}^i$ and $\mathbf{d}_j^i$ denote the centroid and the jth document with the ith term deleted. This approximate procedure may be described as shown in algorithm B (Fig. 2), where it will be seen that only $N(M + 1)$ document-centroid similarities in all need to be evaluated.

An analysis of algorithm B has been presented by Crawford[5]. Apart from describing ways of speeding up the calculation of the $\cos(\mathbf{c}, \mathbf{d}_j)$ and $\cos(\mathbf{c}^i, \mathbf{d}_j^i)$ values, he notes that during the calculation of each Q_i value, a summation is made over all of the $\cos(\mathbf{c}^i, \mathbf{d}_j^i)$ values even though very few of the documents will actually contain i. The assumption is then made that the $\cos(\mathbf{c}^i, \mathbf{d}_j^i)$ values for those documents not containing i will be virtually the same as the $\cos(\mathbf{c}, \mathbf{d}_j)$ values that were used to compute Q initially. This assumption may be used to form the basis for a modified version of algorithm B

```
$Q := 0$;
FOR $i$: = 1 TO $N - 1$ DO
     FOR $j$: = $i + 1$ TO $N$ DO
          $Q$: = $Q + \cos(\mathbf{d}_j, \mathbf{d}_k)$;
FOR $i$: = 1 TO $M$ DO
     BEGIN
          $Q_i$: = 0;
          FOR $j$: = 1 TO $N - 1$ DO
               FOR $k$: = $j + 1$ TO $N$ DO
                    $Q_i$: = $Q_i + \cos(\mathbf{d}_j^i, \mathbf{d}_k^i)$;
          $DV_i$: = $(Q_i - Q)/Q$
END.
```

Fig. 1. Algorithm A.

```
FOR $i$: = 1 TO $M$ DO
    BEGIN
        $c_i$: = 0;
        FOR $j$: = 1 TO $N$ DO
            $c_i$: = $c_i + d_{ji}$;
        $c_i$: = $c_i/N$
    END;
$Q$: = 0;
FOR $i$: = 1 TO $N$ DO
    $Q$: = $Q + \cos(\mathbf{c}, \mathbf{d}_j)$;
FOR $i$: = 1 TO $M$ DO
    BEGIN
        $Q_i$: = 0;
        FOR $j$: = 1 TO $N$ DO
            $Q_i$: = $Q_i + \cos(\mathbf{c}^i, \mathbf{d}_j^i)$;
        $DV_i$: = $(Q_i - Q)/Q$
    END.
```

Fig. 2. Algorithm B.

which obviates the need to calculate such coefficients and which is accordingly much faster in operation, although the calculated DV_i values are not identical with those obtained from using algorithm B.

This note describes a modified version of algorithm A that is sufficiently fast in operation to permit the calculation of exact term discrimination values, rather than the approximate values used hereto.

2. THE ALGORITHM

The algorithm is based upon two observations. Firstly, contributions to all of the M Q_i values are made as each interdocument similarity coefficient is calculated, rather than by calculating Q_i, and hence DV_i, for each of the terms in sequence. Secondly, a rather different approach is taken to the evaluation of $(Q_i - Q)$ in the numerator of the expression (1) above for the discrimination value of some term i. The numerator may be written as

$$\sum \cos(\mathbf{d}_j^i, \mathbf{d}_k^i) - \sum \cos(\mathbf{d}_j, \mathbf{d}_k) \qquad 1 \le j,\ k \le N,\ j <> k$$

this formulation emphasizing the manner of operation of algorithm A which first calculates Q as the sum of the $\cos(\mathbf{d}_j, \mathbf{d}_k)$ values, then calculates each Q_i as the sum of the $\cos(\mathbf{d}_j^i, \mathbf{d}_k^i)$ values, and finally subtracts one from the other to yield DV_i. Alternatively, the numerator may be written as

$$\sum [\cos(\mathbf{d}_j^i, \mathbf{d}_k^i) - \cos(\mathbf{d}_j, \mathbf{d}_k)] \qquad 1 \le j,\ k \le N,\ j <> k$$

this formulation emphasizing the fact that it is not Q_i *per se*, but the difference of Q_i from Q, that is important in obtaining DV_i.

With these two considerations in mind, let

$$\text{DOTPRODJK} = \sum d_{ji}d_{ki}, \qquad \text{SUMSQJ} = \sum d_{ji}^2, \qquad \text{and} \qquad \text{SUMSQK} = \sum d_{ki}^2$$

for some pair of documents $\mathbf{d}_j$, $\mathbf{d}_k$, with the summation in all three cases being from $i = 1$ to M. Thus

$$\cos(\mathbf{d}_j, \mathbf{d}_k) = \text{DOTPRODJK}/(\text{SUMSQJ}*\text{SUMSQK})^{1/2}.$$

Then let α, β and γ be defined as

$$(\text{DOTPRODJK} - d_{ji}d_{ki})/[(\text{SUMSQJ} - d_{ji}^2)*(\text{SUMSQK} - d_{ki}^2)]^{1/2} - \cos(\mathbf{d}_j, \mathbf{d}_k),$$
$$\text{DOTPRODJK}/[(\text{SUMSQJ}*(\text{SUMSQK} - d_{ki}^2)]^{1/2} - \cos(\mathbf{d}_j, \mathbf{d}_k),$$

and

$$\mathrm{DOTPRODJK}/[(\mathrm{SUMSQJ} - d_{ji}^2)*\mathrm{SUMSQK}]^{1/2} - \cos(\mathbf{d}_j, \mathbf{d}_k)$$

respectively. The modified algorithm may then be described as shown in algorithm C, where α, β and γ correspond to the presence of the ith term in both $\mathbf{d}_j$ and $\mathbf{d}_k$, in $\mathbf{d}_k$ alone and in $\mathbf{d}_j$ alone, respectively. The fourth, and by far the most frequent, case in which the ith term does not appear in either document may be neglected since $\cos(\mathbf{d}_j, \mathbf{d}_k)$ and $\cos(\mathbf{d}_j^i, \mathbf{d}_k^i)$ are then identical and cancel each other out.

It will be seen that this procedure involves the calculation of only $N(N - 1)/2$ inter-document similarity coefficients, an M-fold reduction in the computational load associated with algorithm A. However, the innermost loop of the algorithm will apparently be performed M times for each $\cos(\mathbf{d}_j, \mathbf{d}_k)$ value, and the algorithm would accordingly seem to involve $O(MN^2)$ increments of the Q_i values. In fact, the number of increments occasioned by each similarity calculation will be equal to the number of distinct terms assigned to the two documents since, as noted above, the great bulk of the M terms will not have been assigned to either of them and may be neglected: thus a pair of documents having t_j and t_k terms assigned will occasion $t_j + t_k - 1$ increments at most. The dependency upon M may be eliminated by storing the document vectors as sparse vectors in which only the non-zero elements are stored: in this case, the total number of increments will be proportional to tN^2, where t is the mean number of indexing terms assigned to each document, and algorithm C (Fig. 3) is computationally feasible for the relatively small-sized document test collections currently used in information retrieval research. It does, however, entail storage requirements additional to those of algorithms A and B, owing to the need to allocate M locations for the summing of the $Q_i - Q$ contributions as they are calculated.

The running time, but not the complexity, of algorithm C may be reduced if an inverted file to the document collection is available: the use of the inverted file for the calculation of discrimination values has been described by Crawford[5]. An inverted file contains a set of lists, one for each of the terms in the indexing vocabulary that is used to characterize the documents, such that the ith list contains the identifiers of those documents containing the ith term. The union of the lists corresponding to the terms in the jth document will thus eliminate the many documents that have no terms in common with j, and that cannot affect the DV_i values, without the need to match them against the jth document. There has recently been a considerable amount of interest in the use of alternative inverted file searching algorithms for the matching of documents and queries[9], and two of these algorithms, those used in the SIRE[10] and CUPID[11] experimental retrieval systems, were modified so as to calculate discrimination values. The resulting procedures were, however, considerably more complex than the simple union of the inverted file lists, and they did not prove to be noticeably more efficient in operation.

```
FOR $i$: = 1 TO $M$ DO $Q_i$: = 0;
$Q$: = 0;
FOR $j$: = 1 TO $N - 1$ DO
    FOR $k$: = $j + 1$ TO $N$ DO
        BEGIN
            $Q$: = $Q + \cos(\mathbf{d}_j, \mathbf{d}_k)$;
            FOR $i$: = 1 TO $M$ DO
                IF $d_{ki} > 0$ THEN
                    IF $d_{ji} > 0$ THEN $Q_i$: = $Q_i + \alpha$
                    ELSE $Q_i$: = $Q_i + \beta$
                ELSE
                    IF $d_{ji} > 0$ THEN $Q_i$: = $Q_i + \gamma$
                    ELSE SKIP
        END;
FOR $i$: = 1 TO $M$ DO
    $DV_i$: = $Q_i/Q$.
```

Fig. 3. Algorithm C.

3. THE RELATIONSHIP BETWEEN TERM FREQUENCY AND TERM DISCRIMINATION

Rather than calculating the set of DV_i values for some indexing vocabulary as a precursor to the selection of discriminating terms, an alternative and efficient strategy would be to determine whether a simple and general relationship exists between the frequency of a term in a document collection and the corresponding term discrimination value[12]. Should such a relationship exist, one could then select as indexing terms only those words, or combinations of words, that had the requisite frequencies of occurrence. This section attempts an analysis of the term discrimination model in terms of the algorithm presented above.

For some term i occurring in n of the documents of a collection of size N, DV_i is given by

$$DV_i = n(n - 1)\alpha + n(N - n)\beta + n(N - n)\gamma$$

(where a factor of $1/2Q$ has been ignored since it is common to all of the M discrimination values). Rearranging this expression

$$DV_i = n^2(\alpha - \beta - \gamma) + n(N\beta + N\gamma - \alpha).$$

If it is assumed that n is a continuous, rather than a discrete, variable, and that α, β and γ are all independent of n, rather than being related to it *via* all of the d_{ji} terms, this expression may be differentiated with respect to n, giving, firstly,

$$2n(\alpha - \beta - \gamma) + N(\beta + \gamma) - \alpha$$

and then

$$2(\alpha - \beta - \gamma).$$

There is hence only a single point of inflection at a frequency of

$$(\alpha - N(\beta + \gamma))/2(\alpha - \beta - \gamma)$$

and whether this will be a maximum or a minimum will be determined by the relative values of α, β and γ. If it is assumed that very long document representatives are being used, so that SUMSQJ $\gg d_{ji}^2$ and SUMSQK $\gg d_{ki}^2$, then both β and γ will be zero-valued while α will be given by $-d_{ji}d_{ki}/(\text{SUMSQJ}*\text{SUMSQK})^{1/2}$; this corresponds to α having a negative value and a maximum being observed at $n = \frac{1}{2}$. In practice, of course, β and γ will have small positive values and thus the DV_i values may be expected to fall off rapidly with increasing n, owing to the increasingly large and negative $n^2(\alpha - \beta - \gamma)$ contribution. The discrimination of the high-frequency terms must thus be low, and the exact form of the discrimination-frequency curve will depend upon the precise sets of α, β and γ values observed for some collection: in practice, it has been found that a maximum in this curve is obtained for the low-to-medium frequency terms[2–8].

Entirely comparable results are obtained if other similarity coefficients, such as the Dice, Jaccard or Overlap coefficients, are used as the measure of inter-document similarity during the calculation of the DV_i values; quite different results, however, are obtained if alternative types of inter-document similarity measure are used. In the case of the dot product, which is given by

$$\sum d_{ji}d_{ki}, \qquad 1 \leq i \leq M,$$

both β and γ will be zero-valued and the maximum will be obtained at exactly $n = \frac{1}{2}$. This is, of course, quite unobservable and the observed relationship between discrimination and frequency would be a monotonically decreasing one. In the case of the

Euclidean distance, conversely, the distance between a pair of points is given by

$$\sum (d_{ji} - d_{ki})^2 \qquad 1 \le i \le M.$$

In the special case of binary indexing, where the individual term weights d_{ji} are either 0 or 1, the α contribution to DV_i will be zero, and thus a point of inflection will be obtained in the discrimination-frequency curve at $n = N/2$. This point is a minimum, but it should be remembered that the measure is one of inter-document distance, rather than inter-document similarity, so that it corresponds to maximizing the separation, i.e. to the best discriminators. A term frequency of $N/2$ is most unlikely to be observable in practice and the discrimination would hence be expected to increase monotonically with frequency. In the general case of weighted, rather than binary, terms, some α contributions will generally be present, but these are unlikely to be sufficiently large to prevent the shape of the discrimination-frequency curve from being similar to that obtained with binary weighting.

Thus the observed relationship between term discrimination and term frequency would appear to be dependent not so much upon the characteristics of indexing terms as upon the measure of inter-document similarity that is used for the calculation of the discrimination values. In particular, the identification of the most discriminating terms, and thus those that are most appropriate for indexing purposes, with those of intermediate frequencies would seem to arise from the use of a certain class of similarity coefficients; the use of the dot product (or Euclidean distance) would instead suggest that the most discriminating terms were those of infrequent (or frequent) occurrence in the collection.

Some experiments to test this analysis were carried out, using a small collection of 1000 documents from the INSPEC data base. The titles and abstracts of these documents were compared with a stopword list, the non-trivial words stemmed, using the suffix-stripping routine described by Porter[13], and the cumulated stems used to represent each of the documents. The term discrimination values for each of the 4091

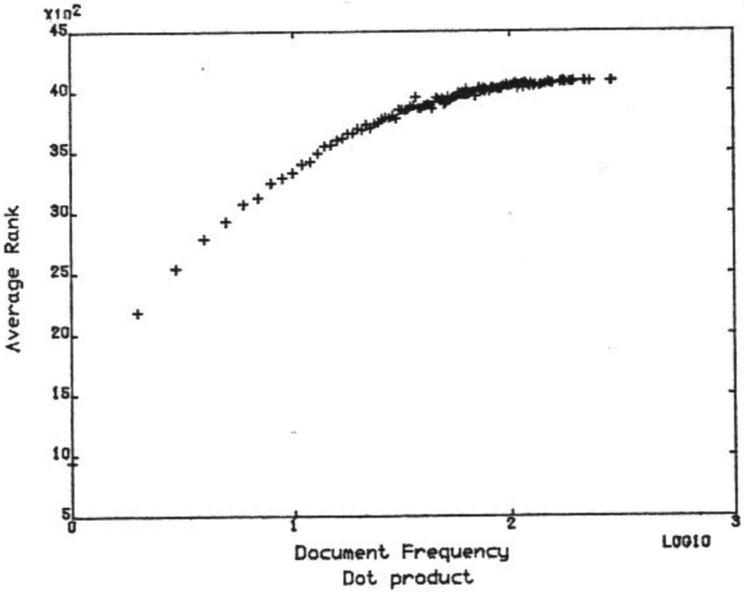

Fig. 4. Relationship between term discrimination and term frequency using the dot product.

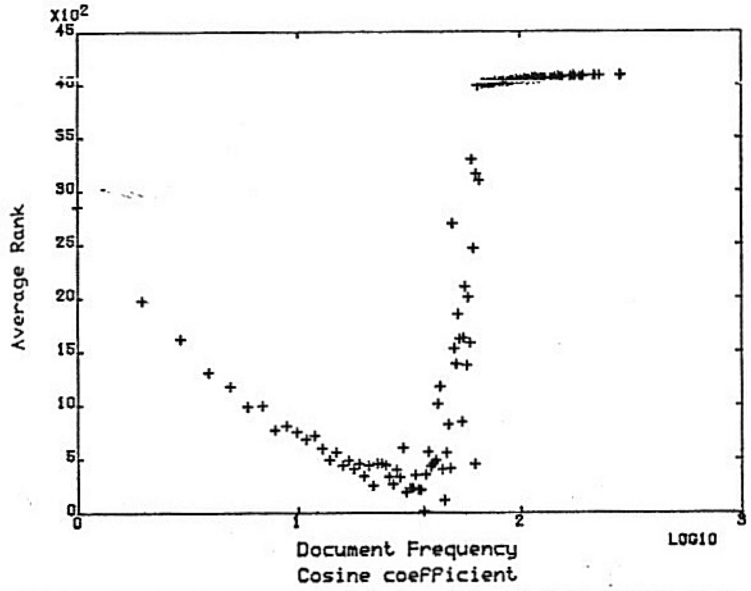

Fig. 5. Relationship between term discrimination and term frequency using the cosine coefficient.

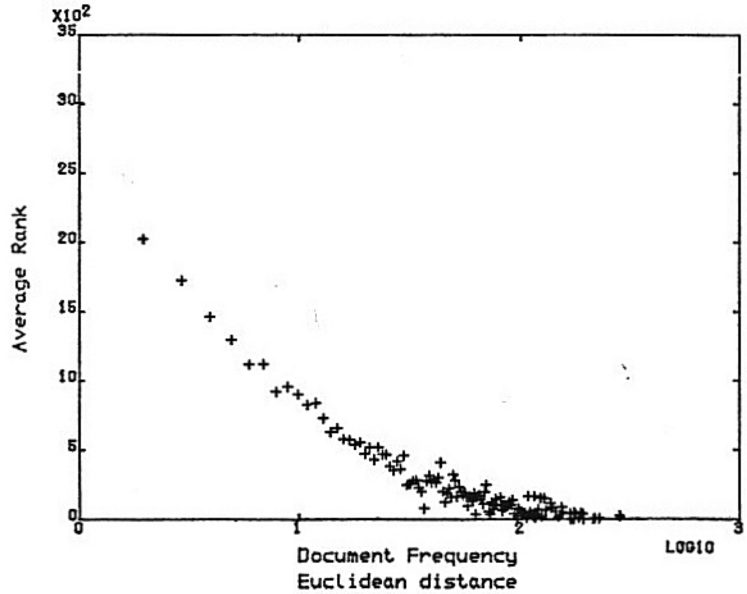

Fig. 6. Relationship between term discrimination and term frequency using the Euclidean distance.

terms in the collection were calculated, using algorithm C with the cosine coefficient, the dot product, and the Euclidean distance as the three measures of inter-document similarity. The relationship between discrimination and frequency was studied as suggested by Salton *et al.*[6]. After the discrimination values had all been calculated, the terms were sorted into order of decreasing discrimination value, so that the most highly discriminating term was given rank 1 and the least discriminating term the rank 4091. The average ranks were then calculated for all of the terms with frequency 1, 2, 3 . . . and a plot obtained of average rank against term frequency. The three plots are shown in Figs. 4–6, where it will be seen that a well-marked minimum is obtained with the cosine coefficient, while the dot product and Euclidean distance result in plots that increase or decrease monotonically as the term frequency increases.

4. CONCLUSIONS

An algorithm has been described for the calculation of term discrimination values in document retrieval systems: it is sufficiently fast in operation to permit the use of exact term discrimination values, rather than the approximate values used in previous studies. An analysis of the algorithm is presented which suggests that the relationship between term discrimination and term frequency is crucially dependent upon the inter-document similarity measure used for the calculation of the discrimination values.

Acknowledgements—Thanks are due to INSPEC for providing data, and to Prof. M. F. Lynch and Mr. L. G. Hendry for helpful discussions.

REFERENCES

[1] K. SPARCK JONES and R. G. BATES, *Research on Automatic Indexing*. Two volumes. Computer Laboratory, University of Cambridge (1977).

[2] G. SALTON and M. J. McGILL, *Introduction to Modern Information Retrieval*. McGraw-Hill, New York (1983).

[3] K. BONWIT and J. ASTE-TONSMANN, Negative dictionaries. In Scientific Report ISR-21, Department of Computer Science, Cornell University (1970).

[4] G. SALTON, *A Theory of Indexing*. Society for Industrial and Applied Mathematics, Philadelphia (1975).

[5] R. G. CRAWFORD, The computation of discrimination values. *Information Processing and Management* 1975, **11**, 249.

[6] G. SALTON, C. S. YANG and C. T. YU, A theory of term importance in automatic text analysis. *Journal of the American Society for Information Science* 1975, **26**, 33.

[7] G. SALTON and A. WONG, On the role of words and phrases in automatic text analysis. *Computers and the Humanities* 1976, **10**, 69.

[8] G. SALTON, A. WONG and C. S. YANG, A vector space model for automatic indexing. *Communications of the ACM* 1975, **18**, 613.

[9] S. A. PERRY and P. WILLETT, A review of the use of inverted files for best match searching in information retrieval systems. *Journal of Information Science* 1983, **6**, 59.

[10] T. NOREAULT, M. KOLL and M. J. McGILL, Automatic ranked output from Boolean searches in SIRE. *Journal of the American Society for Information Science* 1977, **28**, 333.

[11] M. F. PORTER, Implementing a probabilistic information retrieval system. *Information Technology: Research and Development* 1982, **1**, 131.

[12] S. E. ROBERTSON, Term frequency and term value. *ACM SIGIR Forum* 1981, **16**, 22.

[13] M. F. PORTER, An algorithm for suffix stripping. *Program* 1980, **14**, 130.