USING LATENT SEMANTIC ANALYSIS TO IMPROVE ACCESS TO TEXTUAL INFORMATION

Susan T. Dumais George W. Furnas Thomas K. Landauer
Bell Communications Research

Scott Deerwester *Richard Harshman*
University of Chicago University of Western Ontario

ABSTRACT

This paper describes a new approach for dealing with the vocabulary problem in human-computer interaction. Most approaches to retrieving textual materials depend on a lexical match between words in users' requests and those in or assigned to database objects. Because of the tremendous diversity in the words people use to describe the same object, lexical matching methods are necessarily incomplete and imprecise [5]. The latent semantic indexing approach tries to overcome these problems by automatically organizing text objects into a semantic structure more appropriate for matching user requests. This is done by taking advantage of implicit higher-order structure in the association of terms with text objects. The particular technique used is singular-value decomposition, in which a large term by text-object matrix is decomposed into a set of about 50 to 150 orthogonal factors from which the original matrix can be approximated by linear combination. Terms and objects are represented by 50 to 150 dimensional vectors and matched against user queries in this "semantic" space. Initial tests find this completely automatic method widely applicable and a promising way to improve users' access to many kinds of textual materials, or to objects and services for which textual descriptions are available.

INTRODUCTION

Many large and interesting collections of textual materials are now available electronically - e.g., books, newswires, documentation, bulletin boards, mail, etc. While many of the technological barriers to information access and display have been removed, the psychological problem of being able to find what you want remains. Methods for organizing and accessing textual information range from electronic analogs of familiar paper-based techniques, like tables of contents, hierarchies, or indices [11], [16] to richer associative connections that are feasible only with computers, like full-content addressability [12], hypertext [3], or information lenses [10]. While these tools may provide some retrieval advantages over existing paper and pencil technology, many benefits of electronic storage and retrieval are unrealized.

The method we describe provides an automatic way to organize textual materials into a semantic structure we believe will be useful for many information retrieval and navigation tasks. We use the term "text object" to refer to any unit of text that users might want to retrieve. In some contexts, the appropriate text units could be as small as a few words, sentences or paragraphs, while in others multi-paragraph units or even entire documents may be relevant. The text object could also constitute a pointer to some other textual or non-textual object.

DEFICIENCIES OF KEYWORD RETRIEVAL METHODS

A standard approach to finding relevant textual material depends on matching individual words in users' requests with individual words in database texts. Text objects that contain one or more words in common with those in the users' query are returned as relevant. Keyword or content-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 [1], [12].

We believe that a principled attack on these problems depends on understanding human verbal behavior and its implications for human-computer interaction. In previous work [5], [6], we showed that there is tremendous diversity in the words people use to describe the same object or concept (*synonymy*), and that this places strict, and low, limits on the expected performance of keyword systems. If a requester uses different words from the author or organizer of the information, relevant materials will be missed. Conversely, the same word can have more than one meaning (*polysemy*), which leads to irrelevant materials being retrieved.

Because human word use is characterized by extensive synonymy and polysemy, straightforward term-matching schemes are seriously deficient. The basic problem is that people often want to access information based on meaning, but the individual words they use to do so do not uniquely and sufficiently express meaning. Previous attempts to overcome the diversity in human word usage in information retrieval have included: restricting the allowable vocabulary and training intermediaries to generate search keys from this vocabulary; augmenting user's original query terms with related terms (e.g. from a special thesaurus); or constructing explicit models that reflect the semantics of the domain. Not only are these methods expert-labor intensive, but they are often not very successful [12].

LATENT SEMANTIC INDEXING (LSI)

The "latent semantic indexing" (LSI) approach we propose tries to overcome the problems of word-based access by treating the observed word to text-object association data as an unreliable estimate of the true, larger pool of words that could have been associated with each object. We assume there is some underlying "latent" semantic structure in word usage data 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". A description of terms, objects and user queries based on the underlying latent semantic structure (rather than surface level word choice) is used for representing and retrieving information.

The particular latent semantic indexing analysis that we have tried uses singular-value decomposition, a technique closely related to eigenvector decomposition and factor analysis [7]. We take a large matrix, X, 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, for example the $t\times o$ matrix of terms and objects, X, 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 which 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 this matrix, by containing only the first k independent linear components of X, captures the major associational structure in the matrix and throws out the 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 words (t), minor differences in terminology are ignored. In this reduced space, the closeness of objects is determined by the overall pattern of term usage, so objects can be classified together 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. As a result, terms that did not actually appear in an object may still end up close to it, if that is consistent with the major patterns of association in the data. Position in the space then serves as the new kind of "semantic indexing" [4].

The result can be represented geometrically with location of the terms and objects in k-space given by the vectors from the T and O matrices, respectively. In this space the cosine or dot product between vectors corresponds to their estimated similarity. Retrieval proceeds by using the terms in a query to identify a point in the space, and text objects in its neighborhood are returned to the user. Since the similarity between objects in the space is graded, objects can be ranked by their similarity to the query and users can view as many as needed. Importantly, nearby objects need not share any terms with the query since location in the space is determined by overall patterns of term usage across objects.

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 [13], [15]. Factor analysis has also been explored previously for automatic indexing and retrieval [2]. 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 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. Koll [9] has discussed many of the same ideas we describe above regarding concept-based information retrieval. His system differs from ours in that it lacks the formal mathematical underpinnings provided by the singular value decomposition approach, and it has only been tested on very small datasets.

EXAMPLE OF LSI

The simple example presented in Figure 1 helps to illustrate some of the advantages of LSI over keyword matching. The sample database consists of titles from nine Bellcore technical memoranda (Figure 1A). For ease of exposition, the sample database contains only titles, although text objects are typically more complicated and interesting. Note that this sample dataset consists of two classes of titles: five about human computer interaction (c1-c5) and four about graph theory (m1-m4). Words occurring in more than one title were selected for indexing and are italicized. Figure 1B shows the term by title matrix for this dataset where each cell entry indicates the frequency with which each term appears in each title.

Such a matrix can be used directly for keyword-based retrievals or as the initial input for the LSI analysis (i.e. as the X matrix on which SVD is performed).

We now consider how keyword and LSI systems would respond to a query or request like the following: *"I want memos about human computer interaction"*. In standard keyword systems, retrieval is based on term overlap between the query and titles (or text objects, in the more general case). The words *human* and *computer* occur in both the query and titles, so a system that simply matched terms would return titles c1, c2 and c4 in response to the query. But titles c3 and c5, which are also relevant, would be missed because they do not share any terms with the query. This is the synonymy problem - some authors write about human computer interaction, others about user interfaces, and so on.

The latent semantic indexing (LSI) method can improve this performance. Singular-value decomposition (SVD) is used to approximate the original term and title matrix by means of a smaller number of orthogonal dimensions. For this example, we carefully chose terms by titles to get a good approximation using just two dimensions. Figure 2 shows the two-dimensional representation of terms and document titles; their coordinates are simply the first two columns in the corresponding T and O matrices. Terms are shown as filled circles and numbered from 1 to 12. Document titles are represented by open squares with the terms contained in them indicated parenthetically by the appropriate numbers - e.g., title c1 contains the terms human, interface and computer. Similarity in this space is measured by cosines (or dot products). Note that in this representation titles c1 and c5 are near each other even though they share no terms. This is because the derived "meaning" of terms and objects (their location in the space) depends on the total pattern of term usage.

To respond to a query, it must first be represented as a "pseudo-object" in the space. This is done by taking a weighted vector average of the terms in the query - here labeled q. We calculate the cosine between this query vector (q) and each of the title vectors (c1-c5, m1-m4). The region within the dashed lines contains all titles whose cosine with the query, q, is .9 or greater. In this example, the system returns all relevant titles and no irrelevant ones. Titles c3 and c5, which are missed by ordinary keyword techniques, are very similar to the query in this representation.

TESTS AND APPLICATIONS OF THE LSI METHOD

The LSI method has also been tried on more realistic cases with promising results. First, we examined performance in two standard information science document collections for which user queries and relevance judgments were available. For each of a set of queries posed by real users, attempts have been made to identify all relevant documents in the collection. These datasets provide a useful testbed for the systematic evaluation of different indexing and retrieval methods. We also applied the LSI method to two local information retrieval problems - one helps users find technical memoranda relevant to either specific queries or to a more general interest profile, and the other finds domain experts within a large research organization.

Information Science Datasets

Performance of information retrieval systems is typically summarized in terms of two parameters - precision and recall. *Recall* is the proportion of documents relevant to a query that are retrieved by the system (ranging from 0 to 1); and *precision* is the proportion of documents in the set returned to the user that are relevant (again, ranging from 0 to 1). An ideal retrieval system would have both high recall and high precision - that is, it would return all and only relevant documents.

The first database consisted of 1033 medical reference abstracts and titles. 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 LSI and .45 for term matching. This difference was largest at high levels of recall. The 13% average improvement over raw term matching shows that LSI captured some structure in the data which was missed by raw term matching.

The second standard dataset consisted of 1460 information science abstracts that have been consistently difficult for automatic retrieval methods. Automatic indexing found 5135 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 dataset, LSI offered no improvement over term matching methods; precision for both methods was below .30, even for the lowest levels of recall. We suspect that poor performance in this dataset is due to low reliability of relevance judgments, poorly stated queries, and the relative homogeneity of the documents.

Bellcore Technical Memoranda

One local test involved a set of Bellcore technical memoranda like those illustrated in Figure 1. For this test, a text object consisted of title, abstract, author names, and additional author-provided keywords. Approximately 2000 memos were automatically characterized by the 3424 terms which occurred in more than two memoranda, and then evaluated using a 100-factor SVD solution. We selected documents written by people we knew well and looked for the most similar documents in the derived factor space. We were encouraged by the occurrence of many cases like the example in which "nearby" documents included some documents which shared no keywords with the query document, and by the lack of dramatic failures. We are now in the process of collecting more systematic relevance judgments for this dataset.

This application also suggests that LSI might be especially useful in filtering or selection services. People can be represented in the derived factor space as "pseudo-documents". For example, they can be placed at the centroid of the documents they have written or previously found useful. As new documents are added into the space, they can be compared to the points representing people, and suggested to "nearby" people as potentially of interest. Since people can be characterized by their positions in a latent semantic space, many of the vocabulary problems encountered in the use of keyword profiles or filters are overcome.

Bellcore Advisor

Another test was conducted using an on-line service which provides information about Bellcore experts on topics of potential interest to other experts or clients [14]. The objects of interest in this case are people and not document abstracts as in our previous examples. For this database, 480 groups of people were characterized by descriptions of projects and work groups written for administrative purposes. These 480 "research groups" were automatically described using 2662 terms, and a 50-dimensional factor solution was obtained in which both research groups and terms were represented. In response to requests for technical help or advice in some domain, the system returns pointers to "nearby" research groups. A test of the utility of the resulting space was to compare how well people's descriptions of their own research interests predicted what work group they actually belonged to using distances derived from either the 50-dimensional factor space or from raw term overlap methods. By this measure, the LSI method was found to be markedly superior - the average rank of the appropriate work group was 5 for LSI compared to 30 for raw term matching.

CONCLUSIONS

The latent structure approach is useful for helping people find textual information in large collections. It helps overcome vocabulary problems that pose severe limits on human-computer interaction [5],[6],[8] by automatically extracting underlying semantic factors. In terms of helping users find objects of interest, the LSI approach compares quite favorably with existing keyword-based methods, and suggests some new possibilities for customization and selection. The latent structure method is widely applicable and involves almost no human intervention. The term by object matrix can be automatically constructed for any set of text objects, and the underlying concepts are identified by a completely automatic statistical process. The matrix decomposition step is currently computer intensive, but it need only be done once for each dataset. The resulting semantic space provides a compact representation of the original dataset with much of its redundancy and noise squeezed out.

There are many outstanding research issues. We think we can greatly improve performance by encorporating short phrases, differential term-weighting, and allowing Boolean combinations of terms. In the latent structure domain, we can explore different similarity measures, use relevance feedback techniques to better place the query, and examine other methods of uncovering latent structure, including highly parallel "learning machines".

REFERENCES

1. 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*, 289-299.
2. Borko, H. and Bernick, M.D. Automatic document classification. *Journal of the ACM*, April 1963, *10(3)*, 151-162.
3. Conklin, J. Hypertext: An introduction and survey. *Computer*, Sept 1987, 17-41.
4. 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*, in press.
5. Furnas, G.W., Landauer, T.K., Gomez, L.M., and Dumais, S.T. Statistical semantics: Analysis of the potential performance of key-word information systems. *Bell System Technical Journal*, 1983, *62(6)*, 1753-1806.
6. 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.
7. 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.
8. Good, M.D., Whiteside, J.A. Wixon, D.R., and Jones, S.J. Building a user-derived interface. *Communications of the ACM*, 1984, *27(10)*, 1032-1043.
9. Koll, M. An approach to concept-based information retrieval. *ACM SIGIR Forum, XIII*, 1979, 32-50.
10. Malone, T.W., Grant, K.R., Turbak, F.A., Brobst, S.A., and Cohen, M.D. Intelligent information sharing systems. *Communications of the ACM*, 1987, *30(5)*, 390-402.
11. Orwick, P., Jaynes, J.T., Barstow, T.R., and Bohn, L.S. DOMAIN/DELPHI: Retrieving documents online. In *Proceeding of CHI'86*, 1986, 114-121.
12. Salton, G. and McGill, M.J. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.
13. Sparck Jones, K. *Automatic keyword classification for information retrieval.* Buttersworth, 1971.
14. Streeter, L.A. and Lochbaum, K.E. An expert expert-locating system based on automatic representation of semantic structure. In *Proceedings of IEEE Conference on AI Applications.* San Diego, CA, March 1988.
15. Voorhees, E. The cluster hypothesis revisited. *SIGIR*, 1985, 188-196.
16. Weyer, S. The design of a dynamic book for information search. *International Journal of Man*

Machine Studies, 1982, *17*, 87-107.

Technical Memo Example

(A) Database of Titles

c1: *Human* machine *interface* for *computer* applications
c2: *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: *User*-perceived *response time* and error measurement

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

(B) Term by Title Matrix

	Titles								
	c1	c2	c3	c4	c5	m1	m2	m3	m4
Terms	--	--	--	--	--	--	--	--	--
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

Figure 1. (A) A sample dataset consisting of the titles of nine technical memoranda. Terms occurring in more than one title are italicized. There are two classes of objects - five titles about human-computer interaction (c1-c5) and four about graphs (m1-m4). (B) This dataset can be described by means of a term by title matrix where each cell entry indicates the frequency with which a term occurs in a title. This matrix was used as the data, X, on which SVD was performed.

Figure 2. A two-dimensional plot of 12 terms and 9 titles from the SVD analysis of the example in Figure 1. Terms are represented by filled circles; titles are shown as open squares with component term numbers indicated parenthetically. The dotted cone represents the region whose points are within a cosine of .9 from the query, q.