

## Improving the Retrieval of Information from External Sources

### **Abstract:**

*A major barrier to successful retrieval from external sources (e.g., electronic databases) is the tremendous variability in the words people use to describe objects of interest. The fact that different authors use different words to describe essentially the same idea means that relevant objects will be missed; conversely, the fact that the same word can be used to refer to many different things means that irrelevant objects will be retrieved. We describe a statistical method called Latent Semantic Indexing (LSI) which models the implicit higher-order structure in the association of words and objects and improves retrieval performance by up to 30%. Additional large performance improvements of 40% and 67% can be achieved using differential term weighting and iterative retrieval methods.*

Address correspondence to:  
Susan T. Dumais  
Bellcore  
445 South St. - rm 2L-371  
Morristown, NJ 07960-1910  
email: std@bellcore.com  
phone: (201) 829-4253

## 1. Introduction

Although a good deal of research in cognitive psychology examines how people retrieve information from their own memories, much less work addresses the issue of retrieving information from external sources such as other people, books, libraries, and electronic databases. One problem that is immediately evident in attempting to retrieve information from external sources is the mismatch between a searcher's language and that of the target information. How often have you looked in the index of a book or the library catalog and been unable to find what you wanted? This problem is not evident in traditional memory modeling because memory probes are in the same language as the memory representation.

Most approaches to retrieving electronically available textual materials depend on a lexical match between words in users' requests and those in database objects. Typically only text objects that contain one or more words in common with those in the users' query are returned as relevant. Word-based retrieval systems like this are, however, far from ideal - many objects relevant to a users' query are missed, and many unrelated or irrelevant materials are retrieved. A particularly salient example of the failure to find relevant materials is reported by Blair and Maron (1985) in a study of a state-of-the-art online legal retrieval system. Two lawyers, with the aid of expert search intermediary, searched the database for all materials relevant to a case they were litigating. The system contained the full-text of 40,000 documents, corresponding to roughly 350,000 paged of hard-copy text. The lawyers were asked to search until they thought they had found 75% of the relevant materials. The surprising result was that they found only 20% of the known relevant materials.

We believe that fundamental characteristics of human verbal behavior underly these retrieval

difficulties. Furnas et al. (1987), for example, have shown that people generate the same main keyword to describe well-known objects less than 20% of the time. Comparably poor agreement has been reported in studies of inter-indexer consistency Tarr and Borko (1974), and in the generation of search terms expert intermediaries Fidel (1985) or by novices Bates (1986). Because of the tremendous diversity in the words people use to describe the same object or concept (*synonymy*), requesters will often use different words from the author or indexer of the information, and relevant materials will be missed. Conversely, since the same word often has more than one meaning (*polysemy*), irrelevant materials will be retrieved.

Several methods have been developed by researchers and practitioners in information retrieval (library science) to help overcome the problem of variability in human word usage and improve retrieval performance. These methods have included: restricting the allowable indexing and retrieval vocabulary and training intermediaries to generate terms from these restricted vocabularies; hand-crafting domain-specific thesauri to provide synonyms for user's search terms; constructing explicit models of domain-relevant knowledge; and automatically clustering terms and documents. The rationale for restricted or controlled vocabularies is that they are by design relatively unambiguous. However they have high costs and marginal (if any) benefits compared with automatic indexing based on the full content of texts. The use of a thesaurus is intended to improve retrieval by expanding terms that are too specific. Unfortunately this also has the unwanted effect of retrieving irrelevant information. Overall one can expect small retrieval improvements for carefully constructed thesauri in limited domains. More standard AI techniques for knowledge representation are also beginning to be used for information retrieval. Such methods are currently applicable to small, stable domains and have not been systematically compared with more standard methods.

Automatic statistical methods for analyzing the relationships among words and documents are promising and much more widely applicable. We have developed a method called **Latent Semantic Indexing (LSI)** which tries to overcome the problems of variability in word usage by *automatically* organizing objects into a semantic structure more appropriate for information retrieval (Deerwester, et al., 1990; Dumais, et al., 1988). The LSI method begins by viewing the words or terms contained in a document as incomplete and unreliable indicators of the content of the document. We assume that there is some underlying or "latent" structure in the pattern of word usage that is partially obscured by the variability of word choice. We use statistical techniques to estimate this latent structure and get rid of the obscuring "noise". In this model, the similarity of terms and documents is determined by the overall pattern of word usage in the entire collection, so documents can be similar to each other regardless of the precise words they contain. A description of terms, objects and user queries based on the underlying latent semantic structure, rather than surface level words, is used for representing and retrieving information. What this means from a user's perspective is that documents can be similar to a query even if they share no terms in common.

## **2. Overview of Latent Semantic Indexing**

### **2.1 Theory.**

The particular latent semantic indexing analysis that we tried uses singular-value decomposition (SVD), a technique closely related to eigenvector decomposition and factor analysis (Forsythe, Malcolm and Moler, 1977; Cullum and Willoughby, 1985). We take a large matrix of term to text-object association data and decompose it into a set of, typically 50 to 150, orthogonal factors from which the original matrix can be approximated by linear combination. More

formally, any rectangular matrix,  $X$ , for example a  $t \times o$  matrix of terms and objects, can be decomposed into the product of three other matrices:

$$\underset{t \times o}{X} = \underset{t \times r}{T_0} \cdot \underset{r \times r}{S_0} \cdot \underset{r \times o}{O_0'}$$

such that  $T_0$  and  $O_0$  have orthonormal columns,  $S_0$  is diagonal, and  $r$  is the rank of  $X$ . This is so-called *singular value decomposition* of  $X$  and it is unique up to certain row, column and sign permutations.

If only the  $k$  largest singular values of  $S_0$  are kept along with their corresponding columns in the  $T_0$  and  $O_0$  matrices, and the rest deleted (yielding matrices  $S$ ,  $T$  and  $O$ ), the resulting matrix,  $\hat{X}$ , is the unique matrix of rank  $k$  that is closest in the least squares sense to  $X$ :

$$\underset{t \times o}{X} \approx \underset{t \times o}{\hat{X}} = \underset{t \times k}{T} \cdot \underset{k \times k}{S} \cdot \underset{k \times o}{O'}$$

The idea is that the  $\hat{X}$  matrix, by containing only the first  $k$  independent linear components of  $X$ , captures the major associational structure in the matrix and throws out noise. It is this reduced model, usually with  $k=100$ , that we use to approximate the term to text-object association data in  $X$ . Since the number of dimensions in the reduced model ( $k$ ) is much smaller than the number of unique terms ( $t$ ), minor differences in terminology are ignored. In this reduced model, the closeness of objects is determined by the overall pattern of term usage, so objects can be near each other regardless of the precise words that are used to describe them, and their description depends on a kind of consensus of their term meanings, thus dampening the effects of polysemy. In particular, this means that text objects which share no words with a user's query may still be near it if that is consistent with the major patterns of word usage. We use the term "semantic" indexing to describe our method because the reduced SVD representation captures the major

associative relationships between terms and text objects.

One can also interpret the analysis performed by SVD geometrically. The location of the terms and objects in  $k$ -space is given by the row vectors from the  $T$  and  $O$  matrices, respectively. In this space the cosine or dot product between vectors corresponds to their estimated similarity. The position of term (document) vectors in this space reflects the correlations in their usage across documents (terms). This can be contrasted with lexical word-matching methods in which words are treated as if they are independent. Retrieval typically proceeds by using the terms in a query to identify a vector in the space, and all text objects are then ranked by their similarity to the query. However, since both terms and documents are represented in the same space, queries can be formed using any combination of terms and objects, and any combination of terms and objects can be returned in response to a query.

Unlike many factor analytic applications, we make no attempt to interpret the underlying dimensions or factors, nor to rotate them to some intuitively meaningful orientation. Our analysis does not require us to be able to describe the factors verbally but merely to be able to represent terms, text objects and queries in a way that escapes the unreliability, ambiguity and redundancy of individual terms as descriptors.

The idea of aiding information retrieval by discovering latent proximity structure has several lines of precedence in the information science literature. Hierarchical classification analyses have sometimes been used for term and document clustering (Sparck Jones, 1971; Jardin and van Rijsbergen, 1971). Factor analysis has also been explored previously for automatic indexing and retrieval (Baker, 1962; Atherton and Borko, 1965; Borko and Bernick, 1963; Ossorio, 1966). Koll (1979) has discussed many of the same ideas we describe above regarding concept-based

information retrieval, but his system lacks the formal mathematical underpinnings provided by the singular value decomposition model. Our latent structure method differs from these approaches in several important ways: (1) we use a high-dimensional representation which allows us to better represent a wide range of semantic relations; (2) both terms and text objects are explicitly represented in the same space; and (3) objects can be retrieved directly from query terms.

## **2.2 Practice.**

We have applied LSI to several standard information science test collections, for which queries and relevance assessments were available. The text objects in these collections are bibliographic citations (consisting of titles, authors and the full text of document abstracts), or the full text of short articles. A set of user queries and relevance judgements (judgements about the relevance of every document in the collection to each query) is associated with each test collection. Table 1 gives a brief description and summarizes some characteristics of the datasets and queries used in our experiments. As noted above, the "documents" in these collections consisted of the full text of document abstracts or short articles.

<b>MED:</b>	document abstracts in biomedicine received from the National Library of Medicine
<b>CISI:</b>	document abstracts in library science and related areas published between 1969 and 1977 and extracted from Social Science Citation Index by the Institute for Scientific Information
<b>CRAN:</b>	document abstracts in aeronautics and related areas originally used for tests at the Cranfield Institute of Technology in Bedford, England
<b>TIME:</b>	articles from Time magazine's world news section in 1963
<b>ADI:</b>	small test collection of document abstracts from library science and related areas

	<b>MED</b>	<b>CISI</b>	<b>CRAN</b>	<b>TIME</b>	<b>ADI</b>
number of documents	1033	1460	1400	425	82
number of terms (occurring in more than one document)	5831	5743	4486	10337	374
number of queries	30	35	225	83	35
average number of documents relevant to a query	23	50	8	4	5
average number of terms per doc	50	45	56	190	16
average number of docs per term	9	13	16	8	4
average number of terms per query	10	8	9	8	5
percent non-zero entries	0.86	0.88	1.10	1.80	4.38

**Table 1. Characteristics of Datasets**

Results were obtained for LSI and compared against word-based retrieval methods. Each document is indexed completely automatically, and each word occurring in more than one document and not on a stop list of common words is included in the LSI analysis. The LSI analysis begins with a large term by document matrix in which cell entries are a function of the frequency with which a given term occurred in a given document. A singular value decomposition (SVD) is performed on this matrix, and the  $k$  largest singular values and their corresponding left and right singular vectors are used for retrieval. Queries are automatically indexed using the same preprocessing as was used for indexing the original documents, and the query vector is placed at the weighted sum of its constituent term vectors. The cosine between the resulting query vector and each document vector is calculated. Documents are returned in decreasing order of cosine, and performance evaluated on the basis of this list.



Performance of information retrieval systems is often summarized in terms of two parameters - precision and recall. *Recall* is the proportion of all relevant documents in the collection that are retrieved by the system; and *precision* is the proportion of relevant documents in the set returned to the user. Precision is calculated for several levels of recall, and averaged across queries. (A signal detection analysis could also be applied to these data, as Swets (1963) and others have noted. This is typically not done in the information retrieval context because there can be millions of correct rejections but only a handful of relevant documents or hits.)

The results for the MED collection are shown in Figure 1. Precision is plotted as a function of recall for 9 levels of recall (from .10 to .90). These data represent average data from the 30 queries available with the MED collection.

```

frame ht 4 wid 5
coord x 0,1 y 0,1
label left "Precision"
label bottom "Recall"
label top "MED (Entropy) Precision-Recall Curve" "Average over queries"
ticks bot out from 0 to 1 by .1
draw lsi solid # lsi-med
draw key dashed # keyword-med
copy "med.entropy.90" thru % next lsi at 2,1 %
copy "med.keywd.entropy" thru % next key at 2,1 %
"LSI-100" rjust size -3 at 0.18,0.85
"Word-matching" rjust size -3 at 0.18,0.75

```

**Figure 1. MED (Entropy) Precision-Recall**

These are typical precision-recall curves, with precision (proportion of irrelevant information) decreasing as recall (proportion of relevant information found) increases. The important thing to notice is the difference between LSI and the word-matching methods. A 90-dimensional LSI representation results in roughly 30% better performance in discriminating relevant from

irrelevant document. Over all the test collections, LSI averaged about 20% better than word-based methods, with a range of comparable to 30% better than that obtained using standard vector methods. (See Deerwester, et al., 1990 for details of these evaluations.)

### **3. Improving Performance**

Although LSI improves retrieval compared word-matching methods, the overall level of performance is still far from perfect, especially at high levels of recall. There are several well-known techniques for improving performance in standard word-based retrieval systems. We applied many of these methods to our LSI representation. One of the most important and robust methods involves differential *term weighting* (Sparck Jones, 1972). Another method of improvement involves an iterative retrieval process based on users' judgments of relevant items - often referred to as *relevance feedback* (Salton & Buckley, 1990). The LSI approach also involves choosing the *number of dimensions* for the reduced space.

#### **3.1 Choosing the number of dimensions.**

Choosing the number of dimensions for the reduced dimensional representation is an interesting problem. Our choice, thus far, has been determined simply by what works best. We believe that the dimension reduction analysis removes much of the noise, but that keeping too few dimensions would lose important information. We suspect that the use of too few dimensions has been a deficiency of previous experiments that have employed techniques similar to SVD (Atherton and Borko, 1965; Borko and Bernick, 1963; Ossorio, 1966; Koll, 1979). Koll, for example, used only seven dimensions to describe the relations among terms and documents. We evaluated retrieval performance using a range of dimensions. Figure 2 shows performance for

the MED database using different numbers of dimensions in the reduced LSI representation.

Performance is average precision over recall levels of 0.25, 0.50 and 0.75.

```

frame ht 4 wid 5
coord x 0,1100 y .4,.8
label left "Average" "Precision"
label bottom "Dimensions"
label top "MED (LogEntropy)" "Performance vs. number of dimensions"
ticks bot out from 100 to 1100 by 200
draw lsi solid
draw key dashed
copy "med1033.sum.pr" thru % next lsi at 2,1 %
copy "med1033.keywd" thru % next key at 2,1 %
"LSI" rjust size -3 at 250,0.68
"word-matching" rjust size -3 at 250,0.52

```

**Figure 2. MED number of dimensions**

The solid line shows LSI performance as the number of dimensions in the reduced representation varies from 10 to 1033. The dashed line shows word-matching performance. It is clear from this figure that performance improves considerably after 10 or 20 dimensions, peaks between 70 and 100 dimensions, and then begins to diminish slowly. This pattern of performance (initial large increase and slow decrease to word-based performance) is observed with other datasets as well. Theoretically, we expect the performance to increase only while the added dimensions continue to account for meaningful, as opposed to chance, co-occurrence. That LSI works well with a relatively small (compared to the number of unique terms) number of dimensions shows that these dimensions are, in fact, capturing a major portion of the meaningful structure. As noted above, eventually performance must approach the level of performance attained by standard word-matching methods, because with sufficient parameters SVD will exactly reconstruct the original term by document matrix.

We have found that 100-dimensional representations work well for these test collections.

However, the number of dimensions needed to adequately capture the structure in other collections will probably depend on their breadth. Most of the test collections are relatively homogeneous, and 100 dimensions appears to be adequate to capture the major patterns of word usage across documents. In practice, the use of statistical heuristics for determining the dimensionality of an optimal representation will be important

### 3.2 Term weighting.

One of the common and usually effective methods for improving retrieval performance is to give different terms different weights (Sparck Jones, 1972). The raw frequency of occurrence of a term in a document (i.e. the value of a cell in the raw term-document matrix) can be transformed. Such transformations normally have two components. Each term is assigned a *global weight*, indicating its overall importance in the collection as an indexing term. The same global weighting is applied to an entire row (term) of the term-document matrix. It is also possible to transform the term's frequency in the document; such a transformation is called a *local weighting*, and is applied to each cell in the matrix. The value for a term  $t$  in a document  $d$  is  $L(t,d) \times G(t)$ , where  $L(t,d)$  is the local weighting for term  $t$  in document  $d$  and  $G(t)$  is the term's global weighting.

Some popular **local weightings** include: Term Frequency, Binary, and  $\log(\text{Term Frequency} + 1)$ . Term Frequency is simply the frequency with which a given term appears in a given document. Binary weighting replaces any term frequency which is greater than or equal to 1 with 1.  $\log(\text{Term Frequency} + 1)$  takes the log of the raw term frequency, thus dampening effects of large differences in frequencies.

Four well-known **global weightings** are: Normal, GfIdf, Idf, and Entropy. Each is defined in terms of the term frequency ( $tf_{ij}$ ), which is the frequency of term  $i$  in document  $j$ , the document frequency ( $df_i$ ), which is the number of documents in which term  $i$  occurs, and the global frequency ( $gf_i$ ), which is the total number of times term  $i$  occurs in the whole collection.

- Normal:  $\frac{1}{\sqrt{\sum_j tf_{ij}^2}}$
- GfIdf:  $\frac{gf_i}{df_i}$
- Idf:  $\log_2 \left[ \frac{ndocs}{df_i} \right] + 1$ , where  $ndocs$  is the number of documents in the collection
- 1 - Entropy or Noise:  $1 - \sum_j \frac{p_{ij} \log(p_{ij})}{\log(ndocs)}$  where  $p_{ij} = \frac{tf_{ij}}{gf_i}$ , and  $ndocs$  is the number of documents in the collection

All of the global weighting schemes basically give less weight to terms that occur frequently or in many documents. Entropy is based on information theoretic ideas and is the most sophisticated weighting scheme, taking the distribution of terms over documents into account.

We explored the effects of six different term weighting schemes in each of the test collections. We performed analyses using: no global weighting (i.e. raw term frequency,  $tf_{ij}$ ), combinations of the local weight  $tf_{ij}$  and each of the four global weights discussed above (GfIdf, Idf, Entropy, and Normal), and one combination of a local log weight ( $\log(tf_{ij} + 1)$ ) and a global entropy weight (LogEntropy). The original term by document matrix was transformed according to the relevant weighting scheme, and a reduced dimensional SVD was calculated and used for the analysis. Sixty dimensions were used for the ADI collection, and 100 dimensions were used for

the remaining collections. In all cases, query vectors were composed using the same weight used to transform the original matrix.

Figure 3 presents a summary of the term weighting experiments for the CRAN collection. The entries in the table are average precision over three levels of recall (.25, .50 and .75) and over all queries.

```

frame ht 4 wid 5
coord x 0,1 y 0,1
label left "Precision"
label bottom "Recall"
label top "Cranfield subset" "Global weightings - LSI"
draw lentropy solid # logentropy 100
draw entropy dashed # entropy 100
draw idf2 solid # idf2 100
draw nonorm dashed # nonorm 100
draw tfidf solid # tfidf 100
draw norm dashed # norm 100
copy "cran.tiny.lentropy.100" thru % next lentropy at 2,1 %
copy "cran.tiny.entropy.100" thru % next entropy at 2,1 %
copy "cran.tiny.idf2.100" thru % next idf2 at 2,1 %
copy "cran.tiny.tfidf.100" thru % next tfidf at 2,1 %
copy "cran.tiny.nonorm.100" thru % next nonorm at 2,1 %
copy "cran.tiny.norm.100" thru % next norm at 2,1 %
"LogEntropy" rjust size -3 at 0.18,0.75
"Entropy" rjust size -3 at 0.18,0.71
"Idf" rjust size -3 at 0.18,0.68
"Raw TF" rjust size -3 at 0.18,0.59
"GfIdf" rjust size -3 at 0.18,0.54
"Normal" rjust size -3 at 0.18,0.41

```

**Figure 3. CRAN global term weightings**

Differential term weighting has large effects on performance. Normalization and GfIdf are worse than no weighting, and Idf, Entropy and LogEntropy all result in large improvements in performance with LogEntropy being the best. Roughly comparable results are obtained with the other test collections as well. In all cases, LogEntropy results in the best retrieval performance,

with an average advantage over raw term frequency of 40%.

### **3.3 Relevance feedback.**

The idea behind relevance feedback is quite simple. Users are very unlikely to be able to specify their information needs adequately, especially given only one chance. With increases in computer speed, interactive or iterative searches are common, and users can reformulate queries in light of the system's response to previous queries (e.g. Oddy, 1977; Williams, 1984). There is surprisingly little experimental evidence to assess the success of user's successive attempts at query reformulation. Another approach to query reformulation is to have the system *automatically* alter the query based on user feedback about which documents are relevant to the initial request (Salton & Buckley, 1990). This automatic system adaptation is what is usually meant by *relevance feedback*.

Simulations have shown that relevance feedback is quite effective (Stanfill and Kahle, 1986; Salton and Buckley, 1990). Systems can use information about which documents are relevant in many ways. Typically what is done is to increase the weight given to terms occurring in relevant documents and to decrease the weight of terms occurring in non-relevant documents. Our tests using LSI have involved a method in which the initial query is *replaced* with the vector sum of the documents the user has selected as relevant. We do not currently make use of negative information; for example, by moving the query away from documents which the user has indicated are irrelevant.

The document sets and user queries described in Table 1 were used in these experiments. We compared performance with the original user queries against two simulated cases of "feedback"

and against a "centroid" query. Retrieval is first performed using the original query. Then this query is replaced by: the first relevant document, the weighted average or centroid of the first three relevant documents, or the centroid of all relevant documents. The "centroid" condition represents the best performance that can be realized with a single-point query. While this cannot be achieved in practice (except through many iterations), it serves as a useful upper bound on the performance that can be expected given the LSI representation.

Large performance improvements were found for all but the MED dataset, where initial performance was already quite high. Performance improvements average 67% when the first three relevant documents are used as a query and 33% when the first document is used. These substantial performance improvements can be obtained with little cost to the user. A median of 7 documents have to be viewed in order to find the first three relevant documents, and a median of 1 document has to be seen in order to find the first relevant document. Relevance feedback using documents as queries imposes no added costs in terms of system computations. Since both terms and documents are represented as vectors in the same  $k$ -dimensional space, forming queries as combinations of terms (normal query), documents (relevance feedback) or both is straightforward.

The "centroid" query results in performance that is quite good in all but one collection. This suggests that the LSI representation is generally adequate to describe the interrelations among terms and documents, but that users have difficulty in stating their requests in a way that leads to appropriate query placement. In the case of the CISI collection (where there are an average of 50 relevant documents) a single query does not seem to capture the appropriate regions of relevance. We are now exploring a method for representing queries in a way that allows for



multiple disjoint regions to be equally relevant to a query (Kane-Esrig et al., 1989).

How well relevance feedback (or other iterative methods) will work in practice is an empirical issue. We are now in the process of conducting an experiment in which users modify initial queries by: rephrasing the original query; relevance feedback based on user-selected relevant documents; or a combination of both methods. See Dumais and Littman 1990 for a description of the interface and evaluation method.

#### **4. Summary and Conclusions**

LSI is a modification of the vector retrieval method that explicitly models the correlation of term usage across documents using a reduced dimensional SVD. The technique's tested performance ranged from roughly comparable to 30% better than standard vector methods, apparently depending on the associative properties of the document set and the quality of the queries. These results demonstrate that there is useful information in the correlation of terms across documents, contrary to the assumptions behind many vector-model information retrieval approaches which treat terms as uncorrelated.

Performance in LSI-based retrieval can be improved by many of the same techniques that have been useful in standard vector retrieval methods. In addition, varying the *number of dimensions* in the reduced space influences performance. Performance increases dramatically over the first 100 dimensions, reaching a maximum and falling off slowly to reach the typically lower word-based level of performance. Idf and Entropy global *term weighting* improved performance by an average of 30%, and improvements with the combination of a local Log and a global Entropy weighting (LogEntropy) were 40%. In simulation experiments, *relevance feedback* using the

first 3 relevant documents improved performance by an average of 67%, and feedback using only the first relevant document improved performance by an average of 33%. Since the first three relevant document are found after searching through only a median of 7 documents, this method offers the possibility of dramatic performance improvements with relatively little user effort. An experiment is now underway to evaluate the feedback method in practice, and to compare it with other methods of query reformulation.

*REFERENCES*

1. Atherton, P. and Borko, H. A test of factor-analytically derived automated classification methods. AIP rept AIP-DRP 65-1, Jan. 1965.
2. Baker, F.B. Information retrieval based on latent class analysis. *Journal of the ACM*, 1962, 9, 512-521.
3. Bates, M.J. Subject access in online catalogs: A design model. *Journal of the American Society for Information Science*, 1986, 37 (6), 357-376.
4. Blair, D.C. and Maron, M.E. An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Communications of the ACM*, 1985, 28(3), 289-299.
5. Borko, H and Bernick, M.D. Automatic document classification. *Journal of the ACM*, April 1963, 10(3), 151-162.
6. Cullum, J.K. and Willoughby, R.A. *Lanczos algorithms for large symmetric eigenvalue computations - Vol 1 Theory* (Chapter 5: Real rectangular matrices). Birkhauser, Boston, 1985.
7. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman R. A. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 1990, 41(6), 391-407.
8. Dumais, S. T., Furnas, G. W., Landauer, T. K., and Deerwester, S.. Using latent semantic analysis to improve information retrieval. In *CHI'88 Proceedings*, pp. 281-285.

9. Dumais, S.T. and Littman, M.L. InfoSearch: A program for iterative information retrieval using LSI. *CHI'90* Poster.
10. Fidel, R. Individual variability in online searching behavior. In C.A. Parkhurst (Ed.). *ASIS'85: Proceedings of the ASIS 48th Annual Meeting, Vol. 22*, October 20-24, 1985, 69-72.
11. Forsythe, G.E., Malcolm, M.A., and Moler, C.B. *Computer Methods for Mathematical Computations* (Chapter 9: Least squares and the singular value decomposition). Englewood Cliffs, NJ: Prentice Hall, 1977.
12. Furnas, G.W., Landauer, T.K., Gomez, L.M., and Dumais, S.T. The vocabulary problem in human-system communication. *Communications of the ACM*, 1987, 30(11), 964-971.
13. Jardin, N. and van Rijsbergen, C.J. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 1971, 7, 217-240.
14. Kane-Esrig, Y., Casella, G., Streeter, L. A., and Dumais, S. T. Ranking documents for retrieval by Bayesian modeling of a relevance distribution. In *Proceedings of the 12th IRIS (Information System Research In Scandinavia) Conference*, Aug 1989.
15. Koll, M. An approach to concept-based information retrieval. *ACM SIGIR Forum*, XIII32-50, 1979.
16. Oddy, R. N. Information retrieval through man-machine dialogue. *Journal of Documentation*, 1977, 33, 1-14.
17. Ossorio, P.G. Classification space: A multivariate procedure for automatic document indexing and retrieval. *Multivariate Behavioral Research*, October 1966, 479-524.

20. Salton, G. and Buckley, C. Improving retrieval performance by relevance feedback. *JASIS*, 1990, *41(4)*, 288-297.
21. Sparck Jones, K. *Automatic Keyword Classification for Information Retrieval*, Butterworth, London, 1971.
22. Sparck Jones, K. A statistical interpretation of term specificity and its applications in retrieval. *Journal of Documentation*, 1972, *28(1)*, 11-21.
23. Stanfill, C. and Kahle, B. Parallel free-text search on the connection machine system. *Communications of the ACM*, 1986, *29(12)*, 1229-1239.
24. Swets, J. Information retrieval systems. *Science*, 1963, *141*, 245-250.
25. Tarr, D. and Borko, H. Factors influencing inter-indexer consistency. In *Proceedings of the ASIS 37th Annual Meeting, Vol. 11*, 1974, 50-55.
27. Voorhees, E. The cluster hypothesis revisited. *SIGIR*, 1985, 188-196.
28. Williams, M. D. What make RABBIT run? *International Journal of Man-Machine Studies*, 1984, *21*, 333-352.