Describing query expansion using logic-induced vectors of performance measures

M. H. Heine
School of Information Studies, University of Northumbria
Newcastle upon Tyne NE1 8ST, U.K.
michael.heine@unn.ac.uk

Abstract

Practical information retrieval is pervaded by Boolean variety as much as it is by term selection. One of the problems of building models that describe IR is therefore that of characterising the effect of Boolean variety on retrieval performance. An approach to this problem is described based on the use of vectors to characterise retrieval performance and focusing on the changes in direction and magnitude of these vectors under the joint influences of term addition (i.e. query expansion) and choice of logic operator. Several novel multivariate constructs are proposed based on this 'logic/vector' approach, including: (1) the centroid vector of the vectors defined by *AND*ing, *OR*ing and *AND NOT*ing a new term to a predecessor search expression, (2) the area defined by the vertices of these three terms, when two or three performance variables are used, and (3) changes to the vector angles induced by these logic operators. The way in which these constructs change during query expansion offers a characterisation of that process and may suggest stopping rules for it. An analysis of MEDLINE data is described which generates several initial hypotheses using these constructs.

1. Introduction

When one talks of modelling an information retrieval (IR) process, it is customary to see the domain of discourse as comprising:

1. a database of documents or document surrogates[1] which is being searched
2. a person who acts so as to extract informing content from that database
3. an articulant in the form of a structure (the 'search expression') describing, jointly, that person's perception of what it is that is needed or 'lacking', using the database's query language
4. procedures for implementing the 'housekeeping' side of '3', i.e. for accepting search articulants, and outputting documents to the human searcher. These procedures are essentially in the domain of computer science
5. experimental procedures for adjudicating on the appropriateness or 'relevance' of the documents delivered by the system, in reference to the original need that prompted the person's search for them
6. values of performance measures (PMs) chosen to characterise the effectiveness of individual instances of document retrieval

The evaluation of the effectiveness of IR processes very often takes the form of choosing a pair of PMs, namely Recall (R) and Precision (P). However, it can be argued that the decision to see effectiveness in such specific terms is unnecessarily and arbitrarily restrictive. Why should we choose just two variables, and choose these to be probabilistic variables, and why these two particular probabilistic variables? A more general, hospitable, and arguably more user-oriented view, would not be so restricted, i.e. would recognise additional or alternative variables. For example, variables might be constructed which portray: (1) the degree of redundancy of information in the set of retrieved documents, (2) the novelty of retrieved documents to the searcher, (3) the value ('utility') of retrieved documents to the searcher, (4) the recency ('topicality') and/or likely rate of obsolescence of the information received, or (5) the size of the set of retrieved documents. A general formalism for IR should, it is suggested, be expressed in terms of an arbitrary number of such variables, and not solely 'R and P'. In this paper we introduce one such general approach, i.e. one that is not specifically restricted to R and P. However, in the absence of experimental data that is multivariate, we shall illustrate the approach using experimental data which is expressed in traditional terms, i.e. in terms of R-P values. A large set of reviewing sources have been published on the topic of evaluation in general, of which [3,16] offer wide-ranging accounts. A critical review of R and P, and suggested novel extensions to the classical definitions of these variables which recognise a 'learning searcher', is offered in [9]. See also [11,14] for relevant discussion.

*Bi*variate performance measures (BPMs) which map pairs of values of R and P to a single figure of merit have also been proposed, e.g. the well-known expression $E(\beta)$ [15][2]. Another such measure is the perhaps lesser known expression $\sqrt{RP}$ [19]. A systematic review of BPMs has recently been offered in [3]. However, to our knowledge, a general multivariate approach to the evaluation of document retrieval systems has not yet been proposed.

2. Using vectors to describe retrieval effectiveness

Our interest here is in the potential usefulness of *vectorial language* as it may be applied to *any* experimental results, whatever the choice of PMs by the experimenter. We limit our attention to (a) ratio-level variables, and (b) a *minimum* of two variables. Although the use of a vectorial language is a general one, we shall illustrate its use using the classic PMs of R and P, in view of the widespread acceptance of these variables. Our interest is also focused on *the use of this language to characterise changes*

[1] Henceforth we shall simply say 'documents', while recognising that the vast majority of modern document databases outside the World Wide Web are surrogate in character, notwithstanding the rapid growth of full-text electronic sources both linear and hypertextual.

[2] We henceforth use $1\text{-}E(\beta)$ rather than $E(\beta)$, so as to be consistent with the notion that a PM should increase as retrieval performance increases.

brought about by a choice of Boolean operator. The changes concerned are, in the first instance, those that arise from adding a single search term to an existing search expression (perhaps just a single term) using one or other of the familiar Boolean operators *AND*, *OR* and *NOT*. In effect, this is to seek to describe the effects of query expansion, term by term, under the variety in retrieval performance induced by these three logical operators. The approach is an extension to the treatment of retrieval PMs given in [10] which largely used the language of directed line segments. We note also that our use of Boolean operators is 'classical', the complication of *weighted* Boolean operators as introduced in [17] is not considered.

The use of a vector approach *in the performance-variable setting* is in contrast to the use of vectors to describe *queries and documents*, as pioneered by Salton (see, e.g. [16]), and later extended into the 'Generalized Vector Space Model' by Wong *et al.* [20]. In those approaches, the vectors concerned served to show the presence or absence of attributes, via either binary values or real-valued 'weights', rather than, as proposed here, as *vectors of PM values*. In view of this novelty, the account given here is not claimed to be fully rigorous and general. It is not entirely clear, for example, whether what is described for vectors based on two or three PMs can always be generalised to a larger number of PMs. Our use of the vector cross product in several places (this product being defined only for vectors in R^3) suggests the need for more fundamental, linear algebraic, development. It is hoped that 'theoretical' colleagues will aid in this, that 'experimental' colleagues will find the methodology useful in analysing IR data from experiments, and that IR engineers will see the approach as a useful aid to the design of query-expansion algorithms.

3. Logic trails through spaces defined by performance measures

Imagine that a searcher has entered a single search term, *T*1 say[3], in a document database and that the retrieval system then delivers some set of documents by way of reply. Imagine also that an observation process exists that is accepted as *valid* by some chosen set of stakeholders (e.g., the database manufacturer, an authorship community, a searcher community, the information management profession) and that this process leads to assertions as to the *effectiveness* of this search by evaluating particular PMs, labelled *V*1, *V*2, *V*3, *V*4, etc. As intuitive 'handles', we might imagine that *V*1 represents *R*, *V*2 represents *P*, *V*3 represents redundancy in the retrieved set (according to some more precise, instrumental definition), *V*4 represents novelty to the searcher (again dependent on some instrumental definition), etc. Then a search statement based on this term of first choice to the searcher:

***S*1** = ***T*1**

will define a *vector* of values of the *Vi*. Such a vector is illustrated in Figure 1, limited in scope to values of *V*1 and *V*2.

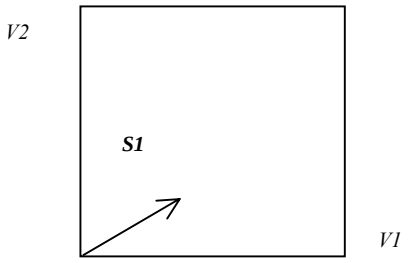

Figure 1

The searcher, if dissatisfied with the set of retrieved records delivered by ***S*1** might then choose a second term (*T*2, say) and choose it to have a logical relationship with *T*1 within the search expression. (The new term might (1) occur to the searcher using personal prior knowledge, (2) be suggested as a result of inspecting informing retrieved records, or (3) be prompted algorithmically by query expansion algorithm.)

If a conjunction relationship is chosen, the new search expression is:

***S*2** = ***T*1 *AND* *T*2**.

Alternatively, disjunction might be chosen:

***S*3** = ***T*1 *OR* *T*2**

or even the negated form:

***S*4** = ***T*1 *AND* (NOT *T*2)**,

i.e. ***S*4** = ***T*1 *NOT* *T*2**.[4]

For search expression such as these, new vectors of *V*1 and *V*2 values will be defined. These are illustrated in Figure 2 which assumes, in accordance with the well-known outcomes of Cranfield-like experiments. that conjoining *T*1 to another term in general increases *P* but decreases *R*, and that disjoining *T*1 to another term usually achieves the reverse. This is, of course, to put on one side controversy as to the *validity* or *R* and *P* (reviewed, for example, in [9]) and it is also to ignore such effects as: (1) *R* and *P* increasing together[5], and (2) ***T*1 *AND* *T*2** yielding exactly the same values of *V*1 and *V*2 as ***T*1 *OR* *T*2**.[6]. The performance

[3] Our notation attempts to distinguish between the text 'literal' *T*1 (e.g. the character string "PHENYLALANINE") and the logical variable defined by the mapping of documents to 'True' or 'False' according to whether they contain this literal or not. This distinction overcomes the confusion of students who ask 'How can a keyword evaluate to True?'. Wong *et al.* make this distinction [20], as did the author in his review of the signal-detection model of IR [6], and also more recently Dominich [4]. We use roman italic type for representations of literals (e.g. *T*1) and for PMs, boldface italic type for search expressions (e.g. ***T*1**), and boldface roman for vectors (e.g. **A**).

[4] For brevity we use *NOT* here to signify the *combination* of the use of the binary operator *AND* with the unary operator *NOT*. An analogy is with unary minus and binary minus in arithmetic where 'a-b' is an acceptable shorthand for 'a+(-b)'.

[5] For formal arguments showing that this is possible, see [1,5].

vector arising from disjoining *T*1 to a *negated* 'good' term, i.e. from using ***S4***, is also shown, adverse in its influence as might be expected.

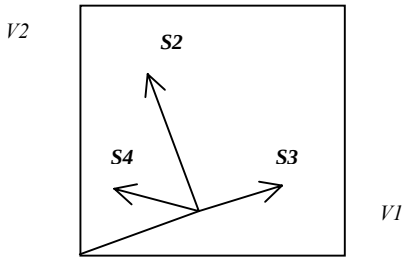

Figure 2

Further query expansion would involve further terms and choices of Boolean operator, e.g.:

S5 = (***T*1 *AND T*2) *OR T*3**

and **S6** = ((***T*1 *AND T*2) *OR T*3) *AND T*4**

The cumulative sequence of effects now might now be illustrated (plausibly) by the 'trajectory' shown in Figure 3, the lines being directed line segments. The theoretical problem is then one of modelling such movements through the space defined by *V*1, *V*2, *V*3, …, both the *actual* trajectories generated in specific operational settings, and the *possible* trajectories taking Boolean variety and search term choice into account. In this paper, we touch on only the latter, while noting that there may be significant conceptual problems in modelling the former which will be the subject of a future paper.

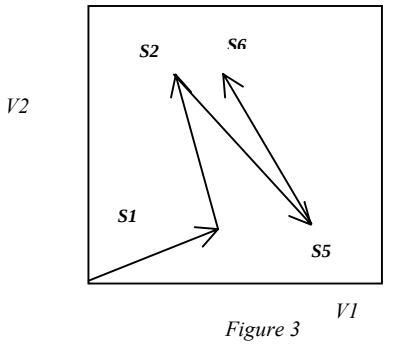

Figure 3

In each of the above cases, we assumed that the new terms brought into the search expression were 'good' terms. We use this phrase to label terms that show an increase in a chosen PM variable when they are joined in some specified way to some specified search expression. The concept is thus triply contingent, i.e. it depends on a choice of PM, the prescription of a previous search expression ***Si***, and the prescription of some logical operator linking the term to ***Si***. In the above examples, a 'good term' would be one which showed improved search performance as measured by *M* when **S1** was replaced by one of the ***S2*** or ***S3*** forms, but a deterioration in search performance when **S1** was replaced by the ***S4*** form (without going into details here as to exactly how *M* was defined.) A term-generation (i.e. 'query expansion') algorithm thus seeks to add new terms to a query where the terms are of two types:

- Good terms: the term is such that *M increases* when the search expression is extended by *AND*ing or *OR*ing it to that term, and *decreases* when the query is extended by negating the term;
- 'Usefully bad' terms: the term is such that M *increases* when the search expression is extended by negating the term concerned.

To gain a better foothold on the problem, it may be helpful to define concepts as follows:

- a 'term' is used in this paper to stand for any document representative, e.g. an author's name, a year of publication, a natural-language word or other character string or image (possibly taken from a controlled vocabulary of words or images, or from the text of the document.)
- a '*query*' is any *unstructured set* of search terms, {*T*1, *T*2, *T*3, …}.[7]
- a '*good search term*' is defined as above. a '*proper query*' is a query consisting solely of good search terms.
- a '*complete proper query*' is a query that contains *all* good search terms, so that any search term which is *excluded* from a complete search query will, when negated and *AND*ed to a term in that query, lead either to *improved* or *constant* search performance.
- '*query expansion*' is a process that generates a sequence of nested queries such as: {*T*1}, {*T*1, *T*2}, {*T*1, *T*2, *T*3}, {*T*1, *T*2, *T*3, *T*4}, …., where term *T*1 is good and further terms *T*2, *T*3, … can be either good or usefully bad.

---

[6] This will happen when ***T*1 *NOT T*2** and ***T*2 *NOT T*1** both fail to retrieve any documents. In effect, *T*1 and *T*2 are then seen by the indexer as exact synonyms.

[7] This usage may be distinct from other usages of this term, where it is sometimes used as a synonym for 'question', and can be identified with: (1) a search expression, (2) the behavioural situation that has prompted the database search, or (3) a character string representing a fragment of natural language descriptive of '2'. The commonly used phrase 'relevance to a query [or question]' raises significant complications as between the second and third of these usages, even if these are brushed aside in Cranfield-like experiments.

The present discussion is limited to: (1) query expansion where each successive query is a proper query, and (2) none of the queries are complete. In other words, we do not consider further here the device of negating usefully bad terms, and make no claim that the queries considered exhaust the set of good terms. By way of a further disclaimer, we also do not consider the more general concept of *query refinement processes,* whereby query improvement can entail *discarding* search terms as well as adding new search terms to an existing set. Such processes are more general than those of query expansion, and presumably will be central to algorithms supporting retrieval improvement under relevance feedback.

The term 'process', used as part of the definition of 'query expansion', might refer to: (1) observation regimes focusing on real-time usage of operational databases; (2) experimental regimes centred on usages of a test collection; or (3) feedback algorithms serving to prompt the searcher with new candidate terms, perhaps drawing information from data on term clustering or document clustering within the database, or within a predecessor set of retrieved documents, while doing so.

4 Some constructs that can describe query expansion under Boolean variety

Some constructs that may provide useful objects within theoretical schema are now described. We assume that our queries are proper, but future workings of the theory may permit this assumption to be discarded. We attempt to categorise these constructs according to whether we are expanding the query by:

- One good search term, subject to Boolean variety in the way that the new term is employed
- Several good search terms, under a fixed (chosen) Boolean operation.

4.1 Constructs defined by expanding a query by one term under Boolean variety

Consider a proper query *Q* made up of (good) terms *T*1, *T*2, …, *Tm*; (*m*>0), the terms concerned having contributed to some initial search expression, ***S*1**. For example, ***S*1** might have been chosen to be the all-*AND*ed form: ***T*1 *AND T*2 *AND* … *AND Tm***. If now this query is expanded by the addition of a single further (good) term *T*n, to *Q*′, then three extensions to ***S*1** become possible, namely:

***S*1 *AND T*n**
***S*1 *OR T*n**
***S*1 *NOT T*n**.

Each of these logical expressions will be associated with a performance vector (*v*1, *v*2, …) where, as before, *v*1 stands for a value of the variable *V*1, *v*2 for a value of *V*2, and so on.[8] Attaching primes to the values of the *vi* so as to distinguish the three performance vectors:

The vector (*v*1, *v*2, *v*3, …) is associated with the search expression ***S*1 *AND T*n**,
The vector (*v*1′, *v*2′, *v*3′, …) is associated with the search expression ***S*1 *OR T*n**,
The vector (*v*1″, *v*2″, *v*3″, …) is associated with the search expression ***S*1 *NOT T*n**,

Each vector can (optionally) also be associated with a *single* (scalar) value of *M*, the multi-variable PM which has been chosen to describe retrieval effectiveness. This value of M characterises the addition of a term *T*n to the preceding query and when the preceding search expression is subject to variety in logical form.[9]

At its simplest, *M* might be defined as a weighted sum of the *vi* values, with weights, k_i say, serving to express a stakeholder's view as to the *relative* significance of *N* variables *Vi*, i.e.:

$$\sum_{i=1}^{N} k_i \; Vi$$

Alternatively, and using Frants et al's I_x notation, *M* could be defined as a multivariate development of, say, 1-*E*(β) or √*RP*, as in the following,:

$$I_{11}^{'} = 1 - I_{11} = \left(1 \Big/ \sum_{i=1}^{N} k_i Vi^{-1} \right)^{-1} \text{; where } \sum k_i = 1.$$

—which is a multivariate (weighted) version of Frants et al's I_{11} expression; or

$$I_2^{'} = \frac{\sqrt[N]{k_1 V1 \;\; k_2 V2 \;\; k_3 V3 \;\; \ldots \;\; k_N \; VN}}{\sqrt[N]{k_1 k_2 \;\; k_3 \;\; \ldots \;\; k_N}}$$

—which is a multivariate (weighted) version of Frants et al's I_2 expression.

$$I_{13} = B \prod_{i=1}^{N} Vi^{k_i} e^{u_i}; \;\; \text{where } \sum k_i = 1$$

[8] The associations will be achieved by a scientific, not a mathematical regime, instanced by the protocols embedded in TREC and Cranfield experiments with their attendant assumptions. Given some choice of performance variables *Vi*, these vectors will, in particular, be determined by a choice of database, and a choice of method for partitioning the database by the informativeness of its records relative to the information need prompting the search. (Later, we will question whether such a partitioning is possible *in principle*, and suggest that performance measures should alternatively partition *the set of retrieved documents.*

[9] At the risk of pretentiousness, we might describe the potential influence of term *Tn* as a vector of values of the variables:

$\{OP_{AND} \varepsilon \{0,1\}, OP_{OR} \varepsilon \{0,1\}, OP_{NOT} \varepsilon \{0,1\}, V1, V2, V3, \ldots\}$

where the 0s and 1s here serve simply as index values recording which of the three logical operators is being employed, and where it is understood that only one non-zero index value may be present.

—a new expression suggested by the Cobb-Douglas production function used in econometrics but apparently not yet used in an IR setting might also prove to be 'valid', i.e. attractive to stakeholders The term e^{u_i} in the latter expression is introduced to accommodate differences in *significance* attached by different stakeholders to the value of *M*, e.g. by different searchers based in the same behavioural situation, in contrast to the k_i weights which reflect relative *validities* in the constituent *Vi*. When just two PMs are used, and those PMs are chosen to be *R* and *P*, then $E(\beta)$ or $1- E(\beta)$, or $\sqrt{RP}$ can, of course, be used for *M* just as they are.

A construct showing the overall effect of *T*n as an adjunct to ***S1,*** i.e. one which blends the three vectors of *vi* values, is their *resultant vector*. Denoting this vector by **r**, its definition is:

$$\mathbf{r} = (v1+ v1'+ v1'',\ v2+v2'+ v2'',\ v3+v3'+ v3'', \ldots)$$

Its *magnitude*[10] *r,* is given by:

$$r = \sqrt{[(v1+ v1'+ v1'')^2 + (v2+v2'+ v2'')^2 + (v3+v3'+ v3'')^2 + \ldots]}$$

and its projections onto *V*1, *V*2, *V*3, etc can be written as $r \cos \alpha$, $r \cos \beta$, $r \cos \gamma$, etc, where $\cos \alpha$, $\cos \beta$, $\cos \gamma$, … , are the respective 'direction cosines' of **r**. For example, the angle α between ***r*** and its projection on to the variable *V*1 is given by: $\cos \alpha = (v1+ v1'+ v1'')/r$.

Using **A**, **B** and **C** as abbreviations for the vectors (*v*1, *v*2, *v*3, …), (*v*1′, *v*2′, *v*3′, …), (*v*1″, *v*2″, *v*3″, …), respectively, and calling the scalar values of *M* appropriate to each of these vectors *M*(**A**), *M*(**B**) and *M*(**C**) respectively, a *weighted resultant vector* of **r** can also be defined:

$$\mathbf{r}_c = [M(\mathbf{A})\ \mathbf{A} + M(\mathbf{B})\ \mathbf{B} + M(\mathbf{C})\ \mathbf{C}] / [M(\mathbf{A}) + M(\mathbf{B}) + M(\mathbf{C})].$$

the so-called *centroid*, $\mathbf{r}_c$.

Angles can also usefully be defined, e.g. between:

- the resultant **r** and **A**, **B** and **C**;
- the centroid $\mathbf{r}_c$ and **A**, **B** and **C**, and
- the vector defined by the original search expression ***S1***, and **A**, **B** and **C**.

We refer to the last of these angles as 'torsions' since this term seems to convey the idea of a turning movement (within the outcome space determined by the variables *Vi*) induced by a change in the search expression arising from the introduction of the new term together with one or other of the three operators.

We note also that the *Euclidean Distance* between the resultant vector **r** and each of these constituent vectors can be defined.[11] For example, the Euclidean Distance between the resultant **r** and the vector which is generated by the particular search expression ***S1 AND Tn*** having coordinates (*v*1, *v*2, *v*3, …) is:

$$\sqrt{[((v1+ v1'+ v1''+\ldots) - v1)^2 + ((v2+v2'+ v2''+\ldots) - v2)^2 + ((v3+v3'+ v3'', \ldots) - v3)^2]}$$

i.e. $$\sqrt{[(v1'+ v1''+\ldots)^2 + (v2'+ v2''+\ldots)^2 + (v3'+ v3'', \ldots)^2]}.$$

The *area* defined by **A**, **B** and **C** constitutes yet another construct that can represent the impact of query expansion on retrieval search performance, when *AND*, *OR* and *NOT* are applied independently to a new term joined to ***S1,*** and there are just two or three PMs. This area is illustrated in Figure 4 for the case where there are just two PMs, *V*1 and *V*2. Assuming that *T*n is a 'good' term, *AND*ing it to ***S1*** will extend the vector to **A** by means of the directed line segment *AB*, i.e., loosely speaking, 'backwards and upwards' and hence *increase* this area. Similarly, *OR*ing *T*n to ***S1*** will be a useful action in extending search performance 'downwards and to the right', and hence again *increase* this area. The search expression ***S1 NOT Tn***, notwithstanding its adverse effect on search performance when *T*n is 'good', will also *increase* this area (by taking the (*V*1,*V*2) coordinate 'backwards and downwards'). This measure is thus an indicator of the effectiveness of the query expansion by drawing in term *T*n but as yet not specifying the logic to be used, i.e. this measure 'controls for logic'. For convenience, we henceforth refer to this area as the 'Boolean Area associated with the addition of a new query term'.

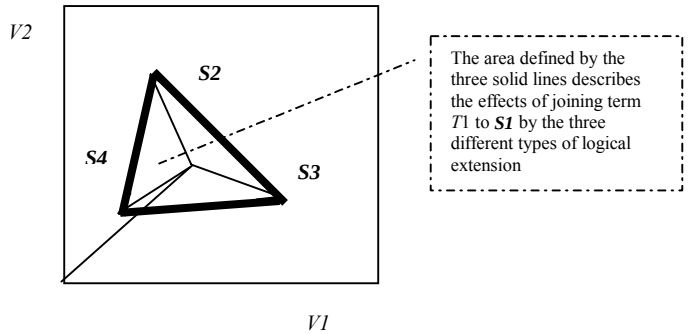

Figure 4

A *theoretical* question here, which we do not attempt to answer, is how the behaviour of different BPMs relates to the increases in Boolean Area. Examinations of contour plots for $1-E(\beta)$ or $\sqrt{RP}$ suggest that although an *AND*ing with a new term will tend to take performance 'backwards and upwards' (when *R* and *P* are used), this does not necessarily lead to an *improvement* in retrieval performance as conceived by these measures. Also, an *OR*ing that takes performance 'downwards and to the right' does not necessarily lead to an increase in $1-E(\beta)$ or $\sqrt{RP}$. An alternative line of thought asks what *functional* relationships there are

[10] Interpreted as a vector's so-called '*p-norm*', with *p*=2.

[11] This usage is different from the author's earlier usage (e.g. [6]) where this term is defined as $\sqrt{(R^2+P^2)}$.

between the area (as we have defined it) and the scalar values of $M(\mathbf{A})$, $M(\mathbf{B})$ and $M(\mathbf{C})$, for a given choice of M, and even whether requirements might be laid down for these relationships which M should be required to satisfy? For example, should M be required to be of such a form that it increases monotonically with area, when the search sequence: ***S*1 *AND T*1**, ***S*1 *AND T*2**, ***S*1 *AND T*3**, … , is followed, with ***S*1** held constant? Ultimately, the resolution of such issues lies not in mathematics but in the stakeholders' hands, since the system is created for their benefit. (A more important experimental matter would appear to be the identification of those PMs *that have the most validity for the stakeholders*, and then to explore the *variability* in the vector of values of those PMs under query expansion.)

To avoid misunderstanding, we emphasise that the term 'independently', as used above, refers to the separate applications of the *AND*, *OR* and *NOT* logic operators. If we had used this term in its stricter *probabilistic* sense, the three extensions of ***S*1** that we have described are possibly *dependent*, i.e. can generate sets of documents ('events') that are dependent. This observation follows from the fact that the elementary logical conjuncts (ELCs) that define the disjunctive normal forms (DNFs) of the three expanded search expressions partly overlap. This is clearer if the three expanded search expressions are written out:

***S*1 *AND T*n** is *already* in DNF.

***S*1 *OR T*n**, if re-expressed in DNF, is: **(*S*1 *AND T*n) *OR* (*S*1 *NOT T*n) *OR* (*NOT T*n *AND S*1)**

***S*1 *NOT T*n**, if re-expressed in DNF is: **(*S*1 *NOT T*n) *OR* (*NOT S*1 *AND T*n)**

It is then apparent that if ***S*1 *AND T*n** retrieves even a single document, this must also affect the retrieval action of ***S*1 *OR T*n**. (The same would also hold if ***T*n *NOT S*1** retrieved even a single document.) Accordingly, the 'events' defined by the two document sets *must* be dependent in that case. Similarly, if the search expression: ***S*1 *NOT T*n** retrieves even a single document, this will also the document set retrieved by ***S*1 *OR T*n**. Dependence as between the two document sets is therefore again determined.

4.1.1 Summary of basic constructs

We summarise here the devices that appear to be useful in portraying the net effect of expanding Q to Q' through the addition of a single term, Ti. We use **S** to stand for a vector of values vi determined by a search expression ***S*** which uses all of the terms in the query Q, and s for the magnitude of the vector **S**.

Constructs 1 and 2: The *resultant* vector $\mathbf{r} = \mathbf{A}+\mathbf{B}+\mathbf{C}$, with (positive) magnitude r; or alternatively, taking M-values into account, the *centroid* vector: $\mathbf{r}_c = [M(\mathbf{A})\,\mathbf{A} + M(\mathbf{B})\,\mathbf{B}+M(\mathbf{C})\,\mathbf{C}] / [M(\mathbf{A})+ M(\mathbf{B})+ M(\mathbf{C})]$.

Construct 3: The *Euclidean distances* between **r** and each of **A**, **B**, **C**.

Constructs 4, 5 and 6: The *angles* between **r** and **A**, **r** and **B**, and **r** and **C**. We refer to these angles as the '*torsions*' on **S** induced by the query expansion. For example, the angle between **r** and **A** is $\cos^{-1}(\mathbf{r} \bullet \mathbf{A}/\ r\ )$, where '$\bullet$' denotes dot product.

Construct 7: 'Boolean area': the *area enclosed by the vertices of* **A**, **B** *and* **C**, in the case where there are exactly two PMs $V1$ and $V2$, or exactly three PMs, $V1$, $V2$ and $V3$. In the latter case, this area can be found from the expression: $\frac{1}{2}\,|\,(v1'\text{-}v1,\ v2'\text{-}v2,\ v3'\text{-}v3) \times (v1''\text{-}v1,\ v2''\text{-}v2,\ v3''\text{-}v3)$,where '$\times$' here denotes vector cross product, and '|…|' denotes the unsigned value of the result.[12] Boolean Area is a function of an ordered pair of logical variables.

Notes to clarify Construct 7:

1) This area will evaluate to 0 if any two of **A**, **B** and **C** are identical, in which case the three vectors do not define a plane This will occur in practice when, for example, ***S*1 *AND T*n**, and ***S*1 *OR T*n** each determine exactly the same set of vi values, which will happen whenever ***S*1 *NOT T*n** and ***NOT S*1 *AND T*n** each retrieves zero documents.
2) A *vector* area can, if wished, also be associated with **A**, **B** and **C**, defined (when there are three variables Vi) by: $\frac{1}{2}\,[\,\mathbf{A} \times \mathbf{B} + \mathbf{B} \times \mathbf{C} + \mathbf{C} \times \mathbf{A}\,]$.
3) The discussion here is at present limited to the use of three PMs, $V1$, $V2$ and $V3$. Development of the discussion for a larger number of PMs is in progress.
4) In the experimental work described below, we evaluated the areas defined by the following logical 'triples':
   ***T*1 *AND T*2, *T*1 *OR T*2, *T*1 *NOT T*2** (i.e. expanding the query from $\{T1\}$ to $\{T1,T2\}$)
   ***T*1 *AND T*2 *AND T*3, *T*1 *OR T*2 *OR T*3, *T*1 *NOT T*2 *NOT T*3** (i.e. expanding the query from $\{T1\}$ to $\{T1,T2,\ T3\}$)
   ***T*1 *AND T*2 *AND T*3 *AND T*4, *T*1 *OR T*2 *OR T*3 *OR T*4, *T*1 *NOT T*2 *NOT T*3 *NOT T*4** (i.e. expanding the query from $\{T1\}$ to $\{T1,T2,\ T3,\ T4\}$)
   ***T*1 *AND T*2 *AND T*3 *AND T*4 *AND T*5, *T*1 *OR T*2 *OR T*3 *OR T*4 *OR T*5, *T*1 *NOT T*2 *NOT T*3 *NOT T*4 *NOT T*5** (i.e. expanding the query from $\{T1\}$ to $\{T1,T2,\ T3,\ T4,\ T5\}$)
   These areas do not exactly conform to the definition of Boolean Area defined as Construct 7, but: (1) may assist in generalising that definition, and (2) portray search performance in a slightly different way.

[12] The *volume of the parallelepiped* defined by **A**, **B** and **C**, i.e. the determinant whose rows are made up **A**, **B** and **C**, may also be useful when there are three Vi variables. This volume is given by the unsigned value of $\mathbf{A} \bullet (\mathbf{B} \times \mathbf{C})$,where '$\bullet$' stands for the scalar or 'dot' product of two vectors.

4.2 Further constructs defined by expanding a query by several terms when the Boolean operator is fixed

If we permit variety *in the choice of a successor term* while holding constant the logical operator to be used on that term rather than permitting variety *in the choice of operator for a particular successor term*, some differences arise in the constructs that can be employed. We now have a 'vector bundle' [13] for each logic type [9], with each bundle defined by an arbitrary number of vectors each of which is associated with a choice of successor terms, in contrast to the scant three vectors defined by *AND*, *OR* and *NOT* variation applied to a single term, as above. Figure 5 illustrates the effects of predicating a *set* of terms *Ti* on a fixed search expression ***S1***, in the three cases where the members of the set are *AND*ed to ***S1***, *OR*ed to ***S1***, and *negated* after ***S1***.

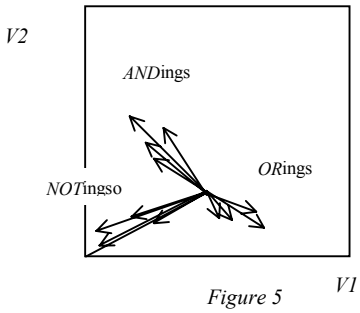

Figure 5

Some constructs suggested by the preceding discussion are, for each vector bundle: (1) a set of *M* values, with associated statistics; (2) for each variable *Vi*, sets of angles between each member of the bundle and the *Vi* variable, with associated statistics; and (3) a set of torsions induced by each of the terms within it to a chosen predecessor search statement ***S1***, and again statistics of these torsions. Since each of the three vector *bundles* has a *resultant* vector, and also a *centroid* vector, we can also evaluate the area defined by connecting the vertices of the three resultants or centroids.

4.2.1 Summary

For brevity, we do not write out the above formally, since they are suggested by the constructs given in *4.1.1*. In any case, the schema described in *4.1* would appear to be more relevant to real world IR practice as discussed in this paper, i.e. to *query expansion*, where the query is expanded term by term rather than with a process of flooding a search process with a set of new terms all at once.

5. Some questions for experiments

Various empirical questions are suggested by the preceding discussion. For example, as query expansion takes place, building on to an initial single-term query *T*1 single term by single term:

- How rapidly do *M(***A***) M(***B***) and M(***C***)* increase[13] and towards what apparent limits?
- How rapidly does the area defined by **A**, **B** and **C** increase, and to what apparent limit?
- What torsions are exhibited and what are their statistics?
- How sensitive are the effects observed to the choice of query expansion algorithm (or process)?

In each case, we should of course, for realism, add such contingencies as "for a given set of variables *Vi*, a given database, and a given observational regime." The sensitivities of the above effects to variation in choice of database, etc, then generates further hypotheses.

6. Some hypotheses suggested by the author's Medline data, using some of the previous constructs

In order to obtain some initial, selective, answers to the above questions, the author's own Medline data set [5-6] was reexamined. This data set was obtained using an observational regime that accepted the validity of the classical measures of *R* and *P*, i.e. it adopted a Cranfield viewpoint wherein information is seen as capable of prior-to-search instrumental definition.[14] However, the regime *contrasted* with Cranfield-like experiments (and therefore with the TREC projects) in that it did not accept the validity of third-party relevance judges, i.e. of the concept that such persons could function within an *experimental* regime divorced from real world settings of need to which their individual subjectivities were party. The settings recognised in the experiment were chosen to be the preparation and publications of medical review papers, where the authors might reasonably be expected to have exercised competent and reasonably exhaustive 'including' and 'excluding' judgements as to the sources that were relevant, and where the act of citation by the review paper constituted an observable, instrumental, definition of relevance. Two additional claims on validity in the data sets are: (1) *a complete database*[15] *was used rather than a test-collection*, and (2) *queries were defined algorithmically* rather than by subjective arbitration.

---

[13] We assume here that *M* is defined such that improved search performance leads to a *higher* value of *M*. Since $E(\beta)$ decreases when search performance improves, our remark would apply to $1- E(\beta)$ in the two-variable case.

[14] For revision of this view, see [7].

[15] A subset of Medline was identified according to the time-period of publication defining eligibility for inclusion in the review, this data being obtained in corrrespondence with each review's author. (Frequencies of non-relevant documents for individual ELCs were identified in a large (512444 records) sample of Medline, and then multiplied by a proportionality constant appropriate to the size of the subset.) The size of the subset of the database was typically about 2×10^6 —well in excess of that used in classical test collections, but comparable with the standard TREC task of about 500,000 documents..

The results below pertain to averaged data for each of two query expansion algorithms. Averaging of probabilities was undertaken *before* the analyses. (The probabilities concerned were those of each ELC taking the value 'True', for (1) the set of relevant documents, and (2) the set of non-relevant documents.) Also, the Generality values attaching to each partitioning of the database, were averaged before the analyses were undertaken. Accordingly, the results are claimed to be indicative and suggestive only, and not conclusive. They are offered solely to suggest possible general effects that might later be verified, falsified or clarified following more extensive experimentation, i.e. they are seen only as 'hypothesis suggesting'.[16]

The query definition and expansion algorithms were defined by the following rules:

1. Choose and prioritise terms according to their *relative specificity* (i.e. frequency in the set of relevant documents divided by frequency in that section of the database regarded as within the time-scope of the reviewing author). We refer to this algorithm as the 'RS algorithm'. The sample size was 31, i.e. the algorithm was applied to 31 partitionings of Medline.
2. Choose and prioritise terms by first clustering all those included in the set of relevant documents and occurring twice or more, and then use single-link, Euclidean distance, nearest-neighbour clustering, the shallowest-clustered—i.e. most broadly applied—terms being seen as the most eligible. We refer to this algorithm as the 'CL algorithm'. The sample size was 29.[17]

In each case, the query was expanded from one term to five terms. In general, the successive choices of terms differed significantly as between the two algorithms.[18] The performance of individual search expressions based on the queries so defined was expressed in terms of an (*R*,*P*) pair of values, which was then mapped to values of 1- $E(\beta)$, with β=0.5, and $\sqrt{RP}$.

Some of the results of the analysis are shown in the following tables. Tables 1-3 show the variations in 1- $E(\beta)$ and the angle shown for the three logic forms under query expansion, for the two methods of query definition and expansion. The poor performance of Medline as shown by the experimental regime is readily apparent, falling very much below the "*R*=50%, *P*=50%" suggested in many IR writings, and suggesting significant needs for improved retrieval technology. Table 4 shows the changes in area under query expansion, for the two methods. Figures 6-7 show selective directed line segment bundles determined in accordance with the discussion of *4.2* for the two methods of query expansion and where the search expression builds from ***T1*** by: (1) choosing one of four successor terms, and (2) one of the three logical operators. Interestingly (but perhaps as expected) *AND*ing a poor term to a good term can produce a roughly comparable effect to *NOT*ing the poor term to a good term. As previously, we do not introduce the additional complication of weighted Boolean operators.

Search expression	Query definition and expansion method	Angle between (*R*,*P*) vector and *R*-axis	1- $E(\beta)$ (β=0.5)	$\sqrt{RP}$
T1	RS-expansion	2.81º	0.0185	0.0675
	CL-expansion	0.788º	0.00460	0.0315
T1 AND T2	RS-expansion	46.3º	0.130	0.128
	CL-expansion	23.0º	0.0881	0.120
T1 AND T2 AND T3	RS-expansion	82.5º	0.134	0.112
	CL-expansion	77.1º	0.268	0.214
T1 AND T2 AND T3 AND T4	RS-expansion	88.2º	0.119	0.151
	CL-expansion	85.3º	0.198	0.183
T1 AND T2 AND T3 AND T4 AND T5	RS-expansion	89.6º	0.0361	0.0862
	CL-expansion	87.8º	0.121	0.143

Table 1

Search expression	Query definition and expansion method	Angle between (*R*,*P*) vector and *R*-axis	1- $E(\beta)$ (β=0.5)	$\sqrt{RP}$
T1	RS-expansion	2.81º	0.0185	0.0675
	CL-expansion	0.788º	0.00460	0.0315
T1 OR T2	RS-expansion	1.00º	0.0109	0.0660
	CL-expansion	0.141º	0.00108	0.0175
T1 OR T2 OR T3	RS-expansion	0.459º	0.00649	0.0582
	CL-expansion	0.0738º	0.000750	0.0167
T1 OR T2 OR T3 OR T4	RS-expansion	0.200º	0.00307	0.0416
	CL-expansion	0.0640º	0.000813	0.0195
T1 OR T2 OR T3 OR T4 OR T5	RS-expansion	0.0625º	0.000415	0.0140
	CL-expansion	0.0436º	0.000613	0.0178

Table 2

[16] For more discussion of the use of ELCs as data elements, see [4], although the latter paper was mostly concerned to re-express the Swets model in discrete rather than continuous terms, and to interpret a document 'ranking' process in logic terms, namely as a process of successive disjunction applied to a weak order of the set of ELCs.

[17] The CL algorithm failed with two of the partitionings, verified in correspondence with the author of the CLUSTAN program.

[18] For example, with one partitioning of Medline, the RS algorithm generated the following terms (in the order given): NATRIURESIS, DESOXYCORTICOSTERONE, ALDOSTERONE, VASOPRESSIN, GLOMULAR FILTRATION RATE, while the CL algorithm generated the following terms (also in the order shown): ADRENALECTOMY, OSMOLAR CONCENTRATION, WATER, DESOXYCORTICOSTERONE, DIURESIS. There was just one query term in common, in this case.

Search expression	Query definition and expansion method	Angle between (*R*,*P*) vector and *R*-axis	1- *E*(β) (β=0.5)	$\sqrt{RP}$
***T*1**	RS-expansion	2.81º	0.0185	0.0675
	CL-expansion	0.788º	0.00460	0.0315
***T*1 *NOT T*2**	RS-expansion	2.95º	0.0114	0.0408
	CL-expansion	0.814º	0.00149	0.0101
***T*1 *NOT T*2 *NOT T*3**	RS-expansion	3.75º	0.00997	0.0317
	CL-expansion	0.834º	0.000895	0.00595
***T*1 *NOT T*2 *NOT T*3 *NOT T*4**	RS-expansion	4.08º	0.00470	0.00139
	CL-expansion	0.856º	0.000674	0.00443
***T*1 *NOT T*2 *NOT T*3 *NOT T*4 *NOT T*5**	RS-expansion	4.36º	0.00470	0.0139
	CL-expansion	0.893º	0.000589	0.00379

Table 3

*Query expansion**	*Boolean Area*	*Change in Boolean Area*
{*T*1} (RS-method)	(not defined)	(not applicable)
{*T*1} (CL-method)	(not defined)	(not applicable)
{*T*1,*T*2} (RS-method)	0.0194	(not applicable)
{*T*1,*T*2} (CL-method)	0.0103	(not applicable)
{*T*1,*T*2,*T*3} (RS-method)	0.0789	0.0595
{*T*1,*T*2,*T*3} (CL-method)	0.0933	0.0830
{*T*1,*T*2,*T*3,*T*4} (RS-method)	0.263	0.184
{*T*1,*T*2,*T*3,*T*4} (CL-method)	0.173	0.0797
{*T*1,*T*2,*T*3,*T*4,*T*5} (RS-method)	0.350	0.0870
{*T*1,*T*2,*T*3,*T*4,*T*5} (CL-method)	0.226	0.0530

Table 4

* See Section 4.1.1, 'Construct 7' Note 4, for clarification as to how Boolean Area is defined here.

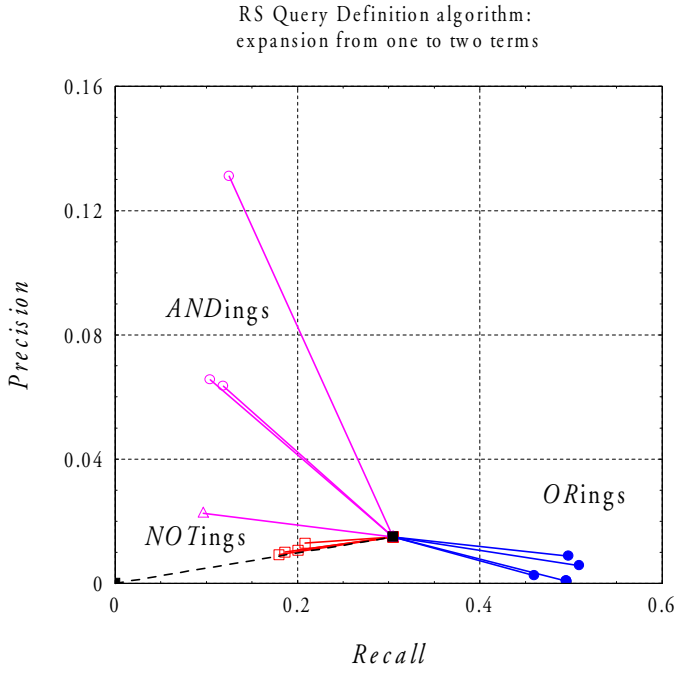

Figure 6

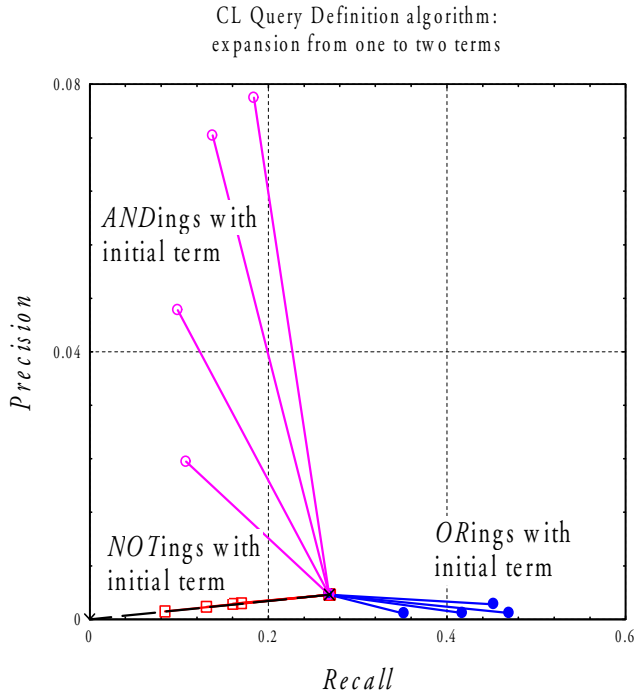

Figure 7

6.1 Hypotheses suggested by the Medline experiments

The Medline data shown in Tables 1-4 and Figures 6-7 suggested the following conjectures, or more generously, hypotheses. As indicated by earlier discussion, they are derived from and therefore may be restricted to conditions under which query expansion is algorithmic rather than under human operator control, and from 1 to 5 terms, and also where terms are chosen so as to be 'good' as defined in the text. ('Usefully bad' terms were not included in the expansion.) These hypotheses are seen as provisional and suggestive of further experimental work that might support, falsify or clarify them, and are not claimed to be conclusive.

Hypothesis 1: The variability in Precision associated with *AND*ing a given term with a new term is much greater than that obtained by *OR*ing. (See Figures 6 and 7)

Hypothesis 2: *AND*ing a weaker term to an effective term *can* give roughly comparable Precision to that obtained by *NOT*ing with that weaker term. (See Figures 6.)

Hypothesis 3: Using *AND*-logic, the angle between the P=0 vector (i.e. R-axis) and the (R,P) vector increases monotonically and rapidly up to almost 90˚.

Hypothesis 4: Using *OR*-logic, the angle between the P=0 vector (i.e. R-axis) and the (R,P) vector decreases monotonically and rapidly down to almost 0˚, but within a narrow angular range of between 0˚ to 5˚.

Hypothesis 5: Using *NOT*-logic, the angle between the P=0 vector (i.e. R-axis) and the (R,P) vector remains fairly constant within the range 0˚ to 5˚.

Hypothesis 6: Using *AND*-logic, the PMs 1-$E(\beta)$ and $\sqrt{RP}$ both reach maxima when between 2 and 4 terms are used; but steadily decrease when *OR*-logic is used.

Hypothesis 7: The area defined by the triples of search expressions written out in Section 4.1.1, Construct 7, Note 4, increases monotonically and almost linearly.

Of these, perhaps Hypotheses 1-3 and 6 would seem to be of most practical interest. Hypothesis 6 appears to conform with everyday observations that database searchers' tend to search with a small number of search terms: see, e.g. [12].

Further suggestions

Although the approach to describing retrieval effectiveness we have described has been illustrated for retrieval against a fixed set of 'relevant documents', there is no reason why it can not be used to describe retrieval where the searcher is a 'learning searcher', i.e. where the relevance criteria vary with successive searches of the database, i.e. where the searcher adapts his or her needs to the information retrieved by each search expression. Spink and her colleagues [18] refers to such studies as 'longitudinal studies'. Search expression adaptation is of course associated with much published work in the relevance feedback area, traceable to Salton's work in this field the 1960s, and with the concept of document retrieval as part of a heuristic process (see, e.g. [2]).[19]

In regard to formal methodology, what would seem to be helpful is a more rigorous development than we have offered here, one which will recognise the combinatorial aspects of term selection and logical operator selection.

Software written by the author to support the type of analysis described in this paper is freely available on a non-commercial basis from the author by e-mail. Offers of collaboration in the development of the code (in C++) would be welcomed.

Acknowledgement: Criticisms by referees of an earlier version of this paper are acknowledged with thanks.

References

1. Bookstein, A. The anomalous behaviour of precision in the Swets model and its resolution. *Journal of Documentation*, 30, 1974, 374-380.
2. Bookstein, A. Information retrieval: a sequential learning process. *Journal of the American Society for Information Science*, 34, 1983, 331-341.
3. Frants, V. I., Shapiro, J. and Voiskunskii, V. G. *Automated information retrieval.* San Diego, Academic Press, 1997.
4. Dominich, S. A unified mathematical definition of classical IR. *Journal of the American Society for Information Science*, 51, 2000, 614-625..
5. Heine, M. H. The inverse relationship of precision and recall in terms of the Swets model. *Journal of Documentation*, 29, 1973, 81-84.
6. Heine, M. H. Information retrieval from classical databases from a signal-detection standpoint: a review. *Information Technology: Research, Development, Applications,* 3, 1984, 95-112.
7. Heine, M. H., and Tague, J. M. An investigation of the optimization of search logic for the Medline database. *Journal of the American Society for Information Science*, 42(4), 1991, 267-278.
8. Heine, M. H. An investigation of the relative influences of database informativeness, query size and query term specificity on the effectiveness of Medline searching. *Journal of Information Science*, 21, 1995, 173-185.
9. Heine, M. H. *Reassessing and Extending the Precision and Recall Concepts*, in www.ewic.org.uk/ewic (Revised 18 Jan 2000) Revised version of '*Time to dump 'P and R'?' Proceedings of the Mira '99: Final Mira Conference on Information Retrieval Evaluation, Glasgow, April 1999.* 16pp.
10. Heine, M. H. Measuring the effects of *AND*, *OR* and *NOT* operators in document retrieval systems using directed line segments. *Paper presented at the Workshop on Logical and Uncertainty Models for Information Systems, of the Fifth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, London, 5-6 July, 1999*.

[19] However, we suggest that there is often an ambiguity as to whether the feedback is directed at improved searching against a fixed or variable 'set of relevant documents', i.e. whether it is the *query* that is 'learning' or the *searcher*.

11. Janes, J. W. Other people's judgments: a comparison of users' and others' judgments of document relevance, topicality and utility. *Journal of the American Society for Information Science*, 45(3), 1994, 160-171.
12. Kim, H. User differences in interactive Web-based OPAC evaluation. MPhil Thesis. University of Sheffield, UK, Department of Information Studies. 1998.
13. Luke, G. and Mishchenko, A.S. *Vector bundles and their application.* Dordrecht, Kluwer, 1998.
14. Park, T. K. Toward a theory of user-based relevance: a call for a new paradigm of enquiry. *Journal of the American Society for Information Science*, 45(3), 1994, 135-141.
15. van Rijsbergen, C. J. *Information retrieval*. London, Butterworths, 1975.
16. Salton, G. and McGill, M. J. *Introduction to modern information retrieval.* N. Y.,McGraw-Hill, 1983.
17. Salton, G., S. Fox and H. Wu Extended Boolean information retrieval. Cornell University: Department of Computer Science, Aug. 1982. (Technical Report TR 82-511)
18. Spink, A. and T. D. Wilson, D. Ellis, N. Ford. Modeling users' successive searches in digital environments. *D-Lib Magazine*, April 1998. [www.dlib.org/dlib/april98/04spink.html]
19. Voiskunskii, V. G. Choosing an optimal version of search: experimental investigation. *Nauchno-Teknicheskaya Informatsiya (NTI)*, ser.2, no9, 10-19, 1982.
20. Wong, S. K. M. and W. Ziarko, V. V. Raghavan, P. C. N. Wong. Extended Boolean query processing in the generalized vector space model. *Information Systems*, 14(1), 1989, 47-63.