SECTION II

Technology

CHAPTER 4

Latent Semantic Analysis

Susan T. Dumais
Microsoft Research, Redmond, Washington

Introduction

Latent Semantic Analysis (LSA) was first introduced in Dumais, Furnas, Landauer, and Deerwester (1988) and Deerwester, Dumais, Furnas, Landauer, and Harshman (1990) as a technique for improving information retrieval. The key insight in LSA was to reduce the dimensionality of the information retrieval problem. Most approaches to retrieving information depend on a lexical match between words in the user's query and those in documents. Indeed, this lexical matching is the way that the popular Web and enterprise search engines work. Such systems are, however, far from ideal. We are all aware of the tremendous amount of irrelevant information that is retrieved when searching. We also fail to find much of the existing relevant material. LSA was designed to address these retrieval problems, using dimension reduction techniques.

Fundamental characteristics of human word usage underlie these retrieval failures. People use a wide variety of words to describe the same object or concept (*synonymy*). Furnas, Landauer, Gomez, and Dumais (1987) showed that people generate the same keyword to describe well-known objects only 20 percent of the time. Poor agreement was also observed in studies of inter-indexer consistency (e.g., Chan, 1989; Tarr & Borko, 1974) in the generation of search terms (e.g., Fidel, 1985; Bates, 1986), and in the generation of hypertext links (Furner, Ellis, & Willett, 1999). Because searchers and authors often use different words, relevant materials are missed. Someone looking for documents on "human-computer interaction" will not find articles that use only the phrase "man-machine studies" or "human factors." People also use the same word to refer to different things (*polysemy*). Words like "saturn," "jaguar," or "chip" have several different meanings. A short query like "saturn" will thus return many irrelevant documents. The query "Saturn car" will return fewer irrelevant items, but it will miss some documents that use only the terms "Saturn automobile." In searching, there is a constant tension between being overly specific and missing relevant information, and being more general and returning irrelevant information.

A number of approaches have been developed in information retrieval to address the problems caused by the variability in word usage. *Stemming* is a popular technique used to normalize some kinds of surface-level variability by converting words to their morphological root. For example, the words "retrieve," "retrieval," "retrieved," and "retrieving" would all be converted to their root form, "retrieve." The root form is used for both document and query processing. Stemming sometimes helps retrieval, although not much (Harman, 1991; Hull, 1996). And, it does not address cases where related words are not morphologically related (e.g., physician and doctor). *Controlled vocabularies* have also been used to limit variability by requiring that query and index terms belong to a pre-defined set of terms. Documents are indexed by a specified or authorized list of subject headings or index terms, called the controlled vocabulary. *Library of Congress Subject Headings, Medical Subject Headings*, Association for Computing Machinery (ACM) keywords, and Yellow Pages headings are examples of controlled vocabularies. If searchers can find the right controlled vocabulary terms, they do not have to think of all the morphologically related or synonymous terms that authors might have used. However, assigning controlled vocabulary terms in a consistent and thorough manner is a time-consuming and usually manual process. A good deal of research has been published about the effectiveness of controlled vocabulary indexing compared to full text indexing (e.g., Bates, 1998; Lancaster, 1986; Svenonius, 1986).

The combination of both full text and controlled vocabularies is often better than either alone, although the size of the advantage is variable (Lancaster, 1986; Markey, Atherton, & Newton, 1982; Srinivasan, 1996). Richer *thesauri* have also been used to provide synonyms, generalizations, and specializations of users' search terms (see Srinivasan, 1992, for a review). Controlled vocabularies and thesaurus entries can be generated either manually or by the automatic analysis of large collections of texts.

With the advent of large-scale collections of full text, statistical approaches are being used more and more to analyze the relationships among terms and documents. LSA takes this approach. LSA induces knowledge about the meanings of documents and words by analyzing large collections of texts. The approach simultaneously models the relationships among documents based on their constituent words, and the relationships between words based on their occurrence in documents. By using fewer dimensions for representation than there are unique words, LSA induces similarities among terms that are useful in solving the information retrieval problems described earlier.

LSA is a fully automatic statistical approach to extracting relations among words by means of their contexts of use in documents, passages, or sentences. It makes no use of natural language processing techniques for analyzing morphological, syntactic, or semantic relations. Nor does it use humanly constructed resources like dictionaries, thesauri, lexical reference systems (e.g., WordNet), semantic networks, or other knowledge representations. Its only input is large amounts of texts.

LSA is an unsupervised learning technique. It starts with a large collection of texts, builds a term-document matrix, and tries to uncover some similarity structures that are useful for information retrieval and related text-analysis problems. Several recent *ARIST* chapters have focused on text mining and discovery (Benoît, 2002; Solomon, 2002; Trybula, 2000). These chapters provide complementary coverage of the field of text analysis.

LSA Overview

Mathematical details of the LSA approach to information retrieval are presented in Deerwester et al. (1990) and Berry, Dumais, and

O'Brien (1995). Here we highlight the main steps and briefly outline the matrix algebra underlying LSA.

The LSA analysis consists of four main steps. The first two steps are also used in vector space models. Step 3, dimension reduction, is the key difference in LSA.

1. *Term-Document Matrix.* A large collection of text is represented as a term-document matrix. Rows are individual words and columns are documents or smaller units such as passages or sentence, as appropriate for each application. Individual cell entries contain the frequency with which a term occurs in a document. Note that the order of words in the document is unimportant in this matrix representation; thus the name "bag of words" representation is often used.

2. *Transformed Term-Document Matrix.* Instead of working with raw term frequencies, the entries in the term-document matrix are often transformed. The best performance is observed when frequencies are cumulated in a sublinear fashion (typically $log(freq_{ij} + 1)$), and inversely with the overall occurrence of the term in the collection (typically an inverse document frequency or entropy-based score).

3. *Dimension Reduction.* A reduced-rank singular value decom position (SVD) is performed on the matrix, in which the k largest singular values are retained, and the remainder set to 0. The resulting reduced-dimension SVD representation is the best k-dimensional approximation to the original matrix, in the least-squares sense. Each document and term is now represented as a k-dimensional vector in the space derived by the SVD. The SVD technique is closely related to eigen analysis, factor analysis, principal components analysis, and linear neural networks.

4. *Retrieval in Reduced Space.* Similarities are computed among entities in the reduced-dimensional space, rather than in the original term-document matrix. Because both documents and terms are represented as vectors in the same space, document-document, term-term, and term-document similarities are all straightforward to compute. In addition, terms and/or documents

> can be combined to create new vectors in space, which can be compared in the same way. For example, to find documents similar to a query, a new query vector is formed at the *centroid* (i.e., weighted average) of its constituent term vectors and then compared to documents vectors to find the most similar documents. This process by which new vectors are added to the LSA space is called *folding-in*. The cosine or angular distance between vectors is used as the measure of their similarity for many information retrieval applications because it has been shown to be effective in practice.

We present only a brief mathematical overview of LSA here. Additional details about the SVD can be found in Gollub and van Loan (1989), and details of the application of the SVD to information retrieval in Deerwester et al. (1990) and Berry et al. (1995). Information retrieval problems begin with a rectangular $t\ x\ d$ matrix of terms and documents, X. Any rectangular matrix can be decomposed into the product of three other matrices using the singular value decomposition (Gollub & van Loan, 1989). Thus,

$$X = T{*}S{*}\, D^T \quad \text{(1) SVD of a matrix X,}$$

where T is a $t\ x\ r$ matrix with orthonormal columns, D is a $d\ x\ r$ matrix with orthonormal columns, and S is an $r\ x\ r$ diagonal matrix with the entries sorted in decreasing order. The entries of the S matrix are the singular values, and the T and D matrices are the left and right singular vectors, corresponding to term and document vectors for information retrieval problems. This is simply a re-representation of the X matrix using orthogonal indexing dimensions. LSA uses a truncated SVD, keeping only the k largest singular values and their associated vectors, so

$$X \approx T_k{*}S_k{*}\, D_k{}^T \text{ (2) reduced-dimension SVD, as used in LSA.}$$

This is the best least squares approximation to X with k parameters, and is what LSA uses for its semantic space. The rows in T_k are the term vectors in LSA space and the rows in D_k are the document vectors in LSA space. Document-document, term-term, and term-document similarities are computed in the reduced dimensional approximation to X.

A geometric analogy helps highlight the differences between traditional vector retrieval systems and the reduced-dimension LSA approach. The vector retrieval model (Salton & McGill, 1983) has a natural geometric interpretation as shown in the left panel of Figure 4.1. Terms form the dimensions or axes of the space. Documents are represented as vectors in this term space, with the entries in the term-document matrix determining the length and direction of the vectors. Note that, in this representation, terms are orthogonal because they form the axes of the space. An important consequence of this is that if a document does not contain a term, it has similarity 0 with a query consisting of just that term. If you ask a query about *cars*, you will not retrieve any documents containing *automobile* (and not car). In Figure 4.1, for example, Doc 3 cannot be retrieved by Term 1.

LSA can also be thought of geometrically, as shown in the right panel of Figure 4.1. The axes are those derived from the SVD; they are linear combinations of terms. Both terms and documents are represented as vectors in this *k*-dimensional LSA space. In this representation, the derived indexing dimensions are orthogonal, but terms are not. The location of term vectors reflects the correlations in their usage across documents. An important consequence is that terms are no longer independent; therefore, a query can match documents, even though the documents do not contain the query terms. For example, Doc 3 can now be retrieved by Term 1 (which does not occur in Doc 3).

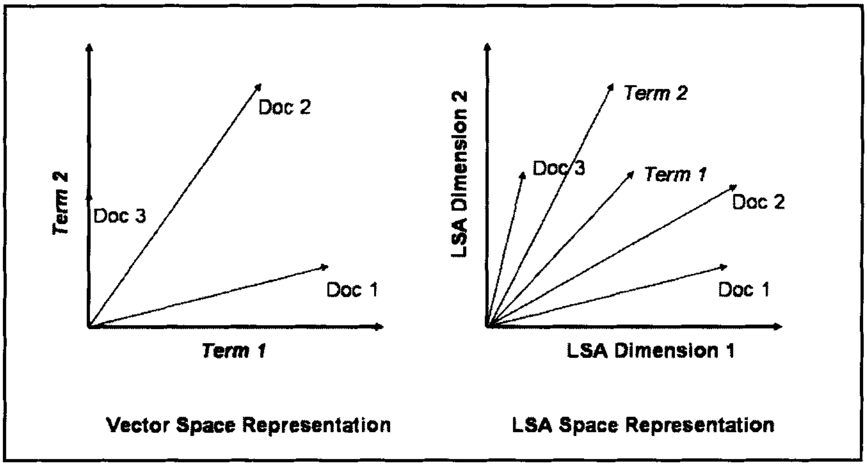

Figure 4.1 Comparison of Vector Space (left) and LSA (right) representations

Deerwester et al. (1990), Berry et al. (1995), and Berry, Drmac, and Jessup (1999) describe the computational aspects of LSA in more detail, including computational complexity, updating, and efficient sparse matrix techniques for computing the SVD.

Several online resources are available for LSA. These resources provide links to papers, demonstrations, and software. The Telcordia (formerly Bellcore) LSI page, http://lsi.research.telcordia.com, provides demonstrations, papers, and software. The University of Colorado LSA page, http://lsa.colorado.edu, provides several demonstrations, including essay assessment and tools for term and sentence analyses. The University of Tennessee LSI site, http://www.cs.utk.edu/~lsi, contains papers, test corpora, and software for text analysis and efficient SVD algorithms.

Applications of LSA

Information Retrieval

LSA was originally developed for, and has been most commonly applied to, information retrieval problems. In this chapter's discussion of information retrieval, the phrase "word matching" is used synonymously with "vector retrieval." This highlights the fact that vector retrieval depends on literal word overlap whereas LSA can retrieve documents even when they do not contain query terms. For both LSA and vector retrieval, the same step 2 matrix is used. For vector retrieval, similarity between queries and documents is computed using the full dimensional term-document matrix. For LSA retrieval, dimension reduction is performed (step 3) and similarity is computed using the reduced-dimension representation. Deerwester, Dumais, Landauer, Furnass, and Beck (1988) evaluated LSA using several information retrieval test collections for which user queries and relevance judgments were available. They compared LSA retrieval to traditional vector matching.

Performance of information retrieval systems is summarized using two measures, precision and recall. *Recall* is the proportion of relevant documents in the collection that are retrieved by the system. *Precision* is the proportion of relevant documents in the set returned to the user.

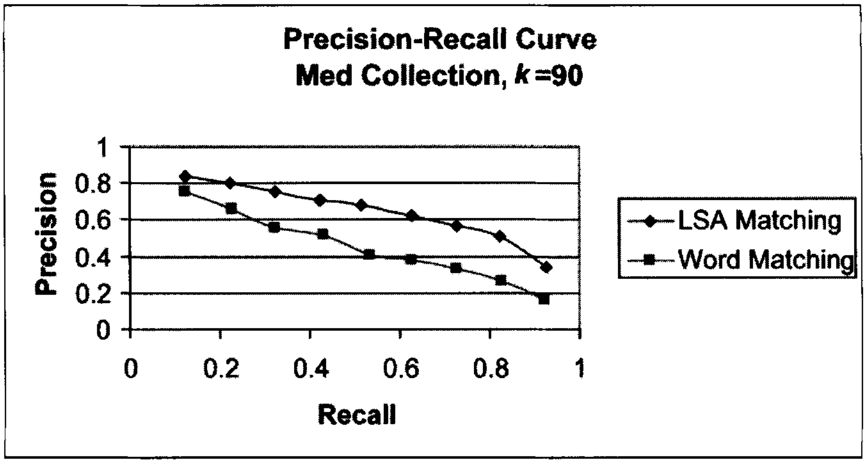

Figure 4.2 Example precision recall curve for medical collection

Precision is calculated at several levels of recall to generate a curve showing the tradeoff between precision and recall.

Figure 4.2 shows an example result for a small test collection with 1,033 medical abstracts (documents) and 5,831 terms. Precision is plotted as a function of recall, averaged over the 30 queries for this collection. As is typical in retrieval applications, precision drops as recall increases. Finding the first few relevant documents is easy, but finding the last few relevant documents requires examining many irrelevant documents. As can be seen, LSA performance is substantially better than the standard word-matching control for the entire range of recall values, with an average advantage of about 30 percent. At 50 percent recall, for example, 68 percent of the documents returned by LSA are relevant, compared with 40 percent of the documents returned by simple word matching. Performance is much like this for several other test collections (see Deerwester et al., 1990, for a review), including some of the larger Text REtrieval Conference (TREC) collections (Dumais, 1995). Sometimes, however, performance with LSA is no better than word matching (e.g., the CISI collection in Deerwester et al., 1990; the TREC collection in Husbands, Simon, & Ding, 2000). The reasons for the inconsistent performance of LSA are not clear and require further research. The diversity and size of the collection and the number of singular values that are extracted have been mentioned as possible issues. Husbands et al.

(2000), for example, found advantages for LSA with the Med collection but not for the much larger and more diverse TREC-6 collection. They developed a technique to normalize the length of the reduced-dimension term vectors, and found improved performance for the TREC-6 collection and all others they tested. Lochbaum and Streeter (1989) compared LSA with word matching and looked at techniques for combining the two approaches, which seems like a promising but not well explored technique.

The LSA approach also involves the parameter *k*, the number of dimensions used in the reduced space. In Figure 4.2, 90 dimensions were used in the LSA analysis. For the vector analysis 5,831 dimensions (one for each term) were used. Figure 4.3 shows LSA performance as a function of number of dimensions for the medical collection described earlier. The measure of performance shown in this figure is average precision; that is, the precision averaged over the nine levels of recall shown in Figure 4.2. For $k = 90$, the average precision is 0.71. Similar values are computed for other values of *k*. Word-matching performance, which is constant across dimensions, is also shown for comparison.

With too few dimensions, LSA performance is poor, and with too many dimensions, performance is the same as word matching. In between these two is a substantial range over which LSA performance is better than word matching performance. For the medical collection, performance peaks at about 90 dimensions. This pattern of initial poor LSA performance with very few dimensions, an increase in performance over a substantial range, and then a decrease to word matching level is observed for other collections as well (see Landauer & Dumais, 1997, Figure 4.3). Choosing the right dimensionality is required for successful application of the LSA approach to information retrieval. Choosing the appropriate value of *k* can be difficult when relevance judgments are not available ahead of time, but this is the subject of active research described in more detail in the section on computational issues with LSA. However, for a fairly large range of values of *k*, LSA performance is substantially better than the standard word-matching approach.

Several techniques have been used to improve the precision and recall of information retrieval systems. One of the most important and robust techniques involves *term weighting*, the transformations in step 2 (e.g., Sparck Jones, 1972). LSA performance can also be improved by using transformations of the term-document matrix such as the popular *tf*idf*

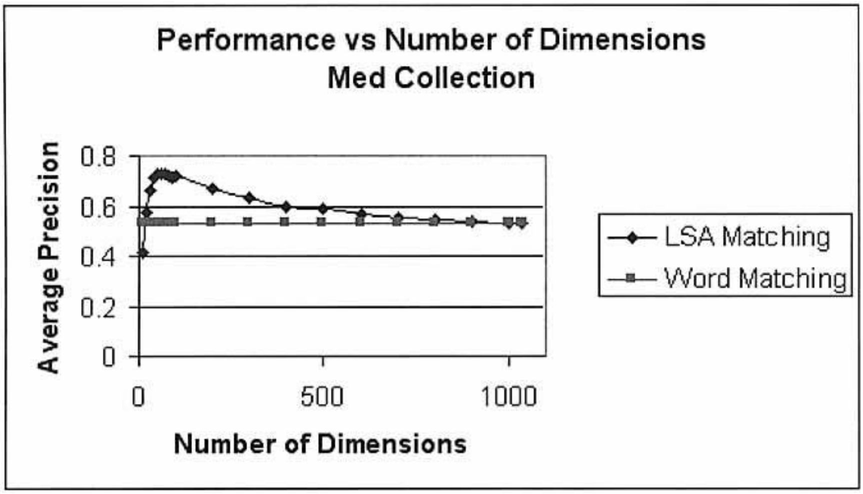

Figure 4.3 Performance as a function of number of dimensions

approaches. Dumais (1991) reports that the best performance is observed when frequencies are cumulated in a sublinear fashion (*log(freq*$_{ij}$ + 1)), and inversely with the overall occurrence of the term in the collection (inverse document frequency or entropy scores). Another approach to improving information retrieval uses *relevance feedback*, which involves iterative retrieval based on user evaluation of items retrieved (e.g., Salton & Buckley, 1990). Relevance feedback can also be used to improve LSA performance (Dumais, 1991).

The success of LSA in information retrieval applications is attributable to the dimension-reduction step. By adding the constraint that the observed term-document relationships must be modeled by many fewer parameters than there are unique words, LSA requires that relationships among words be represented. This reduced space is what is referred to as the "semantic" space, because relationships among words (and documents) are captured. One important consequence of this in the context of information retrieval is that a query can be very similar to a document even though the two do not share any words. In an encyclopedia collection to be described in the section on vocabulary tests, for example, the words "physician" and "doctor" never co-occur in a single article, but they are quite similar in the reduced LSA space. This is because they occur in many of the same contexts (with words like patient, hospital, sick, recovery, surgery, nurse, etc.), and when dimension constraints are imposed, the vectors for doctor and physician are near each other in the reduced LSA space. This inferred similarity among words can also be thought of as a kind of query expansion (Xu & Croft, 1996). Not only does a query word match documents that contain it, but it matches documents that contain similar words as well. Query expansion is typically done on the fly, but with LSA there is no need to explicitly augment a query; that process happened implicitly during the dimension-reduction step.

LSA has also been used for a variety of information filtering and analysis tasks. In addition, LSA has recently been used to model aspects of human memory that depend on the kinds of semantic relations captured by the dimension-reduction approach. We now describe these applications in more detail.

Information Filtering

In information retrieval, the collection is relatively stable, and new queries are issued constantly. In information filtering (also known as routing or selective dissemination of information), the queries are fixed and new documents are added to the collection constantly. The task is to match new documents against these standing queries or profiles of interest, which reflect persistent information needs. The user profile is specified in words describing the user's interest and/or known relevant documents. The nature of the profile and the number of known relevant documents can vary depending on the application. In routing, many relevant documents are known ahead of time and the task is to rank a set of new documents (e.g., a daily or weekly alerting service). In filtering, at most a few relevant documents are known and the task is to mark new documents as relevant or not relevant as they come along (e.g., a real-time alerting service). For filtering, a binary decision must be made about every document as it arrives. Robertson and Soboroff (2001) provide a more detailed description of filtering tasks and performance measures.

Applying LSA to information filtering is straightforward. Any LSA space can be used as a starting point. Typically, a user profile is a vector located at the centroid of words and/or documents in the description of a user's interests. The profile vector can be compared to any term or document. As new documents arrive they are added into the LSA space. New documents are located at the centroid of their constituent terms. If a new document vector is similar enough to the user profile vector, it is returned to the user. The user profile can be adjusted if relevance judgments about the returned documents become available during the search.

Foltz and Dumais (1992) conducted an early evaluation of LSA for use in filtering. They compared several methods for predicting which technical memoranda people would like to receive. They varied the matching algorithm (LSA vs. vector) and the method by which the profile was created (free-form interest statement vs. relevant documents). Their "LSA match-document profile" approach, which combined LSA with some knowledge of previously relevant technical memoranda, was the most successful technique for all performance measures examined.

Dumais (1995) evaluated the LSA approach to filtering (called routing in TREC) on the larger standard TREC-3 collection. For this evaluation, fifty profiles were compared to a stream of 336,000 new documents. The LSA space was created by analyzing 38,000 training documents related to one or more of the topics. User profiles were represented using a free-form interest statement (called topics in TREC) or known relevant documents. Dumais also found that creating a user profile using known relevant documents (the 'lsir2' run) was more effective than using the topic description. Precision over the first ten documents was 0.62 for the topic profile and 0.69 for the document profile, and overall 10 percent more relevant documents were retrieved. Dumais also explored combinations of the topic and document profiles by taking linear combinations of the two vectors, and observed small advantages in precision. Compared with other systems that completed the TREC routing task, the LSA relevant topic profile did quite well. LSA was better than the median on forty-one of the fifty routing topics and the best system for nine of them.

Hull (1994), Schütze, Hull, and Pedersen (1995), and Hull, Pedersen, and Schütze (1996) also looked at LSA for information filtering. They found a small but consistent advantage for LSA compared to no dimension reduction. Schütze et al. (1995) compared different techniques for representing documents (LSA, important terms, LSA and important terms) and for learning the profiles (centroid, logistic regression, neural network, linear discriminant). The TREC-2 and TREC-3 routing topics were evaluated in these experiments. Schütze et al. used a local LSA analysis in which a separate LSA space was computed for every topic using the 2,000 best matching documents for the topic description. This analysis is an interesting variant of the approach described earlier where all topics were represented in the same global LSA space. The best average precision scores were obtained when the LSA representation was combined with a discriminative classification approach (discriminative analysis or neural nets). Discriminative approaches use information about both positive and negative instances to learn a topic model. Non-discriminative approaches, like the centroid method, use just the relevant items. In many experiments advanced discriminative methods from machine learning are more accurate at classifying new test instances.

Zelikovitz and Hirsh (2001) use another technique from machine learning to improve information filtering using an LSA representation. They begin with the documents that are relevant to each topic, but augment this training data with many additional documents that they call background documents. Although these additional documents do not have explicit labels vis-à-vis the filtering task, they do contain many words and contexts, which should help in establishing a useful LSA space. It is generally easy to obtain many documents but harder to obtain relevance judgments. They compared LSA analyses with and without additional background documents on four test collections (technical papers, Web page titles, WebKB, and twenty newsgroups). They found consistently lower error rates when the background knowledge was used, and the advantages were larger when there was less labeled training data.

A slight twist on the text filtering problem was explored by Dumais and Nielsen (1992) in their work on the automatic assignment of reviewers to papers. The system was tested by evaluating several methods for assigning reviewers for a hypertext conference. They first built LSA spaces using several different collections of materials from the hypertext domain (abstracts submitted to the conference, three hypertext text books, ACM hypertext compendium and a human-computer interaction bibliography). They then represented each reviewer as a vector in the LSA space, located at the centroid of the abstracts of papers he or she had written. Conference submissions were also added to the LSA space in the same manner. The reviewers nearest each submitted paper were suggested for that paper. LSA assignments of papers to reviewers were compared to the reviewers' assessments of their interest in each paper and assignments by three human experts. The best LSA space was that based on all the sources combined. The relevance of the automatically assigned papers was as high as those assigned by one human expert and somewhat worse than those provided by the two other human experts. Performance was improved even beyond the level of human experts when reviewers were allowed to select from a larger set of abstracts suggested by LSA.

Cross-Language Retrieval

LSA was designed to overcome the vocabulary mismatch problem between searchers and document creators. An extreme example of mismatch occurs when queries and documents are in different languages, the so-called cross-language retrieval problem. In cross-language retrieval, queries in one language are used to retrieve documents in other languages as well as the original language. Cross-language LSA (CL-LSA) has been applied to this problem with good results. The technique of LSA applies directly to this problem by using a slightly different notion of the term-document matrix (Landauer & Littman, 1990).

In many cross-language applications, parallel corpora are available (the same documents are available in two or more languages) and can be used to train a multilingual semantic space. For ease of exposition, we talk about French and English documents, but the approach works for any pair of languages and, indeed, for more than two languages. When a parallel corpus is available, a dual-language document is created by concatenating the French and English versions of the document, to form a dual-language document. For any dual-language document, some of the rows in the term by dual-language documents matrix are French words and others are English words. The dual-language documents form the contexts that LSA exploits to learn the relations among French and English terms. The SVD analysis is computed on the term by dual-language documents matrix. The resulting LSA space contains both French and English terms, with those sharing many contexts being near each other. The dual-language LSA space is a kind of learned interlingua. The space also contains the dual-language documents used for training, but these documents are not of interest for retrieval. Instead, they are replaced by monolingual French and English documents, which are folded-in to the LSA space at the centroid of their constituent words. The LSA space now consists of French and English documents and words. Queries in English (French) can retrieve the most similar documents regardless of the language in which they are written. Unlike many approaches to cross-language retrieval, CL-LSA does not require any dictionaries, lexical resources, or translation of either documents or terms. The relationships among words are inferred using the parallel corpus. The dual-language LSA space reflects these relationships and is used for cross-language retrieval. Sheridan and Ballerini's (1996) similarity thesaurus approach to cross-language retrieval is related to CL-LSA.

Early work with CL-LSA used a mate-retrieval task for evaluation (Dumais, Littman, & Landauer, 1998; Landauer & Littman, 1990). In the mate-retrieval task, an English (French) document is used as a query, and compared with each French (English) document. Landauer and Littman's work used parallel documents from the Hansard collection of Canadian Parliamentary texts. They worked with 2,482 paragraphs containing at least five sentences. They used randomly selected 900 dual-language documents to build the dual-language LSA space. The remaining 1,582 documents were used for testing. These documents were first folded-in to the LSA space. For the retrieval test, each French (English) document was issued as a query and the closest English (French) documents returned. For CL-LSA approach, a document in one language returned its mate in the other language as the most similar document 98.4 percent of the time. When the same test was performed using standard word matching without dimension reduction, the mate was returned first only 48.6 percent of the time. Dumais et al. (1998) explored extensions of the CL-LSA approach to situations in which machine translation was used to generate a parallel corpus and to situations in which short queries rather than full documents were used as queries.

The queries in these experiments are longer than the ad hoc queries that users generate, but the results are still a strong indication that the LSA technique captures cross-language relations among terms. More traditional retrieval experiments using short queries and explicit relevance judgments have been conducted (Carbonell, Yang, Frederking, Brown, Geng, & Lee, 1997; Rehder, Littman, Dumais, & Landauer, 1997). Carbonell et al. (1997) compared LSA to the generalized vector space model (GVSM). In GVSM, documents form the axes of the retrieval space, and terms are located based on their usage in documents. Performance was evaluated using thirty queries they developed for the United Nations Multilingual Corpus. Average precision was somewhat better with GVSM than LSA (0.39 vs. 0.38). Littman and Jiang (1998) replicated these experiments using the same collection but correcting an error in the LSA implementation. They found that LSA outperforms GVSM over a number of different values of *k* (e.g., 0.45 vs. 0.36 at 200 dimensions).

Evans, Handerson, Monarch, Pereiro, Delon, and Hersh (1998) explored the use of LSA and CL-LSA to associate terms with specialized indexing concepts. Experiments associating English terms with Spanish medical concepts appear to be promising, but no comparisons to monolingual approaches are reported. Vinokourov, Shawe-Taylor, & Cristianini (2002) recently applied a variant of LSA that uses canonical correlation rather than SVD for cross-language retrieval (cross-language kernel canonical correlation analysis, or CL-KCCA). Using the Hansard corpus, they find that KCCA outperforms LSA (e.g., 0.99 versus 0.95 for mate retrieval using $k = 400$). This technique appears promising, but is more computationally expensive than LSA.

The CL-LSA method has been applied to many languages, including English-French (Dumais et al., 1998; Landauer & Littman, 1990), English-Spanish (Carbonell et al., 1997; Evans et al., 1998; Oard & Dorr, 1998), English-Greek (Berry & Young, 1995), Portuguese-English (Orengo & Huyck, 2002) and English-Japanese (Jiang & Littman, 2001; Landauer, Littman, & Stornetta, 1992; Mori, Kokubu, & Tanaka, 2001). It has also been applied to language triples such as English-French-Spanish and English-French-German where three-way document-aligned corpora were available (Littman, Jiang, & Keim, 1998; Rehder et al., 1997). Littman et al. (1998) developed an important extension to allow the same ideas to be applied when fully aligned corpora are not available, but pairwise alignments are. Their extension allows for French-Spanish retrieval, even when only partially aligned corpora (French-English and English-Spanish) are available for training. In this example, English forms a kind of bridge language.

Other IR-Related LSA Applications

LSA has been used for a wide range of other IR-related applications, and we briefly mention the major ones. Schütze and Silverstein (1997) used LSA for document clusteiing. Document clustering seeks to discover relationships among documents. This method requires that the similarity between all document-document pairs be computed, which can be highly inefficient when each document is represented by thousands of features. Schütze and Silverstein looked at two methods for reducing the dimensionality of the problem—one used LSA and the other used a word selection algorithm based on global term frequencies. Accurate performance could be achieved using only a small number of dimensions (20 to 100). With this amount of dimension reduction, using LSA with projection provided results that were two orders of magnitude more efficient than initial computations. Retrieval performance for forty-nine queries was also explored. LSA with $k = 20$ achieved the highest average precision and average rank.

Gordon and Dumais (1998) used LSA for the kind of literature-based discovery that Swanson pioneered (Swanson, 1989; Swanson, Smalheiser, & Bookstein, 2001) in his research on treatment for Raynaud's Disease. They applied LSA to 560 documents published during the years 1983–1985 containing the term *Raynaud's*. The nearest words to the term *Raynaud's* in the LSA space were identified. These words were compared to the top 40 terms and phrases obtained from several statistical techniques proposed by Gordon and Lindsay (1996). A high percentage of terms LSA found as similar to *Raynaud's* had been identified by Gordon and Lindsay's methods (e.g., nine of the top ten terms; fifteen of the top twenty). A rank correlation of the top forty phrases by both methods showed that the position on one list predicts the position on the other ($r = 0.57$). LSA closely reproduces the set of terms that Gordon and Lindsay (1996) showed were a useful starting point for literature-based discovery.

Because LSA does not depend on the literal matching of query words to document words, it is useful in applications where the query or document words are noisy, as occurs with optical character recognition, handwriting recognition, or speech input. When document scanning errors occur, for example, the word *Dumais* can be misrecognized as *Duniais*. If the variants of a word occur in the same contexts (e.g., with words such as information retrieval, LSA, human-computer interaction, *ARIST*), then they will wind up near each other in the reduced dimension LSA space and queries about Dumais can retrieve documents containing only *Duniais*. Nielsen, Phillips, & Dumais (1992) used LSA to index a small collection of abstracts input by a commercially available pen machine in its standard recognizer mode. Even though word-error rates were almost 9 percent, information retrieval using the LSA representation was not disrupted (compared to matching on the uncorrupted texts). Kurimo (2000) and Wolf and Raj (2002) used an SVD-based representation for spoken documents to overcome the noisy input that happens when queries are spoken rather than typed.

Soboroff, Nicholas, Kukla, and Ebert (1997) used LSA's low dimensional representation to help visualize authorship and writing-style patterns. Instead of using terms and documents, they use n-grams and documents. The LSA representation does a good job of grouping documents by authors. Others have also used similarities in an LSA space to visualize citation relationships (Chen, 1999), knowledge domains (Chen & Paul, 2001), and search results (Börner, 2000; Miller, 1997).

All of the applications just described start with a term-document matrix (step 1). The dimension reduction ideas from LSA can be applied to more general problems. We briefly mention two other examples that are closely related to information access, but do not use the standard term-document matrix: link analysis and collaborative filtering.

Link Analysis. PageRank (Brin & Page, 1998) is a technique to compute the importance of items in large graphs based on the structure of the graph. PageRank has been applied most notably to compute the importance of pages on the Web. The analysis starts with a large *N* by *N* connectivity matrix, where *N* is the number of Web pages. For LSA, the matrix consists of terms and documents; whereas for link analysis, the matrix contains documents (pages) on both dimensions. A cell entry *i,j* is non-zero if a link exists from page *i* to page *j*, and 0 otherwise. PageRank assigns to a page a score proportional to the number of times a random surfer would visit that page, if the surfer surfed indefinitely from page to page, following all outlinks from a page with equal probability. More formally, the PageRank of a page *i* is equal to the PageRank of all the inlinks to the page divided by the number of outlinks from the page:

$$PageRank(p) = (1 - d) + d \times \sum_{\substack{\text{all } q \text{ linking}\\ \text{to } p}} \left(\frac{PageRank(q)}{c(q)}\right)$$

where *d is a damping factor between* 0 *and* 1,
c(q) is the number of out-going links in a page q.

PageRank can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix. That is, PageRank (and the HITS algorithm) use only the first eigenvector; LSA uses more. The PageRank idea is closely related to Garfield's work on the impact factor of journals determined by citation patterns. However, Garfield (1972) considered only the average number of citations a paper received in a fixed time period in determining the impact factor of a journal, which amounts to considering only inlink information. PageRank extends this idea by giving different weights to different inlinks based on their PageRank, and by normalizing the number of links on a page.

Kleinberg's (1998) work on hyperlink-induced topic search (HITS) is similar to the work on PageRank. However, instead of propagating importance directly from one page to another, he uses the intermediate notion of hub and authority pages. Authority pages are those pointed to by many others. Hub pages are those that are linked to many authorities. Thus, hubs and authorities are mutually reinforcing. An iterative algorithm is used to compute these scores until convergence. In addition, this approach is typically used on only a small portion of the Web. Instead of computing a global importance score for every page, a query is first issued, and importance scores are computed for only a small subgraph seeded with the search results. Because of the query-dependent nature of the graph, this technique is slower than PageRank, which precomputes all measures. However, the graphs involved are much smaller, so HITS scores can be calculated quickly.

Several researchers have explored techniques for combining content and link information for improved information retrieval or classification. Bharat and Henzinger (1998) used HITS techniques to rank search results. They pruned the HITS node expansion using the content-based similarity of nodes. Cohn and Hofmann (2001) used a probabilistic version of LSA and HITS (to be described in the section on relationship of LSA to other technologies) for combining context and link information.

They used a mixture model to perform a simultaneous decomposition of the matrices associated with word occurrences (content) and link (connectivity) patterns. They applied the model to two text-classification problems and also explored applications to understanding the flow between topics and intelligent Web crawling. Richardson and Domingos (2002) explored a content-guided variant of PageRank that combines content and link information.

Collaborative Filtering. In collaborative filtering applications, the preferences or opinions of others are used to predict preferences of a particular individual. For example, in predicting which movies I would like, the movie ratings of people who are similar to me are used. Thus, in collaborative filtering, the matrices of interest are people by objects, rather than terms by documents. Dimension-reduction techniques can be applied to the collaborative filtering problem as they have been to the information retrieval problem. Hofmann and Puzicha (1999) conducted experiments with the EachMovie dataset containing almost three million ratings for movies. A variant of LSA (called the aspect model) was the best technique for predicting movie preferences of individuals in their experiments. Azar, Fiat, Karlin, McSherry, and Saia (2000) also describe an LSA approach to collaborative filtering.

Modeling Human Memory

More than fifty years ago Vannevar Bush (1945) speculated about "memex," a machine that would be an extension of the personal memory belonging to an individual, and would work in a fashion analogous to the human brain, that is, by association. More recently, Anderson has called attention to the analogy between information retrieval and memory processes (Anderson, 1989; Anderson & Schooler, 1991). Although LSA was initially developed to improve information retrieval, analyses of memory and psycholinguistic phenomena show that LSA captures a great deal of the similarity of meanings evidenced in these behavioral tasks. We review only a sampling of the applications here, focusing on two (essay grading, vocabulary tests) and briefly mentioning several others. Landauer, Foltz, and Laham (1998), Landauer (2002), and Dumais (2003) provide more comprehensive overviews of LSA and its applications to human memory and discourse processing.

Essay Grading. Landauer, Laham, Rehder, and Schreiner (1997), and Foltz, Laham, and Landauer (1999) described how an LSA-based system could be used to score the quality of free-form essays. Because essays are difficult and time consuming to score, they are not widely used in educational assessment. Earlier attempts to develop computational techniques to aid in the scoring of essays focused primarily on measures of writing style such as grammar, spelling, and punctuation (e.g., Page, 1994). The LSA approach, in contrast, focuses on measuring the conceptual content and knowledge conveyed in essays.

To assess the quality of essays, LSA is first trained on a sample of domain-representative text. The standard reduced-dimension LSA semantic space is automatically derived from these tests. Next, essays with known quality scores are folded-in to the space. Ungraded essays are then compared to the essays that have been graded. Several techniques for assigning a grade to a new essay are based on the grades of similar essays. For example, an essay could be assigned the score of the closest gold-standard ideal essay written by an expert, or it could be assigned an average of the *k* most similar essays weighted by their similarity (see Landauer et al., 1998, for details). The approach has been applied to essays on a wide range of topics including heart anatomy, physiology, social studies, physics, and law, as well as general opinion and argument essays.

In one study reported by Foltz et al. (1999), essays from the Educational Testing Service (ETS) Graduate Management Achievement Test (GMAT) were graded. Performance of the fully automated LSA approach was compared to the performance of two trained ETS graders. The correlation between the grades assigned by two trained ETS graders was between 0.86 and 0.87 for different essays. LSA grades were automatically assigned as described earlier. The correlation between these grades with ETS scorers was also 0.86. Thus, LSA is able to perform with nearly the same reliability as trained ETS graders. Larkey (1998) used a related statistical text-analysis technique along with stylistic measures to automatically score essays, with similarly impressive results. These automatic techniques work quite well in assigning appropriate grades and agree with human graders to the same extent that the humans agree with each other. A striking aspect of these results is that the LSA representation is based on analyses that do not take into account any syntactic or word order information. Human graders certainly have access to syntactic information, yet it does not help them in assigning consistent scores to the essays.

Vocabulary Tests. Landauer and Dumais (1996, 1997) first explored the ability of the LSA dimension-reduction representation to simulate aspects of human knowledge and meaning relations. They used the ETS Test of English as a Foreign Language (TOEFL), a multiple test choice of synonymy. The test consists of eighty multiple choice items, including a target word or short phrase and four alternatives for each target. Students select the alternative closest in meaning to the target, e.g., Target: *constantly;* Alternatives: *accidentally, continually, instantly, rapidly*. Students from non-English speaking countries take this test for admission to many U.S. colleges. Summary data provided by ETS show that these students correctly answer 64 percent of the eighty questions.

For LSA performance, the LSA space was derived by analyzing approximately five million words of text from the high-school level encyclopedia, *Grolier's Academic American Encyclopedia*. All analyses were done automatically as described here—a term-article matrix was built, cell entries were transformed, a reduced-dimension SVD was computed, and the resulting *k*-dimensional vectors were used for matching. To take the TOEFL test, the similarity between the target word and each of the four alternatives is computed. The answer with the highest similarity was returned as LSA's synonym guess. In the above example, the similarity between the target and four stems are: *continually* 0.28, *rapidly* 0.22, *instantly* 0.08, and *accidentally* 0.07, so *continually* was selected as the synonym by LSA. LSA's performance on this task was 64 percent, exactly the same as the students who took the test. In addition, for incorrect items, the correlation between the relative frequency of student responses and the LSA cosine is 0.44, indicating similar error patterns.

Landauer and Dumais (1997) also examined the rate at which LSA acquired knowledge, and the influence of direct versus indirect exposure to words. They built several different LSA spaces using different subsets of the encyclopedia content as training, and looked at accuracy on the TOEFL test for these different representations. The model related the number of exposures to a word and the total number of words seen to test performance. LSA learning parameters were compared to the acquisition rates observed in children (middle school children acquire the meanings of new words at an average of ten to fifteen words per day). They concluded that LSA could acquire new knowledge at a rate consistent with what is observed in children. Their model also showed that indirect exposures were as important as direct exposures for learning.

Turney (2001b) reported good TOEFL performance (74 percent) using a variant of a word-matching technique he calls PMI-IR. His algorithm uses pointwise mutual information (PMI) applied to the results of a Web search (IR). For the synonym test, the PMI-IR score for each alternative reflects the extent to which it is statistically independent of the target:

$$\text{Score (alternative}_i) = \log (p(\text{target} \\ \text{AND} \\ \text{alternative}_i)/p(\text{target})p(\text{alternative}_i))$$

The counts were obtained from a large search engine, AltaVista. The scoring function was further modified to take into account the proximity of the words, negation, and context words for sense ambiguation. The final scoring function results in a TOEFL score of 74 percent. The simple co-occurrence score was 62 percent—slightly worse than the 64 percent reported by Landauer and Dumais, but well above their word-matching score of 16 percent. Several differences in the experiments could account for the improvements. The most important difference is the amount of text used for the analysis. Landauer and Dumais (1997) used 30,473 encyclopedia articles, representing five million words of text. Turney (2001b) used a much larger collection, roughly 500 million Web pages, which is more than four orders of magnitude larger. Additional experiments looking at PMI-IR on smaller collections, or LSA on larger collections, are required to better understand the nature of the differences. From a practical perspective, it is not surprising that using the vast resources of the Web can improve information access. From the more theoretical perspective of modeling aspects of human memory, the tremendous amounts of data available on the Web are not characteristic of the amount of text processed by humans.

Semantic priming. When people are asked to decide whether a letter string is a word, they do so faster if they have just read a sentence that is related to the word but does not contain the word (Till, Mross, & Kintsch, 1988). Landauer and Dumais (1997) showed that an LSA representation can model this semantic priming effect. Lund and Burgess (1996) modeled other priming data using a high-dimensional semantic model, HAL (hyperspace analog of language), that is related to LSA. The correlation between semantic distance (measured by distances in HAL space) and human decision times was significant ($r = 0.35$) and of the same magnitude as the correlation between human similarity estimates and the priming effect ($r = 0.31$).

Textual coherence. Kintsch and his colleagues developed methods for representing texts in a propositional language (e.g., van Dijk & Kintsch, 1983). They showed that the comprehension of text depends strongly on its coherence, as measured by the overlap between the arguments in the propositions. The propositional analysis is typically carried out by hand. Foltz, Kintsch, and Landauer (1998) used LSA to measure textual coherence automatically. In one experiment they found that LSA coherence scores correlated highly with human test scores ($r = 0.95$), but did not correlate with simple word overlap scores ($r = 0.05$). Lemaire, Bianco, Sylvestre, and Noveck (2001) also used LSA to model text comprehension. Dunn, Almeida, Waterreus, Barclay, and Flicker (2002) used LSA to score prose recall. They compared LSA against two common scoring methods, which use correctly recalled story and thematic units. LSA scores were highly correlated with existing scoring techniques. And, LSA was able to detect recall deficits in patients with cognitive impairments.

Similarity neighborhoods. Griffiths and Steyvers (2002) proposed a probabilistic variant of LSA that they used to model the relationships among words. In human memory, most words are related to a number of different topics (as shown, for example, in *Roget's Thesaurus*). The number of different topics in which a word occurs is described by a power law—many words are associated with only one topic and some words are associated with many. Griffiths and Steyvers used dimension-reduction techniques to automatically infer topics (like LSA's dimensions) from word usage data. The resulting model revealed the same kind of power relationship observed in the distribution of words across topics as seen in thesauri.

Word sense disambiguation. Schütze (1998) used an approach based on second-order co-occurrences to induce similarity among words. A word space is derived by analyzing second-order co-occurrences. Word contexts are represented in the same space. Clustering of the context vectors is used to identify word senses. Accuracies of up to 94 percent are reported for naturally and artificially ambiguous words. Gallant (1991) also used a dimension-reduction technique for word sense disambiguation.

Tutoring. Graesser and colleagues (e.g., Graesser, Wiemer-Hastings, Wiemer-Hastings, Person, & the Tutoring Research Group, 2000) used LSA as a component in an intelligent tutoring system. LSA is capable of discriminating different classes of ability and of tracking the quality of student contributions in a tutorial dialog. LSA's evaluations overall are comparable to those provided by intermediate experts in computer science, but not as high as more accomplished experts.

Analogical reasoning. Ramscar and Yarlett (2003) describe how LSA can be used to model the retrieval of analogies from long-term memory. They distinguish between two main processes in analogy—retrieval and mapping. In their model, LSA is used for retrieval and a separate process is used for mapping. Although LSA as currently formulated is not sensitive to the structural characteristics required for mapping, its global knowledge is a good model of analogical reminding that is useful in retrieval.

The examples described here show that LSA has been used successfully to model aspects of human memory and discourse processing. The success of LSA in these tasks is remarkable because several sources of information that are available to humans are ignored by the statistical models such as word order, syntactic relationship, morphology, and correspondence to physical objects. French and Labiouse (2002) recently described three examples where a technique related to LSA (Turney's PMI-IR, Turney, 2001b) failed to replicate human behavior. One task asked humans or the PMI-IR system to rate lawyers as horses, fishes, birds, slimeballs, etc. For these tasks, PMI-IR produced similarities that did not correspond well to human ratings. French and Labiouse argued that systems lacking cultural and perceptual associations will not be able to answer such questions. Turney (2001a), however, provides evidence that richer queries run against vast collections of text, such as the Web, can be used to handle such questions by conditioning the counts on a context. The extent to which techniques like LSA that mine implicit relationship from large amounts of text can be used to model aspects of human cognition is an ongoing topic of research.

Relationship of LSA to Other Techniques

LSA was designed to overcome the variability in vocabulary used by authors and searchers. The matrix of observed term-document relations is used to estimate a model of similarity having fewer parameters than the original matrix.

In the information retrieval literature, the idea of improving retrieval by discovering latent proximity structure predates work on LSA. Hierarchical classification analyses were used for term and document clustering (Jardin & van Rijsbergen, 1971; Salton, 1968; Sparck Jones, 1971). Latent class analysis (Baker, 1962) and factor analysis (Borko & Bernick, 1963; Ossorio, 1966) were explored for automatic document indexing and retrieval. These earlier approaches typically focused on representing either terms or documents, but not both in the same space. One exception to this was a proposal by Koll to represent both terms and documents in the same concept space (Koll, 1979; see also Salton, Buckley, & Yu, 1982, and Wong, Ziarko, Raghavan, & Wong, 1987). Although Koll's approach is similar in spirit to LSA, the concept space he used was of very low dimensionality, and the dimensions were hand chosen and not truly orthogonal as they are with the SVD. In addition, all of these early attempts were limited by lack of computer-processing power and availability of large collections of text in machine-readable form. These problems are now largely solved, so progress in the field has been rapid.

Probabilistic models have been successfully used for information retrieval and filtering applications (e.g., Bookstein & Swanson, 1974; Robertson, 1977; van Rijsbergen, 1979).[2] In the last few years, several research groups have explored probabilistic alternatives to the algebraic approach (the SVD) used in LSA. Probabilistic approaches have several advantages theoretically (e.g., they allow for a natural combination with other attributes, and for formal analysis of error bounds), although some computational challenges still remain. Probabilistic models begin by assuming that a small number of topics (sometimes called aspects or factors) are used to generate the observed term-document matrix. Topics play much the same role as dimensions do in LSA, with the main difference being the objective function that is optimized in the two cases. With LSA/SVD, the sum of squared errors is minimized, whereas an alternative function is chosen for probabilistic models. Much of the recent work tries to better understand why LSA works using a variety of alternative formalisms and extensions.

Papadimitriou, Raghavan, Tamaki, and Vempala (1995) proposed a simple generative probabilistic model of how a term-document collection is generated. The goal was to show that dimension-reduction approaches could capture the structure, given certain statistical properties of the corpus. Topics were represented as probability distributions over terms; and documents were represented as a combination of a small number of topics. The corpus was generated by repeatedly drawing sample documents. Papadimitriou et al. made some additional simplifying assumptions, namely that terms almost always occur in the same topic and that documents are about a single topic. They showed that, for documents generated according to this model, the k-dimensional representation produced by LSA/SVD results in sharply defined groups of documents generated by each topic with high probability. The similarity of documents generated by the same topic was much higher than those generated by different topics. However, it is not clear to what extent real text collections fit the two simplifying assumptions used in the generative model (i.e., documents were assumed to be about a single topic with high probability, and terms were almost always associated with a single topic). Azar et al. (2001) extended these ideas by relaxing several assumptions. They showed that when a matrix is a slightly perturbed block matrix, the SVD does a good job of approximation. They further allow an additional error matrix of independent random values, so their results are more widely applicable. However, neither group has looked at the extent to which actual text collections are well described by these generative assumptions, or the extent to which the SVD representation of text collections conforms to the predicted error bounds.

Hofmann (1999) developed a different probabilistic model, which he calls probabilistic LSA (or PLSA), in the context of information retrieval applications. Documents are represented as a multinomial probability distribution over topics (which are assumed but not directly observed). The generative model for a term-document pair is the following: select a document with probability $P(d)$, select a latent class or topic with probability $P(z \mid d)$, and generate a term with probability $P(t \mid z)$. Expectation maximization, a standard machine-learning technique for maximum likelihood estimation in latent variable models, is used to estimate the model parameters. In experiments with four small text-retrieval collections, PLSA provided advantages compared to both LSA and standard vector models. Griffiths and Steyvers (2002) have recently proposed a variant of Hofmann's model that assumes the mixture proportions are distributed as a latent Dirichelet random variable. Their approach has been used to model some interesting aspects of human memory, as noted earlier. Ding (1999) and Girolami (2000) also explored probabilistic variants of LSA.

A number of researchers have proposed alternative approaches and generalizations of LSA. A close similarity is observed between linear neural networks and LSA, as described in Gallant (1991) and Caid, Dumais, and Gallant (1995). Bartell, Cottrell, and Belew (1992) describe the similarities between LSA and multi-dimensional scaling (MDS). MDS allows generalization of the analysis beyond term-document relationships to other sources of document–document similarity information. The many other possible sources of such information include bibliometric relationships or relevance feedback. Story (1996) looked at LSA from the viewpoint of a Bayesian regression model. Ando and Lee (2001) described a generalization of LSA using a subspace-based framework. The basic idea is to repeatedly rescale the vectors to amplify the presence of documents that are poorly represented in earlier iterations. This process results in 8 to 10 percent improvements over LSA in retrieval and clustering applications. Isbell and Viola (1998) described an analysis in which sets of highly related words form the basis of the representation. Documents and queries are represented by their distance to these sets. This technique is efficient to compute and related theoretically to the independent components of documents. De Freitas and Barnard (2001) used a Bayesian mixture model, which allows the encoding of prior knowledge as well as improved regularization techniques. They applied the model to multimedia documents consisting of text and images. Kurimo (2000) looked at random mappings and self-organizing maps as alternatives to LSA's singular value decomposition and applied the technique to the indexing of audio documents. Christianini, Shawe-Taylor, and Lodhi (2001) describe an approach that combines aspects of LSA and support vector machines, a popular discriminative learning technique, for a text classification problem. Instead of taking the usual dot product as a measure of similarity between two documents, they develop a latent semantic kernel that incorporates term co-occurrence information in the similarity measure.

These techniques extend LSA by examining new modeling formalisms, but they all share the focus on dimension reduction.

Finally, several researchers have explored simplifications of LSA that depend on co-occurrence data, often without any dimension reduction. Turney (2001b) developed a technique that combined ideas from pointwise mutual information and information retrieval, as described earlier. This PMI-IR approach, combined with a complex query formulation involving NEAR and NOT operators, scores somewhat better than LSA on the TOEFL vocabulary test. However, it is not clear to what extent the improved performance is based on the underlying analytic technique (PMI-IR) or the vast amounts of content available on the Web. Schütze (1992, 1998) used an approach based on second-order word co-occurrences to induce similarity among words. Instead of forming a representation based on direct or first-order co-occurrences, he used second-order co-occurrences (based on words with which the co-occurring terms occur). Second-order co-occurrence information is less sparse and more robust than first-order co-occurrences. He further combined some ideas from LSA dimension reduction to the resulting word spaces.

Computational Issues with LSA

The term-document matrices that represent information retrieval corpora are large and sparse. Because only the *k* largest singular values are used in the LSA representation, sparse iterative techniques are used to compute the SVD efficiently. Several packages implement Lanczos, subspace iteration, and trace minimization approaches to the solution of such problems. The time for the SVD computation depends on the number of non-zero entries in the term-document matrix and on the number of dimensions retained. With current computer speed and memory, computing the SVDs of problems containing hundreds of thousands of documents is possible. Handling millions of documents is not possible without doing something to reduce the size of the problem, such as sampling. We mention four computational issues encountered when applying LSA to large problems: initial SVD decomposition, updating with new terms and documents, estimating the appropriate value of *k*, and query processing.

Initial decomposition. Some significant work has been conducted in improving the speed with which the SVD can be computed as well as in minimizing memory requirements. In general, the approaches try to sparsify the term-document matrix by sampling. In addition, some approaches involve quantized cell entries so that the arithmetic operations can be performed more quickly.

Papadimitriou et al. (1995) proposed a technique of "random projections" to speed up the SVD computations. By randomly projecting the term-document matrix onto a lower dimensional subspace, the SVD computations can be speeded up while at the same time preserving accuracy within provable bounds. Frieze, Kannan, and Vempala (1998) showed that by taking a weighted sample of the documents with a probability proportional to the length of the document, they could analyze a matrix that depended on *k* rather than the number of terms or documents. Aclioptas and McSherry (2001) also sample matrix entries to achieve more efficient decomposition. They further reduce computation costs by discretizing the matrix entries to 1/0, and using non-uniform sampling to increase the scarcity when the magnitudes of the entries vary significantly. Investigation of image analysis problems (like those described in Turk & Pentland, 1991), showed that they could keep only 7 percent of the data without introducing noticeable error in the SVD. The extent of reduction possible in information retrieval problems has not yet been investigated empirically (although some studies do present theoretical bounds), but work along these lines is interesting and important. Jiang, Kannan, Littman, and Vempala (1999) developed a weighted-sampling technique that is based on the vector length of documents. The term-document matrices generated by this process are somewhat less sparse than those generated by uniform sampling (0.16 percent, as compared with 0.13 percent for the TREC collection), but show lower approximation error and somewhat better retrieval performance.

Jiang and Littman (2000) proposed a technique they call approximate dimension equalization (ADE). They began by noting that the standard vector space model scales nicely, but does not take into account term dependencies. LSA and the generalized vector space model (Wong et al., 1987) take into account term relations, but do not scale as well as the vector model. Jiang and Littman used the typical distribution of singular values for the term-document matrix to approximate the weight that should be assigned to each dimension without having to compute the actual SVD. Jiang and Littman (2000) show that ADE is more efficient than LSA and produces roughly comparable precision and recall in both mono-lingual and cross-lingual tests. Karypis and Han (2000) describe a concept-indexing technique. They first used a fast clustering technique to find the axes of the reduced dimensional space, which is then used for indexing. They report that this technique is roughly an order of magnitude faster than LSA for several retrieval problems and about as accurate.

Updating. So far we have described the SVD analysis of a fixed term-document matrix. Few collections are static, so what happens when new documents and/or terms are added? The most obvious approach is to re-compute the SVD of the new matrix. This approach is often too costly, especially for large collections with rapidly changing content (e.g., the Web). When the amount of new content is small compared to the amount in the original matrix, the final LSA space will not change much. So, although re-computing the SVD is the right thing to do theoretically, it will have little practical impact. Computationally less expensive alternatives include *folding-in* (Berry et al., 1995; Deerwester et al., 1990) and *SVD-updating* (Berry et al., 1995; O'Brien, 1994; Zha & Simon, 1999).

Folding-in is very inexpensive computationally, but results in an inexact representation. New documents (terms) are located at the centroid of their constituent terms (documents). This approach in effect adds new rows (columns) to the SVD matrix, so the underlying dimensions are no longer orthogonal. It also assumes that the original LSA space is a good description of the important dimensions and will not change much as new items are added. Folding-in is how queries are represented, and this makes good sense because the queries are not part of the corpus to be analyzed. Folding-in also works well in representing new documents and terms in many practical applications (Berry et al., 1995).

SVD-updating, first described in Berry et al. (1995) and O'Brien (1994), accounts for the addition of new terms and documents and also maintains orthogonality. Updating is more expensive computationally than folding-in, but less expensive than computing the SVD anew. Similar techniques can be used to remove terms and/or documents from the LSA model, a technique called downdating (Witter & Berry, 1998).

Estimating k. Another computational issue with LSA is how to estimate the number of dimensions *k* that are required for good performance. As described earlier, retrieval performance certainly depends on *k*—with too few dimensions performance is poor, and with too many dimensions performance is again poor. Luckily, a relatively wide region of *k* usually exists where performance is above the full dimensional vector approach. Nonetheless, methods for estimating an optimal *k* would be helpful. Efron (2002) developed a technique call Amended Parallel Analysis (APA) for estimating *k*. APA is a resampling technique that analyzes the departure of the observed singular values from those expected under the hypothesis. The probabilistic approaches by Ding (1999) and Hofmann (1999) can also be used to predict an optimal region for *k*.

Query processing. A final computational issue with LSA is the computation of query-to-document similarities. For standard term-document databases, only documents containing query terms need to be examined; thus, many documents are immediately dismissed. With LSA, every query is related to every document to some extent, so all documents must be examined. Posse and Perisic (2001) developed a technique called Latent Semantic Pursuit (LSP). LSP produces latent concepts via Projection Pursuit, which has better feature extraction capabilities than SVD. LSP also reduces storage reduction, which implies significantly lower query time. Although query processing can be expensive computationally, it can easily be handled in parallel with different machines working on different subsets of documents.

Summary and Conclusions

Latent Semantic Analysis was first introduced more than a decade ago as a technique to improve information retrieval. The main idea was to reduce the dimensionality of the information retrieval problem as a means of overcoming the synonym and polysemy problems observed in standard vector space and probabilistic models. A technique from linear algebra, singular value decomposition, was used to accomplish the dimension reduction. One of the major advantages of LSA in information retrieval and filtering applications is that documents can be retrieved even when they do not match any query words. In many cases, LSA provides retrieval advantages compared with word matching techniques; at other times, performance is the same. LSA has also been applied to many problems related to information retrieval, including text classification, text clustering, and link analysis. It appears to be especially useful for problems where input is noisy (such as speech input) and standard lexical matching techniques fail. Understanding the full range of circumstances under which LSA provides retrieval benefits (e.g., size and breadth of the collections, the distribution of singular values) is an open research area.

LSA has also been used in the cognitive sciences to model aspects of human memory and cognition. LSA often offers considerable advantages over word overlap for modeling human memory. Many of the memory tasks that have been explored involve short queries and short documents, such as vocabulary tests with a single word as the target and a small number of potential synonyms; the comprehension tests involved sentence-to-sentence comparisons. In cases like this, relying on a more robust representation than individual words is advantageous.

In addition to the wide range of applications, a good deal of theoretical work has been aimed at better understanding why LSA works using a variety of alternative formalisms and extensions. Probabilistic aspect models have received the most attention and development. Computational issues have also been addressed, although they continue to be a challenge for large and rapidly changing collections.

Endnotes

1. LSI (Latent Semantic Indexing) was the terminology used in these early papers to refer to the use of dimension reduction ideas to improve the *indexing* of content for information retrieval applications. Subsequently, the same ideas have been applied to a wider range of problems, including modeling of aspects of human memory, and the broader terminology LSA (Latent Semantic Analysis) has been used to describe the approach. We use the more general terminology, LSA, in this chapter.
2. Three classes of models have been extensively explored in information retrieval. *Logical* or Boolean models were the first widely deployed retrieval models. In a Boolean retrieval system, query terms are linked by the logical operators (AND, OR, NOT) and the search engine returns documents satisfying the logical constraints in the query. *Vector space* models were introduced by Salton and his associates. In the vector model, terms form the dimensions of the indexing space. Documents and queries are represented by vectors in this space. Queries are compared with documents using a measure of similarity such as the cosine. LSA is most naturally viewed as a variant of vector

space models. *Probabilistic* models were introduced by Maron and Kuhns (1960). The basic idea is to use information about the distribution of query terms in documents to measure the similarity of queries to documents. Several of the models described in this section are variants of LSA using ideas from the probabilistic approach.

References

Aclioptas, D., & McSherry, F. (2001). Fast computation of low rank matrix approximations. *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC 2001),* 611–618.

Anderson, J. R. (1989). A rational analysis of human memory. In H. L. Roediger, III & F. I. M. Craik (Eds.) *Varieties of memory and consciousness: Essays in honor of Endel Tulving* (pp. 195–210). Hillsdale, NJ: L. Erlbaum.

Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science, 2*(6), 396–408.

Ando, R. K., & Lee, L. (2001). Iterative residual rescaling: An analysis and generalization of LSI. *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 154–162.

Azar, Y., Fiat, A., Karlin, A. R., McSherry, F., & Saia, J. (2001). Spectral analysis of data. *Proceedings of the 33rd ACM Symposium on Theory of Computing (STOC 2001),* 502–509.

Baker, F. B. (1962). Information retrieval based on latent class analysis. *Journal of the ACM, 9,* 512–521.

Bartell, B. T., Cottrell, G. W., & Belew, R. K. (1992). Latent semantic indexing is an optimal special case of multidimensional scaling. *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 161–167.

Bates, M. (1986). Subject access in online catalogs: A design model. *Journal of the American Society for Information Science, 37,* 357–376.

Bates, M. (1998). How to use controlled vocabularies more effectively in online searching. *Online, 12*(6), 45–56.

Benoît, G. (2002). Data mining. *Annual Review of Information Science and Technology, 36,* 265–310.

Berry, M. W., Drmac, Z., & Jessup, E. R. (1999). Matrices, vector spaces, and information retrieval. *SIAM Review, 41*(2), 335–362.

Berry, M. W., Dumais, S. T., & O'Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. *SIAM Review, 37*(4), 573–595.

Berry, M. W., & Young, P. G. (1995). Using latent semantic indexing for multilingual information retrieval. *Computers and the Humanities, 29*(6), 413–419.

Bharat, K., & Henzinger, M. (1998). Improved algorithms for topic distillation in a hyperlinked environment. *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 104–111.

Bookstein, A., & Swanson, D. R. (1974). Probabilistic models for automatic indexing. *Journal of the American Society for Information Science, 25,* 312–318.

Borko, H., & Bernick, M. D. (1963). Automatic document classification. *Journal of the ACM, 10,* 151–162.

Börner, K. (2000). Extracting and visualizing semantic structures in retrieval results for browsing. *Proceedings of the 2000 ACM/IEEE Joint Conference on Digital Libraries,* 234–235.

Brin, S., & Page, L. (1998). Anatomy of a large-scale hypertextual Web search engine. *Proceedings of the 7th International World Wide Web Conference.* Retrieved December 5, 2002, from http://dbpubs.stanford.edu:8090/pub/1998-8

Bush, V. (1945). As we may think. *Atlantic Monthly, 176*(1), 101–108. Retrieved December 5, 2002, from http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm

Caid, W. R., Dumais, S. T., & Gallant, S. I. (1995). Learned vector space models for information retrieval. *Information Processing & Management, 31*(3), 419–429.

Carbonell, J., Yang, Y., Frederking, R., Brown, R. D., Geng, Y., & Lee, D. (1997). Translingual information retrieval: A comparative evaluation. *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97),* 323–345.

Chan, L. M. (1989). Inter-indexer consistency in subject cataloging. *Information Technology and Libraries, 8*(4), 349–58.

Chen, C. (1999). Visualizing semantic spaces and author co-citation networks in digital libraries. *Information Processing & Management, 35,* 401–420.

Chen, C., & Paul, R. J. (2001). Visualizing a knowledge domain's intellectual structure. *Computer, 34*(3), 65–71.

Christianini, N., Shawe-Taylor, J., & Lodhi, H. (2001). Latent semantic kernels. *Proceedings of ICML-01, 18th International Conference on Machine Learning,* 66–73.

Cohn, D., & Hofmann, T. (2001). The missing link: A probabilistic model of document content and hypertext connectivity. *Advances in Neural Information Processing Systems (NIPS*13),* 430–436.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41,* 391–407.

Deerwester, S., Dumais, S., Landauer, T., Furnass, G., & Beck, L. (1988). Improving information retrieval with latent semantic indexing. *Proceedings of the 51st Annual Meeting of the American Society for Information Science, 25,* 36–40.

de Freitas, N., & Barnard, K. (2001). Bayesian latent semantic analysis of multimedia databases. *UBC TR 2001-15.*

Ding, C. H. Q. (1999). A similarity-based probability model for latent semantic indexing. *Proceedings of 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval,* 59–65.

Dumais, S. T. (2003). Data-driven approaches to information access. *Cognitive Science,* 27(3), 491–524.

Dumais, S. T. (1991). Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments and Computers, 23*(2), 229–236.

Dumais, S. T. (1995). Using LSI for information filtering: TREC-3 experiments. In: D. Harman (Ed.), *The Third Text REtrieval Conference (TREC3) National Institute of Standards and Technology Special Publication 500-225* (pp. 219–230). Gaithersburg, MD: National Institute of Standards and Technology.

Dumais, S. T., Furnas, G. W., Landauer, T. K., & Deerwester, S. (1988). Using latent semantic analysis to improve information retrieval. *Proceedings of CHI'88 Conference on Human Factors in Computing Systems*, 281–285.

Dumais, S. T., Littman, M. L., & Landauer, T. K. (1998). Automatic cross-linguistic information retrieval using latent semantic analysis. In G. Grefenstette (Ed.), *Cross-language information retrieval* (pp. 51–62). Boston: Kluwer.

Dumais, S. T., & Nielsen, J. (1992). Automating the assignment of submitted manuscripts to reviewers. *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 233–244.

Dunn, J. C., Almeida, O. P., Waterreus, A., Barclay, L., & Flicker, L. (2002). Latent semantic analysis: A new method to measure prose recall. *Journal of Clinical and Experimental Neuropsychology, 24*, 26–35.

Efron, M. (2002). *Amended parallel analysis for optimal dimensionality estimation in latent semantic indexing* (SILS Technical Report TR-2002-03). Chapel Hill, NC: University of North Carolina at Chapel Hill.

Evans, D. A., Handerson, S. K., Monarch, I. A., Pereiro, J., Delon, L., & Hersh, W. R. (1998). Mapping vocabularies using "latent semantics." In G. Grefenstette (Ed.), *Cross-language information retrieval* (pp. 63–80). Boston: Kluwer.

Fidel, R. (1985). Individual variability in online searching behavior. *Proceedings of the 48th Annual Meeting of the American Society for Information Science*, 69–72.

Foltz, P. W., & Dumais, S. T. (1992). Personalized information delivery: An analysis of information filtering methods. *Communications of the ACM, 35*(12), 51–60.

Foltz, P. W., Kintsch, W., & Landauer T. K. (1998). The measurement of textual coherence with latent semantic analysis. *Discourse Processes, 25*(2/3), 285–307.

Foltz, P. W., Laham, D., & Landauer, T. K. (1999). The intelligent essay assessor: Applications to educational technology. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning, 1*(2).

French, R. M., & Labiouse C. (2002). Four problems with extracting human semantics from large corpora. *Proceedings of the 24th Annual Conference of the Cognitive Society*, 316–320.

Frieze, A., Kannan, R., & Vempala, S. (1998). Fast monte-carlo algorithms for finding low-rank approximations. *39th Annual Symposium on Foundations of Computer Science*, 370–378.

Furnas, G. W., Landauer, T. K., Gomez, L. M., & Dumais, S. T. (1987). The vocabulary problem in human-computer interaction. *Communications of the ACM, 30*, 964–971.

Furner, J., Ellis, D., & Willett, P. (1999). Inter-linker consistency in the manual construction of hypertext documents. *ACM Computing Surveys, 31* (4es) [Online supplement], article no. 18. Retrieved January 21, 2003, from http://doi.acm.org/10.1145/345966.346008

Gallant, S. I. (1991). A practical approach for representing context and for performing word sense disambiguation using neural networks. *Neural Computation, 3*(3), 293–309.

Garfield, E. (1972). Citation analysis as a tool in journal evaluation. *Science, 178,* 471–479.

Girolami, M. (2000). Document representations based on generative multivariate Bernoulli latent topic models. *Proceedings of the 22nd Annual Colloquium on Information Retrieval Research,* 194–201.

Gollub, G. H., & van Loan, C. F. (1989). *Matrix computations* (2nd ed.). Baltimore, MD: The Johns Hopkins University Press.

Gordon, M. D., & Dumais, S. T. (1998). Using latent semantic indexing for literature based discovery. *Journal of the American Society for Information Science, 49,* 674–685.

Gordon, M. D., & Lindsay, R. K. (1996). Toward discovery support systems: A replication, re-examination, and extension of Swanson's work on literature based discovery of a connection between Raynaud's and Fish Oil. *Journal of the American Society for Information Science, 47,* 116–128.

Graesser, A., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N., & the Tutoring Research Group. (2000). Using latent semantic analysis to evaluate the contributions of students in AutoTutor. *Interactive Learning Environments,* 129–148.

Griffiths, T. L., & Steyvers, M. (2002). A probabilistic approach to semantic representation. *Proceedings of 24th Annual Cognitive Science Conference,* 381–386.

Harman, D. (1991). How effective is suffixing? *Journal of the American Society for Information Science, 42,* 7–15.

Hofmann, T. (1999). Pròbabilistic latent semantic analysis. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 50–57.

Hofmann, T., & Puzicha, J. (1999). Latent class models for collaborative filtering. *Proceedings of the International Joint Conference in Artificial Intelligence,* 688–693.

Hull, D. A. (1994). Improving text retrieval for the routing problem using latent semantic indexing. *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 282–290.

Hull, D.A. (1996). Stemming algorithms: A case study for detailed evaluation. *Journal of the American Society for Information Science, 47,* 70–84.

Hull, D. A., Pedersen, J. O., & Schütze, H. (1996). Method combination for document filtering. *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 279–288.

Husbands, P., Simon, H., & Ding, C. (2000, October). On the use of singular value decomposition for text retrieval. *Proceedings of the 1st SIAM Computational*

Information Retrieval Workshop. Retrieved January 21, 2003, from http://www.nersc.gov/research/SCG/cding//hsd4.ps

Isbell, C. L., & Viola, P. (1998). Restructuring sparse high-dimensional data for effective retrieval. *Advances in Neural Information Processing, NIPS-11*, 480–486.

Jardin, N., & van Rijsbergen, C. J. (1971). The use of hierarchical clustering in information retrieval. *Information Storage and Retrieval, 7*, 214–240.

Jiang, F., Kannan, R., Littman, M. L., & Vempala, S. (1999). *Efficient singular-value decomposition via improved document sampling* (Technical Report CS-99-5). Durham, NC: Duke University Department of Computer Science.

Jiang, F., & Littman, M. L. (2000). Approximate dimension equalization in vector-based information retrieval. *Proceedings of the Seventeenth International Conference on Machine Learning*, 423–430.

Jiang, F., & Littman, M. L. (2001). Approximate dimension reduction at NTCIR. *Proceedings of the Second NTCIR Workshop on Research in Chinese and Japanese Text Retrieval and Text Summarization*. Retrieved January 3, 2003, from http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings2/michael.pdf

Karypis, G., & Han, E.-H. (2000). Fast supervised dimensionality reduction algorithm with applications to document categorization and retrieval. *Proceedings of the 9th ACM International Conference on Information and Knowledge Management*, 12–19.

Kleinberg, J. (1998). Authoritative sources in a hyperlinked environment. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 668–677.

Koll, M. (1979). An approach to concept-based information. *ACM SIGIR Forum, 13*, 32–50.

Kurimo, M. (2000). Fast Latent Semantic Indexing of spoken documents by using self-organizing maps. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP'2000*, 2425–2428.

Lancaster, F. W. (1986). *Vocabulary control for information retrieval* (2nd ed.). Arlington, VA: Information Resources.

Landauer, T. K. (2002). Applications of latent semantic analysis. *Proceedings of 24th Annual Cognitive Science Conference*, 44.

Landauer, T. K., & Dumais, S. T. (1996). How come you know so much? From practical problem to theory. In D. Hermann, C. McEvoy, M. Johnson, & P. Hertel (Eds.), *Memory in context* (pp. 105–126). Hillsdale, N.J.: L. Erlbaum.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review, 104*, 211–240.

Landauer, T. K, Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes, 25*(2/3), 259–284.

Landauer, T. K., Laham, D., Rehder, B., & Schreiner, M. E. (1997). How well can passage meaning be derived without using word order? A comparison of latent semantic analysis and humans. *Proceedings of the 19th Annual Meeting of the Cognitive Science Society*, 412–417.

Landauer, T. K., & Littman, M. L. (1990). Fully automatic cross-language document retrieval using latent semantic indexing. *Proceedings of the Sixth*

Annual Conference of the UW Centre for the New Oxford English Dictionary and Text Research, 31–38.

Landauer, T. K., Littman, M. L., & Stornetta, W. S. (1992). *A statistical method for cross-language information retrieval.* Unpublished manuscript.

Larkey, L. S. (1998). Automatic essay grading using text categorization techniques. *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 90–95.

Lemaire, B., Bianco, M., Sylvestre, E., & Noveck, I. (2001). Un modèle de compréhension de textes fondé sur l'analyse de la sémantique latente. In H. Paugam-Moisy, V. Nyckees, J. Caron-Pargue (Eds.), *La cognition entre individu et société* (pp. 309–320). Paris: Hermès.

Littman, M. L., & Jiang, F. (1998). A comparison of two corpus-based methods for translingual information retrieval (Technical Report CS-1998-11). Durham, NC: Department of Computer Science, Duke University.

Littman, M. L., Jiang, F., & Keim, G. A. (1998). Learning a language-independent representation for terms from a partially aligned corpus. *Proceedings of the Fifteenth International Conference on Machine Learning*, 314–322.

Lochbaum, K., & Streeter, L. A. (1989). Comparing and combining the effectiveness of latent semantic indexing and the ordinary vector space model for information retrieval. *Information Processing & Management, 25*(6), 665–676.

Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence data. *Behavior Research Methods, Instruments, and Computers, 28,* 203–208.

Markey, K., Atherton, P., & Newton, C. (1982). An analysis of controlled vocabulary and free text search statements in online searches. *Online Review, 4,* 225–236.

Maron, M. E., & Kuhns, J. L. (1960). On relevance, probabilistic indexing and information retrieval. *Journal of the ACM, 7,* 216–244.

Miller, M. H. (1997). Representing search results in three dimensions with local latent semantic indexing. *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 338–339.

Mori, T., Kokubu, T., & Tanaka, T. (2001). Cross-lingual information retrieval based on LSI with multiple word spaces. *Proceedings of the Second NTCIR Workshop on Research in Chinese and Japanese Text Retrieval and Text Summarization.* Retrieved January 21, 2003, from http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings2/mori-ir.pdf

Nielsen, J., Phillips, V. L., & Dumais, S. T. (1992, August). *Retrieving imperfectly recognized handwritten notes* (Bellcore Technical Memorandum TM-ARH-021781). Piscataway, NJ: Bellcore.

Oard, D. W., & Dorr, B. J. (1998). Evaluating cross-language text filtering effectiveness. In G. Grefenstette (Ed.), *Cross-language information retrieval* (pp. 151–161). Boston: Kluwer.

O'Brien, G. (1994). Information management tools for updating an SVD-encoded indexing scheme. Unpublished master's thesis, University of Tennessee.

Orengo, V. M., & Huyck, C. (2002, September). Portuguese-English experiments using latent semantic indexing. *Proceeding of the CLEF 2002 Workshop.*

Retrieved January 9, 2003, from clef.iei.pi.cnr.it:2002/workshop2002/WN/9.pdf

Ossorio, P. G. (1966). Classification space: A multivariate procedure for automatic document indexing and retrieval. *Multivariate Behavior Research* 1, 479–524.

Page, E. B. (1994). Computer grading of student prose using modern concepts and software. *Journal of Experimental Education, 62,* 127–142.

Papadimitriou, C. H., Raghavan, P., Tamaki, H., & Vempala, S. (1995). Latent semantic indexing: A probabilistic analysis. *17th Annual Symposium on Principles of Database Systems,* 159–169.

Posse, C., & Perisic, I. (2001). Latent semantic pursuit. *Proceedings of the 1st SIAM International Conference on Data Mining, Textmine Workshop,* 31–38.

Ramscar, M., & Yarlett, D. (2003). Semantic grounding in models of analogy: An environmental approach. *Cognitive Science, 27*(1), 41–71.

Rehder, B., Littman, M. L., Dumais, S. T., & Landauer, T. K. (1997). Automatic 3-language cross-language information retrieval with latent semantic indexing. *NIST Special Publication 500–240: The Sixth Text Retrieval Conference (TREC-6),* 233–240.

Richardson, M., & Domingos, P. (2002). The intelligent surfer: Probabilistic combination of link and content information in PageRank. *Advances in Neural Information Processing Systems (NIPS*14),* 1441–1448.

Robertson, S. E. (1977). The probability ranking principle in IR. *Journal of Documentation, 22,* 294–304.

Robertson, S. E., & Soboroff, I. (2001). The TREC 2001 filtering track report. In E. Voorhees (Ed.), *NIST Special Publication 500–250: The Tenth Text REtrieval Conference* (pp. 26–37). Gaithersburg, MD: National Institute of Standards and Technology.

Salton, G. (1968). *Automatic information organization and retrieval.* New York: McGraw Hill.

Salton, G., & Buckley, C. (1990). Improving retrieval by relevance feedback. *Journal of the American Society for Information Science, 41,* 288–297.

Salton, G., Buckley, C., & Yu, C. (1982). An evaluation of term dependence models in information retrieval. *Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 151–173.

Salton, G., & McGill, M. (1983). *Introduction to modern information retrieval.* New York: McGraw Hill.

Schütze, H. (1992). Dimensions of meaning. In *Proceedings of Supercomputing '92,* 787–796.

Schütze, H. (1998). Automatic word sense disambiguation. *Computational Linguistics, 24*(1), 97–124.

Schütze, H., Hull, D. A., & Pedersen J. (1995). A comparison of classifiers and document representation for the routing problem. *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 229–237.

Schütze, H., & Silverstein, C. (1997). A comparison of projections for efficient document clustering. *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 74–81.

Sheridan, P., & Ballerini, J. P. (1996). Experiments in multilingual information retrieval using the SPIDER system. *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 58–65.

Soboroff, I. M., Nicholas, C. K., Kukla, J. M., & Ebert, D. S. (1997). Visualizing document authorship using n-grams and latent semantic analysis. *Workshop on New Paradigms in Information Visualization and Manipulation*, 1997, 43–48.

Solomon, P. (2002). Discovering information in context. *Annual Review of Information Science and Technology, 36*, 229–264.

Sparck Jones, K. (1971). *Automatic keyword classification in information retrieval.* London: Butterworths.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28,* 11–21.

Srinivasan, P. (1992). Thesaurus construction. In W. B. Frakes & R. Baeza-Yates (Eds.), *Information retrieval: Data structures and algorithms* (pp. 161–218). Englewood Cliffs, NJ: Prentice Hall.

Srinivasan, P. (1996). Optimal document-indexing vocabulary for Medline. *Information Processing & Management, 32,* 503–514.

Story, R. E. (1996). An explanation of the effectiveness of latent semantic indexing by means of a Bayesian regression model. *Information Processing & Management, 32*(3), 329–344.

Svenonius, E. (1986). Unanswered questions in the design of controlled vocabularies. *Journal of the American Society of Information Science, 37,* 331–340.

Swanson, D. R. (1989). Online search for logically-related noninteractive medical literatures: A systematic trial-and-error strategy. *Journal of the American Society for Information Science, 40*(5), 356–358.

Swanson, D. R., Smalheiser, N. R., & Bookstein, A. (2001). Information discovery from complementary literatures: Categorizing viruses as potential weapons. *Journal of the American Society for Information Science, 52*, 797–812.

Tarr, D., & Borko, H. (1974). Factors influencing inter-indexer consistency. *Proceedings of the 37th Annual Meeting of the American Society for Information Science*, 50–55.

Till, R. E., Mross, E. F., & Kintsch W. (1988). Time course of priming for associate and inference words in discourse context. *Memory and Cognition, 16*, 283–299.

Trybula, W. (2000). Text mining. *Annual Review of Information Science and Technology, 34*, 385–419.

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience, 3*(1), 71–86.

Turney, P. D. (2001a). Answering subcognitive Turing test questions: A reply to French. *Journal of Experimental and Theoretical Artificial Intelligence, 13*(4), 409–419.

Turney, P. D. (2001b). Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. *Proceedings of the Twelfth European Conference on Machine Learning* (ECML2001), 491–502.

van Dijk, T. A., & Kintsch, W. (1983). *Strategies of discourse comprehension.* New York: Academic Press.

van Rijsbergen, C. J. (1979). *Information retrieval* (2nd ed.). London: Butterworths.

Vinokourov, A., Shawe-Taylor, J., & Cristianini, N. (2002). Inferring a semantic representation of text via cross-language correlation analysis. Neural Information Processing Systems *NIPS 2002.* Retrieved April 9, 2003, from http://www.cs.cmu.edu/Groups/NIPS/NIPS2002/NIPS2002preproceedings/papers/AP10.html

Witter, D. I., & Berry, M. W. (1998). Downdating the latent semantic indexing model for conceptual information retrieval. *The Computer Journal, 41*(8), 589–601.

Wolf, P., & Raj, B. (2002). The MERL SpokenQuery information retrieval system: A system for retrieving pertinent documents from a spoken query. *IEEE International Conference on Multimedia and Expo (ICME 2002),* 317–320

Wong, S. K. M., Ziarko, W., Raghavan, V. V., & Wong, P. C. N. (1987). On modeling of information retrieval concepts in vector spaces. *ACM Transactions on Database Systems, 12*(2), 299–321.

Xu, J., & Croft, W. B. (1996). Query expansion using local and global document analysis. *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 4–11.

Zelikovitz, S., & Hirsh, H. (2001). Using LSI for text classification in the presence of background text. *Proceedings of the Conference on Information and Knowledge Management, CIKM'01,* 113–118.

Zha, H., & Simon, H. (1999). On updating problems in latent semantic indexing. *SIAM Journal on Scientific Computing, 21*(2), 782–791.