Information Technology: Research and Development (1982), *1* (189–196)

A PROCEDURE FOR THE ESTIMATION OF TERM SIMILARITY COEFFICIENTS

T. NOREAULT

University of Pittsburgh, Pittsburgh, Pennsylvania, USA

AND

R. CHATHAM

Bolt, Beranek and Newman, Cambridge, Massachusetts, USA

(Received 22 December 1981, revised 9 March 1982)

ABSTRACT

Term or document clustering has been shown effective in enhancing the performance of information retrieval systems. One barrier to the incorporation of these clustering techniques into operational systems has been the high cost of calculating the matrix of similarity measures. This paper suggests an approximation procedure which greatly reduces the effort expended in such calculation. The reduction is accomplished by estimation of the similarity measures. The procedure results in a substantial reduction in computational effort and a low error in the estimated similarity measures.

1. INTRODUCTION

Willett (1981) has demonstrated an efficient technique for the calculation of either document–document or term–term similarity measures. His algorithm provides substantial savings over either the brute force approach or a technique suggested by Croft (1977). The brute force approach requires the calculation of $N(N-1)/2$ similarity coefficients and is obviously not satisfactory when the N is very large. Croft pointed out that the vast majority of the coefficients were zero, and suggested an approach which avoids calculating the zero elements. Utilizing an inverted file Croft's algorithm only calculates the similarity coefficients which will be non-zero. The weakness with Croft's approach is that some, possibly many (Harding and Willett, 1980), of the coefficients will be calculated more than once. Croft's algorithm calculates the similarity measure every time the terms co-occur in a

document. Hence, if the terms co-occur frequently the coefficient will be calculated many times.

Willett's algorithm eliminates all redundant calculations of similarity coefficients. This results in substantial savings when the indexing exhaustivity (number of terms posted to a document) is high. This algorithm will be discussed in greater detail in the next section. In addition to there being a high number of zero coefficients, a large percentage of the coefficients which are greater than zero are insignificant (very low similarity coefficients). The procedure suggested in this paper will allow the estimation of the similarity measure in a manner which eliminates the need to calculate many of these small coefficients. The estimation procedure provides a substantial saving in the number of coefficients calculated while only introducing a very small error rate.

2. WILLETT'S ALGORITHM

All discussion of the algorithms will be in terms of calculating term–term coefficients. There are only minor differences between calculating these and document–document coefficients. These differences will be discussed at the end of this section. To calculate the similarity coefficients for a single term (X), Willett's algorithm uses the inverted file to identify all of the documents in which X occurs. Each of these documents can then be thought of as a vector, with each component representing the presence or absence of a term in that document. Summing up all of these vectors produces a vector in which each component represents the number of times that that term (represented by that position in the vector) co-occurs with term X (assuming binary data). The change to use frequency data is relatively minor.

The difference between calculating term–term as opposed to document–document similarities is in the nature of the vectors. Document vectors are relatively constant in length, whereas term vectors have a great deal of variability in their length, given that the number of postings for terms follows a Zipfian distribution. In Willett's algorithm, these differences are unimportant since both vectors must be used. The estimation procedure suggested in this paper performs most effectively with term–term vectors, but there are variations which should perform well with document vectors.

Willett suggests several alternative ways of implementing the summing of the vectors: linked list, hashing, and direct addressing. All of the work done in this paper was done using a linked list. The linked list was chosen for two reasons:

1. The size of the machine, DECsystem KA-10, required the smallest possible representation.
2. The estimation techniques keep the size of the problem small enough so that the linked representation will be most efficient.

3. DATABASE

The database was supplied by the Institute of Electrical Engineers and consists of 10885 documents from Computer and Control Abstracts. There are 15961 unique terms with the average term being posted to 21.5 documents. The terms, selected from the title and abstract, were stemmed (see Tars, 1976) and a short stop list was

used to remove common terms. The average document had 39.9 unique terms posted to it. The distribution of terms according to frequency of their postings can be seen in Table 1.

Table 1. Distribution of terms by frequency of postings

Postings	*No. of terms*	*Sum of postings*
1–29	14314	50289
30–44	346	12704
45–59	230	11917
60–199	683	75680
⩾200	388	193058
Total	15961	343648

As shown in Table 1, a very small percentage of the unique terms account for a large number of the occurrences of terms in the database. For example, those terms posted to 200 or more documents account for only 2 per cent of the unique terms but represent 56 per cent of the occurrences of a term in a document. This property will prove very useful in developing the estimation procedure.

4. ESTIMATION PROCEDURE

Both Croft's and Willett's algorithms make use of the fact that only a small fraction, 1.7 per cent for the database used in this study, of the coefficients are greater than zero. Croft improves on the brute force technique by calculating none of the zero coefficients. Willett also calculates none of the zero coefficients, but additionally only calculates each coefficient once. The estimation procedure given in this paper provides a further reduction, in the number of coefficients calculated, by not calculating coefficients which would have very low similarity scores.

In calculating the similarity coefficients to be used in, for example, a clustering algorithm, one usually establishes a cutoff point below which a coefficient will be considered zero (i.e., van Rijsbergen and Croft, 1975). The cutoff point is established primarily to reduce the effort required of the clustering algorithm. If the very small coefficients are not used in the clustering, the calculation of these insignificant (below the cutoff point) coefficients then becomes wasted effort. The approach used in estimating the coefficients uses a sample of the available information to calculate an estimate for the coefficient. This sampling reduces the amount of effort since a number of the small coefficients do not make it into the sample.

Dice's coefficient was used for these experiments in order to be consistent with the work done by Willett. Most of the similarity measures can be calculated from the information needed to calculate Dice's coefficient, and work by McGill *et al.* (1979) suggests that most of the different similarity coefficients will function in a similar manner. The formula for the similarity between terms X and Y is:

$$\text{DICE} = 2*A/(LX+LY) \quad (1)$$

where LX and LY are the numbers of postings for terms X and Y, respectively, and A is the number of documents to which both terms X and Y are posted.

The estimation procedure can be best explained by an example. To calculate all of the coefficients for a particular term:

Suppose a term X is posted to 500 documents, making LX equal to 500. Willett's algorithm would take the sum of all 500 document vectors to get the value of A for each of the coefficients. The value of LX is known, and the value of each of the LYs can be found by a simple lookup in the inverted file. The approach we are suggesting is to take a sample of the document vectors, say of size SS. The randomly selected vectors included in the sample are then summed together to get SA (Sample A). An estimate of A (EA) is then obtained by $EA = SA*(LX/SS)$. The expression (LX/SS) can be thought of as the sampling factor.

The resulting EA turns out to be slightly biased. Experimentation demonstrated that the MIN(EA,(LY/2)) was a slightly better estimator of A. The reason for this is that there are many more coefficients below the cutoff point than there are above, so that a small percentage overestimation amounted to a large number of mistakes.

The revised formula used to estimate Dice was:

$$\text{EDICE} = 2*\text{MIN}(EA,(LY/2))/(LX+LY) \tag{2}$$

5. RESULTS

Deciding how to apply the sampling requires examination of the distribution of coefficients above the cutoff point. Two cutoff points will be examined, 0.1 and 0.2. These points were chosen to be low enough to show a bad case, since the higher the cutoff point the more effective the estimation procedure will be. The significant coefficients primarily occur among terms which are infrequently posted. Table 2 shows the relationship between postings and non-zero coefficients.

Table 2. Number of coefficients by frequency of terms

Postings	*No. of coeff.*	*No. > 0.1*	*No. > 0.2*
1–29	770121	87791	42882
30–44	155985	753	50
45–59	125927	283	42
60–199	569545	771	81
⩾200	605359	2033	71
Total	2226953	91631	43126

As can be seen from Table 2, the very frequent terms account for many of the non-zero coefficients but few significant coefficients. In fact, the simplest estimation scheme would be to carry out Willett's algorithm only for terms posted to less than 30 documents. This would amount to a 63 per cent reduction in the number of coefficients calculated, and a loss of less than 5 per cent of the significant coefficients (4.2 per cent and 0.6 per cent for cutoffs of 0.1 and 0.2, respectively). It is likely that the performance of this estimation procedure would prove satisfactory in many situations.

If the simplistic scheme described above proved to be unsatisfactory, then one could use the sampling estimation procedure described earlier. Results given here will describe the effectiveness of the estimation using samples of size 30 and 60. These sample sizes were chosen with two considerations in mind:

1. That the sample be large enough that it is likely to be representative.
2. That the number of vectors sampled be small enough that a real reduction in effort would result from the sampling.

Table 3. Number of coefficients calculated in sample

Posting	*Sample 30*	*Sample 60*
30–44	138466 (11%)	—
45–59	92887 (16%)	—
60–199	263164 (54%)	422962 (26%)
⩾200	158670 (74%)	248076 (59%)

The first number in each sample pair in Table 3 is the number of coefficients calculated in the sample. The second is the percentage reduction in the number of coefficients this represents from the number which would have been calculated using Willett's algorithm. Very substantial savings result from using the estimation procedure on highly posted terms.

Three consequences can result from the sampling estimation procedure. Coefficients can be assumed to be:

1. Insignificant when they are actually significant (lost).
2. Significant when they are actually insignificant (added).
3. Significant when they are significant (retained).
4. Insignificant when they are insignificant.

The last of these categories, those coefficients correctly assumed to be insignificant, accounts for the vast majority of the terms and will not be reported in the tables as it is more meaningful to examine what is happening to the significant category. The only insignificant terms which are of interest are those which are incorrectly estimated to be significant.

Table 4. Sampling size 30 with 0.1 cutoff

Postings	*No. lost*	*No. added*	*No. retained*
30–44	128	416	625
45–59	70	462	213
60–199	203	4191	568
⩾200	719	9593	1314
Total	1120	14662	2720

The performance of the sampling size of thirty with a cutoff point of 0.1,

primarily shows weakness in estimating too many coefficients as significant (Table 4). This is a small percentage of the insignificant coefficients, 0.6 per cent, but a large absolute number. The simplistic scheme compares quite favorably to the performance at this sampling size. The sampling loses only a slightly lower percentage of significant coefficients while estimating a large number of insignificant coefficients to be significant.

Table 5. Sampling size 30 with 0.2 cutoff

Posting	*No. lost*	*No. added*	*No. retained*
30–44	6	16	44
45–59	7	19	35
60–199	23	103	58
⩾200	18	123	53
Total	54	261	190

The performance of the sampling estimation is significantly better with a cutoff of 0.2 (Table 5). The major improvement is that only a very few terms are falsely assumed to be significant. There is also a reduction in the percentage of significant items lost. It seems likely to these investigators that anyone wishing to use similarity coefficients with a cutoff of 0.2 or above would find the performance of a sample of size 30 satisfactory.

Table 6. Sampling size 60 with 0.1 cutoff

Posting	*No. lost*	*No. added*	*No. retained*
60–199	147	548	624
⩾200	585	3681	1448
Total	732	4229	2072

As can be seen in Table 6, the larger sample results in more effective performance. Note that for a sample size of 60, sampling is conducted only for terms having more than 60 postings. There are fewer significant coefficients lost and also fewer insignificant coefficients estimated to be significant. This higher effectiveness requires that more coefficients be calculated than would be calculated with a sample of size 30.

Table 7. Sampling size 60 with 0.2 cutoff

Posting	*No. lost*	*No. added*	*No. retained*
60–200	16	42	65
⩾200	14	84	57
Total	30	126	122

The larger sample with the higher cutoff point (Table 7) provides the most effective performance of the variations of the estimation procedure which were investigated. The estimation procedure is more accurate the larger the coefficient it is trying to estimate. This is an important characteristic since the larger coefficients are the most important ones in terms of affecting performance of a retrieval system.

It is possible to use different sampling sizes with terms having different frequencies of occurrence. For instance, one plausible scheme would be to do the full calculation for any term with a posting of less than 30, a sample of size 30 for terms with postings of 30 to 60, and a sample of size 60 for all terms posted above 60. This approach yields a small error rate and a substantial reduction in the number of coefficients which must be calculated.

Table 8. Combined sampling cutoff 0.2

Posting	*Actual no.*	*No. calculated*	*No. added*	*No. lost*	*No. retained*
1–29	770121	770121	0	0	42882
30–59	281912	231353	13	35	79
⩾60	1174904	691038	30	126	122
Total	2226953	1692512	43	161	43083

With this combined sampling approach, we have achieved a 24 per cent reduction in the number of coefficients calculated. Less than 0.1 per cent of the terms estimated to be significant are actually insignificant. Only 0.4 per cent of the significant terms were lost. In addition, the average error for the estimated coefficients was —0.02, less than a 10 per cent error in estimation. The correlation between the estimated coefficients and the real values was 0.61. This estimation procedure is so effective because 99 per cent of the significant coefficients, those in the 1–29 range, were fully calculated and not estimated. The cost in terms of error rate seems justified in light of the reduction in effort.

6. DISCUSSION

The results of this study indicate that estimation from samples can be effectively applied to the calculation of similarity coefficients. A substantial reduction in the computational effort occurs with a marginal error rate. In addition to reducing the number of coefficients calculated, the estimation procedure also reduces the space requirements for the calculation of the similarity values. This reduction could prove important in implementing the algorithm on small machines or with very large databases. For very large databases, the direct addressing suggested by Willett would prove to be very expensive in terms of memory use.

The problem with applying the estimation procedure described in this work to document–document similarity coefficients is that the size of the document vectors is relatively constant. Remember, we used the number of postings for a term as a basis for deciding whether to estimate the coefficients or fully calculate them. The sampling for the document–document coefficients would not be a random sample of the term vectors, but a stratified one based on how frequently each term is posted.

Empirical work is needed to decide on the most effective sampling scheme for document–document coefficients.

The use of an estimation procedure has been demonstrated to be of value to the calculation of similarity coefficients. It should prove particularly attractive in operational systems since it allows control over the amount of effort to be expended in the construction of the similarity measures.

APPENDIX A
Pseudo-code for Estimation Procedure

```
FOR I: = 1 TO NUMBER-OF-TERMS DO              {FOR EVERY TERM}
    BEGIN
        LX: = GETDOCVECTOR (I,DOCVECT);  {GET POSTINGS FOR
                                                TERM (I)}
        UPPERBOUND: = LX;
        IF LX<30 THEN COUNT: = LX             {COMBINED SAMPLING
                                                SCHEME}
            ELSE IF LX<60 THEN COUNT: = 30
                ELSE COUNT: = 60;
        LISTHEAD: = NULL;
        FOR J: = 1 TO COUNT DO
            BEGIN
                X: = RANDOM (1,UPPERBOUND);          {SELECT A
                                                      DOCUMENT}
                DOC: = DOCVECT(X);
                SWITCH (DOCVECT(UPPERBOUND),DOCVECT(X));
                                                {SAMPLING WITHOUT
                                                  REPLACEMENT}
                UPPERBOUND: = UPPERBOUND – 1;
                DOCHEAD: = GETDOC (DOC);        {GET LINKED LIST
                                                  REPRESENTING
                                                  DOCUMENT DOC}
                MERGE (LISTHEAD,DOCHEAD);
            END;
    WHILE LISTHEAD<>NONE DO
        BEGIN
            LY: = GETLENGTH (LISTHEAD.TERM);    {POSTINGS FOR
                                                  TERM Y}
            A: = LISTHEAD.COUNT;
            IF LX>29 THEN A: = MIN (A,(LY/2));      {ESTIMATION
                                                      OF A}
            DICE: = (2*A)/(LX + LY);
            LISTHEAD: = LISTHEAD.NEXT;
        END;
    END;
TYPE NODEPOINTER = NODE;
    NODE = RECORD          {NODE FOR LISTHEAD AND DOCHEAD}
            TERM: STRING;
            COUNT: INTEGER;
            NEXT: NODEPOINTER;
            END;
```

The variables LISTHEAD and DOCHEAD are pointers to the head of the term and document lists, respectively. The procedure MERGE, adds the linked list pointed to by DOCHEAD to LISTHEAD. Nodes are equal if they contain the same term. If a node in DOCHEAD is equal to a node in LISTHEAD then add one to the node in LISTHEAD. If a node in DOCHEAD is not in LISTHEAD add the node to LISTHEAD. Both DOCHEAD and LISTHEAD are in sorted on TERM.

REFERENCES

Croft, W. B. (1977) Clustering large files of documents using the single-link method. *Journal of the American Society for Information Science 28*, 341–344.

Harding, A. F. and Willett, P. (1980) Indexing exhaustivity and the computation of similarity matrices. *Journal of the American Society for Information Science 31*, 298–300.

McGill, M., Koll, M. and Noreault, T. (1979) *An evaluation of factors affecting document ranking by information retrieval systems.* NSF Grant NSF-IST-78-10454 (final report, Oct).

Tars, A. (1976) *Stemming as a system design consideration.* ACM Sigir Forum, Spring, 9–16.

Van Rijsbergen, C. J. and Croft, W. B. (1975) Document clustering: an evaluation of some experiments with the Cranfield 1400 collection. *Information Processing and Management 11*, 171–182.

Willett, P. (1981) A fast procedure for the calculation of similarity coefficients in automatic classification. *Information Processing and Management 17*, 53–60.