A Vector Space Model for Automatic Indexing

G. Salton, A. Wong
and C. S. Yang
Cornell University

In a document retrieval, or other pattern matching environment where stored entities (documents) are compared with each other or with incoming patterns (search requests), it appears that the best indexing (property) space is one where each entity lies as far away from the others as possible; in these circumstances the value of an indexing system may be expressible as a function of the density of the object space; in particular, retrieval performance may correlate inversely with space density. An approach based on space density computations is used to choose an optimum indexing vocabulary for a collection of documents. Typical evaluation results are shown, demonstating the usefulness of the model.

Key Words and Phrases: automatic information retrieval, automatic indexing, content analysis, document space

CR Categories: 3.71, 3.73, 3.74, 3.75

This study was supported in part by the National Science Foundation under grant GN 43505. Authors' addresses: G. Salton and A. Wong, Department of Computer Science, Cornell University, Ithaca, NY 14850; C. S. Yang, Department of Computer Science, The University of Iowa, Iowa City, IA, 52240.

[1] Although we speak of documents and index terms, the present development applies to any set of entities identified by weighted property vectors.

[2] Retrieval performance is often measured by parameters such as *recall* and *precision*, reflecting the ratio of relevant items actually retrieved and of retrieved items actually relevant. The question concerning optimum space configurations may then be more conventionally expressed in terms of the relationship between document indexing, on the one hand, and retrieval performance, on the other.

1. Document Space Configurations

Consider a document space consisting of documents D_i, each identified by one or more index terms T_j; the terms may be weighted according to their importance, or unweighted with weights restricted to 0 and 1.[1] A typical three-dimensional index space is shown in Figure 1, where each item is identified by up to three distinct terms. The three-dimensional example may be extended to t dimensions when t different index terms are present. In that case, each document D_i is represented by a t-dimensional vector

$$D_i = (d_{i1}, d_{i2}, \ldots, d_{it}),$$

d_{ij} representing the weight of the jth term.

Given the index vectors for two documents, it is possible to compute a similarity coefficient between them, $s(D_i, D_j)$, which reflects the degree of similarity in the corresponding terms and term weights. Such a similarity measure might be the inner product of the two vectors, or alternatively an inverse function of the angle between the corresponding vector pairs; when the term assignment for two vectors is identical, the angle will be zero, producing a maximum similarity measure.

Instead of identifying each document by a complete vector originating at the 0-point in the coordinate system, the relative distance between the vectors is preserved by normalizing all vector lengths to one, and considering the projection of the vectors onto the envelope of the space represented by the unit sphere. In that case, each document may be depicted by a single point whose position is specified by the area where the corresponding document vector touches the envelope of the space. Two documents with similar index terms are then represented by points that are very close together in the space, and, in general, the distance between two document points in the space is inversely correlated with the similarity between the corresponding vectors.

Since the configuration of the document space is a function of the manner in which terms and term weights are assigned to the various documents of a collection, one may ask whether an optimum document space configuration exists, that is, one which produces an optimum retrieval performance.[2]

If nothing special is known about the documents under consideration, one might conjecture that an ideal document space is one where documents that are jointly relevant to certain user queries are clustered together, thus insuring that they would be retrievable jointly in response to the corresponding queries. Contrariwise, documents that are never wanted simul-

taneously would appear well separated in the document space. Such a situation is depicted in Figure 2, where the distance between two x's representing two documents is inversely related to the similarity between the corresponding index vectors.

While the document configuration of Figure 2 may indeed represent the best possible situation, assuming that relevant and nonrelevant items with respect to the various queries are separable as shown, no practical way exists for actually producing such a space, because during the indexing process, it is difficult to anticipate what relevance assessments the user population will provide over the course of time. That is, the optimum configuration is difficult to generate in the absence of *a priori* knowledge of the complete retrieval history for the given collection.

In these circumstances, one might conjecture that the next best thing is to achieve a maximum possible separation between the individual documents in the space, as shown in the example of Figure 3. Specifically, for a collection of n documents, one would want to minimize the function

$$F = \sum_{\substack{i=1 \\ i \neq j}}^{n} \sum_{j=1}^{n} s(D_i, D_j), \qquad (1)$$

where $s(D_i, D_j)$ is the similarity between documents i and j. Obviously when the function of eq. (1) is minimized, the average similarity between document pairs is smallest, thus guaranteeing that each given document may be retrieved when located sufficiently close to a user query *without also necessarily retrieving its neighbors*. This insures a high precision search output, since a given relevant item is then retrievable without also retrieving a number of nonrelevant items in its vicinity. In cases where several different relevant items for a given query are located in the same general area of the space, it may then also be possible to retrieve many of the relevant items while rejecting most of the nonrelevant. This produces both high recall and high precision.[3]

Two questions then arise: first, is it in fact the case that a separated document space leads to a good retrieval performance, and vice-versa that improved retrieval performance implies a wider separation of the documents in the space; second, is there a practical way of measuring the space separation. In practice, the expression of eq. (1) is difficult to compute, since the number of vector comparisons is proportional to n^2 for a collection of n documents.

For this reason, a clustered document space is best considered, where the documents are grouped into classes, each class being represented by a class centroid.

[3] In practice, the best performance is achieved by obtaining for each user a desired recall level (a specified proportion of the relevant items); at that recall level, one then wants to maximize precision by retrieving as few of the nonrelevant items as possible.

[4] A number of well-known clustering methods exist for automatically generating a clustered collection from the term vectors representing the individual documents [1].

Fig. 1. Vector representation of document space.

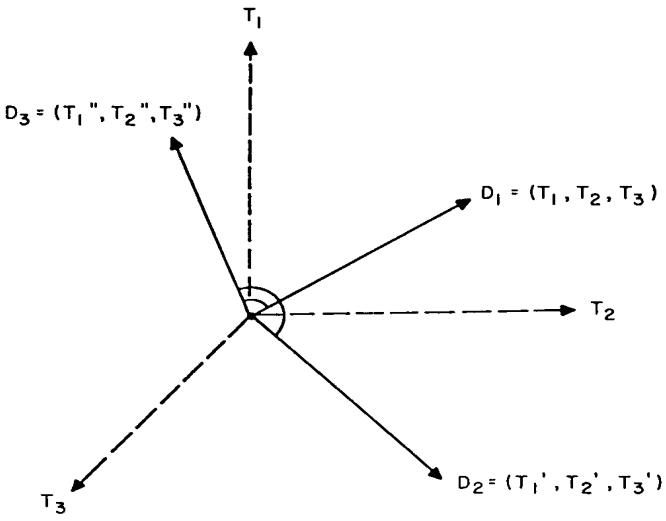


Fig. 2. Ideal document space.

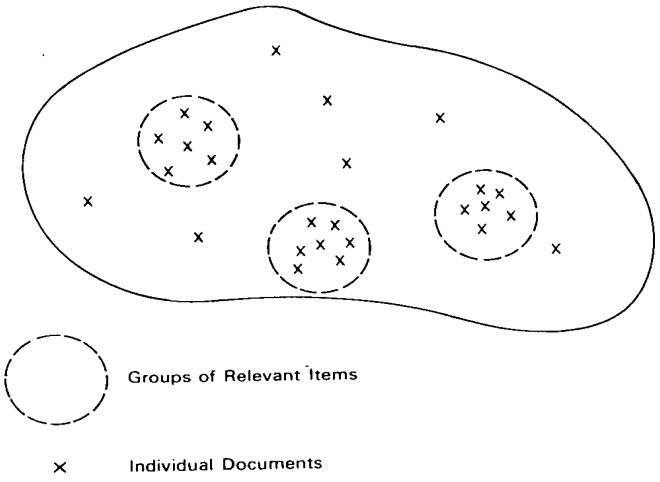


Fig. 3. Space with maximum separation between document pairs.

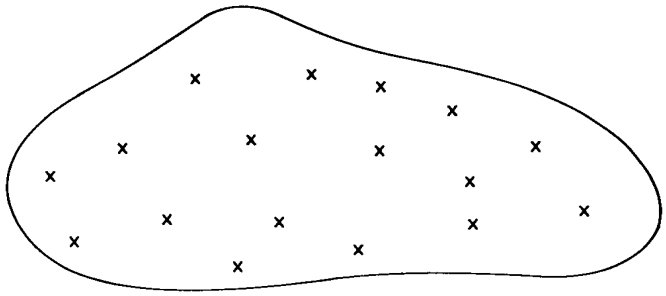
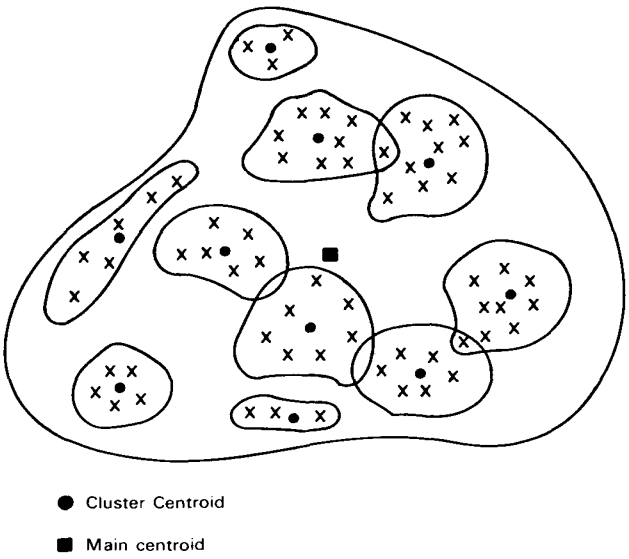


Fig. 4. Clustered document space.



A typical clustered document space is shown in Figure 4, where the various document groups are represented by circles and the centroids by black dots located more or less at the center of the respective clusters.[4] For a given document class K comprising m documents, each element of the centroid C may then be defined as the average weight of the same elements in the corresponding document vectors, that is,

$$c_j = (1/m) \sum_{\substack{i=1 \\ D_i \in K}}^{m} d_{ij} . \tag{2}$$

Corresponding to the centroid of each individual document cluster, a centroid may be defined for the whole document space. This main centroid, represented by a small rectangle in the center of Figure 4, may then be obtained from the individual cluster centroids in the same manner as the cluster centroids are computed from the individual documents. That is, the main centroid of the complete space is simply the weighted average of the various cluster centroids.

In a clustered document space, the space density measure consisting of the sum of all pairwise document similarities, introduced earlier as eq. (1), may be replaced by the sum of all similarity coefficients between each document and the main centroid; that is

$$Q = \sum_{i=1}^{n} s(C^*, D_i), \tag{3}$$

where C^* denotes the main centroid. Whereas the computation of eq. (1) requires n^2 operations, an evaluation of eq. (3) is proportional to n whenever $s(D_i, D_j)$ is proportional to the inner product of the corresponding vectors.

Given a clustered document space such as the one shown in Figure 4, it is necessary to decide what type of clustering represents most closely the separated space shown for the unclustered case in Figure 3. If one assumes that documents that are closely related within a single cluster normally exhibit identical relevance characteristics with respect to most user queries, then the best retrieval performance should be obtainable with a clustered space exhibiting tight individual clusters, but large intercluster distances; that is, (a) the average similarity between pairs of documents within a single cluster should be maximized, while simultaneously (b) the average similarity between different cluster centroids is minimized. The reverse obtains for cluster organizations not conducive to good performance where the individual clusters should be loosely defined, whereas the distance between different cluster centroids should be small.

In the remainder of this study, actual performance figures are given relating document space density to retrieval performance, and conclusions are reached regarding good models for automatic indexing.

2. Correlation between Indexing Performance and Space Density

The main techniques useful for the evaluation of automatic indexing methods are now well understood. In general, a simple straightforward process can be used as a baseline criterion; for example, the use of certain word stems extracted from documents or document abstracts, weighted in accordance with the frequency of occurrence (f_i^k) of each term k in document i. This method is known as term-frequency weighting. Recall-precision graphs can be used to compare the performance of this standard process against the output produced by more refined indexing methods. Typically, a recall-precision graph is a plot giving precision figures, averaged over a number of user queries, at ten fixed recall levels, ranging from 0.1 to 1.0 in steps of 0.1. The better indexing method will of course produce higher precision figures at equivalent recall levels.

One of the best automatic term weighting procedures evaluated as part of a recent study consisted of multiplying the standard term frequency weight f_i^k by a factor inversely related to the document frequency d_k of the term (the number of documents in the collection to which the term is assigned). [2] Specifically, if d_k is the document frequency of term k, the inverse document frequency IDF_k of term k may be defined as [3]:

$$(IDF)_k = \lceil \log_2 n \rceil - \lceil \log_2 d_k \rceil + 1.$$

A term weighting system proportional to $(f_i^k \cdot IDF_k)$ will assign the largest weight to those terms which arise with high frequency in individual documents, but are at the same time relatively rare in the collection as a whole.

It was found in the earlier study that the average improvement in recall and precision (average precision

improvement at the ten fixed recall points) was about 14 percent for the system using inverse document frequencies over the standard term frequency weighting. The corresponding space density measurements are shown in Table I(a) using two different cluster organizations for a collection of 424 documents in aerodynamics:

(i) Cluster organization A is based on a large number of relatively small clusters, and a considerable amount of overlap between the clusters (each document appears in about two clusters on the average); the clusters are defined from the document-query relevance assessments, by placing into a common class all documents jointly declared relevant to a given user query.

(ii) Cluster organization B exhibits fewer classes (83 versus 155) of somewhat larger size (6.6 documents per class on the average versus 5.8 for cluster organization A); there is also much less overlap among the clusters (1.3 clusters per document versus 2.1). The classes are constructed by using a fast automatic tree-search algorithm due to Williamson. [4]

A number of space density measures are shown in Table I(a) for the two cluster organizations, including the average similarity between the documents and the corresponding cluster centroids (factor x); the average similarity between the cluster centroids and the main centroid; and the average similarity between pairs of cluster centroids (factor y). Since a well-separated space corresponds to tight clusters (large x) and large differences between different clusters (small y), the ratio y/x can be used to measure the overall space density [5].

It may be seen from Table I(a) that all density measures are smaller for the indexing system based on inverse document frequencies; that is, the documents within individual clusters resemble each other less, and so do the complete clusters themselves. However, the "spreading out" of the clusters is greater than the spread of the documents inside each cluster. This accounts for the overall decrease in space density between the two indexing systems. The results of Table I(a) would seem to support the notion that improved recall-precision performance is associated with decreased density in the document space.

The reverse proposition, that is, whether decreased performance implies increased space density, may be tested by carrying out term weighting operations inverse to the ones previously used. Specifically, since a weighting system in *inverse* document frequency order produces a high recall-precision performance, a system which weights the terms directly in order of their document frequencies (terms occurring in a large number of documents receive the highest weights) should be correspondingly poor. In the output of Table I(b), a term weighting system proportional to $(f_i^k \cdot DF_k)$ is used, where f_i^k is again the term frequency of term k in document i, and DF_k is defined as $10/(IDF)_k$. The recall-precision figures of Table I(b) show that such a

Table I. Effect of Performance Change on Space Density

	(a) Effect of performance improvement on space density				(b) Effect of performance deterioration on space density			
	Cluster organization A (155 clusters; 2.1 overlap)		Cluster organization B (83 clusters; 1.3 overlap)		Cluster organization A (155 clusters; 2.1 overlap)		Cluster organization B (83 clusters; 1.3 overlap)	
Type of indexing	Standard term frequency weights (f_i^k)	Term frequency with inverse doc. freq. ($f_i^k \cdot IDF_k$)	Standard term frequency weights (f_i^k)	Term frequency with inverse doc. freq. ($f_i^k \cdot IDF_k$)	Standard term frequency weights (f_i^k)	Term frequency with document frequency ($f_i^k \cdot DF_k$)	Standard term frequency weights (f_i^k)	Term frequency with document frequency ($f_i^k \cdot DF_k$)
Recall-precision output*	—	+14%	—	+14%	—	−10.1%	—	−10.1%
Average similarity between documents and corresponding cluster centeroids (x)	.712	.668 (−.044)	.650	.589 (−.061)	.712	.741 (+.029)	.650	.696 (+.046)
Average similarity between cluster centroids and main centroid	.500	.454 (−.046)	.537	.492 (−.045)	.500	.555 (+.055)	.537	.574 (+.037)
Average similarity between pairs of cluster centroids (y)	.273	.209 (−.046)	.315	.252 (−.063)	.273	.329 (+.056)	.315	.362 (+.047)
Ratio y/x	.273/.712 = .383	.209/.668 = .318 (−19%)	.315/.650 = .485	.252/.589 = .428 (−12%)	.273/.712 = .383	.329/.741 = .444 (+16%)	.315/.650 = .485	.362/.696 = .520 (+7%)

* From [2].

weighting system produces a decreased performance of about ten percent, compared with the standard.

The space density measurements included in Table I(b) are the same as those in Table I(a). For the indexing system of Table I(b), a general "bunching up" of the space is noticeable, both inside the clusters and between clusters. However, the similarity of the various cluster centroids increases more than that between documents inside the clusters. This accounts for the higher y/x factor by 16 and 7 percent for the two cluster organizations, respectively.

3. Correlation Between Space Density and Indexing Performance

In the previous section certain indexing methods which operate effectively in a retrieval environment were seen to be associated with a decreased density of the vectors in the document space, and contrariwise, poor retrieval performance corresponded to a space that is more compressed.

The relation between space configuration and retrieval performance may, however, also be considered from the opposite viewpoint. Instead of picking document analysis and indexing systems with known performance characteristics and testing their effect on the density of the document space, it is possible to change the document space configurations artificially in order to ascertain whether the expected changes in recall and precision are in fact produced.

The space density criteria previously given stated that a collection of small tightly clustered documents with wide separation between individual clusters should produce the best performance. The reverse is true of large nonhomogeneous clusters that are not well separated. To achieve improvements in performance, it would then seem to be sufficient to increase the similarity between document vectors located in the same cluster, while decreasing the similarity between different clusters or cluster centroids. The first effect is achieved by emphasizing the terms that are unique to only a few clusters, or terms whose cluster occurrence frequencies are highly skewed (that is, they occur with large occurrence frequencies in some clusters, and with much lower frequencies in many others). The second result is produced by deemphasizing terms that occur in many different clusters.

Two parameters may be introduced to be used in carrying out the required transformations [5]:

$NC(k)$: the number of clusters in which term k occurs (a term occurs in a cluster if it is assigned to at least one document in that cluster); and

$CF(k, j)$: the cluster frequency of term k in cluster j that is, the number of documents in cluster j in which term k occurs.

For a collection arranged into p clusters, the average cluster frequency $\langle CF(k)\rangle$ may then be defined from $CF(k, j)$ as

$$\langle CF(k)\rangle = (1/p) \sum_{j=1}^{p} CF(k, j).$$

Given the above parameters, the skewness of the occurrence frequencies of the terms may now be measured by a factor such as

$$F_1 = |\langle CF(k)\rangle - CF(k, j)|.$$

On the other hand, a factor F_2 inverse to $NC(k)$ [for example, $1/NC(k)$] can be used to reflect the rarity with which term k is assigned to the various clusters. By multiplying the weight of each term k in each cluster j by a factor proportional to $F_1 \cdot F_2$ a suitable spreading out should be obtained in the document space. Contrariwise, the space will be compressed when a multiplicative factor proportional to $1/(F_1 \cdot F_2)$ is used.

The output of Table II(a) shows that a modification of term weights by the $F_1 \cdot F_2$ factor produces precisely the anticipated effect: the similarity between documents included in the same cluster (factor x) is now greater, whereas the similarity between different cluster centroids (factor y) has decreased. Overall, the space density measure (y/x) decreases by 18 and 11 percent respectively for the two cluster organizations. The average retrieval performance for the spread-out space shown at the bottom of Table II(a) is improved by a few percentage points.

The corresponding results for the compression of the space using a transformation factor of $1/(F_1 \cdot F_2)$ are shown in Table II(b). Here the similarity between documents inside a cluster decreases, whereas the similarity between cluster centroids increases. The overall space density measure (y/x) increases by 11 and 16 percent for the two cluster organizations compared with the space representing the standard term frequency weighting. This dense document space produces losses in recall and precision performance of 12 to 13 percent.

Taken together, the results of Tables I and II indicate that retrieval performance and document space density appear inversely related, in the sense that effective indexing methods in terms of recall and precision are associated with separated (compressed) document spaces; on the other hand, artificially generated alterations in the space densities appear to produce the anticipated changes in performance.

The foregoing evidence thus confirms the usefulness of the "term discrimination" model and of the automatic indexing theory based on it. These questions are examined briefly in the remainder of this study.

4. The Discrimination Value Model

For some years, a document indexing model known as the term discrimination model has been used experi-

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.