A Measure Theoretic Approach to Information Retrieval

Sándor Dominich
University of Veszprém, Department of Computer Science,
Egyetem u. 10. 8200 Veszprém, Hungary,
Email: dominich@dcs.vein.hu

ABSTRACT
The vector space model of information retrieval is one of the classical and widely applied retrieval models. Paradoxically, it has been characterised by a discrepancy between its formal framework and implementable form. The underlying concepts of the vector space model are mathematical terms: linear space, vector, and inner product. However, in the vector space model, the mathematical meaning of these concepts is not preserved. They are used as mere computational constructs or metaphors. Thus, the vector space model actually does not follow logically from the mathematical concepts on which it has been claimed to rest. This problem has been recognised for more than two decades, but no proper solution has emerged so far. The present paper proposes just such a solution to this very problem. Firstly, the concept of retrieval is defined based on measure theory. Then, retrieval is particularised using fuzzy set theory. As a result, the retrieval function is conceived as the cardinality of the intersection of two fuzzy sets. This view makes it possible to build a connection to linear spaces. Thus, the classical and the generalised vector space models as well as the latent semantic indexing model gain a correct formal background with which they are consistent. At the same time it becomes clear that the inner product is not a necessary ingredient of the vector space model. Moreover, this view makes it possible to consistently formulate new retrieval methods: in linear space with general basis, entropy-based, and probability-based. Experimental results using standard test collections are also reported.

Categories and Subject Descriptors
H.3.3. **[Information Search and Retrieval]**: Retrieval Models.

General Terms
Algorithms, Measurement, Performance, Experimentation, Theory.

Keywords
General Basis, Measure Theory, Fuzzy Set Theory, Linear Space, Entropy, Probability, Inner Product, Relevance Feedback, Recursion Theorem, Precision-Recall, Mathematical Modelling.

1. INTRODUCTION
The formal mathematical framework for the classical vector space model (VSM) of Information Retrieval (IR) is the orthonormal Euclidean space [19], see [9] for an instructive reading). In a thought provoking theory paper, Wong and Raghavan [24] argue that:

"the notion of vector in the VSM merely refers to data structure… the scalar product is simply an operation defined on the data structure…The main point here is that the concept of a vector was not intended to be a logical or formal tool."

Further on, Wong and Raghavan point out why the VSM approach conflicts with the mathematical notion of a vector space. After a brief mathematical introduction, they rightly observe that the usual similarity functions, more exactly their numerator, which is a scalar product, can be written also in general basis. The metric tensor G, which they refer to as correlation matrix, of the space is interpreted as expressing correlations between index terms t_i, i=1,…,n, viewed as basis vectors, and hence it can be used as a model of term dependences: $G=(\mathbf{t}_i \cdot \mathbf{t}_j)_{n\times n}$, where $\mathbf{t}_i$ denotes the basis vector corresponding to term t_i. No experimental results are reported, and they conclude their paper by envisaging research into establishing the correlation matrix. In [25], an automatic method is proposed to build the correlation matrix G of index terms t_i. The value of the similarity S between a document and query is computed as the product between the query vector $\mathbf{q}$ (expressed in the general basis), the metric tensor G, and the document vector $\mathbf{d}$ in orthonormal basis: $S=\mathbf{q}^T \cdot G \cdot \mathbf{d}$. The method is referred to as the General VSM (GVSM). It is claimed that the expression of the similarity S is a generalisation (in vector space with general basis) of the usual $\mathbf{q}^T \cdot \mathbf{d}$ scalar product. However, this is wrong. The expression $\mathbf{q}^T \cdot G \cdot \mathbf{d}$ simply is an algebraic construct; it would be a scalar product if the document vector $\mathbf{d}$ was also written in the general basis. But in this case, the experimental results would have been identical with those of the classical VSM — because the scalar product is invariant with respect to the change of basis.

It can be seen that in the classical VSM as well as in the generalised VSM there is a discrepancy between the theoretical (mathematical) model and the algorithm applied in practice. They are not consistent with each other: the algorithm does not follow from the model, and conversely, the model is not a formal framework for the algorithm. In order to better render the rightfulness of the concerns with the mathematical modelling as well as of the mathematical subtleties involved, let us consider a simple example.

In the orthonormal Euclidean space, the basis vectors are pair-wise perpendicular to each other and have unit lengths. This property reflects the assumption that the index terms are independent of each other. Let us consider the orthonormal Euclidean space E_2, its unit length and perpendicular basis vectors are $\mathbf{e}_1$=(1,0) and $\mathbf{e}_2$=(0,1). Let us assume that we have the following two index terms: t_1='computer' and t_2='hardware', which correspond to the two basis vectors (or, equivalently, to coordinate axes) $\mathbf{e}_1$ and $\mathbf{e}_2$, respectively (Fig. 1). Consider now a document D being indexed by the term 'computer', and having the following weight vector: $\mathbf{D}$=(3,0). Let a query Q be indexed by the term 'hardware', and have the following weight vector: $\mathbf{Q}$=(0,2).

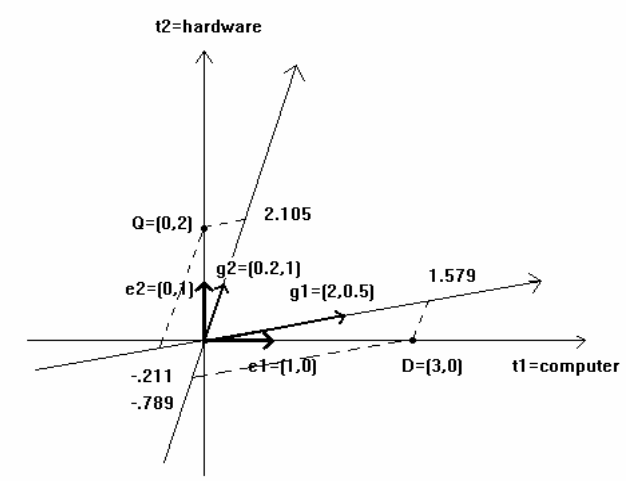


Figure 1. Document and query weight vectors. The document vector D(3;0) and query vector Q(0;2) are represented in the orthonormal basis (e1;e2). They are also represented in the general basis (g1;g2), these basis vectors are not perpendicular to each other, and do not have unit lengths. The coordinates of the document vector will be D(1.579;-0.789), whereas those of the query vector will be Q(-0.211;2.105). The value of the expression D·Q viewed as an inner product between document D and query Q is always zero, regardless of the basis. But the value of the expression D·Q literally viewed as algebraic expression is not zero

If we compute now the value of any typical similarity function ρ (dot product, cosine, etc.) we obtain the value null, because the numerator of every similarity function is the inner product **D·Q** which is equal to zero: **D·Q**=3×0+0×2=0. This means that the document *D* is not retrieved in response to query *Q*. However, because hardware is part of a computer, the user might be interested to see whether document *D* contains information also on hardware; in other words, he/she would not mind if document *D* would be returned in this case. It is well-known that the term independence assumption is not realistic. The terms may depend on each other, and they often do in practice, as in our example too. It is also known that the independence assumption can be counterbalanced, to a certain degree, in practice by, e.g., using thesauri. Can term dependence be captured and expressed in a vector space? One possible answer is as follows: instead of considering an orthonormal basis, let us consider a general basis. The basis vectors of a general basis need not be perpendicular to each other, and need not have unit lengths. In our example, the term 'hardware' is narrower in meaning than the term 'computer'. If orthogonal basis vectors are used to express the fact that two terms are independent, then a 'narrower' relationship can be expressed by taking an angle smaller than 90° (the exact value of this angle can be the subject of experimentation, but it is not important for the purpose of this example). So, let us consider the following two oblique basis vectors: let the basis vector $\mathbf{g}_1$ corresponding to term t_1 be $\mathbf{g}_1$=(2;0.5), and the basis vector $\mathbf{g}_2$ representing term t_2 be $\mathbf{g}_2$=(0.2;1). The coordinates $\mathbf{D}^i$ of the document vector **D** in the new basis are as follows [21]:

$$\mathbf{D}^i = (\mathbf{g}_i)^{-1} \times \mathbf{D} = (\mathbf{g}_1\ \mathbf{g}_2)^{-1} \times \mathbf{D} = \begin{pmatrix} 2 & 0.2 \\ 0.5 & 1 \end{pmatrix}^{-1} \times (3;\ 0)^{\mathrm{T}} =$$

$$= \begin{pmatrix} 0.526 & -0.105 \\ -0.263 & 1.053 \end{pmatrix} \times (3;\ 0)^{\mathrm{T}} = (1.579;\ -0.789),$$

whereas the coordinates $\mathbf{Q}^i$ of the query vector **Q** are as follows:

$$\mathbf{Q}^i = (\mathbf{g}_i)^{-1} \times \mathbf{Q} = (\mathbf{g}_1\ \mathbf{g}_2)^{-1} \times \mathbf{Q} = \begin{pmatrix} 2 & 0.2 \\ 0.5 & 1 \end{pmatrix}^{-1} \times (0;\ 2)^{\mathrm{T}} =$$

$$= (-0.211;\ 2.105).$$

Now, if the numerator of any similarity function (dot product, Cosine measure, Dice's, Jaccard's, Overlap coefficient) is interpreted — as is usual in IR — as being the expression of an inner product between the document vector and query vector, then the inner product of the document **D** and query **Q** is to be computed relative to the new basis $\mathbf{g}_i$=($\mathbf{g}_1$ $\mathbf{g}_2$) using the metric tensor $\mathbf{g}_{ij}$, and we obtain the following value:

$$\mathbf{D}^i \times \mathbf{g}_{ij} \times (\mathbf{Q}^j)^{\mathsf{T}} = (1.579;\ -0.789) \times \begin{pmatrix} g_1 g_1 & g_1 g_2 \\ g_2 g_1 & g_2 g_2 \end{pmatrix} \times$$

$$\times (-0.211;\ 2.105)^{\mathsf{T}} = (1.579;\ -0.789) \times \begin{pmatrix} 4.25 & 0.9 \\ 0.9 & 1.04 \end{pmatrix} \times$$

$$\times (-0.211;\ 2.105)^{\mathsf{T}} = 0,$$

i.e., the inner product of document **D** and query **Q** is equal to zero in the new basis too. This should not be a surprise because, as it is well-known, the scalar product is invariant with respect to the change of basis. This means that, under the inner product interpretation of — the numerator of — similarity, the no-hit case remains valid using general basis. The change of basis represents a "point of view" from which the properties of documents and queries are judged. If the document is conceived as being a vector, i.e., it is the same in any basis (equivalently, its meaning, information content, or properties remain the same in any basis) the inner product is also invariant, and so is the similarity function. But then, what is the point in taking a general basis? Let us assume instead that the — meaning or information content of — document and query do depend on the "point of view", i.e. on the change of basis. Then, the properties of documents and queries may be found to be different in different bases. This is equivalent to not interpreting the similarity function as expressing an inner product, rather being a numerical measure of how much the document and query share. Thus, the similarity, which formally looks like the algebraic expression of an inner product, is literally interpreted as a mere algebraic expression (or computational construct) being a measure of how much the document and query share, and not as expressing an inner product. In our example, we obtain the following value for the similarity between document and query: 1.579×(–0.211)+ +(–0.789)×(2.105)=–1.994. (Subjectively, a numerical measure of similarity should be a positive number, although this is irrelevant from a mathematical (e.g., ranking) point of view.) So, using general basis to express term dependence, and not interpreting similarity as being an inner product, the document *D* is being returned in response to *Q*, as intended.

This example as well as the modelling concerns earlier referred to justify the following question: Is or should the VSM be really based on inner product? In other words, is the inner product an underlying or necessary ingredient of IR?

In the present paper, an answer will be given to this question.

2. FORMAL FRAMEWORK FOR INFORMATION RETRIEVAL

In this part, several — commonly accepted — definitions of IR are recalled as they appeared in major works published in the field over the years. We note that these definitions are not definitions in a strict mathematical or logical sense, they are descriptions of what the concept of IR is or should be.

Salton [16] defines IR as follows: "The SMART retrieval system takes both documents and search requests in unrestricted English, performs a complete content analysis automatically, and retrieves those documents which most nearly match the given request."

Van Rijsbergen [22] gives the following definition: "In principle, information storage and retrieval is simple. Suppose there is a store of documents and a person (user of the store) formulates a question (request or query) to which the answer is a set of documents satisfying the information need expressed by this question."

Some years later, Salton [17] phrases as follows: "An automatic text-retrieval system is designed to search a file of natural-language documents and retrieve certain stored items in response to queries submitted by the user." The meaning of the word "certain" in the above quote is explained later on as follows: "The effectiveness of a retrieval system is usually evaluated in terms of…recall and precision…Both query formulation and document representations can be altered to reach the desired recall and precision levels."

Meadow, Boyce & Kraft [11] define IR as follows: "IR involves finding some desired information in a store of information or database. Implicit in this view is the concept of selectivity; to exercise selectivity usually requires that a price be paid in effort, time, money, or all three. Information recovery is not the same as IR…Copying a complete disk file is not retrieval in our sense. Watching news on CNN…is not retrieval either…Is information retrieval a computer activity? It is not necessary that it be, but as a practical matter that is what we usually imply by the term."

Berry & Browne [5] formulate as follows: "We expect a lot from our search engines. We ask them vague questions … and in turn anticipate a concise, organised response. … Basically we are asking the computer to supply the information we want, instead of the information we asked for. … In the computerised world of searchable databases this same strategy (i.e., that of an experienced reference librarian) is being developed, but it has a long way to go before being perfected."

Baeza-Yates and Ribeiro-Neto [1] write: "In fact, the primary goal of an IR system is to retrieve all the documents which are relevant to a user query while retrieving as few non-relevant documents as possible."

Belew [3], within his cognitive and articulate FOA (Finding Out About) framework, formulates retrieval in a pragmatic way as follows: "We will assume that the search engine has available to it a set of preexisting, 'canned' passages of text and that its response is limited to identifying one or more of these passages and presenting them to the users; see Figure 1.2." (Figure 1.2 shows a user having an information need, this need is being sent to a corpus of documents in the form of a query. Some process retrieves a subset of documents which is then sent back to the user.)

A few years later, Baeza-Yates [2] formulates similarly to his earlier view: "IR aims at modelling, designing, and implementing systems able to provide fast and effective content-based access to large amount of information. The aim of an IR system is to estimate the relevance of information items to a user information expressed in a query."

Taking a closer look at the above definitions given for IR, one can see that, in fact, they do not give *different* interpretations for IR, rather they all define IR *the same way*. All these definitions agree that: *IR means retrieving relevant documents in response to a query expressing a user's information need*. In other words, given documents, users, information needs, and queries — retrieve relevant documents for given queries. Analysing the concepts occurring in this definition, one can group them into two classes as follows:

a) Class 1 (concepts assumed to be given): user, information need, query, document;
b) Class 2 (concepts that express operation, process): relevance, retrieve.

Adopting a mathematical, somewhat axiomatic, approach towards defining IR, Class 1 may be conceived as containing basic concepts, whereas Class 2 as expressing certain connection or relationship between them. Thus, the following purely formal mathematical framework for IR can be formulated (Table 1):

Table 1. Formal mathematical framework for IR

Information Retrieval is a framework given by:	
Basic Concepts	Relationship
User, information need, query, document	For a given user, information need, and query, there exists a corresponding document.

The 'relationship' expresses a requirement, aim or wish. The word 'corresponding' should be understood as a synonym for appropriate, good, relevant. The term 'there exists' is to be interpreted as 'there exists at least one' (encompassing even the case when the only corresponding element is the empty set, i.e., no appropriate documents exist). Further, whether retrieval is query-driven, or query+user driven, or query+user+information need driven, or other mixture of these, is irrelevant from a formal mathematical point of view. Both the basic concepts and relationships should be viewed at an abstract level. In principle, it is actually irrelevant what the particular interpretations of the basic concepts and relationship are, or what the specific realizations or implementations of this latter might be. They may be interpreted abstractly, similarly to the way we interpret, for example, the basic notions in the axiomatic theory of Euclidean Geometry (where the notions of a point, line, and plane may be any three distinct entities). Likewise, the relationship may mean any kind of particular algorithm, relationship, method or process, similar to the free interpretation of the axioms (incidence, etc.) of Euclidean Geometry.

Usually, in practice, IR implies a computerised automatic retrieval system. The correlation degree between query and information need is a human rather than a computer matter (i.e., this degree can hardly be entirely automatised at our present knowledge). The extent to which a query reflects the information need chiefly depends on the user and less on a computer program. Thus, in a computerised automatic retrieval system there only are two basic concepts: query and document, which may be referred to as objects in general. Alternatively, one may say that the concepts 'user' and 'information need' condensate into one single concept: 'query' (of course, at different degrees of adequacy from, e.g., a semantic, subjective

or psychological point of view). So, one can re-define the formal framework of Table 1 as follows (Table 2):

Table 2. Formal mathematical framework for automatic computerised IR

Information Retrieval is a framework given by:	
Basic Concept	Relationship
Object	For a given object, there exists a corresponding object.

3. MEASURE THEORETIC FORMULATION OF INFORMATION RETRIEVAL

The objects in Table 2 may be viewed to form a set, but it is reasonable to assume more than this: the objects are not simply elements of this set, they are not independent and isolated elements of it gathered at random, rather they form some structure or system (from certain points of view such as, e.g., topic, application, etc.). The 'relationship' in Table 2 is an expression of retrieval: some mechanism or process takes an object and associates to it a subset of objects. This subset may even be empty, or it may consist of one or several objects. One way to characterise retrieval is to assume that it is based on what the given object and another object share (formally, whatever the word 'share' is taken to mean). Thus, looking at the framework defined in Table 2 from a purely formal mathematical perspective, IR may be defined as follows:

DEFINITION 3.1. *Let O be a space, and μ a measure on (O,Ω), $\Omega\subseteq 2^O$.* Retrieval *is a function $\rho:\Omega\to 2^O$, $\rho(o)=\{o_j|\mu(o\cap o_j)\geq\theta,\ j=1,\ldots,m\}$, where θ is a threshold value.* ■

Note. In practice, the elements of the set $\rho(o)$ may be sorted on the values $\mu(o\cap o_j)$ ascendingly, also a cut-off value may be applied. Formally, Def. 3.1 does not exclude such a possibility. Considering or not a cut-off or a threshold value, or sorting or not, does not influence the formal mathematical results of the present paper, and their treatment will be disregarded in the following.

Let us particularise the space O as follows. Consider a set $T=\{t_1,t_2,\ldots,t_i,\ldots,t_n\}$ of elements referred to as *terms* (or *index terms*), and let X denote a finite subset of the Cartesian product $T\times\mathbf{N}$ ($\mathbf{N}$ denotes the set of natural numbers), i.e., $X\subset T\times\mathbf{N}$:

$$T\times\mathbf{N} = T\times\{0,1,2,\ldots\} = \{(t_i,n_{ij}) \mid t_i\in T,\ n_{ij}\in\mathbf{N}\},$$
$$X = \{(t_1,0),(t_1,1),(t_1,2),\ldots,(t_1,n_{1N_1}),\ldots,(t_n,0),(t_n,1),$$
$$(t_n,2),\ldots,(t_n,n_{1N_n})\} \quad (1)$$

Let us consider the set O as being the powerset 2^X, i.e., $O=2^X$. Thus, O is a discrete topological space on X. Let $\Omega\subseteq 2^O$ such that $\Omega=\{o_j|o_j=\{(t_i,n_{ij}),t_i\in T\}$, i.e., in every o_j any term t_i occurs at most once. In the sequel, o_j will be referred to as object. Any object o_j can be viewed as being a fuzzy set:

$$o_j = \{(t_i,\varphi_j(t_i)) \mid t_i\in T,\ i\in\{1,\ldots,n\}\} \quad (2)$$

where $\varphi_j(t_i)=n_{ij}$; notation: $o_j=\varphi_j(t_i)$.

EXAMPLE 3.1. Let the set T of terms be as follows:

$$T = \{t_1,t_2\} = \{t_1=\text{people},\ t_2=\text{old}\},$$

and let

$$X = \{(\text{people},0),(\text{people},1),(\text{old},0)\}.$$

Then,

$$O = 2^X = \{\varnothing, o_1, o_2, o_3, o_4, o_5, o_6, o_7\} = \{\varnothing,$$
$$o_1=\{(\text{people},0)\},\ o_2=\{(\text{people},1)\},\ o_3=\{(\text{old},0)\},$$
$$o_4=\{(\text{people},0),(\text{people},1)\},\ o_5=\{(\text{people},0),(\text{old},0)\},$$
$$o_6=\{(\text{people},1),(\text{old},0)\},\ o_7=\{(\text{people},0),(\text{people},1),(\text{old},0)\}\ \},$$

where, for example,

$$o_6=\{(\text{people},1),(\text{old},0)\}=\{(t_1,\varphi_6(t_1)),(t_2,\varphi_6(t_2))\}=\{(t_1,1),(t_2,0)\}.$$

Thus, $\Omega\subseteq 2^O$ will be as follows:

$$\Omega = \{\varnothing, o_1, o_2, o_3, o_5, o_6\} =$$
$$= \{\varnothing,\{(\text{people},0)\},\{(\text{people},1)\},\{(\text{old},0)\}\},$$
$$\{(\text{people},0),(\text{old},0)\},\{(\text{people},1),(\text{old},0)\}\ \}.$$ ■

Any object may, but need not, be normalised, for example as follows:

$$o_j = \{(t_i,\varphi_j(t_i))|t_i\in T,\ i\in\{1,\ldots,n\}\},\ \varphi_j(t_i)=n_{ij}/\max_i n_{ij} \quad (3)$$

Note. A fuzzy set is usually defined as being a set of ordered pairs $(t_i,\varphi_j(t_i))$. As this requirement does not influence the results of the present paper, it will be disregarded.

In practice, an automatic IR system works in 'elementary' steps, it answers a query by performing an 'indivisible' retrieval operation (i.e., an algorithm which is viewed as a unified and consistent process that answers the query, and that cannot be broken down into sub-algorithms answering that query). Thus, the following definition can be introduced.

DEFINITION 3.2. *The triple $(O,\mu,\rho(q))$, $q\in O$, is called* elementary retrieval*; notation: $(O,\mu,\rho(q))=\boldsymbol{A}_q$.* ■

4. INFORMATION RETRIEVAL AND LINEAR SPACES

In fuzzy set theory [26], the intersection of two fuzzy sets, i.e., objects $o_p=\varphi_p(t_i)$ and $o_q=\varphi_q(t_i)$ can be defined as the algebraic product as follows:

$$o_p\cap o_q = \{(t_i,\varphi_s(t_i)) \mid t_i\in T,\ \varphi_s(t_i)=\varphi_p(t_i)\cdot\varphi_q(t_i)\}, \quad (4)$$

their union can be defined as the algebraic sum as follows:

$$o_p\cup o_q =$$
$$\{(t_i,\varphi_s(t_i)) \mid t_i\in T,\ \varphi_s(t_i)=\varphi_p(t_i)+\varphi_q(t_i)-\varphi_p(t_i)\cdot\varphi_q(t_i)\}. \quad (5)$$

The cardinality κ of an object o_j can be defined as the sum of the values of its membership function as follows:

$$\kappa(o_j) = \sum_{i=1}^{n}\varphi_j(t_i) \quad (6)$$

EXAMPLE 3.2. From Example 3.1., $o_6=\{(\text{people},1),(\text{old},0)\}$, and so $\kappa(o_6)=\varphi_6(t_1)+\varphi_6(t_2)=1+0=1$. ■

The empty object $\varnothing$' is the following object $\varnothing'=\{(t_i,0)|t_i\in T\}$. It can be shown that:

THEOREM 4.1. *The cardinality κ is a measure on* (O,Ω).

PROOF. We have to show that:
a) The cardinality of the empty object is equal to zero. This is immediate:

$$\kappa(\varnothing') = \kappa(\{(t_i, 0) \mid t_i \in T\}) = \Sigma_i \varphi_j(t_i) = \Sigma_i 0 = 0$$

b) The cardinality of two disjoint objects o_p and o_q is equal to the sum of their cardinalities. Let

$$o_p \cap o_q = \{ (t_i, \varphi_s(t_i)) \mid t_i \in T,\ \varphi_s(t_i) = \varphi_p(t_i)\cdot\varphi_q(t_i) \} = \varnothing' \Leftrightarrow$$

$$\Leftrightarrow \varphi_p(t_i)\cdot\varphi_q(t_i) = 0,\ \forall p, q.$$

Hence,

$$\kappa(o_p \cup o_q) =$$

$$= \kappa[\{(t_i, \varphi_s(t_i)) \mid t_i \in T,\ \varphi_s(t_i) = \varphi_p(t_i) + \varphi_q(t_i) - \varphi_p(t_i)\cdot\varphi_q(t_i)\}] =$$

$$= \kappa[\{(t_i, \varphi_s(t_i)) \mid t_i \in T,\ \varphi_s(t_i) = \varphi_p(t_i) + \varphi_q(t_i)\}] =$$

$$= \kappa(o_p) + \kappa(o_q).\blacksquare$$

An immediate consequence of Theorem 4.1 is the following:

LEMMA 4.1.

$$\rho(o)=\{o_j|\mu(o\cap o_j)=\kappa(o\cap o_j)=\sum_{i=1}^{n}\varphi(t_i)\cdot\varphi_j(t_i)\geq\theta\} \qquad (7)$$

PROOF. According to Theorem 4.1 the cardinality κ is a measure. From relationship (4) the result is immediate.$\blacksquare$

EXAMPLE 3.3. From Example 3.1., o_2={(people,1)}, o_6={(people,1), (old,0)}. Then, $\mu(o_2\cap o_6)=1\times1+0\times0=1$.$\blacksquare$

4.1 Information retrieval in linear space

It is possible to 'embed' any elementary retrieval **A** in an n-dimensional real linear space **L** as follows. Every term t_i, i=1,…,n, is assigned to the basis vector $\mathbf{b}_i$ of the space **L**. Consider an arbitrary object $o_j=\varphi_j(t_i)$. The values $\varphi_j(t_1),\ldots,\varphi_j(t_n)$ of the membership function can be used to form a vector $\mathbf{v}_j$ of the space **L** as the following linear combination: $\mathbf{v}_j=\varphi_j(t_1)\cdot\mathbf{b}_1+\ldots+\varphi_j(t_n)\cdot\mathbf{b}_n$. Thus, every object o_j may be viewed as corresponding to a vector $\mathbf{v}_j\in\mathbf{L}$ with co-ordinates $\varphi_j(t_i)$, i.e., $o_j\equiv\mathbf{v}_j=(\varphi_j(t_1),\ldots,\varphi_j(t_n))$. Depending on certain properties of the linear space **L** (general, orthogonal or orthonormal basis, with or without inner product), different retrievals **A** can be obtained.

4.1.1 General basis-based retrieval method (GB method)

Given a linear space **L** with a general basis $\mathbf{g}_i$. The space **L** may but need not have an inner product (e.g., Banach space). The following retrieval method is proposed:

i) Let any object o_j correspond to a vector $\mathbf{v}_j$ in this space **L** such that the value $\varphi_j(t_i)$, i=1,…,n, corresponds to the ith coordinate of the vector $\mathbf{v}_j$.

ii) The retrieval function is given by formula (7).

From the point of view of an implementation, the following retrieval algorithm may be formulated. Given a set of index terms $T=\{t_1,\ldots,t_n\}$ and a collection of documents D_j, j=1,…,m. Let $TD_{n\times m}=(f_{ij})_{n\times m}$ denote the frequency term-by-document matrix of term occurrences, i.e., f_{ij} denotes the number of times term t_i occurs in document D_j. Similarly, let us consider a query $Q=(f_1,\ldots,f_i,\ldots,f_n)$, where f_i denotes the number of times term t_i occurs in query Q. Compute, using some weighting scheme, a term-by-document weight matrix $W_{n\times m}=(\mathbf{d}^{<j>})_m=(w_{ij})_{n\times m}$ for documents, and one for the query $\mathbf{q}=(q_1,\ldots,q_i,\ldots,q_n)$. Both $\mathbf{d}^{<j>}$ and $\mathbf{q}$ are viewed as being vectors in a linear space with orthonormal basis. Let us consider a general basis $\mathbf{g}_i$ now:

$$\mathbf{g}_i = (\mathbf{g}_1\ldots\mathbf{g}_n) = \begin{pmatrix} g_{11} & \cdots & g_{1n} \\ . & \cdots & . \\ g_{n1} & \cdots & g_{nn} \end{pmatrix}, \qquad (8)$$

where the coordinates of the basis vector $\mathbf{g}_1$ are $g_{11},\ldots,g_{n1}$, and so on. Compute the coordinates $\mathbf{d}'^{<j>}=(w'_{1j},\ldots,w'_{ij},\ldots,w'_{nj})$ of every document D_j in the general basis as follows:

$$\mathbf{d}'^{<j>} = (\mathbf{g}_i)^{-1}\cdot\mathbf{d}^{<j>}, \qquad (9)$$

where $(\mathbf{g}_i)^{-1}$ denotes the inverse of the basis tensor $\mathbf{g}_i$ (given by (8)). Similarly, the coordinates $\mathbf{q}'=(q'_1,\ldots,q'_i,\ldots,q'_n)$ of the query in the general basis are computed as follows [21]:

$$\mathbf{q}' = (\mathbf{g}_i)^{-1}\cdot\mathbf{q}. \qquad (10)$$

The similarity s_j between a document D_j and query Q is computed using formula (7) of Lemma 4.1 as follows:

$$s_j = \sum_{i=1}^{n} q'_i\cdot w'_{ij} \qquad (11)$$

Note that the expression (11) for the similarity s_j must not have the meaning of an inner product between document and query vectors. It is a measure given by formula (7) according to Def 3.1. It would only be the expression of an inner product if the coordinates of the document vector $\mathbf{d}'$ and those of query vector $\mathbf{q}'$ were in reciprocal bases, i.e., covariant and contravariant, respectively. Or, if expression (11) for s_j contained the metric tensor $\mathbf{g}_{ij}$.

Part 8 reports on experimental results using the GB retrieval method on standard test collections.

EXAMPLE 4.1. See the example given in Introduction.$\blacksquare$

4.2 Latent Semantic Indexing retrieval

Retrieval in the Latent Semantic Indexing (LSI) model [8] [4] can be viewed as an application of Lemma 4.1. Given an approximation matrix A_k of the term-by-document matrix A. Let $\mathbf{b}$ denote a basis of the column space of A_k. The retrieval function is the cardinality measure between (i) the coordinates, in basis $\mathbf{b}$, of the projection of the query vector $\mathbf{q}$ onto the column space of A_k, and (ii) the coordinates of the columns of A_k in basis $\mathbf{b}$.

Let $A=(w_{ij})_{n\times m}$ be a term-by document matrix, where w_{ij} is the weight of term t_i in document D_j, and $\mathbf{q}$ be a query vector. The matrix A is decomposed using singular value decomposition (SVD):

$$A = U\cdot S\cdot V^T,$$

where U and V are orthogonal matrices defining the left and right singular values of A, respectively, whereas S is the diagonal matrix of singular values of A arranged in descending order from the top of the main diagonal downwards. The rank r_A of the matrix A is equal to the number of nonzero diagonal elements of S. The first r_A columns of U (from left to right) are a basis for the column space of A. The rank-k, $k \leq \text{rank}(A)$, approximation of the matrix A is given by considering only the first k singular values in S:

$$A_k = U_k \cdot S_k \cdot V_k^{\mathsf{T}},$$

(equivalently, U_k contains the first k columns of U). The columns of the matrix A_k span a k-dimensional subspace of the column space of A. The columns of the matrix A_k are vectors whose coordinates are given by $S_k \cdot V_k^{\mathsf{T}}$ in the basis defined by the columns of U_k. The query is matched against the columns of A_k using formula (7) of Lemma 4.1, i.e., $\mathbf{q}^{\mathsf{T}} \cdot A_k$, which re-writes as follows:

$$\mathbf{q}^{\mathsf{T}} \cdot A_k =$$

$$= A_k^{\mathsf{T}} \cdot \mathbf{q} = (U_k \cdot S_k \cdot V_k^{\mathsf{T}})^{\mathsf{T}} \cdot \mathbf{q} = (V_k \cdot S_k^{\mathsf{T}} \cdot U_k^{\mathsf{T}}) \cdot \mathbf{q} = (V_k \cdot S_k^{\mathsf{T}}) \cdot (U_k^{\mathsf{T}} \cdot \mathbf{q}) =$$

$$= (S_k \cdot V_k^{\mathsf{T}}) \cdot (U_k^{\mathsf{T}} \cdot \mathbf{q}).$$

The elements of the vector $U_k^{\mathsf{T}} \cdot \mathbf{q}$ are the coordinates in the basis defined by the columns of U_k of the projection $U_k \cdot U_k^{\mathsf{T}} \cdot \mathbf{q}$ of the query vector $\mathbf{q}$ onto the column space of A_k.

If the expression $\mathbf{q}^{\mathsf{T}} \cdot A$ is interpreted as being the numerator of the traditional cosine measure, i.e., as having the meaning of a scalar product (i.e., accepting the document and query invariance principle DQIP), then it should be the same as $\mathbf{q}^{\mathsf{T}} \cdot A_k$, i.e., $\mathbf{q}^{\mathsf{T}} \cdot A = \mathbf{q}^{\mathsf{T}} \cdot A_k$. We have equality when $A = A_k$, which only occurs for $k = \text{rank}(A)$. Otherwise, the expression $\mathbf{q}^{\mathsf{T}} \cdot A_k$ does not have the meaning of a scalar product; it is a cardinality measure.

4.3 Information retrieval in linear space with inner product

If the space $\mathbf{L}$ has an inner product, e.g., it is a Hilbert space, and if the relevance measure μ in formula (7) is interpreted as being the expression of the inner product of the space $\mathbf{L}$, then the following holds:

THEOREM 4.2. *Given a real Hilbert space* $\mathbf{L}$ *for object-documents D having weight vectors* $\mathbf{w} \in \mathbf{L}$*, and an inner product similarity function* $\rho: \mathbf{L} \times \mathbf{L} \to \boldsymbol{R}_+$ ($\boldsymbol{R}_+$ *denotes the set of positive real numbers*)*. Retrieval relative to an object-query Q represented as a vector* $\mathbf{q} \in \mathbf{L}$ *can be performed using a projector* P_A *as follows:* $\mathfrak{R}_Q = \{D | P_A(\mathbf{w}) = \mathbf{v} + \mathbf{q},\ \mathbf{v} \in A = \{\mathbf{q}\}^{\perp}\}$.

PROOF. Retrieval of object-documents D represented as vectors $\mathbf{w}$ in response to an object-query Q represented as a vector $\mathbf{q}$ means constructing the set (inner product similarity function by assumption)

$$\mathfrak{R}_Q = \{D \mid \mathbf{q} \cdot \mathbf{w} \neq 0\}.$$

The orthogonal complement $A = \{\mathbf{q}\}^{\perp}$ (i.e., the set of object-documents which do not share common terms with the object-query) corresponding to the query Q is given by those object-documents D whose vectors $\mathbf{w}$ are perpendicular to $\mathbf{q}$, i.e.,

$$A = \{\mathbf{q}\}^{\perp} = \{\mathbf{w} \mid \mathbf{w} \perp \{\mathbf{q}\}\} = \{\mathbf{w} \mid \mathbf{w} \cdot \mathbf{q} = 0\}.$$

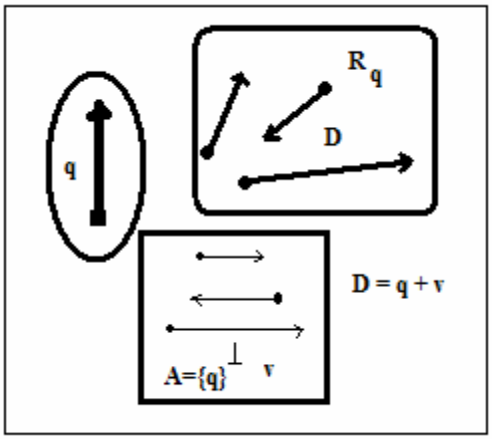
Hence, the set $\mathfrak{R}_Q$ is equal to the set of those object-documents whose vectors form the complement $\mathsf{C}_{\mathbf{L}}(A)$ of the set A relative to the space $\mathbf{L}$ (apart from the query Q itself, of course), i.e.,

$$\mathfrak{R}_Q = \{D \mid \mathbf{q} \cdot \mathbf{w} \neq 0\} = \{D \mid \mathbf{w} \in \mathsf{C}_{\mathbf{L}}(A) \setminus \{Q\}\}.$$

Because the set A is a closed linear subspace of the space $\mathbf{L}$ it follows that any element $\mathbf{w} \in \mathbf{L}$ of the space $\mathbf{L}$ can be uniquely written as $\mathbf{w} = \mathbf{v} + \mathbf{u}$, where $\mathbf{v} \in A$ and $\mathbf{u} \in A^{\perp}$. The projector $P_A(\mathbf{w})$ for the elements $\mathbf{w} \in \mathbf{L}$ of the space $\mathbf{L}$ on the set A is defined as $P_A(\mathbf{w}) = \mathbf{v} + \mathbf{u}$, where $\mathbf{v} \in A$ and $\mathbf{u} \in A^{\perp}$. Thus, the set $\mathfrak{R}_Q$ contains those object-documents D whose vectors $\mathbf{w}$ are so projected on A that $\mathbf{u} = \mathbf{q}$, i.e.,

$$\mathfrak{R}_Q = \{D \mid P_A(\mathbf{w}) = \mathbf{v} + \mathbf{q},\ \mathbf{v} \in A = \{\mathbf{q}\}^{\perp}\}. \blacksquare$$

EXAMPLE 4.2.



■

Van Rijsbergen [23] gives a geometrical treatment of retrieval in Hilbert space.

4.4 Information retrieval in orthonormal real linear space

The classical vectors space model (VSM) of IR is a special case in that the linear space $\mathbf{L}$ is the orthonormal Euclidean space. Salton, Wong and Yang [19] use a linear space as a formal framework for automatic indexing and retrieval. More than a decade later, Salton and Buckley [18] re-use this framework, and formulate as follows:

"In the late 1950's, Luhn first suggested that automatic text retrieval systems could be designed based on a comparison of content identifiers attached both to the stored texts and to the users' queries. The documents would be represented by term vectors of the form $D = (t_i, t_j, \ldots, t_p)$ where each t_k identifies a content term assigned to some sample document D. Analogously, a typical query vector might be formulated as $Q = (q_a, q_b, \ldots, q_r)$. A more formal representation of the term vectors is obtained by including in each term vector all possible content terms allowed in the system and adding term weight assignments to provide distinctions among terms. Thus, if w_{dk} (or w_{qk}) represents the weight of term t_k in document D (or query Q), and t terms in all are available for content representation, the term vectors for document and query can be written as $D = (t_0, w_{d0};\ t_1, w_{d1}; \ldots; t_t, w_{dt})$ and $Q = (q_0, w_{q0};\ q_1, w_{q1}; \ldots; q_t, w_{qt})$. Given the vector representations, a query-document similarity value may be obtained by comparing the corresponding vectors, using for example the conventional

vector product formula similarity $(Q,D)=\Sigma w_{qk}w_{dk}$. When the term weights are restricted to 0 and 1 as previously suggested, the vector product measures the number of terms that are jointly assigned to query Q and document D. In practice it has proven useful to provide a greater degree of discrimination among terms assigned for content representation than is possible with weights of 0 and 1 alone. The weights could be allowed to vary continuously between 0 and 1, the higher weight assignment near 1 being used for the most important terms, whereas lower weights near 0 would characterize the less important terms. A typical term weight using a vector length normalisation factor is $w_{dk}/(\Sigma_{vector}(w_{di})^2)^{1/2}$ for documents. When a length normalized term-weighting system is used with the vector similarity function, one obtains the well-known cosine similarity formula."

One can easily see that a document $D=(t_0,w_{d0}; t_1,w_{d1};\ldots;t_t,w_{dt})$ and query $Q=(q_0,w_{q0}; q_1,w_{q1};\ldots;q_t,w_{qt})$ are objects, and the similarity is a measure of the intersection. Any object o may be conceived as being an element of the real linear space $\mathbf{R}^n$, $n=t+1$, of n-tuples in **R,** i.e., $(w_{o0},w_{o1},\ldots,w_{ot})\in\mathbf{R}^n$. The similarity is the inner product of this space. Because there is a one-to-one correspondence between the linear space $\mathbf{R}^n$ and the n-dimensional Euclidean space E_n, we may say that the formal framework of VSM is E_n. Thus, each term t_i corresponds to a basis vector $\mathbf{e}_i$ in the orthonormal linear space E_n. As E_n is at the same time a Hilbert space relative to similarity, Theorem 4.2 holds.

5. ENTROPY-BASED INFORMATION RETRIEVAL

Apart from the GB retrieval method in linear space with general basis proposed in part 4.1.1, another retrieval method can also be proposed as follows. We particularise the set O of objects as in part 3. Let $H(o_j)$ denote an entropy [20] of o_j as follows ($\varphi_j(t_i)$ can be so normalised as to sum up to unity):

$$H(o_j)= -\sum_{i=1}^{n}\varphi_j(t_i)\cdot\log_b(\varphi_j(t_i)) \qquad (12)$$

where $b>1$.

EXAMPLE 5.1. From Example 3.1., $H(o_2)=H(\{(\text{people},1)\})=\varphi_2(t_1)\cdot\log(\varphi_2(t_1))=1\cdot\log(1)=0$.■

It can be shown that:

THEOREM 5.1. *The entropy H is a measure on* (O,Ω).

PROOF. We have to show that:
a) The entropy of the empty object is equal to zero. This is immediate:

$$H(\varnothing') = H(\{(t_i,0) \mid t_i \in T\}) =$$

$$\lim_{\varphi_j(t_i)\to 0}\left(-\sum_{i=1}^{n}\varphi_j(t_i)\cdot\log(\varphi_j(t_i))\right) = 0.$$

b) The entropy of two disjoint objects o_p and o_q is equal to the sum of their entropies. Let

$$o_p \cap o_q = \{(t_i,\varphi_s(t_i)) \mid t_i \in T,\ \varphi_s(t_i) = \varphi_p(t_i)\cdot\varphi_q(t_i)\} = \varnothing' \Leftrightarrow$$

$$\Leftrightarrow \varphi_p(t_i)\cdot\varphi_q(t_i) = 0,\ \forall p, q.$$

Hence:

$$H(o_p \cup o_q) =$$

$$H[\{(t_i, \varphi_s(t_i)) \mid t_i \in T,\ \varphi_s(t_i) = \varphi_p(t_i) + \varphi_q(t_i) - \varphi_p(t_i)\cdot\varphi_q(t_i)\}] =$$

$$= H[\{(t_i, \varphi_s(t_i)) \mid t_i \in T,\ \varphi_s(t_i) = \varphi_p(t_i) + \varphi_q(t_i)\}] =$$

$$= H(o_p) + H(o_q).\blacksquare$$

An immediate consequence of Theorem 5.1 is the following:

LEMMA 5.2.

$$\rho(o) = \{o_j \mid \mu(o \cap o_j) =$$

$$-\sum_{i=1}^{n}\varphi(t_i)\cdot\varphi_j(t_i)\cdot\log_b(\varphi(t_i)\cdot\varphi_j(t_i)) \geq \theta\}. \qquad (13)$$

PROOF. According to Theorem 5.1 the entropy H is a measure. From relationship (4) the result is immediate.■

5.1 Entropy-based retrieval method (E method)

From the point of view of an implementation, the following retrieval method may be formulated. Given a set of index terms $T=\{t_1,\ldots,t_n\}$ and a collection of documents D_j, $j=1,\ldots,m$. Let $W_{n\times m}=(w_{ij})_{n\times m}$ denote a term-by-document matrix, where w_{ij} is the weight of term t_i in document D_j. Given a query Q, query weights $q_1,\ldots,q_i,\ldots,q_n$ are computed; q_i denotes the weight of term t_i in Q. The similarity s_j between a document D_j and query Q and is computed using formula (13) as follows:

$$s_j = -\sum_{i=1}^{n} q_i\cdot w_{ij}\cdot\log(q_i\cdot w_{ij})\cdot$$

Part 8 reports on experimental results using the E retrieval method on standard test collections.

6. PROBABILITY-BASED INFORMATION RETRIEVAL

Let $p(t_i)$ denote a Kolmogoroff (1950) probability of term $t_i\in T$, $i=1,\ldots,n$, in O. In fuzzy set theory, the fuzzy probability $P(o_j)$ of an object $o_j=\{(t_i,\varphi_j(t_i))|t_i\in T,\ i\in\{1,\ldots,n\}\}$ is defined as follows:

$$P(o_j) = \sum_{i=1}^{n}\varphi_j(t_i)\cdot p(t_i) \qquad (14)$$

It can be shown that:

THEOREM 6.1. *The fuzzy probability P is a measure on* (O,Ω).

PROOF. We have to show that:
a) The fuzzy probability of the empty object is equal to zero. This is immediate:

$$P(\varnothing') = P(\{(t_i, 0) \mid t_i \in T\}) = \sum_{i=1}^{n} 0\cdot p(t_i) = 0.$$

b) The fuzzy probability of two disjoint objects o_p and o_q is equal to the sum of their fuzzy probabilities. Let

$$o_p \cap o_q = \{(t_i,\varphi_s(t_i)) \mid t_i \in T,\ \varphi_s(t_i) = \varphi_p(t_i)\cdot\varphi_q(t_i)\} = \varnothing' \Leftrightarrow$$

$$\Leftrightarrow \varphi_p(t_i)\cdot\varphi_q(t_i) = 0,\ \forall p, q.$$

Hence:

$$P(o_p \cup o_q) =$$

$$= P[\{(t_i, \varphi_s(t_i)) \mid t_i \in T,\ \varphi_s(t_i) = \varphi_p(t_i) + \varphi_q(t_i) - \varphi_p(t_i)\cdot\varphi_q(t_i)\}] =$$

$$= P[\{(t_i, \varphi_s(t_i)) \mid t_i \in T,\ \varphi_s(t_i) = \varphi_p(t_i) + \varphi_q(t_i)\}] =$$

$$= \sum_{i=1}^{n} (\varphi_p(t_i) + \varphi_q(t_i))\, p(t_i) = P(o_p) + P(o_q).\blacksquare$$

An immediate consequence of Theorem 6.1 is the following:

LEMMA 6.2.

$$\rho(o) = \{o_j \mid \mu(o \cap o_j) = P(o \cap o_j) =$$

$$= \sum_{i=1}^{n} \varphi_i(t_i)\cdot\varphi_j(t_i)\cdot p(t_i) \geq \theta\} \qquad (15)$$

PROOF. According to Theorem 6.1 the fuzzy probability P is a measure. From relationship (4) the result is immediate.■

6.1 Probability-based retrieval method

From the point of view of an implementation, several retrieval methods may be formulated. Given a set of index terms $T=\{t_1,\ldots,t_n\}$ and a collection of documents D_j, $j=1,\ldots,m$. Let $TD_{n\times m}=(f_{ij})_{n\times m}$ denote the term-by-document frequency matrix where f_{ij} is the number of occurrences of term t_i in document D_j. The probability $p(t_i)$ of any term t_i may be calculated as follows:

$$p(t_i) = \frac{\sum_{i=1}^{n} f_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{m} f_{ij}}$$

Let $W_{n\times m}=(w_{ij})_{n\times m}$ denote a term-by-document weight matrix, where w_{ij} is the weight of term t_i in document D_j. Given a query Q, query weights $q_1,\ldots,q_i,\ldots,q_n$ are computed; q_i denotes the weight of term t_i in Q. The similarity s_j between a document D_j and query Q and is computed using formula (15) as follows:

$$s_j = P(Q \cap D) = \sum_{i=1}^{n} q_i \cdot w_{ij} \cdot p(t_i)$$

Following the Language Model (Ponte and Croft, 1998), the conditional probability $P(Q|D)$ of document D generating query Q is considered, this may be defined using the formula for conditional probability as follows:

$$s_j = P(Q \mid D) = \frac{P(Q \cap D)}{P(D)} = \frac{\sum_{i=1}^{n} q_i \cdot w_{ij} \cdot p(t_i)}{\sum_{i=1}^{n} w_{ij} \cdot p(t_i)}$$

Alternatively, a hybrid similarity measure can also be defined by combining cardinality and probability as follows:

$$s_j = \frac{\kappa(Q \cap D)}{P(D)} = \frac{\sum_{i=1}^{n} q_i \cdot w_{ij}}{\sum_{i=1}^{n} w_{ij} \cdot p(t_i)}$$

Following the Probabilistic Model (Robertson and Sparck-Jones, 1977), the conditional probability $P(D|Q)$ of a document D given a query Q is considered, this may be defined using the formula for conditional probability as follows:

$$s_j = P(D \mid Q) = \frac{P(Q \cap D)}{P(Q)} = \frac{\sum_{i=1}^{n} q_i \cdot w_{ij} \cdot p(t_i)}{\sum_{i=1}^{n} q_i \cdot p(t_i)}$$

Alternatively, a hybrid similarity measure can also be defined by combining cardinality and probability as follows (referred to as KP method):

$$s_j = \frac{\kappa(Q \cap D)}{P(Q)} = \frac{\sum_{i=1}^{n} q_i \cdot w_{ij}}{\sum_{i=1}^{n} q_i \cdot p(t_i)} \qquad (16)$$

Part 8 reports on experimental results using formula (16) on standard test collections.

7. RELEVANCE FEEDBACK IN INFORMATION RETRIEVAL

Elementary retrievals **A** can be performed in sequence, one after the other. Thus, the following definition may be introduced.

DEFINITION 7.1. *A sequence* $\mathbf{A}_1,\ldots,\mathbf{A}_i,\ldots,\mathbf{A}_n$ *of elementary retrievals is called a* retrieval session.■

The elementary retrievals $\mathbf{A}_i$ in a retrieval session

i) may be independent from each other (e.g., in a random browsing),

ii) may be connected to each other in some way.

In practice, a retrieval session is meant to fulfil a user's information need, and so the elementary retrievals making up the session are/should not independent of each other. They may be connected, e.g., by a process called relevance feedback. Relevance feedback can be implemented in various ways, such as, for example, query reformulation, query expansion, re-computation of query weights. The meaning of relevance feedback may be formulated as follows:

"An initial search is made by the system with a user-supplied query, returning a small number of documents to the user. The user indicates which of the returned documents are useful (relevant). The system then automatically reformulates the original query based upon those user relevance judgments. The new 'feedback query' is then compared to the collection of documents, returning an improved set of documents to the user. The process can continue to iterate…" [7].

Taking a formal approach, relevance feedback means that the object-documents retrieved in one elementary retrieval are used to generate a new object-query to be used in the next

elementary retrieval. Thus, relevance feedback may be made formal as follows:

DEFINITION 7.2. *A function* $\Re$: $2^O \to \Omega$ *is called* relevance feedback.■

In practice, the aim of relevance feedback is to enhance the relevance effectiveness of a retrieval system, i.e., to return object-documents that are more relevant. To what extent can relevance effectiveness be enhanced using relevance feedback? Based on experimental results, Buckley, Salton and Allan [7] found the following answer to this question:

"As expected from our earlier TREC 2 experiments, the more terms added from relevant documents, the better the recall-precision, up to a steady-state value. … For all levels of relevant document information, the effectiveness increase starts to level off after 500 terms have been added. Except for minor random fluctuations after the effectiveness has reached its maximum value, there are no decreases in effectiveness due to adding more terms."

One year later, Buckley and Salton [6] report similar results:

"The test set recall-precision figures, though, don't match the increase of the learning set figures. There is an initial improvement on the first pass, and then the results level off and then tail off. This is not what one would have hoped given the learning set figures."

In other words, the experiments showed that relevance effectiveness in a retrieval session based on relevance feedback increases up to a certain value, and then stabilises on that value. No explanation was given to this phenomenon, it was left as an open question, and no explanation has since emerged. A possible explanation is offered by the following:

THEOREM 7.1. *The repeated application of relevance feedback yields a quantity that is associated to queries and that remains unchanged for consecutive queries.*

PROOF. Let o_0 denote the initial object-query. In the step i+1, i=0,1,2,…, of a retrieval session, i.e., in the $\mathbf{A}_{i+1}$ elementary retrieval, a new object-query o_{i+1} is generated (or obtained) by a process of relevance feedback $\Re$ using the object-documents $\rho(o_i)$ retrieved in the previous elementary retrieval $\mathbf{A}_i$. A formal representation is as follows:

$$\left(\begin{matrix} o_0 \xrightarrow{\rho} \rho(o_0) \\ A_0 \end{matrix}\right) \xrightarrow{\Re} \ldots \left(\begin{matrix} o_i \xrightarrow{\rho} \rho(o_i) \\ A_i \end{matrix}\right) \to$$

$$\xrightarrow{\Re} \left(\begin{matrix} o_{i+1} \xrightarrow{\rho} \rho(o_{i+1}) \\ A_{i+1} \end{matrix}\right) \xrightarrow{\Re} \ldots$$

We may write: $o_{i+1}=\Re(\rho(o_i))$. We can define a function h:$O \to O$ as being the composition of the functions ρ and $\Re$, i.e., $h=\rho \circ \Re$ as follows:

$$O \xrightarrow{\rho} 2^O \xrightarrow{\Re} O$$

$$o \xrightarrow{\rho} \rho(o) \xrightarrow{\Re} \Re(\rho(o)) = o'$$

$$o \xrightarrow{h=\rho \circ \Re} o'$$

$$h\colon O \to O, \quad h(o) = (\rho \circ \Re)(o) = \Re(\rho(o)) = o'.$$

Let $o_0 \in O$. Then, according to the Recursion Theorem [13], there exists a unique function f: $\mathbf{N} \to O$ such that

$$f(0) = o_0$$

$$f(i + 1) = h(f(i))$$

The value of the function f at point i, $f(i)$, corresponds to object-query o_i. The function f is recursive, hence, according to the Fixed Point Theorem [13], there exists a program (i.e., a Turing Machine) P with index e, P_e, which gives the same result as $P_{f(e)}$, i.e., $P_e=P_{f(e)}$. The index e, which corresponds to an object-query o_e, is the code for P (such as, e.g., its Gödel number), and so $P_{o_e} = P_{o_{e+1}}$.■

P_e is a quantity associated to query e, it does not matter what the exact details of the coding are.

Theorem 7.1 can be used to give a possible explanation for the experimental result obtained by Salton *et al*. Relevance feedback is a recursive process, hence it has a fixed point, i.e., there exists a program (process) that associates the same result for a query as for the previous one. In those experiments, the process of retrieval and of computing precision-recall seems to be just such a fixed point program: the stabilisation of recall-precision on a local maximum may be the manifestation of the existence of a fixed point.

8. EXPERIMENTAL RESULTS

Experiments were performed to estimate the relevance effectiveness of the following retrieval methods proposed in the present paper:

1) General basis-based retrieval method (GB method, proposed in part 4.1.1),
2) Entropy-based retrieval method (E method, proposed in part 5.1),
3) Cardinality-probability retrieval method (KP method, proposed in part 6.1, formula 16).

The following standard test collections were used: ADI, MED, TIME, CRAN. They were subjected to the usual Porter-stemming and stop-listing (using computer programs written in the C++ language). Table 3 shows the statistics of these test collections.

Table 3. Statistics of the test collections used in experiments

Test Coll.	No. docs d	No. qrys q	No. terms t	Avg. no t/d	Std. dev t/d	Avg. no. t/q	Std. dev t/q
ADI	82	35	791	21	7	6	2
MED	1033	30	7744	45	20	9	5
TIME	423	83	13479	193	140	8	3
CRAN	1402	225	4009	49	21	8	3

In the experiments for the GB method, the coordinate axes corresponding to the following index terms (stemmed) were taken to be oblique to each other: ADI: writer, book; MED: cell, patient; TIME: boarder, cradl; CRAN: program, computer. For MED and TIME, the two terms were selected because they were the two most frequent terms over the documents. For ADI and CRAN, the two terms were selected because they can be related to each other in general (not just in these test collections). The computational details for the general basis-based GB retrieval method are explained using the test collection MED as an example. The term 'cell' corresponds to

the following orthonormal basis vector in the classical vector space model:

$$\mathbf{e}_{1108} = (0,\ldots, 0, \underset{1108th \text{ position}}{1}, 0, \ldots, 0)$$

whereas the term 'patient' to the following orthonormal basis vector in the classical vector space model:

$$\mathbf{e}_{5637} = (0,\ldots, 0, \underset{5637^{th} \text{ position}}{1}, 0, \ldots, 0)$$

The fact that the terms 'cell' and 'patient' are not independent of one another is modelled by taking oblique basis vectors as follows: instead of the basis vector $\mathbf{e}_{1108}$ a new basis vector $\mathbf{g}_{1108}$ (having unit length) is taken so that it forms an angle of $\alpha°$ with the basis vector $\mathbf{e}_{5637}$ (Fig. 2). Thus, instead of the basis vector $\mathbf{e}_{1108}$ we will have the new basis vector $\mathbf{g}_{1108}$ as follows:

$$\mathbf{g}_{1108} = (0,\ldots, 0, \underset{1108^{th} \text{ position}}{\sin(\alpha)}, 0, \ldots, 0, \underset{5637^{th} \text{ position}}{\cos(\alpha)}, 0,\ldots, 0)$$

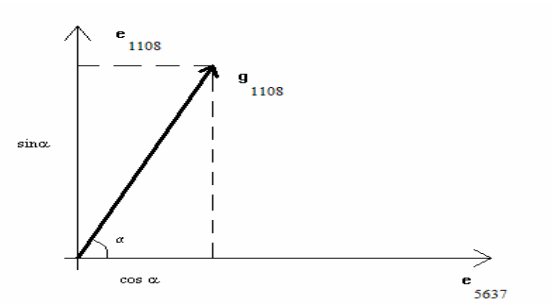


Figure 2. Orthonormal ($\mathbf{e}_{1108}$, $\mathbf{e}_{5637}$), and general ($\mathbf{g}_{1108}$, $\mathbf{e}_{5637}$) basis vectors

The columns $\mathbf{d}^{<j>}=(w_{1j},\ldots,w_{ij},\ldots,w_{nj})$, $j=1,\ldots, m$, of the term-by-document matrix are the coordinates of the documents in orthonormal basis. The new coordinates (i.e., new columns) $\mathbf{d}'^{<j>}=(w'_{1j},\ldots,w'_{ij},\ldots,w'_{nj})$ in the general basis $\mathbf{g}_i$

$$\mathbf{g}_i = (\mathbf{e}_1,\ldots, \mathbf{e}_{1107}, \mathbf{g}_{1108}, \mathbf{e}_{1109},\ldots, \mathbf{e}_{7744})$$

are computed as follows:

$$\mathbf{d}'^{<j>} = (\mathbf{g}_i)^{-1}\cdot\mathbf{d}^{<j>},$$

where $(\mathbf{g}_i)^{-1}$ denotes the inverse of the basis tensor. Thus, the term-by-document matrix in the general basis is obtained. Likewise, the new coordinates $\mathbf{q}'=(q'_1,\ldots,q'_n)$ of the initial query vector $\mathbf{q}=(q_1,\ldots, q_n)$ in the new basis $\mathbf{g}_i$ are computed as follows:

$$\mathbf{q}' = (\mathbf{g}_i)^{-1}\cdot\mathbf{q}.$$

The similarity ρ between the jth document and query Q is computed as follows:

$$\rho = \mathbf{d}'^{<j>}\cdot\mathbf{q}' = \sum_{i=1}^{n} w'_{ij}\cdot q'_i\,.$$

For each test collection, the normalised term frequency weighting scheme was used. The classical vector space (i.e., in orthonormal basis) retrieval method was also implemented and used as baseline. All three retrieval methods as well as the evaluation of retrieval effectiveness were performed using computer programs written in MathCAD. The standard 11-point precision-recall values were computed for test collection for all documents and queries. Table 4 shows the mean average precision values. The results reported in Table 4 correspond to the case when the angle between the corresponding axes was taken to be equal to 60°, this value was meant to reflect the fact that the two terms were not independent of one another (the values 30° and 45° were also used in the experiments, but the results obtained were very similar to those reported in Tables 4).

Table 4. Mean average precision obtained on standard test collections using the following retrieval methods: general basis based method (GB), entropy-based method (E), cardinality-probability method (KP). VSM: baseline B

Test coll	VSM B	E	E over B	KP	GB
ADI	0.33	0.33	0%	0.33	0.33
MED	0.44	0.48	+9%	0.44	0.44
TIME	0.52	0.56	+8%	0.52	0.52
CRAN	0.18	0.20	+11%	0.18	0.18

7. DISCUSSION, CONCLUSIONS

The classical vector space model actually does not follow logically from the mathematical concepts on which it has been claimed to rest. This problem has been recognised for more than two decades, but no proper solution has emerged so far. The present paper proposed just such a solution to this very problem. In the present paper, the concept of retrieval was defined based on measure theory. Then, retrieval was particularised using fuzzy set theory. As a result, the retrieval function was conceived as the cardinality of the intersection of two fuzzy sets. This view made it possible to build a connection to linear spaces. Thus, the classical and the generalised vector space models as well as the latent semantic indexing model became consistent with their formal background. At the same time it became clear that the inner product was not a necessary ingredient of the vector space model. Moreover, this view made it possible to consistently formulate new retrieval methods: in linear space with general basis, entropy-based, and probability-based. Experimental results using standard test collections were also reported.

The threshold value θ in Def. 3.1. was only introduced for formal reasons, it may obviously take on also the vale zero, and its specific value does influence the formal results obtained in the paper (in practice, however, its value does influence the number of retrieved documents).

It should be repeatedly emphasised that even if the space O is assumed to be a linear space, then the similarity function (formula 7) need not be viewed as the expression of inner product. Let us introduce the following principle:

Principle of Object Invariance (POI). The objects are preserved with probability π.

The objects may be documents and queries. The extent to which they are preserved (or equivalently, e.g., their meaning, information content, identity), i.e., the extent to which they remain the same, is characterised by probability π. The case when π=1 means that documents and queries do not change, they remain the same, regardless of the basis of the space; in one word: they are vectors. In this case, the similarity function (formula 7) is the expression of an inner product, hence the

similarity value between documents and queries do not change with the change of basis. If, however, $\pi<1$, the documents and queries (more exactly: their content, meaning, identity, properties) do depend on the basis of the space, in other words they are not vectors. But then, the similarity function is not the expression of inner product. It is important to emphasise that the interpretation of formula 7 as given in Lemma 4.1. (cardinality of intersection) remains valid for any value of the probability π. Thus, the classical vector space model gains a mathematical formulation that is consistent with practice.

It is also noteworthy that the view proposed in Lemma 4.1. allowed for straightforward formulation of two new or novel retrieval models: one based on entropy, the other based on probability. Both models are consistent with their mathematical background. Moreover, the experiments showed that the entropy-based method over-performs the classical vector space method with about 10%.

In [12], generative and discriminative models are overviewed. Such models are: BIR=Binary Independence Retrieval model, 2P=2-Poisson model, LM=Language Model, ME=Maximum Entropy model. In the BIR model, the probability $P(R|DQ)$ of the occurrence of relevance R given the co-occurrence of document D and query Q is estimated. In the 2P model, the probability $P(DQ|R)$ of co-occurrence of document D and query Q is estimated given relevance R. In the LM model, the probability $P(Q|DR)$ of the occurrence of query Q is estimated given the co-occurrence of document D and relevance R. In the ME model, the probability $P(R|DQ)$ of the occurrence of relevance R is estimated given the co-occurrence of document D and query Q. The models differ from each other in the specific methods and algorithms they use to estimate the probabilities. The probability-based methods proposed in part 6 differ from the generative and discriminative models in that the former do not contain relevance. The inclusion of relevance may be a topic to investigate in future.

8. REFERENCES

1. Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley Longman Publishing Co. Inc.

2. Baeza-Yates. R. (2003). Information retrieval in the Web: Beyond current search engines. *International Journal of Approximate Reasoning*, vol. 34, pp: 97-104.

3. Belew, R.K. (2000). *Finding Out About*. Cambridge University Press.

4. Berry, M.W., Drmac, Z., Jessup, E.R. (1999). Matrices, Vector Sapces, and Information Retrieval. *SIAM Review*, vol. 41, no. 2, pp: 335-362.

5. Berry, W.M., Browne, M. (1999). *Understanding Search Engines*. SIAM, Philadelphia.

6. Buckley, C., Salton, G. (1995). Optimisation of relevance feedback weights. *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, Washington, USA, pp: 351-357.

7. Buckley, C., Salton, G., Allan, J. (1994). The effect of adding relevance information in relevance feedback environment. *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Dublin, Ireland, pp: 292-300.

8. Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, vol. 41, pp: 391-407.

9. Dubin, D. (2004). The most influential paper Gerard Salton never wrote. Library Trends, Spring. http://www.findarticles.com/p/articles/mi_m1387/is_4_52

10. Kolmogoroff, A. (1950). *Foundation of Probability*. New York.

11. Meadow, C.T., Boyce, B.R. and Kraft, D.H. (1999). *Text Information Retrieval Systems*. Second edition, Academic Press, San Diego, CA.

12. Nallapati, R. (2004). Discriminative Models for Information *Retrieval. Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval*. Sheffield, United Kingdom, pp:

13. Phillips, I.C.C. (1992). Recursion Theory. In (Abramsky, S., Gabbay, D.M., and Maibaum, T.S.M.; eds.): *Handbook of Logic in Computer Science*, vol. I., Clarendon Press, London, pp:172-183.

14. Ponte, J.M., Croft, W.B. (1998). A Language Modeling Approach to Information Retrieval. *Proceedings of the ACM SIGIR International Conference on the Development and Research in Information Retrieval*, Melbourne, Australia, pp: 275-281.

15. Robertson, S.E., Sparck-Jones, K. (1977). Relevance Weighting of Search Terms. *Journal of the American Society for Information Science*, vol. 27.

16. Salton, G. (1965). Automatic Phrase Matching. In: Hays. D. G. (ed.) *Readings in Automatic Language Processing*, American Elsevier, New York, 1966. pp:169-188.

17. Salton, G. (1986b). Another look at automatic text-retrieval systems. *Communications of the ACM*, vol. 29, no. 7, pp: 648-656.

18. Salton, G., and Buckley, C. (1988). Term-weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, vol. 24, no. 5, pp: 513-523.

19. Salton, G., Wong, A., and Yang, C.S. (1975). A Vector Space Model for Automatic Indexing. *Communications of the ACM*, vol. 18, no. 11, pp: 613-620.

20. Shannon, C. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*, vol. 27, pp. 379-423, 623-656, July, October.

21. Simmonds, J.G. (1982). *A Brief on Tensor Analysis*. Springer Verlag.

22. Van Rijsbergen, C.J. (1979). *Information Retrieval*. Butterworth, London.

23. Van Rijsbergen, C.J. (2004). *The Geometry of IR*. Cambridge University Press, Cambridge, U.K.

24. Wong, S.K.M., Raghavan, V.V. (1984). Vector space model of information retrieval – a re-evaluation. *Proceedings of the 7th ACM SIGIR International Conference on Research and Development in Information Retrieval*. Kings College, Cambridge, England, pp: 167-185.

25. Wong, S.K.M., Ziarko, W., Wong, P.C.N. (1985). Generalized Vector Space Model in Information Retrieval. *Proceedings of the 8th ACM SIGIR International Conference on Research and Development in Information Retrieval*. New York, ACM Press, pp: 18-25.

26. Zimmerman, H.-J. (1996). *Fuzzy set theory—and its applications*. Kluwer Academic Publishers, Norwell-Dordrecht.