Information Retrieval using a Singular Value Decomposition Model of Latent Semantic Structure

George W. Furnas (Bellcore)
Scott Deerwester (University of Chicago)
Susan T. Dumais (Bellcore) *Thomas K. Landauer (Bellcore)*
Richard A. Harshman (University of Western Ontario)
Lynn A. Streeter (Bellcore) *Karen E. Lochbaum (Bellcore)*

ABSTRACT

In a new method for automatic indexing and retrieval, implicit higher-order structure in the association of terms with documents is modeled to improve estimates of term-document association, and therefore the detection of relevant documents on the basis of terms found in queries. Singular-value decomposition is used to decompose a large term by document matrix into 50 to 150 orthogonal factors from which the original matrix can be approximated by linear combination; both documents and terms are represented as vectors in a 50- to 150-dimensioal space. Queries are represented as pseudo-documents vectors formed from weighted combinations of terms, and documents are ordered by their similarity to the query. Initial tests find this automatic method very promising.

1. Introduction

Vocabulary mismatch is one of the principal causes of poor recall in information retrieval. Indexers and searchers invariably choose different subsets of words to specify a given topic, causing retrieval techniques based on lexical matching to miss many relevant documents. The word-use variability has been demonstrated in studies of inter-indexer consistency [TARR74] and in the generation of search terms by either expert intermediaries [FIDEL85] or less experienced searchers [LILEY54] [BATES86]. Systematic simulations using extensive human word choice data have shown severe limitations that result for a variety of keyword-based information access schemes [FURNAS83].

Two general approaches to circumventing the vocabulary mismatch problem are sketched in Figure 1. Term expansion uses term-matching, but augments a user's original terms with related words, e.g., from a special thesaurus, in hopes of hitting more targets in the collection. This approach pays a price in scatter; terms with multiple meanings hit spurious targets, leading to rapid degradation of precision [SPARCK72] .

Another approach avoids simple word-match mediated access altogether, by first structuring a collection of documents so as to reflect the semantics of the domain (e.g., some classification or clustering scheme). Retrieval proceeds by using the query to identify and explore some relevant neighborhood in the structure of documents. The advantage is that, if the model of

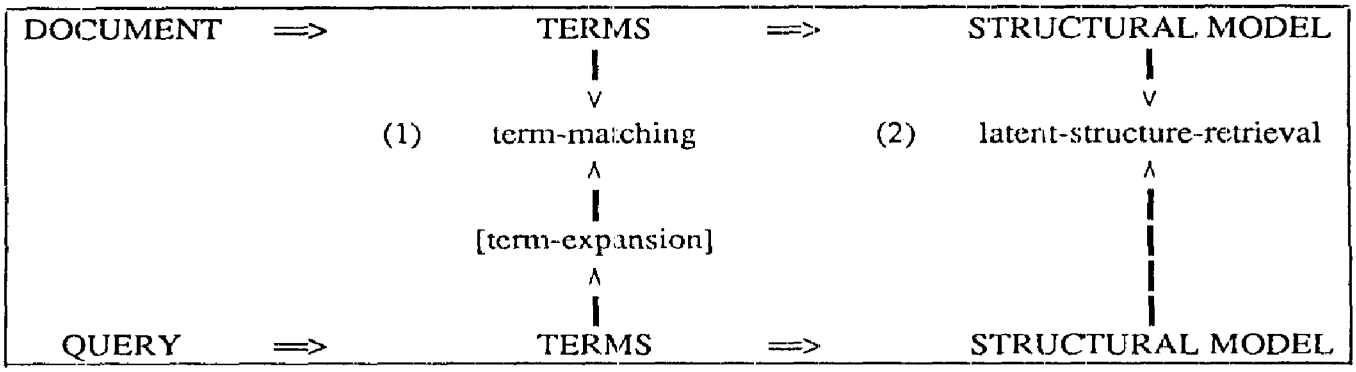
DOCUMENT ==> TERMS ==> STRUCTURAL MODEL
(1) term-matching
[term-expansion]
(2) latent-structure-retrieval
QUERY ==> TERMS ==> STRUCTURAL MODEL

Figure 1. Two frameworks for circumventing variability of word usage, differing in where the document-query comparisons are made. (1) is explicitly based on word matching, but the query is augmented by other terms (e.g., from a thesaurus). (2) Maps both the query and the documents into some semantic structural model and comparisons are made there.

semantic structure is adequate, documents can be retrieved that do not overlap in terminology with the query. Classification analyses of various sorts (e.g., clustering) have been used frequently for structuring document collections (and terms) [SPARCK71] [SALTON68] [JARDIN71],[1] In addition there have been some explorations of Latent Class Analysis [BAKER62], Factor Analysis [ATHERTON65] [BORKO63] [OSSORIO66], and, more recently, attempts at knowledge representation using Artificial Intelligence techniques.

The Term and the Structural Model domains of Figure 1, though conceptually distinct, are importantly inter-related. Comparisons between a query and a document that are nominally in one domain have an implicit corresponding comparison in the other domain. For example, matches made explicitly on the basis of the semantic structural model imply certain matching relations at the term level. Conversely, term-expansion techniques implicitly postulate some hidden underlying semantic structure. This paper presents a framework and specific technique that explicitly links both approaches. It arises naturally from a formulation of the vocabulary problem in terms of statistical sampling. Thus, statistical methods are used to improve sample estimates of term document association (the term matching domain), by estimating parameters of latent semantic structure (structural model domain). Retrieval may be conceived explicitly as term matching using the improved estimates, or in dual fashion, as exploring neighborhoods in the latent structural model.

2. Theory

2.1 The Framework: Truth with Structure Obscured by Error

Consider the familiar rectangular term-document matrix, whose entries tell whether (or how often) a term is contained in a given document. Variability in word choice behavior means that an author or indexer will think to use only a sample of the plausible terms to describe the topic of a document. Thus the terms actually *observed* to be associated in an index with a document are only a sample of the *true* , larger pool of terms that might have been associated with it. In this sense the observed term-document matrix can be thought of as a true association matrix obscured by some sort of sampling error. If one had access to the true matrix, there would be no vocabulary-based recall failures -- if a document was relevant to a

[1] We note that often such clustering approaches have made their principal contribution in increased retrieval speed, rather than increased recall. [SALTON83]

query term, the true matrix would indicate the match.

The problem, of course, is that we do not have access to the true matrix; we must settle for only an estimate. The observed matrix is one such estimate. It is possible, however, that better estimates exist. Indeed the observed matrix is its own best estimate only if there is no structure in the true matrix. However, there is structure, since for example, some closely related documents should contain nearly identical patterns of terms, and synonymous terms should have highly similar patterns of occurrence across documents.

To say that there is structure in an $t \times d$ matrix of terms by documents is to say that there exists a more parsimonious representation -- one with redundancy squeezed out and as a result requiring fewer than $t \times d$ parameters. If one has a good model of the underlying structure, one can make better estimates of the true matrix (Figure 2). The cells of the observed matrix are used to estimate parameters of the underlying model. Since there are fewer parameters than data cells, this model can be more statistically reliable than the raw data, and can be used to reconstruct an improved estimate of the cells of the true matrix. It may even be possible, as in the technique studied in this paper, to extend the model of structure to the query itself, trying to re-estimate the terms properly associated with it.

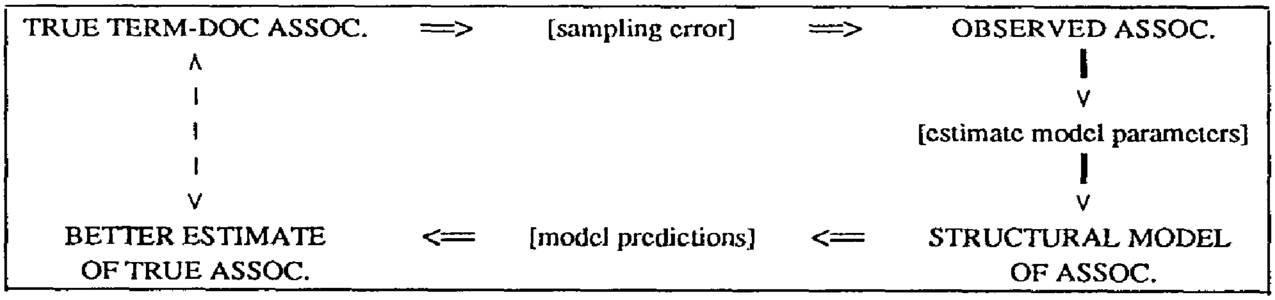
TRUE TERM-DOC ASSOC.
==>
[sampling error]
==>
OBSERVED ASSOC.
[estimate model parameters]
BETTER ESTIMATE OF TRUE ASSOC.
<==
[model predictions]
<==
STRUCTURAL MODEL OF ASSOC.

Figure 2. Using a latent structural model to improve estimates of term document association.

Armed with the new estimate of the true matrix, one could carry out improved term-match mediated retrieval. The re-estimation process has acted much like a thesaurus or term-expansion device, filling out the index entries for documents in the collection, and perhaps expanding the term-list associated with the query. Ideally, of course, it is a careful expansion, estimating the true term-document associations and not the spurious ones that might plague a simple thesaurus.

This approach can also be cast in the second framework, where retrieval is not explicitly mediated by term-match at all. Particularly if it is possible to extend the model of true structure to the query, the comparison of the query against the document collection can be done completely in the world of the underlying structure. For example, if the appropriate model for structure is cluster-like, parameters are estimated from the observed data and a cluster membership is determined also for the query, then we have the familiar cluster-based information retrieval.

Thus the duality between these two approaches is held together by the statistical modeling link between the underlying structure and its corresponding estimate of the observed association matrix. By making this duality explicit, one can exploit statistical techniques for modeling true structure in the presence of obscuring error and, in principle, exploit refinements of both existing frameworks.

2.2 The choice of structural model

The fundamental question in this approach concerns the nature of the true semantic structure latent in the observed relationships between terms and documents. A notion of semantic similarity, between documents and between terms, seemed central to modeling the patterns of term usage across documents. This led us to restrict consideration to proximity models, i.e., models that try to put similar items near each other in some space or structure. Such models include: hierarchical, partition and overlapping clusterings; ultrametric and additive trees; and factor-analytic and multidimensional distance models (see [CARROLL80] for a survey).

In choosing a model we considered the following three criteria:

1. *Adjustable representational richness.* To represent the underlying semantic structure, we need a model with sufficient power. We believe hierarchical clusterings to be too restrictive, since they allow no multiple or crossed classifications and have essentially only as many parameters as objects. Since the right kind of alternative is unknown, we looked for models whose power could be varied, as some compensation for choosing a perhaps inappropriate structure. The most obvious class is dimensional models, like multidimensional scaling and factor analysis, where representational power can be controlled by choosing the number, k, of dimensions (i.e., yielding k parameters per object).

2. *Explicit representation of both terms and documents.* The original data explicitly relate two types of entities, terms and documents, yet most representations chosen so far handle only one at a time (e.g., the dichotomy of term clustering vs. document clustering; though we note [KOLL79] as an exception). In addition to theoretical elegance, there are practical advantages to simultaneously representing both terms and documents. If terms, as well as documents, have positions in the structure, then a query can become a new object placed at something like the centroid of the terms it contains. Retrieval then proceeds by finding those documents that are close to the query. Also, new objects not in the original data matrix can be placed after the fact into the structure analogously; new terms at the centroid of their associated documents, and new documents at the centroid of their associated terms.[2]

   Thus we needed what are called two-mode proximity methods [CARROLL80], that start with a rectangular matrix and construct explicit representations of both row and column objects. Such methods include multidimensional unfolding [COOMBS64] [HEISER81] [DESARBO85], two-mode factor analysis [HARSHMAN70] [HARSHMAN84] [CARROLL70] [KRUSKAL78] , and unfolding in trees [FURNAS80] .

3. *Computational tractability for large datasets.* We wanted the technique to be fully automatic, fitting the semantic structure directly to the term-document matrix. Many of the existing models require computation that goes approximately as N^4 or N^5 (where N is the number of terms plus documents). Since we hoped to work with document sets that were at least in the thousands, models with efficient fitting techniques were needed.

[2] Koll [KOLL79] used this centroid placement technique, after an initial heuristic starting configuration, to construct a representation very similar in spirit to ours. We use the matrix decomposition method, SVD (see below), to construct the final configuration, and a technique analogous to centroid placement to augment it with new objects.

To satisfy these criteria we chose a generalization of the familiar factor-analytic model, called "two-mode factor analysis", based on singular value decomposition (SVD). (See [FORSYTHE77], Chapter 9, for an introduction to SVD and its applications.) SVD can represent both terms and documents as vectors in a space of controllable dimensionality, where the inner-products between points in the space gives their similarity. In addition, a program was available [HARSHMAN84] that fit the model with an algorithm requiring computation only of roughly order $N^2 \times k^3$ (where k is the number of dimensions).[3]

2.3 The Singular Value Decomposition (SVD) Model.

Any rectangular matrix **X**, for example a $t \times d$ matrix of terms and documents, can be decomposed into the product of three other matrices:

$$(1) \qquad \mathbf{X} = \mathbf{T}_m \mathbf{S}_m \mathbf{D}_m{}^t,$$

such that $\mathbf{T}_m$ and $\mathbf{D}_m$ have orthonormal columns and $\mathbf{S}_m$ is diagonal. This is called the *singular value decomposition of* **X**. $\mathbf{T}_m$ and $\mathbf{D}_m$ are the matrices of *left* and *right singular vectors* and $\mathbf{S}_m$ is the $m \times m$ diagonal matrix matrix of *singular values* (where $m = \min(t, d)$)[4]

Figure 3 presents a schematic of the singular value decomposition for a $t \times d$ matrix of terms by documents.

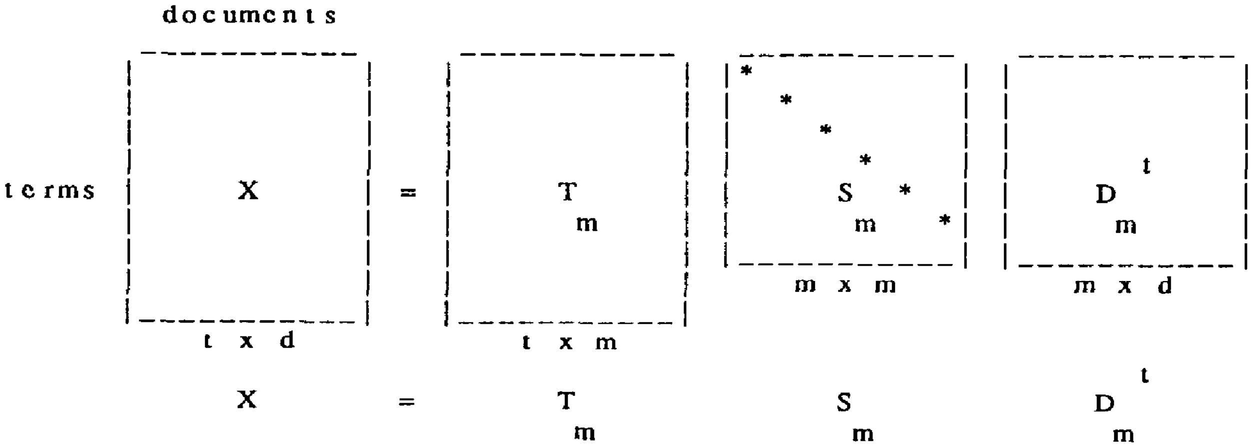

Figure 3. Singular value decomposition of the term x document matrix, **X**. Where :
$\mathbf{T}_m$ has orthogonal, unit-length columns ($\mathbf{T}_m{}^t\mathbf{T}_m = \mathbf{I}$)
$\mathbf{D}_m$ has orthogonal, unit-length columns ($\mathbf{D}_m{}^t\mathbf{D}_m = \mathbf{I}$)
$\mathbf{S}_m$ is the diagonal matrix of singular values
t is the number of rows of **X**
d is the number of columns of **X**
m is the rank of **X** ($\leq \min(t, d)$)

[3] The algorithm is in fact iterative and non-deterministic, so only an estimate can be given.

[4] SVD is closely related to an eigen decomposition of a square symmetric matrix, Y, into VLV^t, where V is orthonormal and L is diagonal. The relation between SVD and eigen analysis is more than one of analogy. In fact, $\mathbf{T}_m$ is the matrix of eigenvectors of the square symmetric matrix $Y = XX^t$, $\mathbf{D}_m$ is the matrix of eigenvectors of $Y = X^t\mathbf{X}$, and in both cases, $\mathbf{S}_m{}^2$ would be the matrix, L, of eigenvalues. Note that there will be zero eigenvalues when the rank of Y is less than t or d. As in the eigen decomposition of a square matrix, singular value decomposition is unique up to certain row and column permutations and assignments of sign (important here principally in that they allow the convention that the diagonal elements of $\mathbf{S}_m$ are all positive and ordered in decreasing magnitude).

In general, if $\mathbf{X}=\mathbf{T}_m\mathbf{S}_m\mathbf{D}_m{}^t$ is of full rank, then the matrices $\mathbf{T}_m$, $\mathbf{D}_m$, and $\mathbf{S}_m$ must be also. However, if only the k largest singular values of $\mathbf{S}_m$ are kept along with their corresponding columns in the $\mathbf{T}_m$ and $\mathbf{D}_m$ matrices, and the rest deleted (yielding matrices $\mathbf{S}_k$, $\mathbf{T}_k$ and $\mathbf{D}_k$), the resulting matrix, $\hat{\mathbf{X}}$, is the unique matrix of rank k which is closest in the least squares sense to $\mathbf{X}$:

$$(2) \qquad \begin{aligned} \hat{\mathbf{X}} &= \mathbf{T}_k\mathbf{S}_k\mathbf{D}_k{}^t \\ &\approx \mathbf{X} \end{aligned}$$

The idea is that this matrix, by containing only the k largest independent linear components of $\mathbf{X}$, captures the major associational structure of the data and throws out much of the noise. It is this reduced model, presented in Figure 4, that we use to approximate our data. (For notational convenience we will henceforth drop the subscript k, writing simply $\mathbf{S}$, $\mathbf{T}$ and $\mathbf{D}$ for $\mathbf{S}_k$, $\mathbf{T}_k$ and $\mathbf{D}_k$ respectively.)

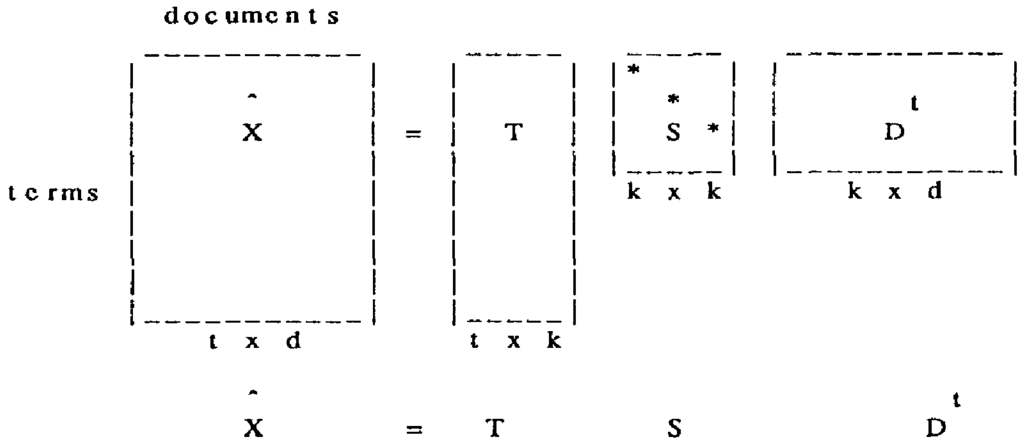

Figure 4. Reduced singular value decomposition of the term x document matrix, $\mathbf{X}$. Notation is as in the previous figure except that k ($\leq m$) is the chosen number of dimensions (factors) in the reduced model.

In choosing the reduced dimensionality, k, we want a value large enough to fit all the real structure in the data, but small enough so that we do not also fit the sampling error or unimportant details. We currently use that value of k which maximizes our sample retrieval performance.

Note that it is useful to interpret the row vectors of the SVD matrices, $\mathbf{T}$ and $\mathbf{D}$, geometrically. If they are taken as coordinates in a k-dimensional space, terms and documents become points in a vector space (the "factor space"). The diagonal matrix, S, serves to stretch or shrink the orthogonal axes of this space, reflecting the relative contribution of those directions to the overall similarity structure.

2.4 Theory of SVD use in Information Retrieval

2.4.1 Term matching paradigm

In a standard vector-based version of term matching, the similarity of two documents is obtained by comparing, e.g., using an inner-product or cosine measure, the corresponding two column vectors of the raw data matrix X. A query is represented as a sort of pseudo-document, i.e., a column vector of term frequencies, $\mathbf{X}_{*q}$, which is similarly compared against columns of $\mathbf{X}$, and the best matches found.

These same calculations may be done using versions of these matrices that have been "cleaned up" by the SVD estimation process. The matrix $\hat{\mathbf{X}}$ of equation (2) would be used in place of $\mathbf{X}$. It may be shown that the appropriate "cleaned up" version of the query column-vector, $\mathbf{X}_{\bullet q}$, is given by $\hat{\mathbf{X}}_{\bullet q} = \mathbf{T}\mathbf{T}^{\mathbf{t}}\mathbf{X}_{\bullet q}$. If these "cleaned up" versions indeed reflect better estimates of the true term document association structure, then retrieval using them should be superior.

2.4.2 Latent structure paradigm

The various comparison calculations made in the vector-matching characterization of the proposed approach used $\hat{\mathbf{X}}$ and $\hat{\mathbf{X}}_{\bullet q}$. These calculations have exact analogs in the latent model, just using the components of the Singular Value Decomposition. For example, the similarity between two documents, in the term-matching paradigm, can be calculated by an inner-product. Calculating these similarities for all pairs of documents is equivalent to the matrix multiplication, $\hat{\mathbf{X}}^{\mathbf{t}}\hat{\mathbf{X}}$. But according to the SVD decomposition of equation (2) this is algebraically equivalent to,

$$(3) \qquad \hat{\mathbf{X}}^{\mathbf{t}}\hat{\mathbf{X}} = (\mathbf{TSD}^{\mathbf{t}})^{\mathbf{t}}\mathbf{TSD}^{\mathbf{t}} = \mathbf{DST}^{\mathbf{t}}\mathbf{TSD}^{\mathbf{t}} = \mathbf{DSSD}^{\mathbf{t}} = (\mathbf{DS})(\mathbf{DS})^{\mathbf{t}}.$$

Thus comparison of document i and document j may be made by taking the inner product of rows i and j of the matrix, $\mathbf{DS}$. The result is equivalent to its dual -- doing a document vector comparison on the large $\hat{\mathbf{X}}$ matrix.

Similarly term-term comparisons, i.e., the inner-product of pairs of term rows, are cells of the matrix $\hat{\mathbf{X}}\hat{\mathbf{X}}^{\mathbf{t}}$, and

$$(4) \qquad \hat{\mathbf{X}}\hat{\mathbf{X}}^{\mathbf{t}} = \mathbf{TSD}^{\mathbf{t}}(\mathbf{TSD}^{\mathbf{t}})^{\mathbf{t}} = \mathbf{TSD}^{\mathbf{t}}\mathbf{DST}^{\mathbf{t}} = \mathbf{TSST}^{\mathbf{t}} = (\mathbf{TS})(\mathbf{TS})^{\mathbf{t}}.$$

So comparison of term i and term j may be made by taking the inner product of rows i and j of the matrix, $\mathbf{TS}$.

Recall finally that the association between term i and document j, which is the ij cell of $\hat{\mathbf{X}}$, is by construction,

$$(5) \qquad \hat{\mathbf{X}} = \mathbf{TSD}^{\mathbf{t}} = (TS^{\frac{1}{2}})(DS^{\frac{1}{2}})^{\mathbf{t}},$$

i.e., the inner product of row i of the matrix, $\mathbf{TS}^{\frac{1}{2}}$, and row j of the matrix, $\mathbf{DS}^{\frac{1}{2}}$.

Thus all useful comparisons can be made using the rows of the matrices $\mathbf{T}$ and $\mathbf{D}$, appropriately scaled by the diagonal matrix $\mathbf{S}$. These matrices are much smaller than $\mathbf{X}$ (since we assume $k \ll t$ and $k \ll d$), so the row vectors form an efficient indexing system for the terms and objects, an index made more efficient both by the elimination of noise and by removing redundancy inherent in the non-independence of term co-occurrence.

These operations may be given a geometric interpretation: If the axes of the space are rescaled by the associated diagonal values of $\mathbf{S}$, the inner-product between term points or document points can be used to make the algebraic comparisons of interest. (The axes must be rescaled by the associated diagonal values of $\mathbf{S}^{\frac{1}{2}}$ for comparisons between a term and a document.)

To complete the latent structure view of the retrieval system, the query must be given a representation within the SVD model. Its representation must yield results consistent with the procedure used in the term-matching conceptualization. The query must be a "pseudo-document" assigned coordinates in the SVD space such that its inner-product to other document points yields the same result as obtained by comparing the full query vector, $\hat{\mathbf{X}}_{\bullet q}$, with the full $\hat{\mathbf{X}}$. That is, we want to calculate, from a new document-like column, $\mathbf{X}_{\bullet q}$ of the **X** matrix, a new row, $\mathbf{D}_{q\bullet}$, of coordinates in the **D** matrix. A little algebra gives,

$$(6) \qquad \mathbf{D}_{q\bullet} = \mathbf{X}_{\bullet q}^{t}\mathbf{T}\mathbf{S}^{-1}$$

Note that with appropriate rescaling of the axes, this amounts to placing the pseudo-document at the center (actually the vector sum) of its corresponding term points. This $\mathbf{D}_{q\bullet}$ then is just like a row of **D** and can be used in the same manner as ordinary document's factor vectors. I.e., equations (3) and (5) will yield comparisons to other documents or to other terms, respectively.

2.5 The procedure

In brief, the Singular Value Decomposition is incorporated in an information retrieval system as follows. A collection of documents has its content terms tabulated to give a frequency matrix, which is taken as **X**. A k-dimensional SVD decomposition of **X** is computed yielding matrices **T**, **S**, and **D**. The rows of **T** and **D** are taken as index vectors for corresponding terms and documents, respectively. The diagonal elements of S (or $S^{1/2}$, as needed), are taken as component-wise weights in ensuing similarity calculations. A query, treated as vector of term frequencies (albeit very sparse), is converted to a pseudo-document, $\mathbf{D}_{q\bullet}$, in the factor space following equation (6). This query factor-vector is then compared to the factor-vectors of all the documents[5], and the documents ordered according to the results.

2.6 A worked-out example

A numerical example may help to make all this clearer and more intuitive. For the example we take a set of nine titles (only) of selected technical memoranda produced at Bellcore as shown in Table 1. Note that five of these concern human-computer interaction, and the other four the discrete math of graph theory. To illustrate one of the positive features of the Latent Semantic Structure Analysis method, consider what would happen if the query " human interaction with computers" were passed against this database. In traditional keyword matching techniques, e.g. vector methods, all the math documents (titles) would be rejected, since none contain any of these terms, but so would documents C3 and C5, which are clearly relevant.

Now let us work through an SVD with dimension reduction for these documents. The term by document table (excluding for convenience terms that occur only in one document and a few stop words) given in Table 1 shows the cell frequencies comprising the **X** matrix. The full Singular Value Decomposition of the Term x Document matrix of Table 1 is given in Table 2.

For expository purposes we want a simple solution for which we can give a graphical display, so instead of the usual 50 - 150 dimensional representation, we will use a 2-dimensional solution, i.e., approximate **X** keeping only the first two singular values and the corresponding columns from the $\mathbf{T}_m$ and $\mathbf{D}_m$ matrices. (These are the **T** and **D** coordinates used to position the 12 terms and 9 documents, respectively, in Figure 5.) This *reduced model* (Table 3), the reader can verify that $\mathbf{X}=\mathbf{T}_m\mathbf{S}_m\mathbf{D}_m'$ (except for small rounding errors) $\mathbf{T}_m$ has orthogonal, unit length columns so $\mathbf{T}_m\mathbf{T}_m'=\mathbf{I}$ and $\mathbf{D}_m$ has orthogonal, unit length columns so $\mathbf{D}_m\mathbf{D}_m'=\mathbf{I}$.

[5] We currently actually do this comparison by cosine, not raw inner-product, for reasons we will not go into here.

TABLE 1

Technical Memo Example

Titles:

c1: *Human* machine *interface* for Lab ABC *computer* applications
c2: A *survey* of *user* opinion of *computer system response time*
c3: The *EPS user interface* management *system*
c4: *System* and *human system* engineering testing of *EPS*
c5: Relation of *user*-perceived *response time* to error measurement

m1: The generation of random, binary, unordered *trees*
m2: The intersection *graph* of paths in *trees*
m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4: *Graph minors*: A *survey*

Terms					**Documents**				
	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Multiplying out the matrices TSD′ gives $\hat{X}$, the estimate of X, which at the bottom of Table 3.

There are two things to note about the $\hat{X}$ matrix. (1) It does not match the original term by document matrix X (it would get closer and closer as more and more singular values were kept). (2) This is what we want; we do not want perfect fit because we think some of the 0's in X should be closer to 1 and vice versa. Note in particular that the cells in bold in the $\hat{X}$ matrix corresponding to the zero entries for "human" and "computer" in the X matrix now contain the values .38 and .36 for "human", and .18 and .24 for "computers" for titles c3 and c5 respectively, and that all these values are considerably higher than any for the comparable terms in any of the math titles. Thus, the method has automatically filled in appropriate term strengths on the basis of structure implicit in overall term by document matrix. Note also, for example, that if one computes the cosine between *user* and *human*, which do not occur in any common document is 0.89 in the reduced SVD space, where it was 0.0 in the raw vector space. This last observation illustrates the method's ability to capture implicit synonymy.

The same effects can be seen graphically in Figure 5, where the geometric interpretation of the two- factor solution shows clearly that all the human-computer papers have been nicely separated from all the math papers. Both terms and documents are represented in this two-dimensional space. The "human interaction with computers" query has been treated as a "pseudo-document" and placed at the weighted vector sum of its component terms. The angle of its vector with that of all relevant documents, whether they share terms with it or not, is less than with any of the math papers.

TABLE 2

T_m =

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

S_m =

3.34								
	2.54							
		2.35						
			1.64					
				1.50				
					1.31			
						0.85		
							0.56	
								0.36

D_m =

0.20	-0.06	0.11	-0.95	0.05	-0.08	0.18	-0.01	-0.06
0.61	0.17	-0.50	-0.03	-0.21	-0.26	-0.43	0.05	0.24
0.46	-0.13	0.21	0.04	0.38	0.72	-0.24	0.01	0.02
0.54	-0.23	0.57	0.27	-0.21	-0.37	0.26	-0.02	-0.08
0.28	0.11	-0.51	0.15	0.33	0.03	0.67	-0.06	-0.26
0.00	0.19	0.10	0.02	0.39	-0.30	-0.34	0.45	-0.62
0.01	0.44	0.19	0.02	0.35	-0.21	-0.15	-0.76	0.02
0.02	0.62	0.25	0.01	0.15	0.00	0.25	0.45	0.52
0.08	0.53	0.08	-0.03	-0.60	0.36	0.04	-0.07	-0.45

3. Tests and Applications of the Method

3.1 Results for Standard Document Sets

We examined performance in two standard document sets for which user queries and relevance judgments are available. Their performance is described briefly here; more details will be available elsewhere [DUMAIS89].

The first database consisted of an often studied corpus of 1033 medical reference abstracts and titles (MED). Automatic indexing found 5823 terms occurring in more than one document. A 100-factor SVD of the 5823 term by 1033 document matrix was obtained and retrieval effectiveness evaluated against 30 queries available with the dataset. The average precision over 9 levels of recall from .10 to .90, was .51 for the SVD approach and .45 for basic inner-product term matching. This 13% improvement over raw term matching shows that the SVD captured some structure in the data which was missed by raw term matching. Improvements were especially large at higher levels of recall, where we would expect word matches to fail.

The second standard dataset consisted of 1460 information science abstracts (CISI) that have been consistently difficult for automatic retrieval methods. Automatic indexing found 5135

TABLE 3

$$\hat{X} = \overset{T}{\begin{bmatrix} 0.22 & -0.11 \\ 0.20 & -0.07 \\ 0.24 & 0.04 \\ 0.40 & 0.06 \\ 0.64 & -0.17 \\ 0.27 & 0.11 \\ 0.27 & 0.11 \\ 0.30 & -0.14 \\ 0.21 & 0.27 \\ 0.01 & 0.49 \\ 0.04 & 0.62 \\ 0.03 & 0.45 \end{bmatrix}} \overset{S}{\begin{bmatrix} 3.34 & \\ & 2.54 \end{bmatrix}} \overset{D'}{\begin{bmatrix} 0.20 & 0.61 & 0.46 & 0.54 & 0.28 & 0.00 & 0.02 & 0.02 & 0.08 \\ -0.06 & 0.17 & -0.13 & -0.23 & 0.11 & 0.19 & 0.44 & 0.62 & 0.53 \end{bmatrix}}$$

$$\hat{X} = \begin{bmatrix} 0.16 & 0.40 & 0.38 & 0.47 & 0.18 & -0.05 & -0.12 & -0.16 & -0.09 \\ 0.14 & 0.37 & 0.33 & 0.40 & 0.16 & -0.03 & -0.07 & -0.10 & -0.04 \\ 0.15 & 0.51 & 0.36 & 0.41 & 0.24 & 0.02 & 0.06 & 0.09 & 0.12 \\ 0.26 & 0.84 & 0.61 & 0.70 & 0.39 & 0.03 & 0.08 & 0.12 & 0.19 \\ 0.45 & 1.23 & 1.05 & 1.27 & 0.56 & -0.07 & -0.15 & -0.21 & -0.05 \\ 0.16 & 0.58 & 0.38 & 0.42 & 0.28 & 0.06 & 0.13 & 0.19 & 0.22 \\ 0.16 & 0.58 & 0.38 & 0.42 & 0.28 & 0.06 & 0.13 & 0.19 & 0.22 \\ 0.22 & 0.55 & 0.51 & 0.63 & 0.24 & -0.07 & -0.14 & -0.20 & -0.11 \\ 0.10 & 0.53 & 0.23 & 0.21 & 0.27 & 0.14 & 0.31 & 0.44 & 0.42 \\ -0.06 & 0.23 & -0.14 & -0.27 & 0.14 & 0.24 & 0.55 & 0.77 & 0.66 \\ -0.06 & 0.34 & -0.15 & -0.30 & 0.20 & 0.31 & 0.69 & 0.98 & 0.85 \\ -0.04 & 0.25 & -0.10 & -0.21 & 0.15 & 0.22 & 0.50 & 0.71 & 0.62 \end{bmatrix}$$

terms occurring in more than one document. A 100-factor SVD solution was obtained for the 5135 term by 1460 document matrix and evaluated using 35 queries available with the dataset. For this particular dataset, the SVD approach offered no improvement over term matching methods; precision for all methods was below .30, even for the lowest levels of recall.

3.2 Results for a Novel IR Application: An Expert-Locating System

Traditional information retrieval has focused on documents. However, the satisfaction of information requirements is frequently accomplished instead by finding a person or organization capable of providing expert knowledge. We [STREETER87] have developed a system that accomplishes this goal using the latent structural model described above. To give a concrete example of the method in use, we describe this application in some detail.

3.2.1 Document Collection and Document Preprocessing

In the expert-locating application, research groups were characterized by a representative collection of technical documents which they had written. For each of the company's first-level working groups, we collected the annual project write-ups each must prepare (approximately 270 content words) and, where possible, abstracts of their technical papers for the previous 18 months.

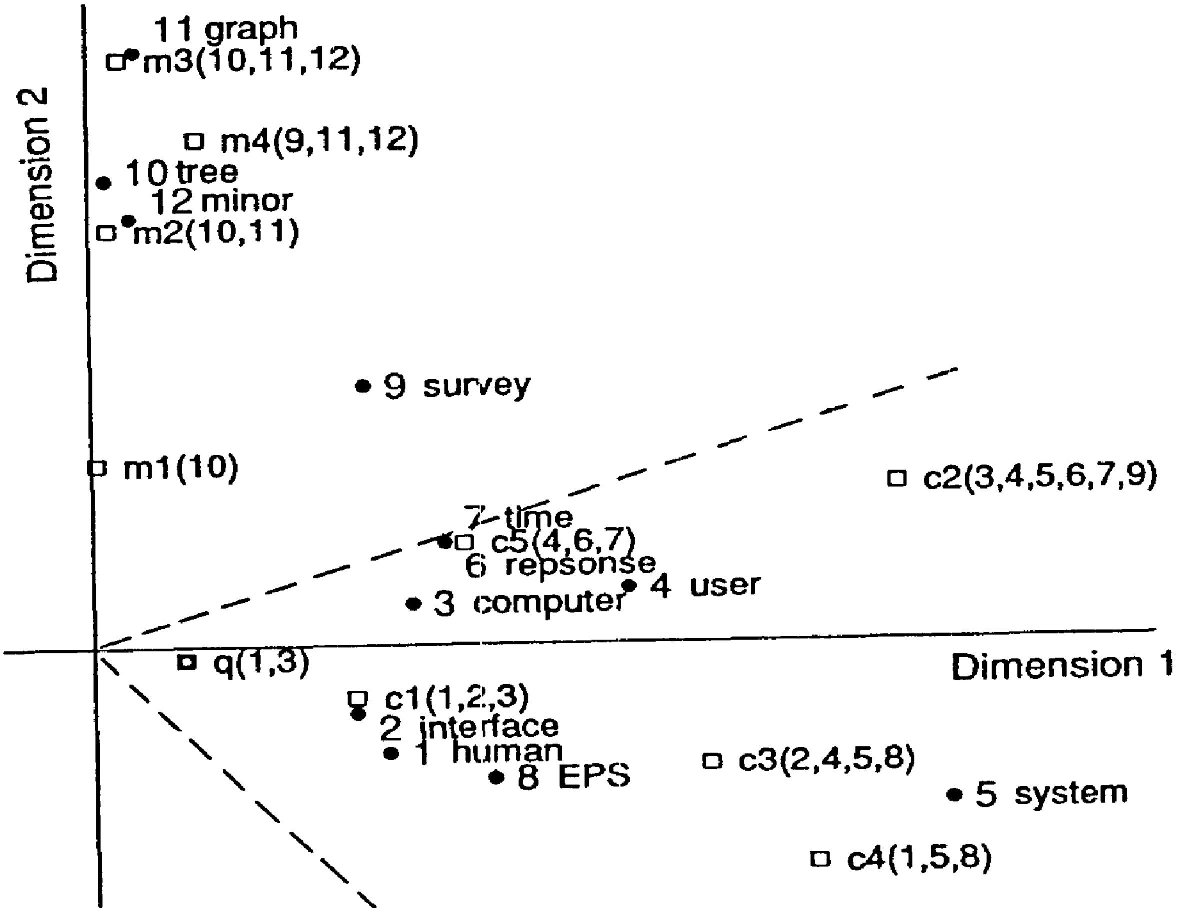

Figure 5. A 2-dimensional plot of 12 Terms and 9 Documents from the example set. Terms are represented by filled circles. Documents are shown as open squares, and component terms are indicated parenthetically. The query ("human computer interaction") is represented as a pseudo-document at point q. Axes are appropriately scaled for Document-Document or Term-Term comparisons The dotted cone contains all points within a cosine of .9 from the query q. All documents about human-computer (c1-c5) are within this cone, but none of the graph theory documents (m1-m4) are nearby. In this reduced space, even documents c3 and c5, which share no terms with the query, are very close to the query direction.

All text was preprocessed to isolate possible compound noun phrases. Potential phrases were word strings falling between any two of a set of 160 delimiters and punctuation marks. Inflectional suffixes (past tense, plurals, progressive tense, and adverbials) were removed from the words. The resulting list of phrases was manually edited to include only noun phrases. Compound phrases ranged from two to eight words. All full compound phrases, single words making up the compounds, and single words that occurred in more than two documents and were not among the most frequent 150 English words were entered into the latent semantic structure analysis. Of the 7,100 terms in the system lexicon, 2,879 were compounds.

3.2.2 SVD Analysis

A singular value decomposition on 7,100 terms and 728 documents, representing 480 research groups in 100 dimensions was performed. The collection of all technical memorandum abstracts and the work descriptions for a single organization were treated as two separate documents for the purposes of analysis. For some of the work descriptions the only available identifying information was the superordinate department level rather than the research group.

3.2.3 Processing a Query

To process a query, inflectional suffixes in the query were first removed and potential phrases identified. The query vector contained all phrases (if short phrases were nested in longer phrases, only the longest was used) and all other words occurring outside of phrases. For each query term or phrase that appeared in the system lexicon, the corresponding 100 dimensional vector from the SVD analysis was retrieved. For the analyses to be reported here, the query vector was the centroid of these 100 dimensional vectors for each of the terms, appropriately scaled by S.

The query vector was then compared to all document vectors (728) in the space. The similarity metric used was the cosine between the query vector and the document vectors. (For this application, we found that the inner product measure produced poorer performance.)

The cosines were then sorted and for each of the N research groups (typically 10) with the best fitting document, the value of the cosine (fit) and the identity of the group were returned to the user. The match of a research group (or department) was taken as that of the maximum match of any of its associated documents.

3.2.4 System Performance and Effects of Experimental Variables

Two separate semantic spaces were constructed, each with 100 dimensions. In one analysis, compound noun phrases, their single word components, and all other single words were the input terms (compound and single word space). In the second analysis, compounds were not included in the input terms (single word space).

To evaluate the method and variables of interest, we collected a new set of 263 technical abstracts, not used to construct the semantic space, which became the test queries.

Three different types of queries were compared (1) both words and compounds from the abstract, (2) only words of the title and (3) only a single keyword or key phrase from the title (selected by us).

The cosine similarity between every query and each of the 728 documents was calculated. The measure of success was the rank with which the system predicted the group that had produced the abstract used to form the query. Table 4 shows the results in terms of the median rank of the correct department. (Note, that the department is one organizational level above the research group; there were 104 departments) Thus, if the method were perfect, the correct department's rank would be 1; by chance the rank would be 52. Treating the entire abstract as a query and using the space derived from including both compound noun phrases and single words, the method predicted the correct department with median rank 3.

TABLE 4
PERFORMANCE STATISTICS
(Chance rank = 52)

Query Terms	SVD Terms	Median Correct Rank	Median Cosine
Words & Compounds	Compounds & Words	3.0	0.57
Single words	Single words	5.0	0.56
Compounds treated as single words	Single words	11.5	0.42
Title words only	Compounds & Words	5.0	0.49
Keyword(s) only	Compounds & Words	8.0	0.46

The relevant findings to note in Table 4 are: (1) representing compound noun phrases in the space improved performance; and (2) longer queries performed better. With regard to the first point, performance was best when both the space and query were based on both compounds and words. When compounds were decomposed into the single words that comprise them, and the term vectors found in the single-word semantic space, performance decreased substantially. It is likely that the advantage of using compounds is related to disambiguation of terms. Note that the latent structure method will tend to place a polysemous term at the usage-weighted average location of its different meanings.

3.2.5 Comparison with Raw Term Space

We compared the performance of the latent structure dimension reduction technique with ordinary vector retrieval based on the raw term space for two sets of queries. One set was the 263 technical abstracts described above, the other a set of current individual project descriptions solicited from 40 people. Table 5 shows the results. For project descriptions, similarities derived from the latent structure analysis outperformed those from raw term space. For the technical abstract queries, performance was approximately equal for the two methods. Ranks determined by the SVD and raw term space methods were only moderately correlated (0.40 for technical abstracts and 0.53 for personal descriptions). Thus, while each method worked moderately well, they did not rank documents in the same order, making it feasible to consider combining them. Because cosines in the higher dimensional raw term space are on the average much smaller than those in the SVD space, the indices were normalized to standard deviation units before combining. Using a maximum rule, that is, selecting whichever method produced the higher normalized cosine index, clearly yielded performance considerably better than that of either method alone.

TABLE 5
COMPARING SVD AND RAW TERM SPACE
PERFORMANCE STATISTICS
Number of Dimensions = 100

Method	Query Collection	Median and 75th percentile rank of true match
SVD	Technical Abstracts	3, 17
Raw Term Space	Technical Abstracts	3, 14
SVD	Personal Descriptions	2, 9
Raw Term Space	Personal Descriptions	4, 23
Max of SVD & Raw Term Space	Technical Abstracts	1, 7
Max of SVD & Term Raw Space	Personal Descriptions	1, 5

4. Conclusions

The technique sketched here appears useful for helping people find textual information in moderately large collections. It uses the estimation of latent structure to re-estimate the term-document association matrix, in this way reducing the vocabulary mismatch problems that limit recall. Its retrieval performance compares favorably with existing systems, and it is capable of retrieving objects missed by other methods.

Making the link between the term-match and latent structure paradigms explicit grounds the model theoretically and statistically, and results in a completely automatic computational process. The duality also affords double richness for future extensions, as improvements natural in either domain can be explored. Refinements from the vector and term-match approach, some of which were explored in the previous example, might include term weighting, stemming, use of short-phrases and booleans. Refinements in the latent structure realm might include finding better notions of "relevance neighborhoods" in the structure, identifying such neighborhoods using relevance feedback, or examining other methods of uncovering latent structure, including highly parallel "learning machines".

REFERENCES

[ATHERTON65] Atherton, P. and Borko, H. A test of factor-analytically derived automated classification methods. AIP rept AIP-DRP 65-1, Jan. 1965.

[BAKER62] Baker, F.B. Information retrieval based on latent class analysis. *Journal of the ACM*, 1962, *9*, 512-521.

[BATES86] Bates, M.J. Subject access in online catalogs: A design model. *JASIS*, 1986, *37 (6)*, 357-376.

[BORKO63] Borko, H and Bernick, M.D. Automatic document classification. *Journal of the ACM*, April 1963, *10(3)*, 151-162.

[CARROLL70] Carroll, J.D. and Chang, J.J. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 1970, *35*, 283-319.

[CARROLL80] Carroll, J.D. and Arabie, P. Multidimensional scaling. In M.R. Rosenzweig & L.W. Porter (Eds.). *Annual Review of Psychology*, 1980, *31*, 607-649.

[COOMBS64] Coombs, C.H. *A Theory of Data*. New York: Wiley, 1964.

[DESARBO85] Desarbo, W.S., and Carroll, J.D. Three-way metric unfolding via alternating weighted least squares. *Psychometrika*, 1985, *50(3)*, 275-300.

[DUMAIS89] Dumais, S.T., Deerwester, S., Furnas, G.W., Landauer T.K., and Harshman, R., Indexing by Latent Structure Analysis. *Journal of the American Society for Information Science*. 1989, in press.

[FIDEL85] 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, Las Vegas, 69-72.

[FORSYTHE77] 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.

[FURNAS80] Furnas, G.W. Objects and their features: The metric representation of two-class data. Ph.D. Dissertation. Stanford University, 1980.

[FURNAS83] Furnas, G.W., Landauer, T.K., Dumais, S.T., and Gomez, L.M. Statistical semantics: Analysis of the potential performance of key-word information systems. *Bell System Technical Journal*, 1983, *62(6)*, 1753-1806.

[HARSHMAN70] Harshman, R.A. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Work Papers Phonetics*, 1970, *16*, 86pp.

[HARSHMAN84] Harshman, R.A. and Lundy, M.E. Data preprocessing and the extended PARAFAC model. In H.G. Law, C.W. Snyder, Jr., J.A. Hattie, and R.P. McDonald (Eds.). *Research Methods for Multimode Data Analysis*, Praeger, 1984b.

[HARSHMAN84] Harshman, R.A. and Lundy, M.E. The PARAFAC model for three-way factor analysis and multi-dimensional scaling. In H.G. Law, C.W. Snyder, Jr., J.A. Hattie, and R.P. McDonald (Eds.). *Research Methods for Multimode Data Analysis*, Praeger, 1984a.

[HEISER81] Heiser, W.J. *Unfolding Analysis of Proximity Data.* Leiden, The Netherlands: Reprodienst Psychologie RUL, 1981.

[JARDIN71] Jardin, N. and van Rijsbergen, C.J. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 1971, *7*, 217-240.

[KOLL79] Koll, M. An approach to concept-based information retrieval. *ACM SIGIR Forum, XIII32-50*, 1979.

[KRUSKAL78] Kruskal, J.B. Factor analysis and principal components: Bilinear methods. In H. Kruskal, J.M. Tanur (Eds.). *International Encyclopedia of Statistics*, New York: Free Press, 1978.

[LILEY54] Liley, O. Evaluation of the subject catalog. *American Documentation*, 1954, *5(2)*, 41-60.

[OSSORIO66] Ossorio, P.G. Classification space: A multivariate procedure for automatic document indexing and retrieval. *Multivariate Behavioral Research*, October 1966, 479-524.

[SALTON68] Salton, G. *Automatic Information Organization and Retrieval.* McGraw Hill, 1968.

[SALTON83] Salton, G. and McGill, M.J. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

[SPARCK71] Sparck Jones, K. *Automatic Keyword Classification for Information Retrieval,* Buttersworth, London, 1971.

[SPARCK72] Sparck Jones, K. A statistical interpretation of term specificity and its applications in retrieval. *Journal of Documentation*, March 1972, *28(1)*, 11-21.

[STREETER87] Streeter, L. A, and Lochbaum, K. E. An expert/expert-locating system based on automatic representation of semantic structure. *Proceedings of the Fourth Conference on Artificial Intelligence Applications*. March 14-18, 1987, San Deigo, CA., pp. 345-350.

[TARR74] Tarr, D. and Borko, H. Factors influencing inter-indexer consistency. In *Proceedings of the ASIS 37th Annual Meeting, Vol. 11*, 1974, 50-55.