On Modeling of Information Retrieval Concepts in Vector Spaces

S. K. M. WONG, W. ZIARKO, V. V. RAGHAVAN, P. C. N. WONG
University of Regina

The Vector Space Model (VSM) has been adopted in information retrieval as a means of coping with inexact representation of documents and queries, and the resulting difficulties in determining the relevance of a document relative to a given query. The major problem in employing this approach is that the explicit representation of term vectors is not known a priori. Consequently, earlier researchers made the assumption that the vectors corresponding to terms are pairwise orthogonal. Such an assumption is clearly unrealistic. Although attempts have been made to compensate for this assumption by some separate, corrective steps, such methods are ad hoc and, in most cases, formally inconsistent.

In this paper, a generalization of the VSM, called the GVSM, is advanced. The developments provide a solution not only for the computation of a measure of similarity (correlation) between terms, but also for the incorporation of these similarities into the retrieval process.

The major strength of the GVSM derives from the fact that it is theoretically sound and elegant. Furthermore, experimental evaluation of the model on several test collections indicates that the performance is better than that of the VSM. Experiments have been performed on some variations of the GVSM, and all these results have also been compared to those of the VSM, based on inverse document frequency weighting. These results and some ideas for the efficient implementation of the GVSM are discussed.

Categories and Subject Descriptors H.3.1 **[Information Storage and Retrieval]**: Content Analysis and Indexing—*indexing methods*; *thesauruses*; H.3.3 **[Information Storage and Retrieval]**: Information Search and Retrieval—*retrieval models*; *search process*

General Terms: Design, Experimentation, Languages, Theory

Additional Key Words and Phrases: Boolean algebra, document representation, generalized vector space, retrieval strategy, term cooccurrence, vector space theory

1. INTRODUCTION

Information Retrieval (IR) systems are designed with the objective of providing, in response to a user query, references to documents that would contain the information desired by the user. In other words, the system is intended to identify which documents the user should read in order to satisfy his (her) information requirements.

Authors' address: Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada S4S 0A2.

In this environment, the information items to be searched are not simply "records" or "tuples" as in conventional database management systems. Instead, we have a collection of documents (e.g., books, journal articles, technical reports, etc.). In order to identify which documents the user should read with respect to his information requirements, some method for the representation of what the documents are about (i.e., knowledge representation of documents) is needed. Since any knowledge representation of a set of objects provides only partial and imprecise characterization of the perceived reality, the representation in the system as to the contents of the documents cannot be expected to be entirely satisfactory.

Another problem which is closely related to the difficulty in representing the contents of documents is that of characterizing the user need. Although the query language used can be very precise relative to the method of representation chosen, it is unlikely that the actual user need can be exactly specified by the language.

An additional problem in this context is one of making an assessment as to whether or not a document meets the actual needs of the user. That is, not only does the system have to represent the documents and user needs, but it must also provide a characterization of the process by which the user comes to a particular decision concerning relevance. A document may or may not be relevant to a user query depending on many variables concerning the document (e.g., its scope, how it is written) as well as numerous user characteristics (e.g., why the search is initiated, user's previous knowledge). In any case, whatever the information retrieval system does, if a document is judged by the user to be of interest, it is *relevant*; it is *nonrelevant* otherwise. Since many factors may influence the judgement concerning relevance in a complex way, it is easy to see that designing an IR system within this frame of reference is very challenging. In fact, it is impossible to come up with a "perfect" scheme for representation and retrieval whereby only and all relevant documents are retrieved.

Finally, even if one manages to adopt a "perfect" scheme for representation, and a query language with full expressive power, the user may still have difficulties. From a practical point of view, any such system would be too complex for a typical user to master.

It is due to all these reasons that IR researchers take the view that the system should adopt fairly simple methods of representation and seek approaches that facilitate the ranking of documents in the order of their estimated usefulness to a user query.

One well-known approach for the design of an IR system, with the above goals and constraints, models documents and queries as elements of a vector space [6, 9, 11]. First, a representation as a vector is developed for each document in the collection. This would require the application of some automatic or manual indexing technique to the full text or some surrogate (e.g., abstract) of the documents in order to identify the index terms or keywords to be used in their representation. Second, each index term involved is assumed to correspond to a vector, and these vectors together are assumed to generate the vector space of interest. The effect of this is that one can express any document as a linear combination of these term vectors. Similarly, when a query is presented, it is also put through the indexing process and a vector representing the query is constructed.

In addition to the representation of documents and queries as vectors, one needs to introduce some notion of *closeness* or *similarity* between a document and a query, between terms, and so on. A natural way to determine closeness between these items in the vector space model is to define a scalar product between the corresponding vectors. The matching of documents against the query, then, consists of computing the scalar product between the respective vectors. Finally, the documents are presented to the user in the decreasing order of this measure.

2. MOTIVATION

Typically, the end result of putting a document collection through the indexing process is a *document-by-term* matrix, where the (i, j)th element of the matrix corresponds to the frequency of occurrence of term j in the ith document [6]. Let this $p \times n$ matrix be denoted by the symbol W, and its elements by w_{ij}.

The most common implementation of the vector space model involves

(a) the interpretation of the rows of matrix W as the component of document vectors along the direction of the various term vectors, and

(b) the assumption that terms are pairwise orthogonal. That is, scalar product $\vec{t}_i \cdot \vec{t}_j$ of any two (normalized) term vectors equals 1 if $i = j$ and equals 0 otherwise.

It is well known that the orthogonality assumption is too restrictive. However, it has been considered acceptable as a first order of approximation, and many useful and interesting results have been obtained despite such a simplifying assumption [6, 9, 11].

While there may be good justification for starting an investigation with a simpler model (which is a special case), it is important to clearly understand the general model. In Raghavan and Wong [4], it is pointed out that earlier work with the vector space model does not fully explain the various concepts and interactions that are critical and this, in turn, has led to some misunderstandings and inconsistent usage of the model.

For example, in order to relax the assumption that terms are pairwise orthogonal, term cooccurrence information has been used, and certain methods of computing term correlations suggest that the columns of W can be viewed as vectors corresponding to the terms, that is, $\vec{t}_i = (w_{1i}, w_{2i}, \ldots, w_{mi})$. However, it is easily shown that representing terms as columns of W is not consistent with representing documents as the rows of W. In other words, if columns of W are interpreted as components of terms along document vectors, then the rows cannot, at the same time, be used to represent document vectors [4].

Raghavan and Wong [4] introduced and explained the various notations and definitions necessary for the understanding of vector space model in the context of information retrieval. While the difficulties involved in the generalization, to the situation where term vectors are not assumed to be orthogonal, were explained in detail, approaches to resolve such difficulties were considered only in passing. In the current work, a particular approach is thoroughly explored.

Two main aspects of dealing with nonorthogonality are

(i) the definition of what it means to say that two terms are orthogonal, and a method of computing the degree of similarity (or correlation) between nonorthogonal terms;
(ii) the incorporation of this information into the retrieval strategy.

As has been the case in the past, the first aspect of computing the term-term correlations is based on cooccurrence data. However, the specific scheme we propose for dealing with nonorthogonality leads to a natural and rigorous framework for identifying a set of orthogonal basis vectors that spans the subspace of interest. As for the second aspect, given the premises of the vector space model, it can be readily shown how the term-term correlations ought to be incorporated into the retrieval process. Thus, in this paper, both of these important phases are given a clean, theoretical justification. In contrast, certain methods reported earlier have been, for the most part, heuristic and without adequate formal support [3, 7, 12].

Because of the emphasis in this work on a theoretical foundation for utilizing term cooccurrence data, it is believed that the significance of our results is comparable to that of several recent investigations on term cooccurrence [2, 10, 13]. This paper is organized as follows. In Section 3, we introduce notions and definitions that are needed for subsequent discussions on the vector space model. In Section 4, an overview of the steps involved in the proposed generalization of the vector space model is provided. Section 5 contains the specific details of the generalized model and the justification for the various prescriptions. The proposed model and a few of its variations are experimentally tested and these results are discussed in Section 6. In the final section, the conclusions of this investigation are summarized.

3. NOTATIONS AND BASIC DEFINITIONS

The basic premise in the vector space model is that the various items of interest in the information retrieval environment are modeled as elements of a vector space. Specially terms, documents, queries, concepts, and so on are all represented as vectors in a vector space.

Let $t_1, t_2, \ldots, t_n$ be the terms used to index the documents in a collection. Corresponding to each term, t_i, suppose there exists a vector $\vec{t}_i$ in a vector space. For the general case of the model, we consider the set of term vectors, $\{\vec{t}_i \mid 1 \le i \le n\}$, to be the *generating set* of the subspace of interest. Thus, any vector in the subspace can be expressed as a linear combination of the $\vec{t}_i$'s.

Let $d_1, d_2, \ldots, d_p$ denote the documents in a collection and let $\{\vec{d}_\alpha = (a_{\alpha 1}, a_{\alpha 2}, \ldots, a_{\alpha n}) \mid 1 \le \alpha \le p\}$ be the set of vectors representing the documents, where $a_{\alpha i}$'s are real numbers. More precisely, $a_{\alpha i}$ is the *component* of $\vec{d}_\alpha$ along the direction of the term vector $\vec{t}_i$. It follows then, that

$$\vec{d}_\alpha = \sum_{i=1}^{n} a_{\alpha i}\vec{t}_i. \tag{3.1}$$

Definition 3.1. A set of vectors $\{\vec{v}_1, \ldots, \vec{v}_k\}$ is *linearly dependent* if there exist some scalars $c_1, c_2, \ldots, c_k$ not all zero, such that

$$c_1\vec{v}_1 + c_2\vec{v}_2 + \cdots c_k\vec{v}_k = \vec{0}.$$

Clearly, if the set $\{\vec{t}_1, \vec{t}_2, \ldots, \vec{t}_n\}$ is linearly independent, then this set of vectors is also a *basis* for the subspace of interest, the subspace is of *dimension n*, and the expansion of $\vec{d}_\alpha$ given in Eq. (3.1) is unique. In contrast, if the set of vectors $\{\vec{t}_1, \vec{t}_2, \ldots, \vec{t}_n\}$ that spans the vector space is linearly dependent, then the dimension of the vector space is n', for some $n' < n$, since a set of linearly independent vectors consisting of n' vectors can always be selected from $\{\vec{t}_1, \vec{t}_2, \ldots, \vec{t}_n\}$.

Definition 3.2. Given a vector space V, the *scalar product* $\vec{u} \cdot \vec{v}$ of any two vectors $\vec{u}, \vec{v} \in V$, is given by $|\vec{u}| \cdot |\vec{v}| \cos \theta$, where $|\vec{u}|$ and $|\vec{v}|$ are the lengths of the vectors and θ is the angle between $\vec{u}$ and $\vec{v}$.

Two term vectors, $\vec{t}_i$ and $\vec{t}_j$, are *orthogonal* if $\vec{t}_i \cdot \vec{t}_j = 0$. If every pair of vectors $\{\vec{t}_i, \vec{t}_j\}$ for $i \neq j$ in the set $\{\vec{t}_1, \vec{t}_2, \ldots, \vec{t}_n\}$ is orthogonal, then the set is linearly independent and forms a basis for the subspace under consideration. The converse, however, is not true. That is, the set $\{\vec{t}_1, \vec{t}_2, \ldots, \vec{t}_n\}$ may be linearly independent, but not necessarily pairwise orthogonal. From the above discussions, it should be clear that, when adopting the vector space model, one cannot assume that term vectors are necessarily pairwise orthogonal.

Now let us consider the issue of ranking documents with respect to a query as a part of the retrieval process. The fact that the set of term vectors $\{\vec{t}_1, \vec{t}_2, \ldots, \vec{t}_n\}$ is considered a generating set implies that not only the documents, but also the queries, can be represented as a linear combination of the $\vec{t}_i$'s. A query vector $\vec{q}$ can, therefore, be expressed by

$$\vec{q} = \sum_{j=1}^{n} q_j \cdot \vec{t}_j. \tag{3.2}$$

Given the expression in (3.1) and (3.2), and assuming that the scalar product between two normalized vectors is a measure of their similarity (cosine similarity function), we have

$$\vec{d}_\alpha \cdot \vec{q} = \sum_{j=1}^{n} \sum_{i=1}^{n} a_{\alpha i} q_j \vec{t}_i \cdot \vec{t}_j, \qquad \text{where} \quad \alpha = 1, 2, \ldots, p. \tag{3.3}$$

We can, then, rank the documents relative to $\vec{q}$ in terms of the values of the above similarity function. Thus, for our purposes, we need to know $a_{\alpha i}$'s, the components of documents along the various term vectors as well as the similarity between every pair of term vectors expressed as $\vec{t}_i \cdot \vec{t}_j$. Note that we may or may not know the vector representation for the $\vec{t}_i$'s explicitly.

4. OVERVIEW

In this section an overview of how we propose to resolve the problems and issues associated with the use of the vector space model is presented. More specifically, we identify the assumptions or hypotheses needed, the kinds of data in the physical problem that are considered as given, and the mapping of these data to the elements of the vector space model. The last issue mentioned above concerns the *interpretation* of the objects and relationships in the physical problem to the formal objects and components of the model.

As mentioned in the introduction, it is common to start with a document-by-term matrix W, which has been obtained from a document collection through a process of indexing. The matrix element $w_{\alpha i}$ of W is the occurrence frequency of term t_i in document d_α. In the context of the vector space model, there are a few different options for interpreting the elements of W [4]. For the purposes of this paper, $w_{\alpha i}$ is interpreted as the component of the document vector $\vec{d}_\alpha$ along term vector $\vec{t}_i$. That is, $a_{\alpha i} = w_{\alpha i}$. It follows from Eq. (3.1) that

$$\vec{d}_\alpha = \sum_{i=1}^{n} w_{\alpha i}\vec{t}_i. \tag{4.1}$$

We further assume that the components of a query vector along the various term vectors is determined in a similar fashion. However, from the point of view of computing the document-query similarity defined by Eq. (3.3), the term-term similarities are still unknown.

On the one hand, one could side-step this issue by assuming that there does not exist any correlation between terms. In other words, the assumption is that the terms are pairwise orthogonal. This is, clearly, not satisfactory. On the other hand, a prescription for obtaining term-term similarity measures can be proposed. Our aim is to take the latter route. In doing so, we would like to also ensure that the prescription is a natural and rigorous extension of the conventional vector space model.

Before we can *measure* term-term similarity, there must first be a determination of what we want to mean by saying that two terms are similar. Alternatively, we may ask under what condition can two terms be considered not similar or, using the terminology of the model, orthogonal. The answer to this question is crucial since whatever notion one applies at this stage will essentially dictate the meaning attached to the similarity between any two vectors within the model. Once a meaning is selected, it remains fixed for all subsequent computations of similarity between vectors. The real question, then, is to what aspect of the physical problem do we want to map the formal concept of orthogonality, which is a part of the model.

A review of earlier work in interpreting term-term similarity suggests various directions:

(i) words in the language can be analyzed from a linguistic point of view to obtain synonyms, antonyms, relationships leading to hierarchical structure, and so on;

(ii) term-term relationships can be based on pseudoclassification, where the term relationships are obtained by analyzing the way in which document representations and similarity computations must be adjusted to obtain retrieval results desired by the user [5, 8];

(iii) term cooccurrence data can be used to determine if the presence of one term implies the presence of another, and this fact can be used for obtaining term-term similarity [2, 10, 13].

It is interesting to note that the direction one chooses also has implication for the extent to which terms are imagined to exist, independently of the document collection at hand. In the first case above, terms are considered to have meaning and relationships between each other, essentially, without particular reference to what documents we have in a collection or how they are indexed. In contrast, the third case views concepts that correspond to the terms and their relationships totally within the context of the particular document representations at hand. That is, the meaning of terms are formal notions specified with respect to the given collection of documents.

The specific choice made for the developments in this paper bases the computation of term similarities on cooccurrence data. In this connection, the following hypotheses[1] are made:

Hypothesis 1. A concept (an index term is a special case) is characterized by a set of documents. More precisely, a concept corresponds to the maximal subset of documents such that every document in the set contains the concept.

Hypothesis 2. A concept i is unrelated to another concept j if the set of documents characterizing concept i does not intersect with the set of documents characterizing concept j.

Hypothesis 3. The greater the overlap between the document sets characterizing two different concepts, the more similar are the two concepts.

The important contribution of the current work is the realization of the vector space model in a way that is consistent with the hypotheses given above. The model then forms a basis for verifying the validity of the hypotheses as measured by the retrieval performance.

In summary, the investigation of these hypotheses involves the following correspondences between model elements and the physical problem:

(i) vectors representing concepts are such that if sets of documents corresponding to two different concepts are disjoint, then the associated vectors are orthogonal;

(ii) the scalar product between vectors associated with two different concepts, essentially, becomes larger as the amount of overlap between the corresponding sets of documents increases;

(iii) the determination of the basis vectors involves the identification of a set of *fundamental* concepts (denoted by $\vec{m}$'s in subsequent sections) that are, together, complete and are pairwise orthogonal;

(iv) terms contained in documents are represented as a linear combination of vectors associated with the fundamental concepts;

(v) finally, documents and queries are represented as a linear combination of terms, which as mentioned in (iv), are in turn a linear combination of fundamental concepts.

[1] These hypotheses are not really new [3, 7, 12], but they have often not been explicitly identified as such.

5. DEVELOPMENT OF THE MODEL

In this section the essential features of the model being proposed are developed and justified. The resulting model is referred to as the *Generalized Vector Space Model* (*GVSM*). The justification for the GVSM is provided both by considering how the limitations associated with Boolean retrieval can be removed and by showing that it is a theoretically sound generalization of the conventional vector space model for handling term correlations. The measure of term correlation is based on term cooccurrence information. The first step in the development is to explain how the elements of Boolean algebra may be modeled as vectors in a vector space. It is then pointed out how terms that are represented as Boolean expressions in a Boolean retrieval system can be modeled as vectors in a vector space. In the vector space corresponding to the Boolean retrieval model, if documents are not identical then they are, by our definition, orthogonal. Finally, these concepts are generalized to handle nonorthogonality and the situation where documents and query are represented by weighted terms.

5.1 Vector Representation of Elements of a Boolean Algebra

Let $x_1, x_2, \ldots, x_n$ be n literals used to generate the free Boolean algebra, denoted B_{2^n}. Any Boolean expression composed of these literals (using operators AND, OR, or NOT) is an element of the algebra.

What we desire is to identify a vector space such that every Boolean expression in B_{2^n} corresponds to a vector in the vector space. In a vector space it is necessary to specify a set of vectors that form a basis. If a basis is known, then any vector in the space can be expressed as a linear combination of the basis vectors. Since the intent is to obtain a way of expressing every possible Boolean expression, it is appropriate to have the set of basis vectors correspond to a set of fundamental expressions which can be combined to generate any element of the algebra. We therefore employ the notion of an atomic expression.

An atomic expression, or a minterm, in the n literals $x_1, x_2, \ldots, x_n$ is a conjunction of the literals where each x_i appears exactly once and is either in complemented or uncomplemented form. Clearly, there are 2^n minterms in all. It is well known that the conjunction of any two minterms is always zero (false) and that any Boolean expression in the literals $x_1, x_2, \ldots, x_n$ can be uniquely expressed as a disjunction of minterms. The representation obtained in this way is the well-known disjunctive normal form.

Let $\{m_k\}_{2^n}$ denote the set of minterms in B_{2^n}. In order to characterize a vector space in which these correspond to the basis vectors, we define a set of 2^n-dimensional vectors $\{\vec{m}_k\}$. These vectors constitute an orthonormal basis of the vector space in R^{2^n} as follows:

$$\begin{aligned}
\vec{m}_1 &= (1, 0, 0, \ldots, 0)\\
\vec{m}_2 &= (0, 1, 0, \ldots, 0)\\
\vec{m}_3 &= (0, 0, 1, \ldots, 0)\\
&\ \vdots\\
\vec{m}_{2^n} &= (0, 0, 0, \ldots, 1)
\end{aligned} \tag{5.1}$$

Given these, it is easily seen that the vector representation of any Boolean expression is given by the vector sum of the basis vectors which correspond to the minterms in the disjunctive normal form of the expression.

The assertion that, for any two vectors $\vec{m}_i$, $\vec{m}_j$, the scalar product $\vec{m}_i \cdot \vec{m}_j$ is zero corresponds to the fact that the conjunction of atomic expressions m_i and m_j is zero. In general, if two vectors are not orthogonal, then the corresponding Boolean expressions have at least one minterm in common.

5.2 Vector Representation of Terms Assuming No Weights

The ideas developed in Section 5.1 can be applied to an information retrieval environment and each index term can be given an explicit vector representation. Let the indexing vocabulary consist of terms $t_1, t_2, \ldots, t_n$. Any literal can appear in a Boolean expression either as $\bar{t}_i$ or t_i, depending on whether it needs to be complemented or not. In particular, conjunctive expressions, where every literal appears in either uncomplemented or complemented form are the atomic expressions.

Let $\{m_k\}_{2^n}$ denote the set of all atomic expressions in the literals $t_1, t_2, \ldots, t_n$. Then, since each t_i is itself an element of the Boolean algebra generated, t_i can be expressed in its disjunctive normal form:

$$t_i = m_{i_1} \quad \text{OR} \quad m_{i_2} \ldots \quad \text{OR} \quad m_{i_r}, \tag{5.2}$$

where the m_{i_j}'s are minterms in which t_i is uncomplemented. Let the set of minterms in Eq. (5.2) be denoted by $\{m\}^i$. We can now define basis vectors analogous to Eq. (5.1) and the term t_i can be written in the vector notation as

$$\vec{t}_i = \sum_{m_k \in \{m\}^i} \vec{m}_k. \tag{5.3}$$

Alternatively,

$$\vec{t}_i = \sum_{k=1}^{2^n} c_{i_k} \vec{m}_k \tag{5.4}$$

where

$$c_{i_k} = \begin{cases} 1, & \text{if } \ m_k \in \{m\}^i \\ 0, & \text{otherwise.} \end{cases}$$

That is, the term vectors are a linear combination of the $\vec{m}_k$'s, the basis vectors, and the vector sum operator is mapped to the Boolean operator OR of Eq. (5.2). Furthermore, the scalar product between any two basis vectors is zero, corresponding to the fact that the *AND*ing of two minterms is "false."

5.3 The Generalized Vector Space Model (GVSM)

In this section we review the essential features of the GVSM [14, 15]. This model is the result of incorporating the idea developed in Section 5.2 into the framework of the conventional vector space model. One of the main steps in this process involves the generalization of the term vector representation in such a way that the expansion coefficients in Eq. (5.4) are not binary. The determination of these coefficients is, however, closely tied in with our hypothesis of what is meant by two terms being not orthogonal (or correlated). This is because, once the coefficients are specified, the scalar product between any two nonbinary vectors $\vec{t}_i$ and $\vec{t}_j$ is defined. Since scalar product being zero implies orthogonality, a nonzero value must represent a measure of nonorthogonality.

5.3.1 *Document and Query Representation in GVSM.* From Eq. (3.1), it is seen that in the VSM the representation of a document is taken to be a sum of term vectors. In the GVSM we continue to use the vector sum operator and hypothesize that a document should be expressed as the vector sum of the associated term vectors. More precisely,

$$\begin{aligned}\vec{d}_\alpha &= w_{\alpha 1}\vec{t}_1 + w_{\alpha 2}\vec{t}_2 + \cdots w_{\alpha n}\vec{t}_n \\ &= \sum_{i=1}^{n} w_{\alpha i}\vec{t}_i.\end{aligned} \tag{5.5}$$

Since the term vectors are of the form specified in Eqs. (5.4) and (5.5) implies that the documents can be represented as a vector sum of the $\vec{m}$'s. That is,

$$\vec{d}_\alpha = \sum_{i=1}^{2^n} c_i\vec{m}_i, \tag{5.6}$$

where the c_i's are yet to be specified. Although the use of Eq. (5.6) for obtaining a vector representation of documents can be taken simply as a hypothesis, the following points are noted in support of that choice.

Let d_α be a document indexed by terms $\{t_{\alpha 1}, t_{\alpha 2}, \ldots, t_{\alpha r}\}$. Imagine also that we are working in a strict Boolean environment where each query is a Boolean expression in t_i's, the literals associated with the terms. Then d_α should be retrieved for a query q, if the disjunctive normal form of q includes the minterm in which precisely the literals $t_{\alpha 1}, t_{\alpha 2}, \ldots, t_{\alpha r}$ are not negated and all other literals are negated. Now, if this case is modeled by our vector model, we would have

$$\vec{d}_\alpha = \vec{m}_{i_\alpha},$$

such that $m_{i_\alpha} = t_{\alpha 1}$ AND $t_{\alpha 2}$ AND ... $t_{\alpha r}$ AND $\bar{t}_{\alpha r+1}$ AND ... AND $\bar{t}_{\alpha n}$. Let m_{i_α} be referred to as the *dominant atom of* $\vec{d}_\alpha$. Since a query can be a vector sum of $\vec{m}$'s, we have the correspondence that d_α is retrieved for query q if and only if $\vec{d}_\alpha \cdot \vec{q} = 1$. It is clear from the foregoing discussion that representing $\vec{d}_\alpha$ by its dominant atom is a special case of Eq. (5.6). We however observe that this representation is inflexible in the sense that the condition for retrieval is too strict. Furthermore, in this case, if two documents are not identical, then they are orthogonal. Thus, Eq. (5.6) has the effect of relaxing the retrieval criterion and broadening the scope of the document.

Our choice of representatior (Eq. (5.6)) can also be justified from another point of view. The point is that this kind of broadening of representation is a natural way in which to reflect the effect of term similarities. In earlier studies involving the use of term-term similarities, the approach employed had some parallels. For example, both Minker et al. [3] and Sparck-Jones [12] proposed certain algorithms to construct clusters of terms. Then, the incorporation of these clusters into the retrieval process consisted of either expanding queries with terms which belong to the same cluster as those already in the query [3], or by replacement of the original terms in documents and queries by the new, broader concepts (corresponding to the clusters) of which they are a part. In our case, we propose to represent each document by a disjunction of certain fundamental concepts. The idea of having m's other than the dominant atom in representing a document can be illustrated with the following example.

Let us consider a situation involving just three terms, t_1, t_2, t_3. Suppose a document d contains terms t_1 and t_3. Furthermore, let there be many documents in our collection that contain exactly t_1 and t_2, implying that we may conclude that d's description should include t_2 as well. We make d's description include t_2 by letting d be represented by more than one basis vector. More precisely, d is represented not just by the dominant atom $m_i = t_1$ AND $\bar{t}_2$ AND t_3, but also by the $m_j = t_1$ AND t_2 AND $\bar{t}_3$. That is, d is a linear combination of the corresponding basis vectors $\vec{m}_i$ and $\vec{m}_j$. We believe that this is an alternative way of incorporating the effect of term dependencies; it is well suited to the premises upon which the GVSM is based.

In a general sense, expressing documents as a combination of $\vec{m}$'s can also be seen as a way to model another aspect of the physical problem. For instance, it is well known that there is certain variability in the way trained indexers describe the same document [16]. If several descriptions are equally valid, then, clearly, building a system based on just one of them may bias the representation towards certain kinds of users. Thus our approach can be seen as a means associating more than one description (m's) with each document and, additionally, having a certain measure of importance accorded to each such description. In fact, this line of thinking is central to a recent paper by Gordon [1], in which the generic algorithm is used as a basis for determining alternative descriptions and corresponding measures of usefulness for each document.

It is also necessary to specify the way in which a query will be represented. Given $q = (q_1, q_2, \ldots, q_n)$, we propose that the query be represented as the vector sum of the $\vec{t}_i$'s involved. That is,

$$\vec{q} = \sum_{j=1}^{n} q_j \vec{t}_j. \tag{5.7}$$

This choice is made for the same reason that documents are represented as a vector sum of terms.

Using these prescriptions, both documents and queries can be expressed as a linear combination of the $\vec{m}$'s, and the computation of $\vec{d}_\alpha \cdot \vec{q}$ is straightforward. All that still remains is to show how Eq. (5.4) is generalized to express the $\vec{t}$'s as a vector sum of $\vec{m}$'s in terms of nonbinary expansion coefficients. As mentioned earlier, this requires the meaning of term correlations to be made precise.

5.3.2 *Vector Representation of Terms Using Term Occurrence Frequencies.* First, a simple example is presented to motivate the approach adopted.

Example 5.1. Let D be a set of documents indexed only by two terms, t_1 and t_2. Let D_F be the maximal subset of documents satisfying F, where F is a Boolean expression in the t's.

We can identify the following disjoint subsets forming a partition of D:

$$D_{t_1\bar{t}_2} = D_{t_1} \cap \bar{D}_{t_2},$$
$$D_{t_1t_2} = D_{t_1} \cap D_{t_2},$$
$$D_{\bar{t}_1t_2} = \bar{D}_{t_1} \cap D_{t_2}.$$

where $D_{t_1\bar{t}_2}$, $D_{t_1t_2}$, and so on, correspond respectively to $D_{t_1 \text{AND} \bar{t}_2}$, $D_{t_1 \text{AND} t_2}$, and so on. (The ANDs are dropped for convenience.) $\bar{D}_{t_i}$ denotes the set complement of D_{t_i} (i.e., $\bar{D}_{t_i}$ is the subset of documents not containing t_i).

Based on intuition, we argue that the correlation between any two index terms depends on the number of documents in which these two terms appear together. This sort of argument based on term cooccurrence information has been the basis for measuring term correlation in earlier studies [2, 10, 13].

Let $c(D_F)$ denote the cardinality of the set D_F. For reasons of clarity, the cardinality $c(D_{t_1t_2})$ of the subset $D_{t_1t_2} = D_{t_1} \cap D_{t_2}$ (which denotes the number of documents containing t_1 and t_2) is first taken as the measure of the "unnormalized" correlation between t_1 and t_2. We develop a finer measure of term correlation using nonbinary weights in the latter part of this section.

In terms of vector notation, the correlation between t_1 and t_2, denoted by $\vec{t}_1 \cdot \vec{t}_2$, can be conveniently expressed as the scalar product of two normalized term vectors, $\vec{t}_1$ and $\vec{t}_2$, namely,

$$\vec{t}_1 \cdot \vec{t}_2 = \frac{c^2(D_{t_1t_2})}{[c^2(D_{t_1\bar{t}_2}) + c^2(D_{t_1t_2})]^{1/2}[c^2(D_{\bar{t}_1t_2}) + c^2(D_{t_1t_2})]^{1/2}},$$

where

$$\vec{t}_1 = \frac{c(D_{t_1\bar{t}_2})\vec{m}_1 + c(D_{t_1t_2})\vec{m}_2}{[c^2(D_{t_1\bar{t}_2}) + c^2(D_{t_1t_2})]^{1/2}},$$

$$\vec{t}_2 = \frac{c(D_{t_1t_2})\vec{m}_2 + c(D_{\bar{t}_1t_2})\vec{m}_3}{[c^2(D_{t_1t_2}) + c^2(D_{\bar{t}_1t_2})]^{1/2}}.$$

and $\vec{m}_1$, $\vec{m}_2$, and $\vec{m}_3$ are the orthonormal basis vectors.

It is evident from the above example that terms can be meaningfully expressed as linear combinations of $\vec{m}$'s. Clearly, $\vec{m}_1$, $\vec{m}_2$, and $\vec{m}_3$ correspond respectively to the atomic expressions $t_1\bar{t}_2$, t_1t_2, and $\bar{t}_1t_2$. In the example, only the presence or absence of a term in a document is considered. This limitation is reflected in the assertion that $c(D_{t_1t_2})$ is a measure of the correlation between $\bar{t}_1$ and $\bar{t}_2$. Furthermore, this example helps in convincing oneself that the expansion of a term vector, say $\bar{t}_i$, need not have a nonzero coefficient for all vectors corresponding to minterms in $\{m\}^i$. This is due to the fact that term cooccurrence, and the cardinalities of other sets used as normalization factors, depend on the particular collection of documents at hand. For instance, if t_1 and t_2 do not cooccur in a given collection, then the expansion of neither $\vec{t}_1$ nor $\vec{t}_2$ will involve $\vec{m}_2$, and $\vec{t}_1 \cdot \vec{t}_2$ will be zero.

More generally, given terms $t_1, t_2, \ldots, t_n$ and a collection of documents of cardinality p, the number of "active" minterms is a *subset* of all possible minterms, which we denoted as $\{m\}_{2^n}$. Since, in the worst case, each document can correspond to a different minterm, the number of active basis vectors is at most p.

Thus, the expansion of $\vec{t}_i$, $1 \le i \le n$, involves only those basis (atomic) vectors restricted to the set of active minterms. It is understood, in subsequent discussions, that $\{\vec{m}\}^i$ refers to this subset of active basis vectors.

Another issue raised in the discussion above is the limitation of using only the cardinalities. A natural generalization, which considers the importance of a term to the documents (i.e., term weights), should be developed. For this purpose the following expression for (unnormalized) $\vec{t}_i$ is proposed:

$$\vec{t}_i = \sum_{m_k \in \{m\}^i} c_{i_k} \vec{m}_k, \tag{5.8}$$

where the unnormalized form of c_{i_k} is given by

$$c_{i_k} = \sum_{d_\alpha \in D_{m_k}} w_{\alpha i}. \tag{5.9}$$

The above concepts are illustrated by the following example.

Example 5.2. Given a set of documents $D = \{d_1, d_2, d_3, d_4\}$ indexed by the set of terms $T = \{t_1, t_2, t_3\}$. The weights of each term in the documents are given by the following matrix:

$$W = \left[\begin{array}{cccc} & t_1 & t_2 & t_3 \\ d_1 & 2 & 0 & 1 \\ d_2 & 1 & 0 & 0 \\ d_3 & 0 & 1 & 3 \\ d_4 & 2 & 0 & 0 \end{array}\right]$$

There are eight fundamental products or minterms, $\{m_1 = t_1t_2t_3$, $m_2 = \bar{t}_1t_2t_3$, $m_3 = t_1\bar{t}_2t_3$, $m_4 = t_1t_2\bar{t}_3$, $m_5 = \bar{t}_1\bar{t}_2t_3$, $m_6 = t_1\bar{t}_2\bar{t}_2$, $m_7 = \bar{t}_2t_2\bar{t}_3$, $m_8 = \bar{t}_1\bar{t}_2\bar{t}_3\}$, generated by the literals t_1, t_2, and t_3. In vector notation, these minterms can be represented explicitly by the following set of orthonormal basis vectors:

$$\begin{aligned}
\vec{m}_1 &= (1, 0, 0, 0, 0, 0, 0, 0)\\
\vec{m}_2 &= (0, 1, 0, 0, 0, 0, 0, 0)\\
\vec{m}_3 &= (0, 0, 1, 0, 0, 0, 0, 0)\\
\vec{m}_4 &= (0, 0, 0, 1, 0, 0, 0, 0)\\
\vec{m}_5 &= (0, 0, 0, 0, 1, 0, 0, 0)\\
\vec{m}_6 &= (0, 0, 0, 0, 0, 1, 0, 0)\\
\vec{m}_7 &= (0, 0, 0, 0, 0, 0, 1, 0)\\
\vec{m}_8 &= (0, 0, 0, 0, 0, 0, 0, 1)
\end{aligned}$$

Each $t_i \in T$ can be expressed in a disjunctive normal form as follows:

$$\begin{aligned}
t_1 &= t_1 \text{ AND } (t_2 \text{ OR } \bar{t}_2) \text{ AND } (t_3 \text{ OR } \bar{t}_3)\\
&= [(t_1 \text{ AND } t_2) \text{ OR } (t_1 \text{ AND } \bar{t}_2)] \text{ AND } (t_3 \text{ OR } \bar{t}_3)\\
&= (t_1 \text{ AND } t_2 \text{ AND } t_3) \text{ OR } (t_1 \text{ AND } \bar{t}_2 \text{ AND } t_3) \text{ OR}\\
&\quad\ (t_1 \text{ AND } t_2 \text{ AND } \bar{t}_3) \text{ OR } (t_1 \text{ AND } \bar{t}_2 \text{ AND } \bar{t}_3)\\
&= m_4 \text{ OR } m_1 \text{ OR } m_5 \text{ OR } m_2,\\
t_2 &= t_2 \text{ AND } (t_1 \text{ OR } \bar{t}_1) \text{ AND } (t_3 \text{ OR } \bar{t}_3)\\
&= m_4 \text{ OR } m_3 \text{ OR } m_5 \text{ OR } m_7,
\end{aligned}$$

and

$$\begin{aligned} t_3 &= t_3 \text{ AND } (t_1 \text{ OR } \bar{t}_1) \text{ AND } (t_2 \text{ OR } \bar{t}_2) \\ &= m_4 \text{ OR } m_1 \text{ OR } m_3 \text{ OR } m_6. \end{aligned}$$

From Eqs. (5.8), (5.9), and the above document matrix W, we obtain the following normalized term vectors:

$$\vec{t}_1 = \frac{2\vec{m}_1 + (1+2)\vec{m}_2 + 0\vec{m}_3}{[2^2 + 3^2]^{1/2}} = 0.55\vec{m}_1 + 0.83\vec{m}_2,$$

$$\vec{t}_2 = \frac{0\vec{m}_1 + (0+0)\vec{m}_2 + 1\vec{m}_3}{[1^2]^{1/2}} = \vec{m}_3,$$

$$\vec{t}_3 = \frac{1\vec{m}_1 + (0+0)\vec{m}_2 + 3\vec{m}_3}{[1^2 + 3^2]^{1/2}} = 0.32\vec{m}_1 + 0.95\vec{m}_3.$$

By substituting the above expressions for term vectors into Eq. (5.5), it follows:

$$\begin{aligned} \vec{d}_1 &= 2\vec{t}_1 + \vec{t}_3 = 2(0.55\vec{m}_1 + 0.83\vec{m}_2) + (0.32\vec{m}_1 + 0.95\vec{m}_3) \\ &= 1.42\vec{m}_1 + 1.66\vec{m}_2 + 0.95\vec{m}_3, \\ \vec{d}_2 &= \vec{t}_1 = 0.55\vec{m}_1 + 0.83\vec{m}_2, \\ \vec{d}_3 &= \vec{t}_2 + 3\vec{t}_3 = \vec{m}_3 + 3(0.32\vec{m}_1 + 0.95\vec{m}_2) \\ &= 0.96\vec{m}_1 + 3.85\vec{m}_3, \\ \vec{d}_4 &= 2\vec{t}_1 = 2(0.55\vec{m}_1 + 0.83\vec{m}_2) = 1.1\vec{m}_1 + 1.66\vec{m}_2. \end{aligned}$$

Similarly, we can transform, for example, the query vector, $\vec{q} = \vec{t}_1 + \vec{t}_2$, into a linear combination of atomic vectors, that is,

$$\begin{aligned} \vec{q} &= (0.55\vec{m}_1 + 0.83\vec{m}_2) + (\vec{m}_3) \\ &= 0.55\vec{m}_1 + 0.83\vec{m}_2 + \vec{m}_3. \end{aligned}$$

Then the cosine similarity $s_i = \vec{d}_i \cdot \vec{q}/|\vec{d}_i|\,|\vec{q}|$ between the normalized document d_α and the query q can be computed as follows:

$$s_1 = \frac{(1.42)(.55) + (1.66)(.83) + (.95)(1)}{[1.42^2 + 1.66^2 + 0.95^2]^{1/2}[0.55^2 + 0.83^2 + 1^2]^{1/2}} = 0.9234,$$

$$s_2 = \frac{(.55)(.55) + (.83)(.83) + (0)(1)}{[0.55^2 + 0.83^2]^{1/2}[0.55^2 + 0.83^2 + 1^2]^{1/2}} = 0.7056,$$

$$s_3 = \frac{(.96)(.55) + (0)(.83) + (3.85)(1)}{[0.96^2 + 3.85^2]^{1/2}[0.55^2 + 0.83^2 + 1^2]^{1/2}} = 0.7819,$$

$$s_4 = \frac{(1.1)(.55) + (1.66)(.83) + (0)(1)}{[1.1^2 + 1.66^2]^{1/2}[0.55^2 + 0.83^2 + 1^2]^{1/2}} = 0.7056.$$

Based on these similarity values ($s_1 > s_3 > s_2 \geq s_4$), d_1 will be retrieved first, d_3 second, and so on.

Before concluding this section, we relate the formulation presented above to earlier work. It should be noted that in Eq. (5.9) the c_{i_k}'s are obtained as the sum of term frequency weights, $w_{\alpha i}$'s, over those documents belonging to D_{m_k}. When the cardinality of D_{m_k} equals one, Eq. (5.8) can be rewritten as

$$\vec{t}_i = \sum_{m_k \in \{m^i\}} w_{\alpha i} \vec{m}_k. \tag{5.10}$$

In this special case, we obtain the following expression for term correlation:

$$\vec{t}_i \cdot \vec{t}_j = \sum_{\alpha=1}^{p} w_{\alpha i} w_{\alpha j}, \tag{5.11}$$

which is in fact similar to the formulas adopted in the experiments at Cornell [6]. Thus, we provide a rigorous framework in which the method of computing term correlations used many years ago can be justified. However, the question of how to incorporate this information into the retrieval process is answered very differently in this work (see Eq. (3.3)).

6. EXPERIMENTAL RESULTS

6.1 General Specifications

Four document collections are used for these experiments: ADINUL, CRN4NUL, MEDNUL, and MEDLARS. The collection characteristics are the following:

—ADINUL is a collection of 82 documents in library science. It consists of the full text of papers presented at American Documentation Institute meeting held in 1963. There are 35 queries.

—CRN4NUL has abstracts of 424 documents on aerodynamics, which were used by the Cranfield Project. The corresponding query collection involves 155 queries.

—MEDNUL is a collection of 450 documents and 30 queries. The documents are in the area of biomedicine.

—MEDLARS is a collection with 1,033 documents, also in biomedicine, and has 30 queries associated with it.

The indexing of the first three collections is done automatically in the SMART system [6], using the word-stem method. The last two collections are subsets of documents prepared by the National Library of Medicine. The query collections include, for evaluation purposes, information as to which documents are relevant to each query.

The standard recall and precision measures are used for comparing the performance of different strategies for weighting index terms. Recall is defined as the proportion of relevant documents retrieved and precision is the proportion of the retrieved documents actually relevant. The overall performance of a strategy is determined by processing the queries with that strategy and computing the average precision over all the queries for recall values 0.1, 0.2, . . . , and 1. The algorithm for averaging is consistent with that implemented in the SMART system.

The comparison of one method with another is accomplished by presenting the percentage improvement of both relative to a base strategy. The standard COSINE matching technique, where both documents and queries are weighted and terms are assumed to be pairwise orthogonal, is used as the base. In the tables consisting of experimental results, the set of precision values corresponding to the base method are labelled VSM.

6.2 Evaluation of the GVSM

As mentioned in Section 4, the term occurrence frequency data are the basis for prescribing the document vectors. Specifically, the αth document is given by

$$\vec{d}_\alpha = \sum_{i=1}^{n} w_{\alpha_i} \vec{t}_i,$$

where w_{α_i}, the component of $\vec{d}_\alpha$ along $\vec{t}_i$, is the number of times term t_i appears in the αth document. Equations (5.8) and (5.9) are used to obtain a vector representation for each term t_i. That is, each $\vec{t}_i$ is expressed as a linear combination of the orthonormal basis vectors $\{\vec{m}\}_{2^n}$. Given the term vectors, term-term similarities are computed in a straightforward manner (as the scalar product), and these are incorporated in the retrieval process, for ranking purposes, by using Eq. (3.3).

The retrieval performance, when this approach is used, is compared to that of the VSM in Tables I(a), I(b), I(c), and I(d) respectively for ADINUL, CRN4NUL, MEDNUL, and MEDLARS collection. The column of precision values for the proposed scheme is labelled GVSM.

Significant improvement over the standard implementation of the vector model is observed. The GVSM gives consistently better results in that the precision values are higher for every recall level and for all four collections tested. The ADINUL, CRN4NUL, and MEDLARS collections yield, respectively, average improvements of 29%, 20.5%, and 31.7%. For the MEDNUL collection, the improvement is an impressive 150.9%.

Although the standard vector model with COSINE similarity is often used as the base strategy, it may not be a fair comparison in our context since the standard model ignores term similarities. Unfortunately, there seems to be very little one can do to correct this. For one thing, any of the methods in earlier literature is not nearly as sound as the proposed scheme from a theoretical viewpoint. Moreover, earlier experiments based on term cooccurrence data have not led to worthwhile improvements in performance. For example, Minker et al. [3] conclude that, when retrieval is carried out using their scheme for query expansion, the only *significant* changes observed in overall performance are degradations, although some small improvements over limited portions of "recall" range can be realized in a few isolated instances. Salton has also noted that the utility of *fully* automatic methods of thesaurus construction is marginal at best, [7]. More precisely, automatic refinement of manually constructed thesaurus is believed to be the most promising. Consequently, earlier proposals for incorporating term-term similarities are not included for comparison.

There are, however, other well-known approaches in the literature that perform better than our base strategy. An example of such a method is to use the term frequency weights in combination with inverse document frequency (IDF) weights. We therefore use this kind of a weighting scheme for comparative

Table I (a) and (b). VSM versus GVSM

ADINUL (82 DOCS 35 QUES)			CRN4NUL (424 DOCS 155 QUES)		
	Precision			Precision	
Recall	VSM	GVSM	Recall	VSM	GVSM
0.1	.3786	.4587	0.1	.6415	.7005
0.2	.3434	.4253	0.2	.5540	.6250
0.3	.3094	.3433	0.3	.4514	.5246
0.4	.2587	.3289	0.4	.3621	.4459
0.5	.2465	.3104	0.5	.3249	.4040
0.6	.1887	.2613	0.6	.2726	.3251
0.7	.1357	.1993	0.7	.2059	.2502
0.8	.1283	.1879	0.8	.1655	.2084
0.9	.1092	.1361	0.9	.1241	.1574
1.0	.1082	.1353	1.0	.1179	.1492
Improvement		29.0%			20.5%

Table I (c) and (d). VSM versus GVSM

MEDNUL (450 DOCS 30 QUES)			MEDLARS (1033 DOCS 30 QUES)		
	Precision			Precision	
Recall	VSM	GVSM	Recall	VSM	GVSM
0.1	.4975	.7918	0.1	.7824	.8280
0.2	.3577	.7187	0.2	.6931	.7685
0.3	.3047	.6462	0.3	.5879	.6931
0.4	.2548	.6061	0.4	.5450	.6358
0.5	.2186	.5898	0.5	.4409	.5907
0.6	.1934	.5210	0.6	.3821	.5263
0.7	.1642	.4467	0.7	.3296	.4469
0.8	.1326	.3658	0.8	.2706	.3866
0.9	.0996	.3100	0.9	.1547	.2841
1.0	.0755	.2270	1.0	.0832	.1549
Improvement		150.9%			31.7%

purposes. The specific form of IDF weight adopted is as follows:

$$\mathrm{IDF}_{ij} = w_{ij} \cdot \left[\log \frac{N}{n_j} + 1\right],$$

where

N = the total number of documents in the collection, and
n_j = the total number of documents that contain term j.

In both VSM (IDF) and GVSM (BQ) methods, the query terms are assumed to be unweighted. The performance results in Tables II(a) and (b) and II(c) and (d) show that VSM (IDF) is better than VSM (see Tables I(a)–(d)) for each collection. Although GVSM (BQ) is not always better, in three out of the four collections GVSM gives a better retrieval performance over VSM. The percentage of

Table II (a) and (b). VSM(IDF) versus GVSM using Binary Queries

ADINUL (82 DOCS 35 QUES)			CRN4NUL (424 DOCS 155 QUES)		
	Precision			Precision	
Recall	VSM(IDF)	GVSM(BQ)	Recall	VSM(IDF)	GVSM(BQ)
0.1	.4714	.4572	0.1	.6817	.7110
0.2	.4523	.4237	0.2	.6129	.6345
0.3	.3941	.3631	0.3	.4973	.5324
0.4	.3509	.3290	0.4	.4107	.4456
0.5	.3453	.3090	0.5	.3690	.4026
0.6	.3038	.2607	0.6	.3050	.3239
0.7	.2230	.1906	0.7	.2269	.2507
0.8	.2143	.1831	0.8	.1804	.2078
0.9	.1842	.1406	0.9	.1359	.1563
1.0	.1825	.1397	1.0	.1287	.1482
Average improvement		−12%			+10%

Table II (c) and (d). VSM(IDF) versus GVSM using Binary Queries

MEDNUL (450 DOCS 30 QUES)			MEDLARS (1033 DOCS 30 QUES)		
	Precision			Precision	
Recall	VSM(IDF)	GVSM(BQ)	Recall	VSM(IDF)	GVSM(BQ)
0.1	.8247	.8199	0.1	.8511	.8400
0.2	.7339	.7497	0.2	.7682	.7868
0.3	.6455	.6714	0.3	.6848	.7194
0.4	.5239	.6416	0.4	.5965	.6454
0.5	.4759	.6199	0.5	.4954	.6026
0.6	.3966	.5474	0.6	.4183	.5345
0.7	.3592	.4584	0.7	.3531	.4398
0.8	.2586	.3577	0.8	.3010	.3751
0.9	.2010	.2967	0.9	.1848	.2746
1.0	.1450	.2200	1.0	.0865	.1576
Average improvement		+26%			+24%

improvement is computed by comparing the average precision over all recall values of a given strategy with the corresponding average of the base strategy. The improvement in the average precision can be imagined as an approximation of the percentage change in the area under the respective precision-recall curves. That is, the average improvement for GVSM is 10% in CRN4NUL, 26% in MEDNUL, and 24% in MEDLARS. The reason for the behavior in the ADINUL collection is possibly explained by the collection statistics presented in the subsequent sections.

In summary, we find the proposed scheme to determine term-term similarities and to incorporate them for ranking purposes to be very effective. Although the performance improvement achieved is very encouraging, the approach does involve a price in the form of computing resources. First, there is the cost associated with starting from the term distributional data and obtaining the vector representation of each document in terms of the $\vec{m}$'s, the fundamental

Table III. Distribution of Atoms in the Document Vectors

Coefficient	Total number of atoms			
of atom	ADINUL	CRN4NUL	MEDNUL	MEDLARS
0.00–0.05	2,027	144,747	133,321	879,485
0.05–0.10	3,744	23,886	7,881	32,632
0.10–0.20	830	4,164	2,078	5,903
0.20–0.30	23	309	298	593
0.30–0.40	0	90	59	90
0.40–0.50	0	105	14	95
0.50–0.60	2	106	15	173
0.60–0.70	25	89	29	271
0.70–0.80	36	54	99	306
0.80–0.90	19	14	197	163
0.90–1.00	0	2	107	28
ATOMS < .05	30%	83%	93%	96%

concepts. This is not a major concern, since it is a one-time cost. Second, there is an increase in the retrieval time, which is due to the fact that the similarity computation, as denoted by Eq. (3.3), is more complicated. This effect, in our implementation, comes through by way of increase in the number of nonzero coefficients in the typical document. That is, the *document-by-fundamental-concept* matrix is not as sparse as W. Since a substantial increase in retrieval time with respect to individual queries is not attractive, we perform some experiments that represent an approximation of the GVSM.

6.3 Experimental Evaluation of GVSM Approximations

6.3.1 *Component Approximation.* Although the document-by-fundamental-concept matrix is not sparse, we expect that many of the matrix elements to be extremely small. If this is the case, then by ignoring coefficients that are considered to be too small, the retrieval process can be speeded up. Conceptually, this is a way of approximating term correlations, which is somewhat analogous to the use, for example, of tree dependence in the context of probabilistic models [2, 13]. Thus our approach to the approximation of the GVSM consists of including only those $\vec{m}$'s having sufficiently large coefficients in the expansion of a document.

In order to determine the cut-off value where a certain component is small enough to be dropped, a frequency distribution of the values in the document-by-fundamental-concept matrix is computed. These values, for the four collections, are presented in Table III.

A careful study of the figures in Table III, as well as the data it is derived from, shows that a vast majority of the matrix elements are small. In the case of ADINUL, there is a distinct gap in the value of the coefficient of the domain $\vec{m}$ in any document versus the other $\vec{m}$'s. This is seen by the fact that there are exactly 82 coefficient values that are greater than 0.5. The values less than 0.3 correspond to nondominant $\vec{m}$'s. While the separation is less distinct in other collections, it is seen that a much higher percentage of values is small in the larger collections. From the distribution, it is decided that 0.05 is a reasonable

Table IV (a) and (b). Component Approximation versus GVSM

ADINUL (82 DOCS 35 QUES)			CRN4NUL (424 DOCS 155 QUES)		
	Precision			Precision	
Recall	GVSM	C-Approx.	Recall	GVSM	C-Approx.
0.1	.4587	.4315	0.1	.7005	.6855
0.2	.4253	.4162	0.2	.6250	.6132
0.3	.3433	.3380	0.3	.5246	.5181
0.4	.3289	.3254	0.4	.4459	.4241
0.5	.3104	.3066	0.5	.4040	.3877
0.6	.2613	.2545	0.6	.3251	.3032
0.7	.1993	.1897	0.7	.2502	.2356
0.8	.1879	.1708	0.8	.2084	.1955
0.9	.1361	.1234	0.9	.1574	.1475
1.0	.1353	.1227	1.0	.1492	.1406
Improvement over VSM		22.8%			15.0%
Improvement over GVSM		−4.7%			−4.5%

Table IV (c) and (d). Component Approximation versus GVSM

MEDNUL (450 DOCS 30 QUES)			MEDLARS (1033 DOCS 30 QUES)		
	Precision			Precision	
Recall	GVSM	C-Approx.	Recall	GVSM	C-Approx.
0.1	.7918	.7846	0.1	.8280	.8322
0.2	.7187	.7016	0.2	.7685	.7527
0.3	.6462	.6403	0.3	.6931	.7149
0.4	.6061	.5817	0.4	.6358	.6503
0.5	.5898	.5706	0.5	.5907	.5908
0.6	.5210	.5101	0.6	.5263	.5296
0.7	.4467	.4413	0.7	.4469	.4654
0.8	.3658	.3589	0.8	.3866	.3847
0.9	.3100	.3080	0.9	.2841	.2779
1.0	.2270	.2061	1.0	.1549	.1517
Improvement over VSM		143.8%			37.4%
Improvement over GVSM		−27.0%			4.0%

value of cut-off. The bottom row of the table shows that for the CRN4NUL, MEDNUL, and MEDLARS collections, the percentage of coefficients dropped is respectively 83, 93, and 96. This approximation would therefore cut down the retrieval time drastically, and the amount of savings is expected to increase with the size of the collection. In particular, the three larger collections should speed up the retrieval time by a factor of 8 to 10.

Table IV (a, b, c, and d) presents the performance results with the above approximation. For these experiments, the document representations are renormalized after coefficients below 0.05 are ignored. The columns of precision values for the approximation are labeled C-Approx. For ADINUL, CRN4NUL, MEDNUL, and MEDLARS, the C-Approx. is better than VSM by respectively 22.8%, 15%, 143.8%, and 37.4%. Furthermore, the approximation results are not significantly different from those of the GVSM. The differences vary, for the four collections, from −4.7% to +0.4%; very small indeed. As far as retrieval time is concerned, it is expected that cut-off can be small enough to achieve a speed comparable to, if not better than, VSM. Consequently, we believe that the proposed scheme is not only effective but can also be made very efficient.

6.3.2 *Dominant-Atom Approximation.* It is mentioned in Section 6.3.1 that we can associate a particular $\vec{m}$, the dominant $\vec{m}$, with each document. The dominant $\vec{m}$ is expected to receive the largest coefficient in the vector representation of the document. If each document is represented as a linear combination of several $\vec{m}$'s, then we have the situation where document-document similarities are assumed to exist. In contrast, if each document is simply represented just by a single $\vec{m}$, then the effect is to assume documents as being uncorrelated (just as a matter of convenience or simplification).

When such a simplification is made, the queries can still be handled as before. That is, term cooccurrences can still be used to obtain term-term similarities and to determine query vectors. Thus we have the special case of the GVSM where term similarities are not ignored, but documents are deemed orthogonal to each other.

This special case is very attractive from a computational point of view, because now the matrix (document-by-fundamental-concept) is diagonal. This means each document is described by a single $\vec{m}$ and the scalar product between a document and a query is simply the coefficient of that $\vec{m}$ in the query vector. It is easy to see that query-document similarity computation will be decreased drastically, assuming that the (one time) preprocessing to represent the index terms in terms of $\vec{m}$'s has been done.

The retrieval performance, when each document is represented only by the $\vec{m}$ with the largest coefficient, is summarized in Table V. The columns of precision values are labelled DA-Approx.

Naturally, the retrieval performance of DA-Approx. is not good in comparison to the GVSM, except in the case of the ADINUL collection. The ADINUL collection is a special case where, in our sense, documents indeed are almost orthogonal to each other. The more interesting result is that, for ADINUL and MEDNUL, DA-Approx. is better than the VSM by 28.3% and 82.5%, respectively. For the other two collections, DA-Approx. is only slightly worse (by about 10%). These results lead us to believe that DA-Approx. can be very attractive in order to provide a first-cut retrieval very quickly. Such a result can then be refined using relevance feedback techniques [6, 9, 11].

DA-approximation is not only interesting in terms of its computational economy, but also because of its theoretical connection to strict Boolean retrieval systems. In Section 5.2 it is mentioned that the $\vec{m}$'s comprise the basis of our vector space and that they are pairwise orthogonal. It follows, then, that DA-approximation is precisely the special case (with respect to Eq. (5.6) of GVSM) in which documents are deemed orthogonal. In a related paper [14], where an approach for extended Boolean queries is developed, the mapping of a strict Boolean system to our vector space results in each document being represented by its dominant atom.

Table V (a) and (b). Dominant-Atom Approximation versus GVSM.

ADINUL (82 DOCS 35 QUES)			CRN4NUL (424 DOCS 155 QUES)		
	Precision			Precision	
Recall	GVSM	DA-Approx.	Recall	GVSM	DA-Approx.
0.1	.4587	.4840	0.1	.7005	.6325
0.2	.4253	.4339	0.2	.6250	.5371
0.3	.3433	.3528	0.3	.5246	.4172
0.4	.3289	.3369	0.4	.4459	.3367
0.5	.3104	.3233	0.5	.4040	.3003
0.6	.2613	.2720	0.6	.3251	.2366
0.7	.1993	.1673	0.7	.2502	.1748
0.8	.1879	.1608	0.8	.2084	.1288
0.9	.1361	.1425	0.9	.1574	.0996
1.0	.1353	.1414	1.0	.1492	.0932
Improvement over VSM		28.3%			−11.7%
Improvement over GVSM		0.0%			−26.4%

Table V (c) and (d). Dominant-Atom Approximation versus GVSM

MEDNUL (450 DOCS 30 QUES)			MEDLARS (1033 DOCS 30 QUES)		
	Precision			Precision	
Recall	GVSM	DA-Approx.	Recall	GVSM	DA-Approx.
0.1	.7918	.7190	0.1	.8280	.7094
0.2	.7187	.6521	0.2	.7685	.6355
0.3	.6462	.5670	0.3	.6931	.5502
0.4	.6061	.5150	0.4	.6358	.4793
0.5	.5898	.4687	0.5	.5907	.4132
0.6	.5210	.3919	0.6	.5263	.3439
0.7	.4467	.3257	0.7	.4469	.2872
0.8	.3658	.2240	0.8	.3866	.2400
0.9	.3100	.1629	0.9	.2841	.1484
1.0	.2270	.1222	1.0	.1549	.0680
Improvement over VSM		82.5%			−9.9%
Improvement over GVSM		−21.5%			−31.9%

7. CONCLUSION

It is noted that there is a lack of a sound theoretical basis for determining term similarities and for incorporating such data in the retrieval process. A rigorous model based on the premises of the vector space theory, called the GVSM, is advanced as a solution. The GVSM involves the derivation of new (fundamental) concepts from the terms used to index documents in a collection and, subsequently, the use of these concepts as the basis vectors of the vector space of interest. It should perhaps be emphasized here that if the collection changes, theoretically the representation of each individual document in the entire collection must be modified. However, the same problem occurs in using IDF weights. Nevertheless, further investigation to resolve this problem would be beneficial.

The similarities between the original terms are then obtained by analyzing the occurrence distribution of the various terms in documents. In the context of the vector space model, the incorporation of term-term similarities into the retrieval process is straightforward.

Experiments are performed to demonstrate that the GVSM is more effective than the standard implementation of the vector space model, where terms are assumed to be pairwise orthogonal. Since the GVSM is computationally quite intense, two approximations to the GVSM are identified and tested empirically. These experiments indicate that, using our theoretical framework, IR systems that are both effective and computationally attractive can be developed.

REFERENCES

1. GORDON, M. D. A learning algorithm applied to document description. In *Proceedings of the 8th Annual International ACM-SIGIR Conference* (June 1985), ACM, New York, 179–186.
2. HARPER, D. J., AND VAN RIJSBERGEN, C. J. An evaluation of feedback in document retrieval using co-occurrence data. *J. Doc. 34* (1978), 189–216.
3. MINKER, J., WILSON, G. A., AND ZIMMERMAN, B. H. An evaluation of query expansion by the addition of clustered terms for a document retrieval system. *Inf. Storage Retrieval 8* (1972), 329–348.
4. RAGHAVAN, V. V., AND WONG, S. K. M. A critical analysis of vector space mode for information retrieval. *J. Am. Soc. Inf. Sci. 37*, 2 (Sept. 1986), 279–287.
5. RAGHAVAN, V. V., AND YU, C. T. Experiments on the determination of the relationships between terms. *ACM Trans. Database Syst. 4*, 2 (June 1979), 240–260.
6. SALTON, G., AND LESK, M. E. Computer evaluation of indexing and text processing. *ACM 15*, 1 (Jan. 1968), 8–36.
7. SALTON, G. Experiments in automatic thesaurus construction for information retrieval. *Information Processing 71*, North-Holland, Amsterdam, 1972, 115–123.
8. SALTON, G. Automatic term class construction using relevance—a summary of work in automatic pseudoclassification. *Inf. Process. Manage. 16*, 1 (1980), 1–15.
9. SALTON, G. *Dynamic Information and Library Processing.* Prentice-Hall, Englewood Cliffs, N.J., 1983.
10. SALTON, G., BUCKLEY, C., AND YU, C. T. An evaluation of term dependence models in information retrieval. In *Proceedings of the 5th Annual International ACM-SIGIR Conference* (1982), ACM, New York, 151–173.
11. SALTON, G., AND MCGILL, M. J. *Introduction to Modern Information Retrieval.* McGraw Hill, New York, 1983.
12. SPARCK-JONES, K. *Automatic Keyword Classifications.* Butterworths, London, 1971.
13. VAN RIJSBERGEN, C. J. A theoretical basis for the use of co-occurrence data in information retrieval. *J. Doc. 33* (1977), 106–119.
14. WONG, S. K. M., ZIARKO, W., RAGHAVAN, V. V., AND WONG, P. C. N. On extending the vector space model for Boolean query processing. In *Proceedings of the 9th Annual International ACM-SIGIR Conference* (1986), ACM, New York, 175–185.
15. WONG, S. K. M., ZIARKO, W., AND WONG, P. C. N. Generalized vector space model in information retrieval. In *Proceedings of the 8th Annual International ACM-SIGIR Conference* (1985), ACM, New York, 18–25.
16. ZUNDE, P., AND DEXTER, M. Indexing consistency and quality. *Am. Doc.* (July 1969).

Received May 1986; revised January 1987; accepted January 1987