The automatic identification of stop words

W. John Wilbur and Karl Sirotkin

National Center for Biotechnology Information, Bethesda, MD, USA

Received 19 September 1991; Revised 9 October 1991

Abstract. A stop word may be identified as a word that has the same likelihood of occurring in those documents not relevant to a query as in those documents relevant to the query. In this paper we show how the concept of relevance may be replaced by the condition of being highly rated by a similarity measure. Thus it becomes possible to identify the stop words in a collection by automated statistical testing. We describe the nature of the statistical test as it is realized with a vector retrieval methodology based on the cosine coefficient of document-document similarity. As an example, this technique is then applied to a large MEDLINE® subset in the area of biotechnology. The initial processing of this database involves a 310 word stop list of common non-content terms. Our technique is then applied and 75% of the remaining terms are identified as stop words. We compare retrieval with and without the removal of these stop words and find that of the top twenty documents retrieved in response to a random query document, seventeen of these are the same on the average for the two methods. We also examine the differences and conclude that where the user prefers one method over the other, the new method with the reduced term set is favored about three times out of four.

1. Introduction

When a document collection is indexed for retrieval by considering each individual term as a potential index term for the documents in which it occurs, it is common practice to remove from consideration a so-called stop list of common functional terms. These terms, such as "the", "if", "but", "and", etc., have a grammatical function but reveal nothing about the content of documents (see [11], p. 279). By removing such terms one can save space in indices and also, in some cases, may hope to improve retrieval. The most frequently occurring stop terms are fairly obvious and relatively few in number. It is clear however that many terms that appear in textual material have more of a syntactic than a semantic role and are not useful indicators of content. It is evident that these noncontent words may be identified by what is an essentially random occurrence in the database, i.e., a word is a stop word if its occurrence in a query and a document is unrelated to whether the document is relevant to the query or not. Our proposal is to replace the concept of relevance by the condition of being highly rated by some objective measure of query–document similarity. To be more precise suppose D is a database and Q is a set of queries we wish to apply to D. Suppose we are given some method of measuring the relatedness of documents in D to the queries in Q. Then we would say:

Stop Word Criterion. The term t is a stop word if it contributes to the production of highly rated pairs (q, d) from $Q \times D$ no more frequently than would be expected on a random basis, given the frequency of t in both Q and D.

It has been for some time our concern to discover the best methods to cluster documents and/or find the nearest neighbors of documents in selected subsets of the Medline database. Consequently we have chosen to study this criterion for stop words in the setting where the set of queries is the database, i.e., $Q = D$. Advantages are a uniform and well defined set of queries and a simplification of the stop word criterion. As our measure of relatedness we employ the cosine coefficient of document–document similarity based on the vector retrieval model. This is chosen because it is known to have good performance and is easily implemented automatically.

In Section 2 we give the details of the vector retrieval model that we use for the calculation of similarity scores between documents. In Section 3 this vector model is used to implement the stop word criterion as a statistical test. An example applying this criterion to a real database of Medline records is given in Section 4. Characteristics of the identified stop words are given. In Section 5 the effect on vector retrieval of the removal of

Correspondence to: W.J. Wilbur, National Center for Biotechnology Information, National Library of Medecine, National Institutes of Health, Building 38A, 8600 Rockville Pike, Bethesda, MD 20894, USA

Journal of Information Science 18 (1992) 45–55
Elsevier

all stop words for this database is described. In particular one finds that where differences in the documents retrieved exist, judgments based on their comparison tend to favor the new method with the reduced term set [1].

2. The vector model

We have used a version of the basic vector space model of information retrieval largely developed by Salton [8,10,11], Salton et al. [9] and Wong and Raghaven [4]. The vector space model assumes that each document may be represented by a vector whose nonzero coordinates correspond to the terms contained in that document. In this way two documents may be compared by comparing the vectors derived from them. More formally one assumes a collection of documents D and a collection of terms T obtained from the set of documents. Each term in the set T is taken to correspond to a dimension in term space, $V(T)$. Here $V(T)$ is the standard vector space over the real numbers of dimension $\|T\|$ ($\|T\|$ denotes the number of terms in T). It is not difficult to see that one may associate with each document, d, in D a vector v_d in $V(T)$. One simply has to specify how, for each term t, the corresponding coordinate of v_d is determined by d and t. Let this coordinate be denoted v_{dt}. If the document d does not contain the term t then one sets v_{dt} equal to zero. If, however, the term t occurs in d, it has been found very useful to define v_{dt} as the product of two weights (see [2] and [11], p. 280)

$$v_{dt} = lw_{td} \times \sqrt{gw_t} \tag{1}$$

The global weight, gw_t, is intended to measure the term t's overall importance in the collection, for the purpose of relating documents or retrieving information. Let N stand for $\|D\|$ (the number of documents in D) and let n_t stand for the number of documents in D which contain the term t for a given t. It is common to define gw_t as the inverse document frequency

$$gw_t = \log_2(N/n_t) \tag{2}$$

[1] Our programs are written in C and calculations were performed on the ISI 3090-300J computer facility and on a Sun 4/60 at the National Institutes of Health.

With this definition, gw_t may be understood as the number of bits of information conveyed about document d when one is told that term t occurs in d. The more infrequently t occurs the greater the information gw_t. The local weight lw_{td} is intended to measure the importance of the term t in the particular document d. As such lw_{td} is appropriately some function of the frequency of the term t in d. We employ the augmented normalized term frequency [1–4]

$$lw_{dt} = 0.5 + 0.5 \times \left(\frac{m_{dt}}{m_d} \right) \tag{3}$$

Here m_{dt} represents the number of times the term t occurs in the document d and m_d is the maximum of m_{dt} over all the terms t that occur in d. The normalization and augmentation in (3) are helpful in decreasing the discrepancy that can occur in word weighting between documents simply because of variation in document length, while still allowing the more frequent words in a document to play a stronger role in its representation.

Given documents d and e, we now have associated document vectors, v_d and v_e, and we define the document similarity by

$$sim(d, e) = \frac{v_d \cdot v_e}{lv_d \times lv_e} \tag{4}$$

This is the so-called cosine coefficient of similarity [11, p. 318]. Here $v_d \cdot v_e$ is the familiar dot product of vector analysis and $l(v_d)$ stands for the length of v_d and is given by the formula

$$l(v_d) = \sqrt{v_d \cdot v_d} \tag{5}$$

Thus $sim(d, e)$ is the cosine of the angle between the vectors v_d and v_e in the Euclidean vector space $V(T)$. The numerator in (4) is the sum of the products of vector components of the form v_{dt} times v_{et} where the term t occurs in both d and e. Such a product has gw_t as a factor to the first power.

Given a document d, the general approach to finding the documents in D that are close to d in content is to compute the similarity coefficient $sim(d, e)$ for all documents e in D and rank the documents in order by the size of $sim(d, e)$. Thus relatedness to d will progressively decrease in going from the top (largest value of $sim(d, e)$), to the bottom of the ordering.

3. The detection of stop words

We have observed that when two documents are found to have a large number of words in common the individual words vary in their frequency of use throughout the database. Either they are very frequent in the database so that many pairs of documents are expected to have them in common, or they tend to be rare, making their common occurrence unexpected. The greater the degree to which a word is shared in pairs of documents above the frequency expected, the more likely such a word reflects the distinctive content of documents. Conversely, as expressed in the stop word criterion of Section 1, if a word is shared in pairs of documents with a frequency accounted for by word-frequency alone, that word is a stop word.

Our task here is to give this criterion a precise expression in terms of the vector model and the notation of the previous section. Let us choose a number m which is to be a lower bound for a similarity score to be considered significant. In the language of the stop word criterion then, a document pair will be considered highly rated if and only if the similarity score is greater than or equal to m. We define the *strength* of a term to be the probability that when it occurs in one member of a highly rated pair, then it also occurs in the other member. For a term t we will denote the strength of t by $s(t)$. Thus $s(t)$ is the probability that given a pair of documents d and e with $d \neq e$, t occurring in d, and

$$sim(d, e) \geq m, \tag{6}$$

then t occurs in e also. While it is computationally intensive, one may compute $s(t)$ for any term t in a straightforward manner. Now let t_r denote a term with the same frequency of occurrence, n_t, in the database but assume that t_r is random in its occurrence in the database, i.e., it is distributed without regard to the similarity scores of document pairs. For such a random term t_r we may estimate the expected strength, $E(s(t_r))$, and the standard deviation of the strength, $S(s(t_r))$. The strength $s(t)$ is to be compared with the result for the randomized term t_r and we consider a term t to be a stop word if

$$s(t) \leq E(s(t_r)) + 2S(s(t_r)). \tag{7}$$

For a given t, the values of $E(s(t_r))$ and $S(s(t_r))$ are obtained by simulating values of $s(t_r)$ in the database D. In order to simulate the value $s(t_r)$ as a realistic estimate of what $s(t)$ would be were the occurrence of t random in the database, we consider t_r to be distributed in the database identical to t, i.e., in the set S_t of the n_t documents containing t. The difference between t and t_r becomes evident only on considering pairs of documents. If d is a document in S_t and $sim(d, e) \geq m$ then we disregard whether t is in e or not and decide whether t_r is in e randomly based only on the frequency n_t. Thus t functions to supply the number of pairs to consider with t_r in the first member and a random choice must then decide whether t_r is in the second member of these pairs. This is the basic idea of the simulation. The details of the simulation are contained in the Appendix.

4. Application to a Medline database

We here apply the method outlined in the previous section to the identification of stop words in a database consisting of 71,311 records from Medline in the area of molecular biology/genetics. All records have titles and abstracts and the set of key terms, T, is obtained from titles and abstracts while the MESH® headings are ignored. This is done so that the results we present may be considered relevant to databases of a more general nature. This particular data set was studied because it is of interest in other ongoing projects in which we are involved. It must also be noted that because it is a large database in which an average document has multiple other related documents it is an ideal setting in which to apply the stop word criterion.

There are two steps in the processing of text to produce T which deserve mention. First, a stop list of 310 common words is eliminated at the outset. This is fairly standard practice and anything we identify as a stop word must of course have evaded this list. Second, we process all terms through an implementation of the Porter [5] stemming algorithm. The result of this processing is a set, T, of 203,040 terms.

For all those terms in T which occurred in at least one member of some highly rated pair, ($R_0 \neq \emptyset$, see Appendix) we simulated the strength

of a related random term 10,000 times to obtain the necessary expectation and standard deviation to apply the stop word criterion as defined by (7). Of those terms occurring more than once in the database this included all but 352 terms that occur in at most three documents each in the database, and four terms that occur in at most seven documents each. All other multiple occurrence terms qualified for processing. We based the simulation on an m of 0.21 which resulted in roughly twenty documents e which satisfied $sim(d, e) \geq m$ for each d in the database. This value of m was chosen because our prior experience with this particular database suggests that when a document is used as a query on this database there are on the average about twenty other documents that are relevant to the query. Our aim of course is to approximate the relevance relation with the set of highly rated pairs.

For the purpose of describing the results we shall refer to the number

$$[\log_2(n_t)] \tag{8}$$

as the *frequency class* of the term t which occurs with frequency n_t. The frequency class is the greatest integer not greater than the log to the base two of the frequency of t and is always nonnegative. Fig. 1 shows the average values, in the different frequency classes, of $s(t)$ and $E(s(t_r))$. The average value for $p(t)$ defined by

$$p(t) = \frac{n_t}{N} \tag{9}$$

is also shown for comparison purposes. While the average value of $p(t)$ appears similar to that of $E(s(t_r))$ this is not so in the relative sense. The average value of $p(t)$ starts out orders of magnitude smaller than that of $E(s(t_r))$ on the left side of the picture and is still less than one half for frequency class nine. Not shown in the Fig. 1 is the average value of $S(s(t_r))$ for each class. This value starts out at approximately 0.007 for class one and slowly decreases to approximately 0.001 by the time one reaches class fifteen. At class nine it is about one tenth of the average value of $E(s(t_r))$. One may conclude that the average value of $p(t)$ would not be a useful substitute for that of $E(s(t_r))$.

While stop terms are classically limited to be among only the most frequent terms, we find stop terms which cover virtually the whole frequency range. Truncation in the distribution is only ob-

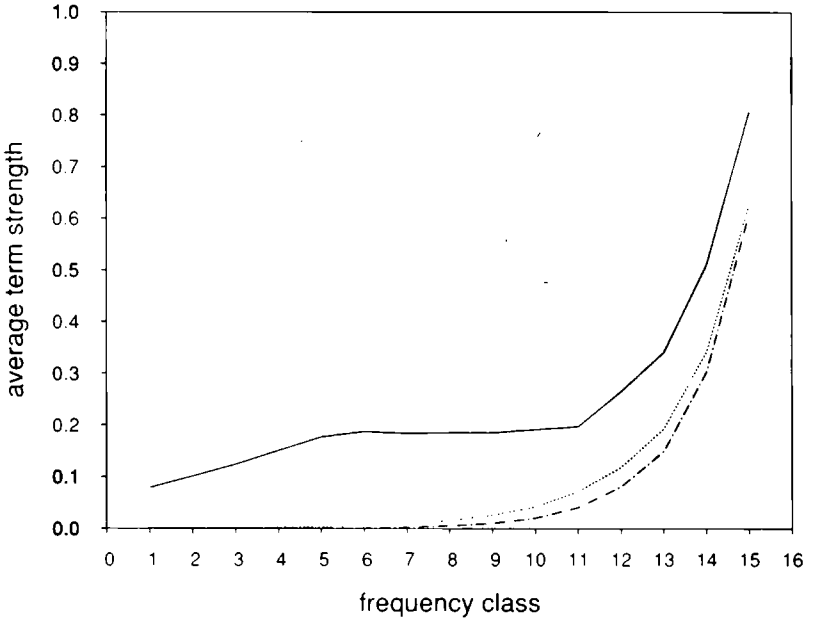

Fig. 1. The three curves shown here are all computed as averages over all the terms in each class for which $s(t)$ is defined. For the classes from one to fifteen this includes all but 356 terms which are in classes one and two. The average of $s(t)$ is shown as the solid curve. The dotted curve is the average of $E(s(t_r))$. The alternate dotted-dashed curve gives the average of n_t/N for comparison purposes.

served at the highest frequencies because at this level potential stop terms have already been placed on the stop list we use in pre-processing to produce T. The number of terms in each of the frequency classes and the number of stop words found in each of these classes in three different categories (strength zero, non-zero but within σ of the average random strength, or between one and two σ's above the average random strength) are given in Table 1. In frequency classes one to fifteen there are 27,324 stop words (total of numbers above class zero in columns 3, 4 and 5 in Table 1) accounting for 383,588 term occurrences in the database (sum of all the frequencies of occurrence of all the stop words above class zero in Table 1). This together with the 125,208 single term occurrences in the database corresponding to the 125,208 terms in class zero that trivially satisfy (7) makes a total of 508,796 term occurrences in records that may be eliminated by eliminating the stop words. There are 50,508 terms that remain with 4,881,880 occurrences in the database. Thus one obtains about a 10% space saving by elimination of the stop words.

There are twelve stop words that occur in the classes twelve to fourteen. We list these in Table 2 along with the eleven stop words in class three in the third grouping (see Table 1), and ten randomly chosen stop words from class one. These are shown to illustrate the kind of words that are found to be stop words. Generally we recognize the more frequent stop words as familiar words that convey nothing about content. The less common, to rare, stop words are more likely to be idiosyncratic constructions from common elements which themselves might have some significance for content. However, their rarity far outweighs the significance of the common element or elements from which they were composed.

Table 1
List of numbers of terms by class and the numbers of stop words in each class as computed by (7) divided into three mutually exclusive groups

class	no. of terms	$s(t) = 0$	$0 < s(t) \le E + S$	$E + S < s(t) \le E + 2S$
0	125,208	125,208	0	0
1	37,192	20,133	0	0
2	16,791	4,814	0	0
3	9,159	1,241	0	11
4	5,417	255	51	241
5	3,330	35	105	85
6	2,177	1	85	40
7	1,356	0	44	21
8	885	0	31	15
9	639	0	45	11
10	378	0	26	1
11	246	0	17	4
12	176	0	6	2
13	60	0	3	0
14	24	0	1	0
15	2	0	0	0

Table 2
List of the twelve most frequent stop words that occur in the classes twelve to fourteen, the eleven of the third grouping in class three, and ten randomly chosen stop words from class one (see Table 1). The frequency of occurrence, strength, and the mean and standard deviation for a similar randomly occurring term is given preceding each term. Note that all terms have been stemmed by the Porter [5] stemming algorithm

n_t	$s(t)$	$E(s(t_r))$	$S(s(t_r))$	t
20465	0.336	0.335	0.001	result
12202	0.216	0.226	0.001	studi
9095	0.167	0.178	0.001	observ
8931	0.174	0.173	0.001	presenc
7749	0.160	0.159	0.001	data
6638	0.128	0.135	0.001	well
5613	0.127	0.125	0.001	follow
5907	0.113	0.128	0.001	possibl
4907	0.104	0.111	0.001	provid
4819	0.121	0.120	0.002	properti
4627	0.096	0.104	0.001	signific
4175	0.096	0.095	0.001	contrast
15	0.0024	0.0005	0.0011	11-base
14	0.0040	0.0007	0.0017	20-amino-acid
15	0.0041	0.0008	0.0018	35k
15	0.0053	0.0010	0.0022	6-kbp
15	0.0048	0.0009	0.0020	700-base
15	0.0055	0.0010	0.0023	different-s
15	0.0051	0.0009	0.0021	murphi
15	0.0069	0.0012	0.0029	p18
15	0.0063	0.0011	0.0026	promoterlik
15	0.0036	0.0007	0.0016	t-lymphoid
15	0.0051	0.0009	0.0021	tail-to-tail
2	0	0.0007	0.0145	4-di-o-benzyl-alpha-l-rhamnopyranosid
2	0	0.0004	0.0073	colombo
3	0	0.0004	0.0049	d1313
2	0	0.0001	0.0014	glutaminyl-glycin
3	0	0.0008	0.0127	phytochrome-regul
3	0	0.0006	0.0086	quinoa
2	0	0.0004	0.0063	testis-deriv
3	0	0.0023	0.0276	vacanc
2	0	0.0003	0.0057	viral-associ
3	0	0.0005	0.0058	wu-kabat

This also tends to be true of the large class of terms that occur only once in the database and it provides some justification for their identification as stop words (beyond the trivial satisfaction of the inequality (7)).

5. Effect of stop word removal on retrieval

In the previous section we described the identification of stop words in a set, D, of 71,311 Medline documents in the area of biotechnology, based on the automatic procedure proposed in this paper. Here we discuss the consequences this may have for identifying close document neighbors by vector retrieval in the same set. For this purpose let us denote as R (the reduced set) the set of terms that remain after all stop words have been removed from T. Then we wish to compare vector retrieval based on T and R with documents themselves as queries.

To compare vector retrieval by T and R we took a random sample, W, of 2000 documents from D. For each document in W we retrieved the top 20 documents by T and R retrieval. We then compared the results of the retrieval by the two methods for each query in W. We found agreement on about 85% of the documents retrieved by the two methods for each depth from one to twenty. The extent of the agreement between the two methods at depths five and twenty is shown in Fig. 2 in frequency histograms. We performed a similar analysis to compare retrieval by the full set of terms, T, with retrieval when only the stop words in class 0 were removed and when only the stop words in classes 1–14 were removed. In both cases the level of agreement was found to be about 90% at all depths from rank one to rank twenty (data not shown). From the opposite point of view about 15% of the documents retrieved by T and R are different. In examining this disagreement we found that for all query documents in the set of 2000 there was at least one disagreement between T and R by the time one had compared documents in the two retrievals down to rank twelve (of the twenty possible). Fig. 3 shows a frequency histogram of the ranks where disagreement first takes place

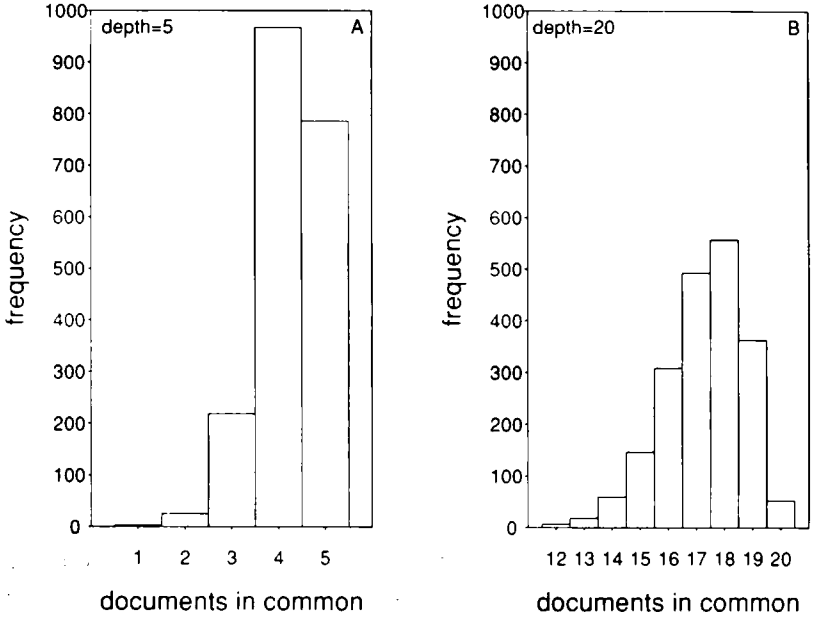

Fig. 2. Panel A shows the frequencies with which queries retrieved from zero up to five documents in common in the top five ranks with retrieval based on the full term set compared with retrieval based on the reduced term set (stop terms removed). Note that the methods retrieve at least one document in common in the top five ranks for all 2000 queries. Panel B shows a similar analysis for the top twenty ranks and at least twelve documents are found in common for all queries.

over the set W of queries. One notes that the first disagreement is most likely to occur at the second ranked document.

In order to further compare the performance of vector retrieval by T and R we extracted for each query in W the pair of documents which corresponded to the first disagreement between T and R in their retrieval by that query. Thus if T and R disagreed on the top documents they retrieved, these would form the pair in question, while if not one would look further down the retrieval order for this first disagreement. We then arranged a system which allowed one to judge which of each pair of documents was most relevant to the query document. This allowed the judge to be presented with three windows on a computer screen, one containing the query document (title, abstract, and MESH terms), and the other two each containing one of the retrieved documents. The windows for the retrieved documents were marked 1 and 2 but the two retrieved documents were placed randomly in these windows and the computer then assigned the operator's judgment to the correct category of its origin in recording the judgment. In this way any bias based on a knowledge of the origin of the documents to be judged was prevented.

Table 3
Independent judgments of document pairs to determine which is most relevant to the query document. P-values are calculated as the probability that a series of judgments would favor the R method as much or more than was observed given the null hypothesis that the judge could not distinguish one source of documents as better than the other.

	Favored T	Favored R	p-value
First 100 triples			
Author 1	7	**30**	0.0001
Author 2	12	**44**	0.00001
Graduate student	19	**30**	**0.08**
First 201 triplets			
Graduent student	35	**89**	7×10^{-7}

With the technique just described the first and second authors of the paper evaluated the first 100 triplets of documents arising from the random sample W. The results are shown in Table 3. These results strongly favor R-based retrieval. In comparing our results it was of interest to note that we agreed in favoring method R on only 17 triplets and agreed in favoring method T on only 2 triplets. This is essentially a random level of agreement. This raised the possibility that we might have had some bias in making our judgments just from a knowledge of the two different

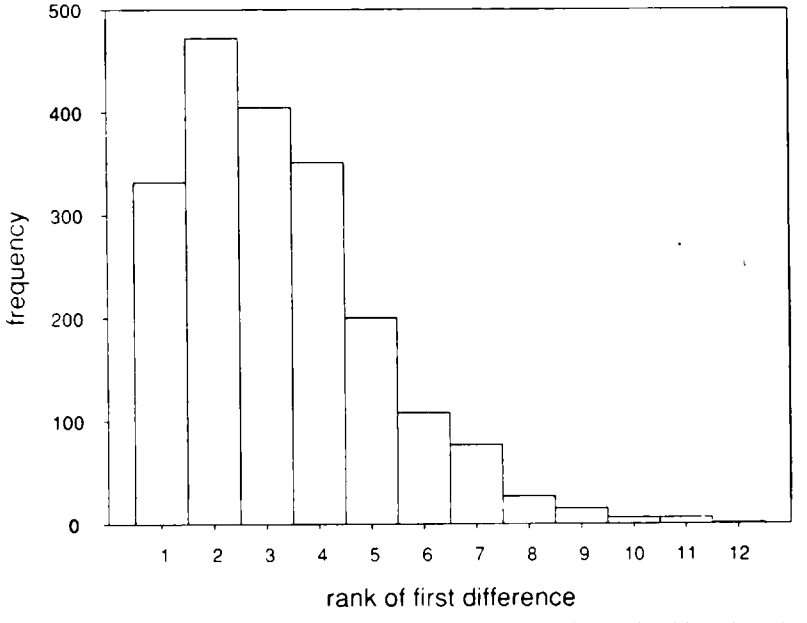

Fig. 3. A comparison of retrieval based on the full term set with retrieval based on the reduced term set and showing the first rank at which a difference was observed and its frequency of occurrence for the 2000 queries. Note that all queries had produced a difference al least by the time one moved down the retrieval order to rank twelve (of the twenty ranks considered).

methods used in the retrieval, even though we did not know which document had been retrieved by which method. To investigate such a possible effect a graduate student in molecular biology was hired to repeat our judgments. The student was not told anything about how the two documents were retrieved and carried out the judgments using the same blinded method we had used. The results are again shown in Table 3. Because the results had not reached statistical significance at the 5% level the student was asked to continue for another 100 judgments. Again nothing was told the student except that the results had not reached significance and more judgments were needed. The student performed a total of 201 judgments and the results strongly favor the method based on R. The results of the student's judgments on the first 100 triplets were again noted to be relatively independent of the judgments made by ourselves. We all three agreed in favoring T on only one triplet and in favoring R on only eight triplets. The fact that different judges are nearly independent in their judgments is consistent with the known high level of disagreement found when relevance judgments made by different individuals are compared (see [6] and [13]).

These results rather clearly favor retrieval based on the reduced set R of 50,508 terms over retrieval based on the full set T of 203,040 terms. While three out of four terms are identified as stop words and eliminated, only about 10% of the total number of term occurrences are thereby eliminated. This means that the stop words tend to be relatively low frequency words and hence by (2) to have relatively high weights. Thus in those documents where they occur they may have a large effect on the similarity scores. When this effect is removed about 15% of the documents retrieved in the top 20 are different (see Fig. 2) and our results from triplet judgments suggest that the new documents obtained in this way will be preferred to the old by a ratio of about three to one in the roughly 50% of cases where one method is preferred.

6. Conclusion

We have presented an automatic method of identifying a high percentage of the words in documents as stop words and when such words are eliminated the results of vector retrieval are not degraded but rather appear to be improved. This was accomplished by replacing the concept of relevance by the condition of being highly rated by an objective document–document similarity measure. This is possible, first because the vector measure is of sufficient accuracy and, second because the database used, i.e. Medline, is a large and richly interconnected data set on which it is not difficult to obtain good statistics. We would expect these results to be reproducible on other large and richly interconnected databases. Further the measure of similarity between documents need not be the vector method but can be any method that is successful in distinguishing the relevant pairs from the nonrelevant a high percentage of the time. Even if one only detects half of the occurrences where a term functions in relating relevant pairs, this is often sufficient to distinguish it above random. Of course, a method that is more efficient than the vector method in detecting relevant document pairs can only be more accurate in defining the stop words. We have not applied our method to a small database or one where few documents have other related documents occurring in the database. One may note however that if a term does not contribute significantly to the identification of highly rated pairs of documents, then the removal of that term is not likely to degrade this identification process.

The approach to finding stop words that we have presented may appear limited in its applicability because it is based on document–document similarity. The question of stop words is clearly broader in that one wishes to know what terms are useful in the setting of general textual queries against the database. Likewise one must decide which terms should be indexed for the purpose of allowing Boolean retrievals. We believe that document–document similarities may provide a useful approach to both questions. If one accepts the cluster hypothesis of van Rijsbergen [7,11], then closely related documents tend to be relevant to the same queries. One suspects this is true for Boolean searches as well as those of a more general nature, as this is a consequence of the information need rather than the particular kind of query employed. Now suppose a word is a stop word by our definition, i.e., it exhibits only a random occurrence in related documents. If we

use this term in a Boolean query we might actually retrieve a document relevant to our query. However with only random occurrences in related documents the cluster hypothesis indicates that this word has only a random chance of occurring in a number of other relevant documents. It is then reasonable to conclude that it had only a chance occurrence in the relevant document we did retrieve and is perhaps more likely to exclude relevant documents than to lead to their retrieval. Such a word then should also be a stop word for Boolean retrieval. In the case of general textual queries one would expect the stop words we identify to remain appropriate stop words. However, the fact that word usage in such queries tends to involve the higher frequency terms in the database [12] may have a noticeable effect on the results observed. Because the stop words we identify are more abundant among the low frequency terms, the change seen in retrieval performance due to removal of such stop words may be less in magnitude than that we observed (Section 5 above) when documents are used as queries. The applicability of our method in this more general setting remains to be tested.

Acknowledgment

The authors would like to thank David Lipman and Dennis Benson for helpful discussions regarding this paper and Shirley Myers for making 201 triplet document comparisons used in this project.

References

[1] C. Buckley, Implementation of the SMART information retrieval system, Technical Report 85-686 (Department of Computer Science, Cornell University, 1985).

[2] C. Buckley and A.F. Lewit, Optimization of inverted vector searches. In: *Proceedings of the Eighth International ACM Conference on Research and Development in Information Retrieval* (Montreal, Quebec, 1985) 97–110.

[3] W.B. Croft, Experiments with representation in a document retrieval system, Technical Report 82-21 (COINS, University of Massachusetts, Amherst, MA, 1982).

[4] D. Lucarella, A document retrieval system based on nearest neighbor searching, *Journal of Information Science* 14 (1988) 25–33.

[5] M.F. Porter, An algorithm for suffix stripping, *Program* 14 (1980) 130–137.

[6] J.J. Regazzi, Performance measures for information retrieval systems: an experimental approach, *Journal of the American Society for Information Science* 39(4) (1988) 235–251.

[7] C.J. van Rijsbergen, *Information Retrieval*, 2nd ed. (Butterworths, London, 1979).

[8] G. Salton, *Automatic Information Organization and Retrieval* (McGraw-Hill, New York, 1968).

[9] G. Salton, A. Wong and C.S. Yang, A vector space model for automatic indexing, *Communications of the ACM* 18(11) (1975) 613–620.

[10] G. Salton and M. McGill, *Introduction to Modern Information Retrieval* (McGraw-Hill, New York, 1983).

[11] G. Salton, *Automatic Text Processing* (Addison-Wesley, Reading, MA, 1989).

[12] K. Sparck Jones, A statistical interpretation of term specificity and its application, *Journal of Documentation* 28 (1972) 11–21.

[13] D.R. Swanson, Historical note: information retrieval and the future of an illusion, *Journal of the American Society for Information Science* 39(2) (1988) 92–98.

[14] S.K.M. Wong and V.V. Raghavan, Vector space model of information retrieval: a reevaluation. In: C.J. van Rijsbergen, ed. *Research and Development in Information Retrieval* (Cambridge University Press, Cambridge, 1984) 167–185.

Appendix

Here we assume the setting and notations outlined in Section 3 and give the details of the simulation of values for $s(t_r)$ based on the properties of the term t, from which the expected strength, $E(s(t_r))$ and the standard deviation of the strength $S(s(t_r))$ are to be obtained. A subtle point is that we must account for the fact that t_r itself contributes to the production of highly rated pairs even when it occurs randomly in the database. Thus if d is a document in S_t and $sim(d, e) \geq m$, even on a random basis we will have t_r occurring in e with a higher probability than

$$p(t) = \frac{n_t}{N}. \tag{10}$$

To account for this fact requires a special approach. Then let R_0 be the set of pairs (d, e) with d in S_t, $d \neq e$, and which satisfy $sim(d, e) \geq m$ even when any contribution from t_r is disregarded. Likewise let R_1 be the set of pairs (d, e) with d and e in S_t, $d \neq e$, and which satisfy $sim(d, e) \geq m$ only if the contribution from t_r is counted. Finally let S_0 be the subset of R_0 con-

sisting of those pairs (d, e) which have e contained is S_t as well as d. Then, using $\| S \|$ to denote the number of elements in a set S, we may write

$$s(t_r) = \frac{\| S_0 \| + \| R_1 \|}{\| R_0 \| + \| R_1 \|}. \tag{11}$$

Now in order to simulate R_0, R_1 and S_0 we must be explicit about how t_r is to enter into the calculation of $sim(d, e)$. Calculations are made by the guidelines:

I. If t_r occurs in d this does not change the length $l(v_d)$, rather t_r is assumed to be identified with one of the elements that was already in d.

II. If t_r occurs in both d and e then $sim(d, e)$ is calculated by assuming that the occurrence of t_r in d replaces a term in d that was not in e and likewise the occurrence of t_r in e replaces a term in e that was not in the original d.

We have chosen these rules of calculation so as not to disturb the distribution of lengths of documents yet to as faithfully as possible represent the effect of the random occurrence of the term t_r in the database. According to these guidelines then, R_0 is just the set of all pairs (d, e) with d in S_t, $d \neq e$, and which satisfy $sim(d, e) \geq m$ when the occurrence of t_r is totally ignored in the calculation. As already noted the size of R_0 puts important constraints on the computation of $s(t_r)$ unrelated to our randomness concerns. The actual randomness in the simulation has to do with how S_0 and R_1 are chosen, or better, simulated: and this is our next consideration.

It will be convenient to let $S_t \times S_{t_r}$ denote the set of all document pairs which have t and t_r in the first coordinate but which have t_r in the second coordinate as a random occurrence. Note that this is a convention rather than a real object, because t_r cannot occur both just where t does and also in a random fashion just because we require this behaviour for our calculation. However, it is a useful convention for purposes of the calculation. Now we note that S_0 consists of those pairs from $S_t \times S_{t_r}$ which are in R_0 purely on a random basis, because t_r's contribution to score may be ignored in this set. Because the number of pairs in $S_t \times D$ and with unequal first and second coordinates $(d \neq e)$ is $n_t (N - 1)$, the probability that an element of $S_t \times S_{t_r}$ is in R_0 on a random basis is

$$pr(t) = \frac{\| R_0 \|}{n_t (N - 1)}. \tag{12}$$

Next we note that if (d, e) is in S_0 then (e, d) must be also and *vice versa*. To account for this fact we assume that P_t is the set of all sets of two documents each of which contain t_r. The set P_t has $n_t(n_t - 1)/2$ elements in it and the probability that an element in P_t corresponds to a pair in S_0 (in either order) we again take as $pr(t)$. Then let A_0 be the number of successes in $n_t(n_t - 1)/2$ independent Bernoulli trials each with probability of success $pr(t)$ (a success corresponds to an element of P_t occurring in R_0). The number $2A_0$ will stand in the place of $\| S_0 \|$. Next consider R_1. The set R_1 consists of pairs (d, e) from $S_t \times S_{t_r}$ that satisfy $sim(d, e) \geq m$ and would not satisfy this relation if t_r were not in both coordinates. Just as for S_0 and for the same reason it is useful to think in terms of the set P_t. For the simulation we let

$$x(t) = 0.5 \times gw_t \tag{13}$$

and consider $x(t)$ to be the average contribution t_r makes to a vector dot product when it occurs in both involved documents. Then we ask for the probability that a document pair (d, e) from $S_t \times S_{t_r}$ which satisfies

$$sim(d, e) < m \tag{14}$$

also satisfies

$$sim(d, e) + \frac{x(t)}{l(v_d) l(v_e)} \geq m. \tag{15}$$

This probability is denoted by $ps(t)$ and is found by processing a large number of document pairs in the database. We let A_1 denote the number of successes in $n_t(n_t - 1)/2 - A_0$ independent Bernoulli trials each with probability $ps(t)$ of success (success now corresponds to an element of P_t that satisfies (14) and (15)). The number $2A_1$ will then stand in the place of $\| R_1 \|$ representing the number of pairs each member of which contains t_r and which satisfy (6) but would

not do so without t_r's contribution. The simulated strength of t_r is then

$$s(t_r) = \frac{2(A_0 + A_1)}{\|R_0\| + 2A_1} \quad (16)$$

This value is simulated a large number of times to obtain the mean, $E(s(t_r))$, and standard deviation, $S(s(t_r))$, and the results are then compared with $s(t)$ according to (7) to determine whether t is a stop word or not.