ELSEVIER

Cognitive Science 27 (2003) 491–524

COGNITIVE SCIENCE

http://www.elsevier.com/locate/cogsci

Data-driven approaches to information access

Susan Dumais*

Microsoft Research, One Microsoft Way, Redmond, WA 98033, USA

Received 30 April 2002; received in revised form 7 January 2003; accepted 21 January 2003

Abstract

This paper summarizes three lines of research that are motivated by the practical problem of helping users find information from external data sources, most notably computers. The application areas include information retrieval, text categorization, and question answering. A common theme in these applications is that practical information access problems can be solved by analyzing the statistical properties of words in large volumes of real world texts. The same statistical properties constrain human performance, thus we believe that solutions to practical information access problems can shed light on human knowledge representation and reasoning.
© 2003 Cognitive Science Society, Inc. All rights reserved.

Keywords: Latent semantic analysis; Information retrieval; Text categorization; Question answering

1. Introduction

Information access tools help people find information in external data sources, such as computers. This paper summarizes research in three practical information access problems—information retrieval, text categorization, and question answering. A common theme in all three applications is that practical information access problems can be solved by analyzing the statistical properties of words in large volumes of real world texts. The same statistical properties constrain human performance, thus we believe that solutions to practical information access problems can shed light on human knowledge representation and reasoning. For each application area, we begin by describing the solution to the information access problem and then examining relationships to results in human knowledge representation.

* Tel.: +1-425-706-8049; fax: +1-425-936-7329.
E-mail address: sdumais@microsoft.com (S. Dumais).

0364-0213/03/$ – see front matter © 2003 Cognitive Science Society, Inc. All rights reserved.
doi:10.1016/S0364-0213(03)00013-2

Anderson and colleagues (e.g., Anderson, 1989; Anderson & Schooler, 1991) have also called attention to the analogy between information retrieval and semantic memory processes, and more generally to the environment as an important source of input for rational analyses of cognitive processes. Jones (1986) also drew parallels between human memory and information retrieval systems, although his focus was primarily to motivate the design of information retrieval systems by examining models of human memory.

The problems of information retrieval, text categorization, and question answering are all concerned with how people access information from external sources such as computers. A number of practical solutions to these problems have been developed. The problem of information retrieval has been the most thoroughly studied in practical settings and is the best developed in its link to human cognition. Latent semantic analysis (LSA) was developed to improve information retrieval. More recently, LSA has been suggested as a model of language understanding and this work has generated significant interest and controversy. In the area of text classification, several machine learning techniques have been developed to automatically classify documents. There has been some exploration of how these models can be used to describe human classification performance. Finally, recent work has developed practical and scalable question answering systems. There has been little work on how this related to human cognition, but we hope that practical solutions to text categorization and question answering will be useful in informing cognitive models.

Several themes emerge from this analysis. The solutions to all three practical problems are based on simple mechanisms applied to large amounts data. The solutions use statistical as opposed to semantic representations and operate primarily by induction from data. To the extent that the data is representative of what humans encounter, the constraints and regularities of the environment can be used to understand human cognition. Finally, we believe that solutions to practical problems can shed light on human knowledge representation and meaning; some of these links have already been made and others require further empirical work.

2. Information retrieval

Most approaches to retrieving information from external sources depend on a lexical match between words in a users query and words in the documents (Salton & McGill, 1983). Indeed, this is the way that all of the popular web search engines work. Some search engines, like Google, augment the text of a web page with the anchor text from in-links, but a lexical match is still performed on this enriched content. Such systems are, however, far from ideal. We are all aware of the tremendous amount of irrelevant information that is retrieved in web searching. We also fail to find large amounts of relevant content as well.

Fundamental characteristics of human verbal behavior underlie these retrieval failures. Furnas, Landauer, Gomez, and Dumais (1987) showed that people generate the same keyword to describe well-known objects only 20% of the time. Poor agreement has also been observed in studies of inter-indexer consistency (e.g., Tarr & Borko, 1974), and in the generation of search terms (e.g., Fidel, 1985; Bates, 1986). Because of the tremendous diversity in the words that people use to describe the same object or concept (*synonymy*), searchers

will often use different words than authors and relevant materials will be missed. Someone looking for information on "tumors" will not find articles that use only the term "neoplasm" and not tumor. Conversely, because the same word often has more than one meaning (*polysemy*), irrelevant materials will be returned. Words like "saturn," "chip," and "bug" have several very different meanings. A short query like "saturn" will return many irrelevant documents.

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 surface-level variability by converting words to their morphological root. For example, the words "cognitive," "cognition," "cognate," and "cognitively" all get stemmed to their root "cognit." The root form is used for both document and query processing. Stemming does not address cases where related words are not morphologically related (e.g., "physician" and "doctor"). Consequently, stemming does not help retrieval by much on balance (Harman, 1991). *Controlled vocabularies* have also been used to limit variability by requiring that query and index terms belong to a pre-defined list of index terms, called the controlled vocabulary. Library of Congress Subject Headings, Medical Subject Headings, Cognitive Science keyword entries, and Yellow Page headings are examples of controlled vocabularies. If the searcher can find the right controlled vocabulary term they do not have to think of all the morphological related or synonymous terms that authors might have used. However, assigning controlled vocabulary terms in a consistent manner is an error prone process. Thus, the effectiveness of controlled vocabulary indexing compared to full text indexing is variable, sometimes improving retrieval sometimes hurting it (Lancaster, 1986). Richer *thesauri* can also been used to provide synonyms, generalizations and specializations of users' search terms. Thesaurus entries can be generated either manually or by the automatic analysis of large collections of texts. Even with these enhancements, it is often difficult to find all documents relevant to a query. With the advent of large scale collections of full text, automatic statistical approaches are being used more and more to analyze the relationships among terms and documents. LSA takes this approach.

Latent semantic analysis was first introduced by Deerwester, Dumais, Furnas, Landauer, and Harshman (1990) as a technique for improving information retrieval.[1] The basic idea is to reduce the dimensionality of the information retrieval by mapping documents and terms to a common conceptual or semantic space. LSA simultaneously models the relationships among documents based on their constituent words, and the relationships between words based on their usage in similar documents. The constraint satisfaction problem, a kind of induction, is accomplished using a technique from linear algebra. By using fewer dimensions for representation than there are unique words LSA induces similarities among terms which is useful in solving the information retrieval problems described above. LSA is a fully automatic statistical approach to extracting relations among words by examining their contexts of use in documents. 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, lexical reference systems (e.g., WordNet), or semantic networks. The only input to LSA is large amounts of meaningful passages of texts. Because of this, it is easy to apply LSA to languages other than English and to cross-language retrieval problems as well (Littman, Dumais, & Landauer, 1998).

2.1. LSA and applications to information retrieval

Mathematical details of how LSA is used for information retrieval are presented in Deerwester et al. (1990), Dumais (1991), and Berry, Dumais, and O'Brien (1995). Here we only highlight the main steps, omitting the formal mathematics.

The analysis consists of four main steps:

1. *Term by document matrix*. A large collection of texts is represented as a term–document matrix. Rows are words and columns are documents. Smaller units such as passages or sentences can be used instead of documents, 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 the matrix representation, thus the name "bag of words" representation is often used to describe this representation. For general information retrieval tasks (e.g., *find journals about cognitive science*) word order can generally be ignored. This aspect of the representation may, however, be limiting for richer modeling of human memory performance.
2. *Transformed term by document matrix*. The cell entries are transformed to cumulate frequencies in a sub-linear fashion (typically $\log(\text{freq}_{ij})$), and inversely with the overall occurrence of the term in the collection (typically an inverse document frequency or entropy-based score). The transformed matrix is used as input to further analysis.
3. *Singular value decomposition (SVD)*. A reduced-rank singular value decomposition (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 rank k approximation to the original matrix, in the least-squares sense. Each document and term is now represented as a k-dimensional vector in the spaced derived by the SVD. The SVD analysis is closely related to eigen analysis, factor analysis, principal components analysis, and linear neural networks.
4. *Computing similarities*. Similarities can be computed among vectors in the reduced-dimensional space. The cosine between vectors is used as a measure of their similarity for many information retrieval applications because it works well in practice (Deerwester et al., 1990; Salton & McGill, 1983). In computing similarities, the dimensions are weighted according to their importance as determined by the SVD. Since both terms and documents are represented in the same space, document–document, term–term, and term–document similarities can be computed. In addition, terms or documents can be folded-in 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* or weighted average of its constituent terms and then compared to documents vectors to find the most similar documents.

By adding the constraint that the observed term–context 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 referred to as the "semantic" space, because relationships among words (and documents) are captured. One important consequence of the dimension reduction for information retrieval is that a query can be very similar to a document even though the two do not share any words. LSA starts with local co-occurrence data, but goes

well beyond that. In an encyclopedia collection to be described in detail below, 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 (e.g., with words like patient, hospital, sick, recovery, surgery, nurse, etc.) and when dimension constraints are imposed, they end up at similar locations.

A geometric analogy helps highlight the differences between traditional retrieval systems and the reduced-dimension LSA approach. Most retrieval systems, especially the vector retrieval model (Salton & McGill, 1983), have a natural geometric interpretation. In the vector space model, the rows of the term–document matrix (terms) are the dimensions of the space. Documents (and queries) are represented as vectors in this space, with the values in the term–document matrix determining the length and direction of the vector. Two vectors can be compared using any number of similarity measures, although the cosine between the vectors has been shown empirically to work quite well. Note that in this representation, terms are orthogonal since they form the axes of the space. In addition, if a document does not contain a term, it has similarity 0 with a query consisting of just that term. For example, a query about *cars* will not retrieve any documents containing *automobile* (but not car).

LSA can also be thought of geometrically. The axes of the LSA space are those derived from the SVD; they are linear combinations of terms. Both terms and documents are represented in this *k*-dimensional LSA space. In this representation, the indexing dimensions are orthogonal, but terms are not. The location of term vectors reflects the correlations in their usage across documents. A query can also be represented a vector in the LSA space and compared to terms or documents. An important consequence of the dimension reduction is that terms are no longer independent. Because of this a query can match documents even when they do not contain the query terms.

LSA has been evaluated by comparing it to the standard vector retrieval approach for several information retrieval test collections. We refer to the vector system as a word matching system to highlight the fact that retrieval depends on lexical word matching. For the test collections, user queries and relevance judgments (judgments about the relevance of every document in the collection to the query) are available, making systematic comparisons among systems possible. The Step 2 matrix is used for both word matching and LSA. For LSA the Step 3 dimension reduction is performed and for word matching the Step 2 matrix is used as is with no dimension reduction.

In information retrieval applications, users submit a query and the system returns a ranked list of documents. For test collections used in evaluations, the relevance of each document to a query is known, so it is straightforward to measure the ability of a system to discriminate relevant from irrelevant documents. The 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. Recall can only be calculated for test collections in which all of the relevant documents are known. For example, if there are 100 documents relevant to a query, then a recall level of .10 occurs when the first 10 relevant documents have been returned to the user. *Precision* is the proportion of relevant documents in the set returned to the user. Precision is calculated at several levels of recall to generate a curve showing the tradeoff between precision and recall. Precision–recall curves are closely related to ROC curves

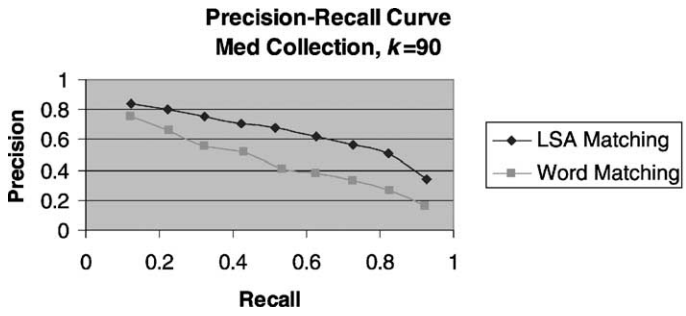

Fig. 1. Example precision recall curve for MED collection. Precision is shown as a function of recall, averaged over 30 test queries. Precision for LSA matching is better than that for word matching over the entire recall range.

that are often used to characterize human perception and memory. Recall is the same as correct detection or hits. Swets (1963) described how d' could be used to measure the performance of information retrieval systems. We report precision–recall data because it is more common in information retrieval research.

Fig. 1 shows the results for a test collection of 1,033 abstracts of medical documents, 5,831 terms, and $k = 90$. Precision is plotted as a function of recall, averaged over 30 queries. As is typical in information retrieval applications, precision drops as recall increases. Finding the first few relevant documents is easy, but to find the last few relevant documents a lot of irrelevant documents must be considered as well. As can be seen, LSA performance is substantially better than standard word matching for the entire range of recall values, with an average advantage of about 30%. For example, at the 50% recall level, 68% of the documents returned LSA are relevant compared with 40% of the documents returned by word matching. Similar performance advantages are observed for several other test collections although the magnitude of the difference is not always as large (Deerwester et al., 1990).

LSA involves a parameter k, the number of dimensions in the reduced space. Fig. 2 shows LSA performance as a function of number of dimensions for the medical collection described

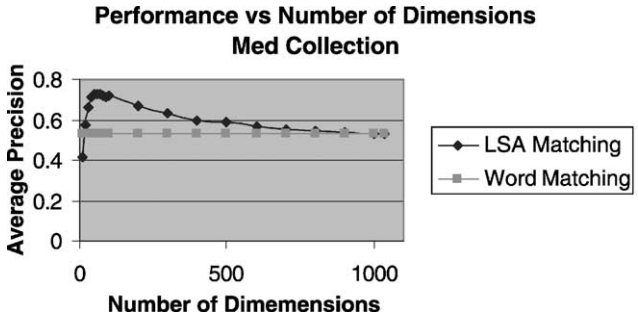

Fig. 2. Performance as a function of number of dimensions. Precision scores averaged over several levels of recall and 30 test questions. LSA performance is poor initially, then surpassed word matching for most of the range. Optimal performance is at $k = 90$ for this collection. Word matching performance is constant since number of dimensions is not a parameter for that model.

above. Word matching performance, which is constant across dimensions, is also shown for comparison. With too few dimensions, LSA performance is poor because there is not enough representational richness. With too many dimensions, performance decreases because LSA models the noise in the data thus reducing generalization accuracy. In between these two, there 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 performance with very few dimensions, an increase ance over a substantial range, and then a decrease to word matching level is observed for other collections as well (see Landauer & Dumais, 1997, Fig. 3).

Choosing the right dimensionality is required for the successful application of LSA to information retrieval and, as we shall see below, to simulate human performance. It is interesting to note that many of the early factor analytic or multidimensional scaling approach to modeling word meaning used very few dimensions (Deese, 1965; Fillenbaum & Rapoport, 1971; Osgood, Suci, & Tannenbaum, 1957). For information retrieval applications, using only a few dimensions is not adequate to capture the richness of meaning in a variety of contexts. In addition, the early psychological models depended on human judgments of word similarities (e.g., word association norms) for creating the semantic space, whereas LSA uses similarities that are automatically computed from naturally occurring discourse.

LSA has been successful in information retrieval applications for which it was initially designed. The dimension reduction constraint was a key part of that success as illustrated in Fig. 2. We now explore the extent to which the semantic relationships captured by LSA on the basis of a large set of natural texts might also characterize important aspects of human semantic memory.

2.2. LSA and human behavior

More than 50 years ago Vannevar Bush (1945) speculated about memex, a machine that should be an extension of the personal memory belonging to an individual, and should work in a fashion analogous to the working human brain, by association. Jones (1986) and Anderson (1989) also drew parallels between human memory and information retrieval systems, and more generally to the environment as an important source of input for rational analyses. LSA can be used to analyze large collections of naturally occurring texts and its reduced dimensional semantic space represents a kind of acquired similarity among words. We review a sample of applications of LSA to model aspects of such learning. Landauer, Foltz, and Laham (1998) and Landauer (2002) provide more comprehensive overviews of LSA and applications to human memory and discourse processing.

2.2.1. Vocabulary tests

Landauer and Dumais (1996, 1997) first explored the ability of the reduced dimensional representation induced by LSA to simulate aspects of human knowledge and meaning relations. They used the Educational Testing Service (ETS) Test of English as a Foreign Language (TOEFL), a multiple choice test of synonymy. The test consists of 80 multiple choice items, including a target word or short phrase and four alternatives for each target. Students are asked

to select the alternative closest in meaning to the target. Example test items include:

Target	Alternatives
Provisions	Stipulations, interrelations, jurisdictions, interpretations;
Unlikely	Improbable, disagreeable, different, unpopular
Physician	Doctor, nurse, pharmacist, chemist

Students from non-English speaking countries take this test for admission to many U.S. colleges. Summary student data was provided by ETS. On average these students answered 64% of the 80 questions correctly.

For the LSA analysis, they trained LSA on approximately 5 million words of text from 30,473 articles in a high-school level encyclopedia, Grolier's Academic American Encyclopedia. Given the encyclopedia texts, all computations are done automatically as described above—a term–article matrix is built, cell entries are transformed, a reduced-dimension SVD is computed, and the resulting *k*-dimensional vectors are used for analysis. To take the TOEFL test, the similarity between the target word and each of the four alternatives is computed. The answer with the highest score is LSA's guess as to the synonym. Performance for this completely automatic LSA system is 64%, exactly the same as the students who took the test. In addition, error patterns are generally similar to those observed for students. For incorrect items, the Pearson product moment correlation between the relative frequency of student responses and LSA cosines is .44, indicating similar error patterns. There are some interesting counterexamples to this trend, however. For the target physician, students and LSA disagree. Students select the correct synonym, doctor, 72% of the time whereas LSA selects nurse which has the highest cosine similarity with doctor. Physician is also related to doctor in the LSA space, but less so than to nurse. The details of the training texts (an encyclopedia in this case) are not critical. Similar LSA TOEFL test performance was found using a similar sized sample of newswire text from the Associated Press (Landauer, Laham, & Foltz, 1998). What is critical is that large amounts of words are used in natural contexts as input to the LSA analysis.

It is important to note that more than simple co-occurrence relations are needed for a computational system to perform well on the TOEFL synonym task. If the term–article matrix is used without any dimension reduction, accuracy on the TOEFL test is only 16%. The proper choice of *k* matters for the synonym test as it did with information retrieval tests. With just two or three dimensions LSA performance is quite poor (14%), and with too many dimensions it is again performs quite poorly (16%). But with 300–350 dimensions performance is 60% or more (see Landauer & Dumais, 1997, Fig. 3). The constraints imposed by dimension reduction are a key to providing useful associative relationships for this task. The performance difference between LSA and word matching is larger in the TOEFL test than in the information retrieval application shown in Fig. 1. We suspect this is because in the TOEFL test, both the query and the target alternatives have very little text associated with them (only a single word in each case). If the target and stem words do not co-occur, word matching techniques will fail to find the right answer.

Turney (2001) reported very good TOEFL performance (74%) using a variant of a word matching technique called PMI-IR. The 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—i.e., $\text{score}(\text{alternative}_i) = \log(p(\text{target and alternative}_i)/p(\text{target})\,p(\text{alternative}_i))$. Because only the rank of the alternatives is needed for the synonym test, the log can be removed as can p(target) which is the same for all alternatives, and the formula simplifies to $\text{score}(\text{alternative}_i) = \text{count}(\text{target and alternative}_i)/\text{count}(\text{alternative}_i)$. The counts were obtained from a large Web search engine, AltaVista. The scoring function was further modified to take into account the proximity of the words, negation, and context words for sense disambiguation. The final scoring function results in a TOEFL score of 74%. The simple co-occurrence score is 62%, which is somewhat worse than the 64% reported by Landauer and Dumais, but well above their word matching score of 16%. There are several differences in the experiments which could account for the differences in performance. The most important difference is the amount of text used for the analysis. Landauer and Dumais (1997) used 30,473 encyclopedia articles representing 5 million words of text. Turney (2001) used a much larger collection of text from the Web. In 2001, AltaVista indexed 500 million Web pages, which is more that 16,000 times the number of encyclopedia articles analyzed. Additional experiments looking at PMI-IR on smaller collections or LSA or larger collections is 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 (a theme we will return to in the section on question answering). From the more theoretical perspective of modeling aspects of human memory, the tremendous amounts of additional data are not characteristic of the amount of text processed by humans.

2.2.2. Rate of vocabulary acquisition

Landauer and Dumais (1997) examined the extent to which LSA could model vocabulary acquisition in children. They looked at two aspects of learning—the rate at which LSA acquired knowledge, and the influence of direct versus indirect exposures to words. Children in middle school acquire the meanings of new words at an average of 10–15 words per day. Even though the LSA model can pass an adult vocabulary test, if it required much more data than a human encounters to achieve the same performance then one would have to conclude that something significant was missing from the data-driven inductive approach. Children acquire some (if not most) of their knowledge of word meanings not through explicit definition but by observing how words are used—that is by reading and listening. Knowledge about words can be acquired either directly (i.e., reading a word in context) or indirectly (i.e., reading in general). Indirect learning allows for the acquisition of other words, but not those being tested.

To explore the rate of acquisition of word meanings from both direct and indirect evidence, Landauer and Dumais conducted several simulation experiments using variants of the encyclopedia collection described above. While it would have been ideal to have a large representative collection of age-appropriate materials, none was available at the time. They varied the total collection size from 2,500 to 30,473 text samples. The length of the average text sample is 151 words, so this corresponds to exposure to 377,000–4,601,000 words of text, respectively. In addition, they varied the amount of direct and indirect exposure to words. Direct exposure was manipulated by varying the number of articles containing a word. Indirect exposure was manipulated by changing the total number of articles analyzed. The variables were experimentally manipulated by randomly replacing TOEFL test words with nonsense words

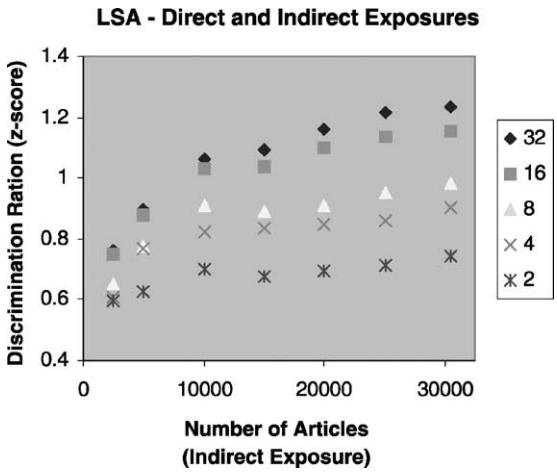

Fig. 3. LSA performance on the TOEFL test as a function of direct and indirect exposures. The amount of direct experience (number of articles) varies from 2,500 to 30,000; the amount of direct experience varies from 2 to 32. The dependent measure is the normalized difference in cosine of the target word to the correct and incorrect alternative. Both direct and indirect exposures affect performance, and there is an interaction such that less direct experience is needed the more indirect experience there is.

(so as to maintain context of use but control exposure frequency) and choosing random nested samples of the articles. The number of articles containing the TOEFL test words and the number not containing the test words were independently varied to explore direct and indirect effects. The number of direct exposures per word was varied from 2 to 32, and the amount of indirect experience varied from 2,500 to 30,473 articles.[2]

Fig. 3 shows the results of these simulation experiments. The amount of indirect experience is shown on the *x*-axis, the curves represent different amounts of direct experience, and the *y*-axis is TOEFL test performance using a *z*-score measure. Their experiments revealed that both direct and indirect exposure to words is important in acquiring word meanings. There is also an interaction—direct experience with a word helps more when there is more experience with other words. The more known about words in general, the easier it is to acquire the meaning of a new word. For example, when the system has experience with 30*k* articles only 4 direct exposures to words results in a *z*-score of .90, whereas when there are only 5*k* samples it takes 32 direct exposures to get to the same level of performance (see Fig. 3). A three-parameter model fits the results nicely ($r^2 = .98$). The model is a simple log function, $z = a \times \log(b \times T) \times \log(c \times S)$, where T is the total number of text samples analyzed and S is the number of samples containing the TOEFL word. Using this model, they estimated the effect of indirect exposures to be three times more important than direct exposures. And they found that LSA acquired the meaning of 10 new words after reading 50 samples, which is about what a school aged child reads per day. They concluded that there is enough information present in the language to which learners are exposed to allow them to acquire knowledge needed to pass multiple choice vocabulary texts. If the human process of induction is roughly comparable in efficiency to LSA in extracting word similarities from discourse (not to mention spoken and visual input they have access to) children could acquire words at the rate observed.

2.2.3. Semantic priming

When people are asked to decide whether a letter string is a word or not, they do so faster if they have just read a sentence that is related to (but does not contain) the word. Landauer and Dumais (1997) showed that an LSA semantic representation can model semantic priming effects. Till, Mross, and Kintsch (1988) presented readers short passages one word at a time and interrupted them at strategic points to make lexical decisions about one or another sense of recently-presented homophones. Landauer and Dumais modeled data from this experiment using the encyclopedia scaling developed for the TOEFL test. The test words from the Till et al. (1988) study were already vectors in the LSA space, and the short passages were located at the centroid of the words they contained. Cosine similarity was used to measure the similarity of the test words to the passages. Three main results were nicely predicted by the LSA model:

(a) When the ambiguous homographs were presented alone they were equally similar to words related to both meanings.
(b) The passages containing the homographs were more similar to the related sense than the unrelated sense.
(c) The passages were significantly closer to the appropriate sense of inferentially related words than to unrelated control words.

Lund and Burgess (1996) modeled other priming data using a high dimensional semantic model, HAL (hyperspace analog of language), that is related to LSA and described in more detail below. They used prime-target pairs from experiments by Chiarello, Burgess, Richards, and Pollock (1990) along with unrelated word pairs as controls. Subjects were presented with a prime word followed by a target word (or nonword) and the decision time was recorded. Distances between words in the HAL space was computed for all pairs. Correlations between semantic distance (measured using three different metrics) and human decision times were highly significant.

2.2.4. Textual coherence

Kintsch and his colleagues (van Dijk & Kintsch, 1983; McNamara, Kintsch, Songer, & Kintsch, 1996) have developed methods for representing texts in a propositional language. They have shown that the comprehension of text depends strongly on its coherence, as measured by the overlap between the arguments in the propositional representations. Propositional analyses are typically carried out by hand.

Foltz, Kintsch, and Landauer (1998) used LSA to automatically measure textual coherence with good success. To measure textual coherence using LSA, they first created a reduced-dimensional space from large amounts of text as described above. Then sentences are represented as points in this same space, at the centroid of their constituent terms. The cosine similarity between successive sentences is used to predict the comprehensibility of text. Comprehension measures from two previous studies (Britton & Gulgoz, 1991; McNamara et al., 1996) were modeled using LSA. Both studies systematically manipulated the coherence of text, as reflected in readers' comprehension scores, and showed that differences in coherence could be model with their propositional techniques.

For data from the Britton and Gulogz (1991) study, Foltz et al. (1998) showed that LSA-derived coherence measures (similarity between successive sentences) was highly correlated

with human test scores ($r^2 = .99$). Simple word overlap measures (based on the same matrix used for input to LSA) were also highly correlated with human performance ($r^2 = .95$). Most of the coherence effects for this test are predicted by simple word-overlap statistics, although dimension reduction helps a bit. The coherence relations between sentences are more subtle in the McNamara et al. (1996) study. In their experiment, words and phrases with similar meaning but different lexical forms were often substituted to provide conceptual bridges between one sentence and the next. For these texts, LSA similarity was strongly correlated with human performance ($r^2 = .89$), but simple word overlap techniques were not ($r^2 = .03$).

LSA can be applied to modeling discourse coherence as reflected in readers' comprehension. It approximates many of the same features found in propositional models of text comprehension. It is interesting to note that LSA similarity measures are based only on the extent to which two units of text discuss semantically related topics. There is no syntactic analysis or parsing of the texts, yet predictions are quite accurate. On the practical side, because LSA is a fully automatic method, it permits the analysis of much larger collections than have previously been used in text coherence research.

2.2.5. Essay tests

Landauer, Laham, Rehder, and Schreiner (1997), Foltz, Laham, and Landauer (1999), and Landauer, Laham, and Foltz (2000) described how an LSA-based system can be used to score the quality of free-form essays. The ability to convey information verbally is an important skill, and one that is not sufficiently well assessed by other kinds of tests. Because essays are difficult and time consuming to score, especially when applied on a national scale, they are not as widely used as possible. Some earlier attempts to develop computational techniques to aid in the scoring of essays focused 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. In LSA-space, two essays can be quite similar even though they share few words, as long as they convey the same meaning.

To assess the quality of essays, LSA is first trained on a sample of domain-representative text. The standard reduced-dimension semantic space is automatically derived from these texts. Next, essays with known quality scores are added to the space, and ungraded essays are compared to the essays of known quality. Essays can be represented either as a whole or at the level of smaller sub-components such as sentences or paragraphs. There are several techniques for assigning a grade to a new essay based on the grades of similar essays, but most are variants of nearest neighbor approaches. 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., 1997 for details). The use of sub-components of essays allows for comparison against a predetermined set of topics or aspects that should be covered (e.g., Foltz, 1996; Foltz, Britt, & Perfetti, 1996). The approach has been applied to essays on a wide range of topics including heart anatomy, physiology, social studies, physics, law, as well as general opinion and argument essays.

In one study reported by Foltz et al. (1999), essays from the ETS Graduate Management Achievement Test (GMAT) were graded. Performance of the fully automated LSA approach was compared to performance of two trained ETS graders. For one question a sample of 695 opinion essays with six possible grades was examined. The correlation between the grades

assigned by two trained ETS graders was .86. LSA grades were automatically assigned as described above. The correlation between the LSA grades and the ETS graders was also .86. For a second question, a set of 668 argument essays with six possible grades was examined. The correlation between two trained graders was .87, and that between LSA and the graders was .86. Thus, LSA is able to perform at near the same reliability as trained ETS graders. Larkey (1998) has used related statistical text analysis techniques along with stylistic measures to automatically score essays, with similarly impressive results. Automatic techniques work quite well in an absolute sense and agree with human graders to the same extent that they 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.

Scoring of essays is just one use of LSA for supporting the analysis of written texts. Kintsch and colleagues have used LSA to match students with text at the optimum level of complexity for learning. Earlier work by Kintsch showed that students learn the most when a text is neither too hard nor too easy. Linking the information represented in the current text with prior knowledge is a key to learning (McKeown, Beck, Sinatra, & Loxterman, 1992; McNamara et al., 1996). Texts that contain only known information are not useful for learning. Neither are texts that are too distant from the reader's current knowledge. Hence, there exists an intermediate zone-of-learnability, where texts are different from what the reader already knows, but not too different.

Wolfe et al. (1998) showed how LSA can be used to match instructional text of varying topical sophistication with students differing in background knowledge. LSA was used to characterize both the knowledge of an individual student and the knowledge conveyed by a text, and then to match the student and the text. For their study, they used texts about heart function that varied in difficulty including those intended for elementary school students, general interest readers, undergraduate anatomy students, and medical pathology students. They tested college undergraduates and medical students. Pre- and post-reading tests were used to assess the knowledge gained by reading. LSA derived scores were good predictors of both prior domain knowledge and learning. The correlation was significant between the LSA measure of prior knowledge and the pre-questionnaire score ($r^2 = .63$) and the pre-essay score as measured by ETS graders ($r^2 = .68$), suggesting that LSA scores can stand in for more expensive methods of assessing prior knowledge. There was also a reliable quadratic relationship between the LSA similarity between the pre-essay score and the text read with the amount learned. This models the finding that students who do not know much to begin with do not learn, and those who already know a lot do not learn much.

2.2.6. Prose recall

Dunn, Osvaldo, Barclay, Waterreus, and Flicker (2002) used LSA to score prose recall. In clinical applications, performance on the logical memory test of the Wechsler Memory Scale can predict subsequent cognitive decline. This test measures memory by means of prose recall, and like essay tests, it is difficult to score. Dunn et al. (2002) compared LSA against two common scoring methods, using correctly recalled story and thematic units. For LSA scoring, the similarity of the original prose and the recalled prose were measured using cosine similarity.

LSA scores were highly correlated with the existing scoring techniques. Furthermore, LSA was able to detect cognitive impairments.

2.2.7. Analogical reasoning

Ramscar and Yarlett (2003) describe a series of experiments showing how LSA contributes to human performance in the retrieval of analogies from long term memory. They distinguish between two main processes in analogy, retrieval and mapping. In their environmental model of analogy, 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.

2.2.8. Similarity neighborhoods

Griffiths and Steyvers (2002) proposed a probabilistic variant of LSA, described in more detail below. They used their analysis to model the distribution of related words. Most words are related to a number of different topics (as 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 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 in the distribution of words across topics as seen in thesauri.

2.2.9. Related models

Related computational approaches have also explored the use of dimension reduction to represent various aspects of word meaning. Although the details are different from LSA, these other models also use simple co-occurrence contexts as input to statistical analyses that produce a consistent global representation using dimension reduction constraints. In all these models, words are similar because they occur in similar contexts.

Schütze (1992) used a high-dimensional representation of words for word sense disambiguation. Burgess and colleagues (Lund & Burgess, 1996; Burgess, 1998; Burgess, Livesay, & Lund, 1998) developed the hyperspace analog to language (HAL). HAL starts with a word–word matrix that indicates which words precede and follow each other in a large corpus of text. They then reduce the dimensionality by keeping only the columns with highest variance. Many of their simulations have been done with substantial dimension reduction (down to 200 words), but others have kept up to 140,000 words.

Hofmann (1999) developed a probabilistic LSA model (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 to select a document with probability $p(d)$, select a latent class or topic with probability $p(z|d)$, and generate a term with probability $p(t|z)$. Expectation maximization, a standard machine learning technique for maximum likelihood estimation in latent variable models, is used to estimate the model parameters. The aspects are very similar to LSA's dimensions, but are derived by maximizing a different objective function. Griffiths and Steyvers (2002) proposed a variant of Hofmann's model that assumes the mixture proportions are distributed as a latent Dirichelet random variable. As described above, their model has been

used to model the distribution of topics associated with words, and could easily be extended to explore other aspect of human memory.

2.3. Summary of information retrieval

LSA was developed initially to improved information retrieval by overcoming the variability in terms used by authors and searchers. The main idea was to use dimension reduction techniques to mitigate the effects of lexical variability. By reducing the dimensionality of the problem, the relationships among words are captured. The LSA approach has been successful in information retrieval applications.

More recently, LSA (and probabilistic variants) have been considered as a theory of human knowledge acquisition and representation, and as a method of extracting semantic content from texts. A number of simulations of cognitive and psycholinguistic phenomena (vocabulary acquisition, semantic priming, textual coherence, essay grading, analogy retrieval, similarity neighborhoods) show that LSA captures a great deal of the similarity of meanings evidenced in a variety of behavioral tasks. There are a number of benefits to modeling lexical knowledge in the data-driven way that LSA proposes. The semantic metric is clearly specified, and it is based on the characteristics of naturally occurring texts in the environment. There is no need to specify new semantic primitives or features; the words are initial features and statistical properties of collections determine the latent semantic analyses. Finally, it scales nicely, perhaps not to web-scale collections but certainly, to handling the volumes of text that humans encounter.

It is worth noting that LSA achieved this performance using text samples whose initial representation was simply a "bag of words"; that is, all information from word order, syntax and grammar is ignored. Because the model cannot see or hear, it cannot make use of phonology, morphology, orthography, or real world perceptual knowledge. While it seems unlikely that the human brain computes the SVD of word-context experiences, it is quite likely that computations transform local experiences (e.g., words in context) into global knowledge. The basic process, the representation of myriad local associative relations between components and larger contexts of experience in a joint space of lower dimensionality, offers a candidate for such a mechanism. This mechanism has been shown sufficient to approximate human knowledge acquisition from natural sources at natural scale. At the very least, LSA demonstrates how traditional associative memory models can be extended to exploit higher-order correlations. Such representations are sufficient for capturing a wide range of interesting psychological behavior.

We now turn to another information access application, text categorization, that has been addressed using data-driven techniques, and describe how these practical solutions might be related to human categorization.

3. Text categorization

As the volume of information available online increases, there is growing interest in helping people better find, filter, and manage these resources. *Text categorization* (also referred to as text classification) is an important component in many information organization and management

systems. Text categorization is the assignment of one or more pre-defined category labels to natural language texts. Categorization is a supervised learning technique in which examples of items in each category are provided—that is, the categories are predefined and the learning task is to model the regularities in this structure. Clustering which involves discovery of categories in an unsupervised fashion is also an active area of research in text analysis but beyond the scope of this paper (for reviews see Kaufmann & Rousseeuw, 1990; Willett, 1988).

The most widespread use of text categorization to date has been for assigning subject categories to documents. Applications include indexing documents and web pages by controlled vocabulary (e.g., topical tags for news content, or web categories like Yahoo!), help desk automation, identification of spam email, alerting, and many others. News services, for example, manually tag articles into categories like *current news*, *sports*, *legal*, *bonds*, *mergers*, *markets*. Web indexers manually tag web pages into a large hierarchy of general subject headings like *entertainment*, *computing*, *lifestyle*, *travel*, *reference*, *shopping*. Medical, legal, and general library ontologies are also widely used for indexing and search. The process of manual indexing is time consuming and costly. Consequently, there is tremendous interest in developing technologies for automatic text categorization. Some early attempts to automate text classification had humans write rules to distinguish categories which could then be applied automatically to new examples (Hayes, Andersen, Nirenburg, & Schmandt, 1990). Building and tuning these rules required considerable skill and ongoing maintenance as categories evolved over time. The more popular approach to automation involves having domain experts provide examples of items in each category, and then using machine learning techniques to learn a model for each category. So-called *inductive learning* techniques start with examples of items in each category, and learn a model that characterizes the category. This learned model can then be used to classify new instances. Such systems can be run completely automatically or with human interaction to verify suggested class labels.

A number of statistical analysis and machine learning techniques have been applied to text categorization, including multivariate regression models, discriminant analysis, nearest neighbor classifiers, probabilistic and language models, decision trees, neural networks, perceptrons, symbolic rule learners, vector representations including LSA, and support vector machines. Sebastiani (2002) provides a nice review of several learning techniques and applications to text classification. The learning techniques used for the practical problem of text categorization represent many of the popular models used in human categorization research, including rule-based, exemplar-based, prototype-based, and category-boundary approaches (Goldstone & Kersten, in press).

Learning algorithms are often distinguished by whether they are discriminative or non-discriminative, and whether they are batch or online. Discriminative approaches explicitly discriminate between positive and negative examples; non-discriminative approaches simply characterize the positive examples. Batch methods build a classifier by analyzing the training set all at once—discriminant analysis and support vector machines are examples of batch algorithms. Online algorithms build classifiers after analyzing only a few instances and update the models constantly—perceptrons, language models, and most centroid methods are examples of online classifiers. The choice of model depends primarily on the accuracy of generalization to new instances, but also on factors like training time, real-time classification speed, etc. The desirable tradeoffs between these factors depend on the application. Differences in the costs

of different kinds of errors (false positives vs. misses) determine how decisions thresholds are set for particular applications.

Most text classification applications involve simple binary classification—in category versus out of category. If there are several categories of interest, several binary classifiers are learned. One can construct a multi-class classifier from these binary classifiers, and the underlying binary approach provides flexibility in doing so. In almost all practical applications, overlapping categories are desirable—that is, a document can belong to more than one category. A medical article may well be about heart disease, human studies, and blood vessels.

Although the details of the learning mechanisms vary, the document representations used are similar to those described above for information retrieval. In addition to the basic term–document matrix, one or more category tags for each document is available. The vector for each word can be binary (word present–absent), counts of how often a word occurs in a document (*tf*), or some transform of the counts (usually one of the $tf \times idf$ family of transforms described earlier). Sometimes all the words are used, but often feature selection is performed using frequency thresholds, mutual information with category, global LSA-based dimension reduction, etc. In some applications, domain-specific features are used in addition to words. Sahami, Dumais, Heckerman, and Horvitz (1998) used text classification to distinguish spam email from regular mail. Their document representation used words as features but also consider domain-specific features like the time the email was sent, the proportion of capital letters, the number of exclamation marks, the number of recipients on the To: line, etc. Similarly, in applications involving the classification of web pages, information about the link structure of the web is often encoded into features (Chakrabarti, Dom, Agrawal, & Raghavan, 1998; Yang, Slattery, & Ghani, 2002).

We now describe two specific approaches to text classification, LSA and support vector machines, and describe applications of these techniques to modeling aspects of human categorization.

3.1. LSA classifiers for text categorization

LSA can be used for text classification as well as information retrieval (e.g., Dumais, 1995; Foltz & Dumais, 1992; Schütze, Hull, & Pedersen, 1995; Zelikovitz & Hirsh, 2001). The same reduced-dimensional space, that was used for information retrieval, is used for classification. In addition, category information has to be represented and a decision threshold established. Dumais (1995) used 50 topically defined categories from the TREC collection, a large-scale text retrieval evaluation corpus. The categories were broad topics, like Welfare Reform, Catastrophic Health Insurance, and Black Monday. The topics were defined by means of 25–742 short news articles on the topic. Dumais used the centroid of the positive examples in each category as a model for that category. More than 330*k* new articles were compared to the category centroid and judged to belong to the category if they exceeded a similarity threshold. The LSA approach was better than the median of 51 systems for 41 of the 50 categories. The use of LSA with a centroid category model is a non-discriminative and online method. The centroid is a kind of prototype model.

Schütze et al. (1995) compared an LSA representation to a term representation using five different learning algorithms (centroid, linear discriminant analysis, logistic regression, linear and non-linear neural networks). The categories were 100 TREC categories like those used

by Dumais (1995), and the test examples were news articles that needed to be classified. The LSA representation provided better performance for 61 of the categories, but there were large differences for individual categories. If there were a small number of good terms that described a category, then the term representation was preferable. If the number of indicative terms was large, then the LSA representation was superior. In this application, LSA was used with both non-discriminative (centroid) and discriminative (discriminant analysis, regression, neural networks) learning approaches.

In the essay work described above, Landauer et al. (1997), and Larkey (1998) used text classification techniques for scoring essays. Landauer et al. (1997) used a *k*-nearest neighbor classifier in combination with an LSA representation. The classes consisted of essays that had the same grade assigned to them. New essays were classified by taking a weighted combination of the grades assigned to its 10 nearest neighbors. LSA assigned scores were somewhat better than human graders' scores in predicting short-answer test scores. The *k*-nearest neighbor approach is a kind of exemplar-based process. Larkey (1998) used a prototype (centroid) approach to essay scoring, but she used words rather than derived LSA dimensions as the feature representation.

An underlying LSA representation can be combined with several learning techniques to form the basis of a text classification system. We now describe in more detail a new and very accurate classification technique, support vector machines.

3.2. Support vector machine (SVM) classifiers for text categorization

3.2.1. SVM overview

SVMs are an interesting new learning technique that has been shown to be quite effective for a number of applications including text classification (Dumais, Platt, Heckerman, & Sahami, 1998; Joachims, 1998; Yang & Liu, 1999). Because of the success of SVMs in text classification and possible applications to human memory, we describe the SVM algorithm in some detail. We then describe applications to text categorization and to support human decision making in examining search results.

SVMs are a general learning technique with solid foundations in statistical learning theory (Joachims, 2002; Vapnikm, 1995). Fig. 4, illustrates the SVM approach for a simple two class problem. In this problem, there are two classes of objects—the circles are positive examples of the category of interest and the squares are negative examples. There are only two features for each object, the *x* and *y* axes. (Note that for text classification problems, the representations usually have hundreds of thousands of dimensions or features.) The learning task is to find a function that best separates the positive from negative examples. In its simplest linear form, an SVM is a hyperplane that separates a set of positive examples from a set of negative examples with maximum margin. In the linear case, the margin is defined by the distance of the hyperplane to the nearest of the positive and negative examples. Maximizing the margin can be expressed as an optimization problem. The output of the SVM learning is the weight vector, $\vec{w}$, which defines a weight for each dimension and b a scaling parameter. Using these learned weights, new instances, $\vec{x}$, can be classified by computing $\vec{w} \cdot \vec{x} - b$. Note that unlike regression techniques that minimize the sum of squared errors over all training examples, finding the optimal SVM involves only points on the margin, in effect important boundary points.

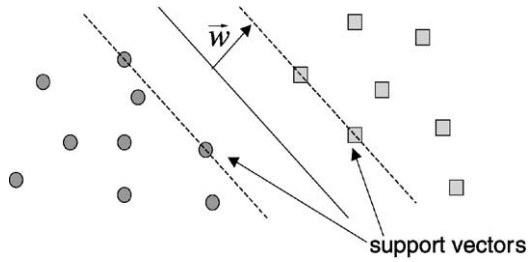

Fig. 4. Graphical representation of a linear support vector machine (SVM). In this problem there are two categories (circles and squares). The SVM learning technique finds the hyperplane that separates the categories with maximum margin, defined by the vector w in this example. Finding the optimal vector involves only points on the margin, called the support vectors.

SVMs have been shown to yield good generalization performance on a wide variety of classification problems, including handwritten character recognition, face detection, and most recently text categorization (Joachims, 1998; Dumais et al., 1998). SVMs are fast to learn using techniques developed by Platt (1998) and Joachims (1999), and very fast at classifying new instances since a simple dot product is all that is required. SVMs use overfitting protection, which is important in text applications since there can often be hundreds of thousands of features in the term–document matrix. Of course, not all classification problems are linearly separable. Cortes and Vapnik (1995) proposed a modification to the optimization formula that allows, but penalizes, examples that fall on the wrong side of the decision boundary. An alternative approach for handling non-separable data is to use kernel methods to transform the input space so that the problem becomes linearly separable in a different space. Kernels, that expand words to include synonyms explicitly or through LSA dimension reduction, have been explored (Cristianini, Shawe-Taylor, & Lodhi, 2001). For text classification problems, the simplest linear version of the SVM provides good classification accuracy (e.g., Joachims, 1998), but more sophisticated kernel functions have been explored as well.

3.2.2. SVMs for text categorization

A common benchmark for evaluating classification algorithms is Reuters, a collection of 12,902 Reuters news articles tagged by editors as belonging to 118 topical categories (e.g., corporate acquisitions, earnings, money market, interest, grain). Joachims (1998) and Dumais et al. (1998) evaluated SVM learning algorithms on this collection with similar overall accuracy—92% accuracy for the largest 10 categories and 87% over all 118 categories. Both found that SVM-based learning was superior to a number of other approaches including Naïve Bayes, 2-dependency Bayes nets, centroids, decision trees, and *k*-nearest neighbors. Yang and Liu (1999) also found SVMs superior to several other classification techniques for the Reuters collection, but they used a somewhat different performance measure, so direct comparisons are difficult. Interestingly, Dumais et al. and Joachims used fairly different representations. Dumais et al. used binary features and selected the 300 features with highest mutual information for each category. Joachims used real valued feature weights ($tf \times idf$), and kept all the features that occurred in more than three articles (almost 10,000 features for this problem). In this application, the benefits of the SVM learning algorithm dominated, and the details of the

feature representation were less important. Cristianini et al. (2001) compared a linear SVM to one with an LSA-based kernel, varying the number of dimensions in the LSA space. They too found roughly comparable overall performance with the two representations.

In other applications the representation does matter. Sahami et al. (1998) used text classification techniques for distinguishing spam or junk email from regular email. They used all the words in the email messages as features, and in addition some domain-specific features such as proportion of capital letters, time of day the message was sent, the number of recipients, whether the message contained attachments, etc. Using the domain-specific features improved performance, reducing false positive from 3 to 0% and improving coverage from 94 to 98%.

Another common benchmark collection for text classification is the TREC collection. Lewis (2001) used SVMs for text classification using 84 TREC categories. Although he did not directly compare SVMs to other techniques, his SVM-based system was one of the most effective systems, performing above the median of 18 other systems on 83 of the 84 categories and was the best performing system on 61 categories.

SVMs have been shown to be effective and efficient in several text classification problems. During learning a weight vector is established, and at test time new instances are compared to this vector. Only support vectors, or important boundary points, are important in determining the weights. The focus on decision boundaries is similar to Ashby's work on category boundary models (e.g., Ashby, 1992). SVMs are also similar to prototype models of categorization (e.g., Reed, 1972) since only a single weight vector is use to evaluate the category membership of new items. New instances, $\vec{x}$, are classified by computing $\vec{w} \cdot \vec{x} - b$, so the most similar items will be those that lie in the same direction as the $\vec{w}$ vector.

3.3. *Text categorizing for organizing search results*

Text classification can be used to support many tasks, including automated tagging, filtering, etc. We explored the use of automatic tagging of web content in combination with novel interface techniques to improve the ability of people to find information quickly. This is an interesting application of text classification that includes decision making as well. Most text retrieval systems return a ranked list of results in response to a user's search request. Such lists can be long and overwhelming. A query on *CHI 2001* on Google returns more than 2 million results. Many are about the ACM: CHI 2001 conference on computer–human interaction, which was the intent of the query. However, there are also many returned web pages on sports (Loyola CHI 2001 schedules, CHI 2001 football draft headquarters, Chi Chi Rodriguez' 2001 schedule), health (California Health Institute, Tai Chi), fraternities, restaurants, etc. Results on all these different topics are intermixed in the list requiring users to sift through a long undifferentiated list to find items of interest. We developed and evaluated several interfaces for structuring search results in order to better understand the cognitive processes that lead to effective analysis of search results. The use of automatic text classification provides the backbone for these systems.

Chakrabarti et al. (1998), Dumais and Chen (2000), Stata, Bharat, and Maghhout (2000) and others have developed automatic classifiers for Web pages. Existing Web directories like Yahoo! or MSN Web Directory or Open Directory are used to provide training examples from which category models can be learned. These directories contain 1–3 million labeled examples

and contain tens of thousands categories organized in a hierarchical fashion. In April 2002, for example, Open Directory (http://www.dmoz.org) contained 3,339,794 sites organized into 386,093 categories.

Dumais and Chen (2000) learned models for the 163 categories representing the top two levels of the MSN Web Directory (13 top-level and 150 second-level categories). Given these models, new web pages can be tagged automatically. Classification accuracy for this problem is about 65%. This is lower than the Reuters news articles described above. For reasons of efficiency in this application they used only the short summaries that were returned in search results rather than the full content of the web page. In addition, even if the full web page is retrieved, many pages contain very little textual information (they contain only images), so information about neighboring pages would need to be incorporated for higher accuracy. The classification errors are mostly failures to classify a page into any of the known categories rather than misclassifying it.

Chen and Dumais (2000) and Dumais, Cutrell, and Chen (2001) studied how best to use the automatically derived category information to organize search results. Fig. 5 shows the two basic interfaces that were used to present search results. In the *Category* interface (left panel), search results was organized into hierarchical categories. The best matching web pages within each category were shown initially, and additional pages could be seen on demand by category expansion. In order to show both category context and individual search results in limited screen space, only the title of each page is shown initially. The summary of each page is available as hover text—if users move their cursor over the page title the summary appears. The *List* interface (right panel) is similar to current systems for displaying web search results in that results are shown in a long list. For comparability to the Category condition, only titles are shown initially with summaries available on demand as hover text. Chen and Dumais found large and reliable advantages for the Category interface in both objective and subjective measures. Participants preferred the Category interface (6.4 vs. 4.2 on a 7-point rating scale), and they were 34% faster at finding relevant information (56 s/task vs. 85 s/task median search times). Chen and Dumais' interpretation of the finding was that grouping of results into categories allowed users to ignore many results and quickly focus in on the subset of results of interest.

In order to better understand this category advantage, Dumais et al. (2001) developed and evaluated several new interfaces for presenting search results. They explored two methods for adding contextual information to the List interface. The first approach presented summaries inline instead of as hover text. The second approach added category names to the inline summaries in the List interface. This second condition looks very much like results presentation in popular web search systems in which results are shown in a ranked list with each entry containing a title, a short summary, and some additional information such as URL or category name.

They also explored methods for removing aspects of the context from the Category interface. The first approach removed the category names (e.g., Computer & Internet) but the results were still grouped by category. With this technique, the visual grouping of objects is maintained, but the richness and directness of the category representation is reduced because the category names are removed. The second approach removed the page titles, leaving only category names. This is essentially a top-down browsing interface. A final condition in which summaries were shown inline was added for comparability with the list conditions.

Three key findings emerged from a series of experiments using seven different interfaces.

(1) Category interfaces were faster than List interfaces, in all cases. This was true even when category names and inline summaries were added to the Lists, and when the Category organization was degraded by removing category names or page titles. Interestingly, the List interface augmented with category names contains exactly the same information as the Category interface, but search times and user preferences favor the Category presentation. Grouping results from the same category together visually appears to be the key. Having metadata in the form of category labels is useful, only if presented in the right way.
(2) The best performance in the Category interfaces was achieved when both category names and page titles were available (Fig. 5). Either alone worked better than any of the list presentations, but the combination of specific results in the category context was the most effective for allowing users to quickly analyze search results.
(3) Inline summaries were more effective than summaries presented as hover text for both the List and Category interfaces. This is somewhat surprising since more scrolling is required, but apparently the cognitive costs of deciding which title to examine in more detail and the physical costs of moving the cursor over the title outweigh the additional scrolling required.

This work shows how automatic text classification techniques can be used to extend the reach of existing directories and provide users with nicely organized presentations of search results. In addition, these results provide interface design guidelines for effectively combining category context with search results. The best overall presentation condition groups results by category (using the automatically derived category metadata), provides a short descriptive title for each category, and shows both the title and description inline. As was the case for information retrieval, good solutions to practical problems (both core text classification algorithms and uses in search interfaces) can be achieved by using simple techniques on large amounts of data.

3.4. Text categorization and human memory

The applied work on text classification shows that simple textual representations in high dimensional space (either the full feature space or a reduced dimension space as with LSA) in combination with inductive learning techniques can be used to solve a variety of text classification problems. The automatic classification techniques accurately mimic human judgments of category membership for a wide range of categorical distinctions such as topical categories, news categories, medical subject headings, email folders, etc. These categories are not the natural kinds typically studied in psychological experiments. In addition, the categories are much richer and less carefully controlled than those typically used in experiments on human memory, although there are certainly some exceptions (e.g., Ross & Murphy, 1999).

A particularly interesting application from the cognitive perspective is the use of text classification techniques for scoring free-form essays (Landauer et al., 2000; Larkey, 1998). Here the categories are the grades to assigned to essays, and the task is to grade a new essay. Unlike much of the work on text classification in which the categories are defined by topical similarity,

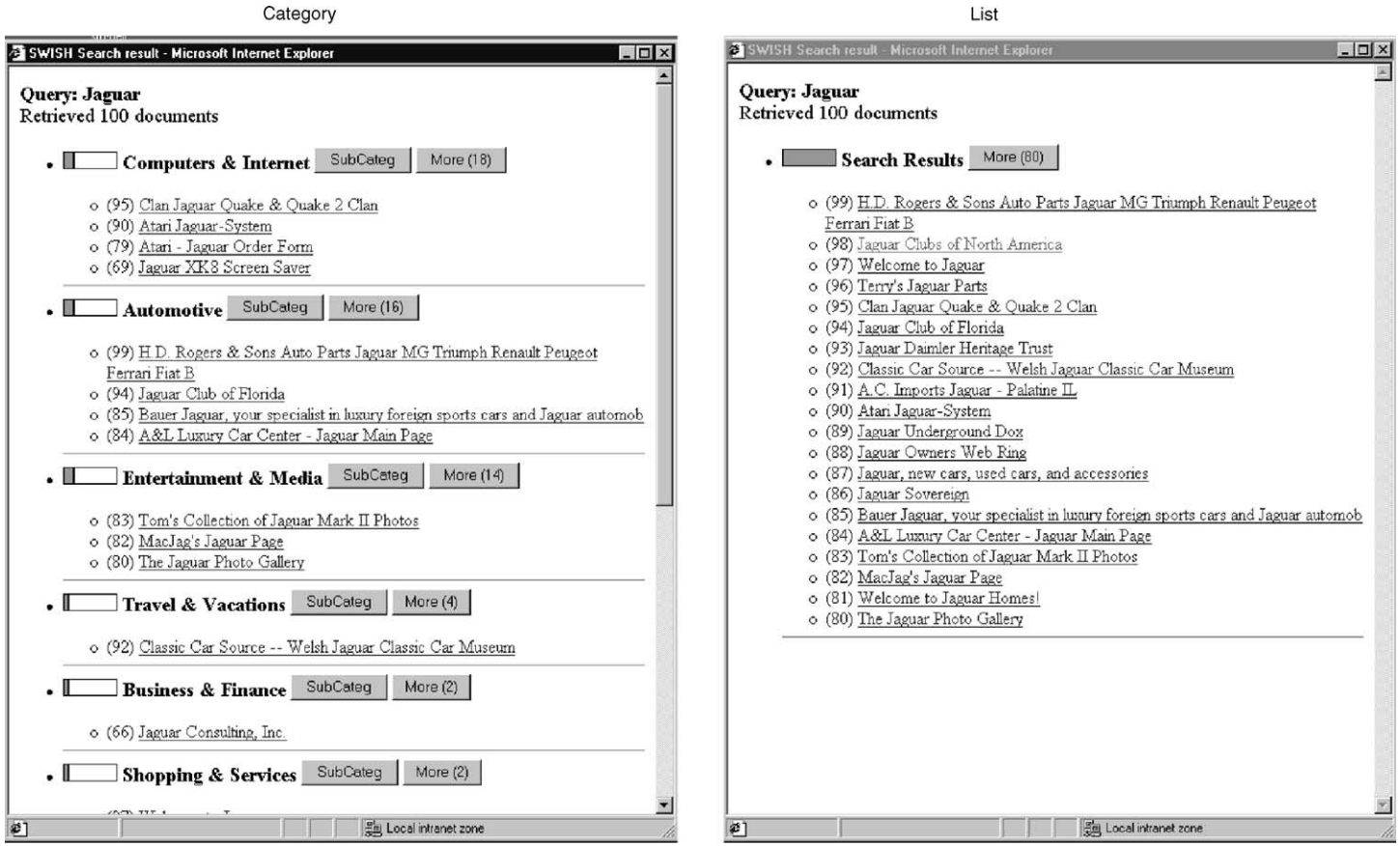

Fig. 5. Category and List interfaces used by Chen and Dumais (2000). The Category interface shows search results grouped by category. The List interface shows results in a single list, ordered by the score assigned by the search engine.

the categories here are defined by quality as well as topical characteristics. Both an LSA-based representation with *k*-nearest neighbor matching (Landauer et al., 2000) and a word-based representation with *k*-nearest neighbor or Naïve Bayes matching have been successful in predicting essay scores.

Other research has used text analysis techniques to model category effects such as typicality and the representation of proper names and common nouns. Laham (1997) explored the use of LSA to model of human category judgments of various kinds. In one experiment, he examined the LSA-based similarity of category members to 15 superordinate category names (e.g., the similarity of apple to fruit vs. flower vs. mammal, etc.). Percent correct was high for animate and inanimate natural kinds like flowers, mammals, fruits and gemstones (92–100%) and lower but still well above chance for man-made artifacts like furniture, vehicles, weapons, tools, toys and clothing (53%). In addition, for natural categories the correlation between the LSA similarity between the concept and superordinate name or most typical member showed high correlations with human judgments for natural kind categories (e.g., $r^2 = .82$ for fruit) and near-zero correlations for artifact categories. Similar effects have been observed in patients with some clinical disnomias.

Burgess and Conley (1998) examined how proper names and common nouns were represented in their high-dimensional HAL model. Unlike many lines of research that focus on the dissociations between proper names and other objects (e.g., Burton & Bruce, 1993; Cohen & Burke, 1993), Burgess and Conley represented both names and nouns in the same semantic HAL space. They then used the inter-item similarities in this space as input to multi-dimensional scaling analyses, and explored similarity neighborhoods in the resulting MDS space. In the first experiment, they showed that proper names grouped together and were separated from common nouns and verbs. In a second experiment they showed that different types of proper nouns (people, cities, states) grouped together in distinct regions of the space. And, in a third experiment they explored the semantic neighborhoods of objects and proper names. Semantic neighborhoods are operationally defined as the nearest neighbors of target words. Common name neighborhoods contained other names (e.g., neighbors of John were David, Peter, Paul, Mike, etc.), whereas neighborhoods of nouns contained a rich set of semantically related words (e.g., neighbors of book included story, game, movie, new, file, etc.). Thus, an error in retrieval of a common name results in the retrieval of another name, whereas an error in the retrieval of a noun results in semantically related words. In addition to the difference in the kinds of neighbors, proper names had higher neighborhood densities (more near neighbors) than frequency-matched common nouns. Burgess and Conley posit that these two differences (rather than differences in representation *per se*) account for many of the observed dissociations between proper names and common nouns. Although the same representation and processes are used, characteristics of real world co-occurrence patterns result in differences that correlate with human performance. Computational approaches are also able to identify different types of proper nouns with high accuracy (e.g., Paik, Liddy, Yu, & McKenna, 1996).

Even without explicit discriminative training, high-dimensional meaning spaces appear capable of providing the basic contextual information for categorization of a wide range of objects. It would be interesting to explore the ability of discriminative techniques like SVM to model aspects of human category judgments. The output of an SVM is a graded similarity score, so typicality effects could follow naturally. The objects with the highest SVM scores will

be those that lie parallel to the learned weight vector, $\vec{w}$. The trick will be to identify objects that can serve as training examples since they need to be explicitly labeled for discriminative training.

3.5. Summary of text categorization

Text categorization is the assignment of one or more pre-defined category labels to natural language texts. Text categorization is used in many practical information management tasks ranging from assigning topical tags to content like news or web pages, to identifying spam email.

Vector retrieval techniques, including LSA, have been used for text classification tasks. Representative texts are analyzed to create a vector space, categories are represented by means of examples or prototypes, similarity is computed between new examples and the category representation, and some similarity threshold is used to make category decisions. Similar models have been used to model some effects observed in human memory research, including typicality and proper noun effects.

More recently, discriminative learning techniques such as support vector machines have been used with excellent generalization accuracy for practical text classification problems. SVMs are able to mimic human assignments of topical category labels with high accuracy. Furthermore, there has been no work on modeling the kinds of natural categories and category effects observed in human memory research, and this should be an interesting direction for future research. The SVMs model identifies a decision surface that maximizes the margin between the closest examples from two categories (see Fig. 4). The extent to which models like this can account for typicality and other effects seen in human category judgments is an interesting research direction.

Another interesting line of research would be to directly compare humans and the inductive learning algorithms on the same learning task. The learning algorithms have no access to the order of the words, the organization of the words in a document, or any images of formatting information in a document. It would be interesting to see how humans do using the same representation. It would not be surprising if the models do not completely agree with human classifiers, but the nature of the differences would be interesting to explore. Empirical research on difference between human and computer models should shed light on the cognitive processes and strategies that humans bring to bear on this task.

4. Question answering

4.1. Introduction

Most text retrieval and text classification systems operate at the level of entire documents. In searching the web, web pages or documents are returned. There has been a recent surge of interest in finer grained analyses focused on obtaining answers rather than entire documents. The goal of a question answering system is to retrieve *answers* to questions rather than full documents or best-matching passages as most information retrieval systems currently

do. The problem of question answering involves aspects of information retrieval, information extraction, machine learning, and natural language processing (see recent workshops on the topic: AAAI, 2002; ACL-EACL, 2002; Voorhees & Harman, 2000, 2001). The TREC Question Answering Track has motivated much of the recent work in the field. The initial efforts in question answering are focused on fact-based, short-answer questions such as "*Who killed Abraham Lincoln?*" or "*How tall is Mount Everest?*"

Automatic question answering from a single, small information source is extremely challenging. Given a source that contains only a small number of formulations of answers to a user's question, a computational system is faced with the difficult task of mapping questions to answers by way of uncovering complex lexical, syntactic, or semantic relationships between questions and answer strings. The need for anaphor resolution, synonymy, even semantic inference, the presence of alternate syntactic formulations, and indirect answers all make answer finding a challenging task. Consider the difficulty of gleaning an answer to the question "*Who killed Abraham Lincoln?*" from a source which contains only the text "*John Wilkes Booth altered history with a bullet. He will forever be known as the man who ended Abraham Lincoln's life.*"

Most approaches to question answering use a combination of information retrieval and natural language processing techniques. Systems typically find some candidate passages using standard information retrieval techniques, and then do more detailed linguistic analyses of both the question and passages to find specific answers. A variety of linguistic resources (part-of-speech tagging, parsing, named entity extraction, semantic relations, dictionaries, WordNet, etc.) are used to support question answering (e.g., Pasca & Harabagiu, 2001; Hovy, Gerber Hermjakob, Junk, & Lin, 2000; Prager, Brown, Coden, & Radev, 2000). The FALCON system by Pasca and Harabagiu is one that is typical of the linguistic approaches and has excellent performance in benchmark tests. The query is parsed to identify the important entities and to suggest a likely answer type. They have developed a rich taxonomy of answer types using lexico-semantic resources like WordNet (Miller, 1995). WordNet encodes more than 100,000 English nouns, verbs, adjectives and adverbs into conceptual synonym sets; this work was done by lexicographers over the course of many years. Candidate matching paragraphs are similarly analyzed to see if they match the expected answer type. Often, relevant passages will not share words with the query, as in the example above. In these cases, their system uses WordNet to examine morphological alternatives, lexical alternatives (e.g., nouns killer or assassin or slayer will match the verb killed), and semantic alternatives (e.g., cause the death of). Additional processes are required to hone in on the best matching sentences or phrases.

In contrast to these rich semantic approaches, we have been working on a question answering system that attempts to solve the difficult matching and extraction problems not by using rich natural language analysis but rather by using large amounts of data. The system, called AskMSR, is described in more detail in the next section and in Brill, Lin, Banko, Dumais, and Ng (2001) and Dumais, Banko, Brill, Lin, and Ng (2002).

4.2. AskMSR data-driven approach to question answering

The data-driven approach of the AskMSR question answering system is motivated by recent observations in natural language processing that, for many applications, significant

improvements in accuracy can be attained simply by increasing the amount of data used for learning. Following the same guiding principle the tremendous data resource that the Web provides is used as the backbone of the question answering system. The system contains four main components (see Dumais et al., 2002 for details).

1. *Rewrite query*. Given a question, the system generates a number of rewrite strings, which are likely substrings of declarative answers to the question. There are fewer than 10 rewrite types, which vary from specific string matching to a simple ANDing of all the query words. For the query "*Who killed Abraham Lincoln?*" there are three rewrites: <LEFT> killed Abraham Lincoln; Abraham Lincoln was killed by <RIGHT>; and who AND killed AND Abraham AND Lincoln. The first two rewrites require that a text match the exact string, such as "killed Abraham Lincoln." The last rewrite it a backoff strategy that simply ANDs together all the query words. The rewrite strings are then formulated as search engine queries and sent to a search engine from which page summaries are collected. Any back end search engine can be used. Matches to more precise rewrites are given higher scores than those to the backoff AND rewrite.
2. *Mine n-grams*. From the page summaries returned for each rewrite, all unigrams, bigrams and trigram word sequences are extracted. The *n-gra*ms are scored according to their frequency of occurrence and the weight of the query rewrite that retrieved it. The system picks up what are in effect common associates of the query strings. The common *n-gra*ms for this example query are: Booth, Wilkes, Wilkes Booth, John Wilkes Booth, bullet, actor, president, Ford's, Gettysburg Address, derringer, assignation, etc.
3. *Filter n-grams*. The *n-gra*ms are filtered and re-weighted according to how well each candidate matches the expected answer-type, as specified by a handful of handwritten filters. Fifteen filters were developed based on human knowledge about question types. These filters use surface-level string features, such as capitalization or the presence of digits. For example, for *When* or *How many* questions, answer strings with numbers are given higher weight, and for *Who* questions, answer strings with capitals are given added weight and those with dates are demoted.
4. *Tile n-grams*. Finally, the *n-gra*ms are tiled together where appropriate, so that longer answers can be assembled from shorter ones. After tiling the answers to the example query are: John Wilkes Booth, bullet, president, actor, Ford. John Wilkes Booth receives a much higher score than the others because it is found in specific rewrites and because it occurs often.

The tremendous redundancy that a large data repository provides is used in two ways in this system. First, the greater the answer redundancy in the text collection, the more likely it is that an answer occurs in a very simple relation to the question. Therefore, the need to handle tough problems like anaphor resolution, synonymy, and the presence of alternate syntactic formulations is greatly reduced. On the web, hundreds of pages contain transparent answer strings like "*John Wilkes Booth killed Abraham Lincoln.*" and finding the answer in this string is easy. The second use of redundancy is in answer mining. Instead of looking at just one or two of the most likely passages, we consider hundreds of matching passages looking for consistently occurring strings. Others have also proposed using the Web to aid in question

answering (Clarke, Cormack, & Lyman, 2001; Kwok, Etzioni, & Weld, 2001), but even these systems do much more sophisticated parsing and indexing than we do.

Redundancy *per se* has been shown to be important in two experiments (Dumais et al., 2002). The experiments use 500 short answer questions from the TREC evaluations (Voorhees & Harman, 2001). The questions were taken from Web query logs and represent a range of question types—e.g., *Who invented the paper clip? Who was the first Russian astronaut to do a space walk? When was Babe Ruth born? What is the life expectancy of an elephant? Where are the headquarters of Eli Lily? How big is the Electoral College? What is the longest word in the English language?* The first experiment varies the number of best-matching passages which are examined for *n-gra*m mining. Accuracy improves from 31 to 62% as the number of passages examined increases from 1 to 200. This performance exceeds that of many more complex systems and is on par with the best systems. The second experiment compares performance using two different collections both known to contain answers to the questions. The collections vary in size by three orders of magnitude; one contains 1 million documents (TREC) and the other 2 billion documents (Web). Accuracy improves from 35% for the small collection to 55% for the large collection. Collection size influences both the ability to find simple rewrites that match and to mine answers from results.

This data-driven approach to question answering is complimentary to more linguistic approaches, but we have chosen to see how far we can get initially by focusing on data *per se* as a key resource for question answering. In benchmark TREC tasks, the algorithmically simple but data rich system performs quite well compared to much more linguistically sophisticated systems (see Brill et al., 2001 for details). These experiments demonstrate that data *per se* can be a valuable resource for question answering engines. An important research direction for question answering is to extend the systems to more than short-answer, fact-based questions.

4.3. Question answering and human memory

Simple techniques operating over large amounts of data provide surprisingly good question answering performance. Clearly, humans have not read the same quantity of information that is available on the web, but human memory is also very rich in data. Human memory consists of many episodic memories, e.g., how many times have you heard about, or read about, or visited the scene of Lincoln's assassination?

There has been some work on the role that question asking play in development and education (e.g., Graesser & Black, 1985), but there is less work on question answering *per se*. It would be interested to see the extent to which a system like AskMSR can mimic human question answering abilities. One simple task would be to evaluate the extent to which questions that are difficult for people also difficult for AskMSR? In AskMSR, question difficulty can be measured by the magnitude of the score for the first answer or by the distribution of scores across answers. For humans, one could look at question answering time or accuracy averaged over many people. Presumably some normative evaluation like this is used for assigning the dollar values to questions on the show *Who wants to be a millionaire*.

Another research direction would be to replace the literal string matching approach used in AskMSR with an LSA-based representation in which surface level differences are replaced with a more robust representation.

4.4. Summary of question answering

The goal of a question answering system is to retrieve *answers* to questions rather than full documents or best-matching passages as most information retrieval systems currently do. Recent research in question answering systems has looked at systems for answering fact-based, short-answer questions such as "*Who killed Abraham Lincoln?*" or "*How tall is Mount Everest?*" While many systems use sophisticated natural language technologies to tackle this problem, some researchers have used much simpler techniques (Brill et al., 2001; Clarke et al., 2001; Dumais et al., 2002). One particular system, AskMSR, was described in detail. This system uses simple string matching and data mining techniques applied to large amounts of data, and has been quite successful in benchmark evaluations. The success of this data-driven approach suggests that similar techniques might be useful in modeling human question answering.

5. Summary

We surveyed three practical problems involving the retrieval of textual information from external sources—information retrieval, text categorization, and question answering. In all three cases, the analysis of the statistical properties of words in large volumes of real world texts formed the basis of practical solutions to the problems. Simple statistical analyses operating over large volumes of representative texts are used to solve practical information access problems. In the case of information retrieval, a technique from linear algebra was used to analyze the contexts in which words occur and uncover latent relationships among words. This LSA analysis improved information retrieval performance in several problems. In text classification, a simple machine learning technique was used to separate classes with maximum margin. This SVM technique consistently outperformed other approaches. Finally, in question answering problems, a simple statistical approach that uses string matching and mining applied to large amounts of data is quite successful in a difficult task.

The same statistical properties of objects that underlie the operational systems also constrain human performance. The same systems that have been used to solve practical information access problems might also be used to simulate human performance. This relationship has been explored extensively in the case of LSA. LSA has been used successfully to model vocabulary acquisition, the rate of vocabulary learning, semantic priming effects, textual coherence, the quality of student essays, the retrieval component of analogies, and prose recall. Although LSA has been successfully used in these many applications, there are several directions that require further exploration. On the computational front, the analysis of large corpora is still computationally difficult and the LSA analysis is not easy to update incrementally. More psychologically, there is currently no way to integrate the vast amounts of spoken information that humans are exposed to or to combine information from perception and language. Extensions of LSA to other psychological effects, and comparisons to other model such as the probabilistic variant proposed by Griffiths and Steyvers (2002) are interesting directions for future research.

There has also been some use of algorithms from text classification for modeling aspects of human category judgments, including typicality and proper noun effects. To date, SVMs have

been used to mimic human assignments of category labels, but not to look more closely at human categorization. Studying the extent to which models like this can account for typicality and other effects (or fail to do so) is an interesting research direction. Working with natural kind categories and directly comparing humans and inductive learning algorithms on the same tasks should be informative. The area of question answering is much newer, and successful open-domain systems have only recently been developed. An interesting initial direction would be to evaluate the extent to which questions that are difficult for people also difficult for question answering systems.

From both engineering and scientific perspectives there are reasons to design learning algorithms that can acquire human-like quantities of human-like knowledge from the same sources as humans. Computer algorithms for information access tasks like information retrieval, text categorization and question answering offer solutions to practical problems, and may offer theoretical insights about human knowledge representation as well as. Empirical research on differences between humans and computer models can shed light on the cognitive processes and strategies that humans bring to bear on practical information access tasks.

Notes

1. Latent Semantic Indexing (LSI) was the term used initially to describe the application of dimension reduction techniques to the problem of information retrieval. Latent semantic analysis (LSA) was used later to describe the application of the same technique to more general problems. We use the more general term in this article since we describe both information retrieval and memory modeling applications.
2. The TOEFL test words did not all occur sufficiently often. Thus the actual number of direct exposures was 2, 3.8, 7.4, 12.8, and 22.2.

Acknowledgments

The work described here was carried out over the last 15 years and supported by Bell Laboratories, Bellcore (now Telcordia) and Microsoft Research. Several collaborators have been involved in all aspects of the work, most notably Tom Landauer, Peter Foltz, George Furnas, Michael Littman (for LSA), John Platt, David Heckerman, Eric Horvitz, Hao Chen, Edward Cutrell (for text classification), Eric Brill, and Michele Banko (for question answering). I would also like to thank my graduate advisor, Rich Shiffrin, who provided numerous insights along the way and continues to be a source of challenging questions.

References

AAAI. (2002). *AAAI Spring Symposium Series Mining answers from text and knowledge bases*.

ACL-EACL. (2002). *ACL-EACL Workshop on Open-domain question answering*.

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: Erlbaum.

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

Ashby, F. G. (1992). *Multidimensional models of perception and cognition*. Hillsdale, NJ: Erlbaum.

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

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

Brill, E., Lin, J., Banko, M., Dumais, S., & Ng, A. (2001). Data-intensive question answering. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)* (pp. 393–400).

Britton, B. K., & Gulgoz, S. (1991). Using Kintsch's computational model to improve instructional text: Effects of repairing inference calls on recall and cognitive structures. *Journal of Educational Psychology*, *83*, 329–345.

Bush, V. (1945). As we may think. *Atlantic Monthly*, *176*(1), 101–108.

Burgess, C. (1998). From simple associations to building blocks of language: Modeling meaning in memory with the HAL model. *Behavior Research Methods, Instruments, and Computers*, *30*(2), 188–198.

Burgess, C., & Conley, P. (1998). Developing semantic representations for proper names. In *Proceedings of the Cognitive Science Society* (pp. 185–190).

Burgess, C., Livesay, K., & Lund, K. (1998). Explorations in context space: Words, sentences and discourse. *Discourse Processes*, *25*(2&3), 211–257.

Burton, A. M., & Bruce, V. (1993). Naming faces and naming names: Exploring an interactive activation model of person recognition. *Memory*, *1*, 457–480.

Chakrabarti, S., Dom, B., Agrawal, R., & Raghavan, P. (1998). Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies. *The VLDB Journal*, *7*, 163–178.

Chen, H., & Dumais, S. T. (2000). Bringing order to the Web: Automatically categorizing search results. In *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems* (pp. 145–152).

Chiarello, C., Burgess, C., Richards, L., & Pollock, A. (1990). Semantic and associative priming in the cerebral hemispheres: Some words do, some words don't . . . sometimes, some places. *Brain and Language*, *35*, 75–104.

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

Clarke, C., Cormack, G., & Lyman, T. (2001). Exploiting redundancy in question answering. In *Proceedings of SIGIR'2001, 24th ACM International Conference on Research and Development in Information Retrieval* (pp. 358–365).

Cohen, G., & Burke, D. M. (1993). Memory for proper names: A review. *Memory*, *1*, 249–263.

Cortes, C., & Vapnik, V. (1995). Support vector networks. *Machine Learning*, *20*, 273–297.

Deese. J. (1965). *The structure of associations in language and thought*. Baltimore, MD: Johns Hopkins University Press.

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.

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).

Dumais, S. T., Banko, M., Brill, E., Lin, J., & Ng, A. (2002). Web question answering: Is more always better. In *Proceedings of SIGIR'2002, 25th ACM International Conference on Research and Development in Information Retrieval* (pp. 291–298).

Dumais, S. T., & Chen, H. (2000). Hierarchical classification of web content. In *Proceedings of SIGIR'2000, 23rd ACM International Conference on Research and Development in Information Retrieval* (pp. 256–263).

Dumais, S. T., Cutrell, E., & Chen, H. (2001). Optimizing search by showing results in context. In *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems* (pp. 277–283).

Dumais, S., Platt, J., Heckerman, D., & Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In *Proceedings of CIKM'98, 7th ACM International Conference on Information and Knowledge Management* (pp. 148–155).

Dunn, J. C., Osvaldo, P. A., Barclay, L., Waterreus, A., & Flicker, L. (2002). Latent semantic analysis: A new method to measure prose recall. *Journal of Clinical and Experimental Neuropsychology*, *24*(1), 26–35.

Fidel, R. (1985). Individual variability in online searching behavior. In *Proceedings of the ASIS 48th Annual Meeting* (pp. 69–72).

Fillenbaum, S., & Rapoport, A. (1971). *Structures in the subjective lexicon*. New York: Academic Press.

Foltz, P. W. (1996). Latent semantic analysis for text-based research. *Behavior Research Methods, Instruments and Computers*, *28*(2), 197–202.

Foltz, P. W., Britt, M. A., & Perfetti, C. A. (1996). Reasoning from multiple texts: An automatic analysis of readers' situation models. In G. W. Cottrell (Ed.), *Proceedings of the 18th Annual Cognitive Science Conference* (pp. 110–115).

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

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.

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.

Goldstone, R., & Kersten, A. (in press). Concepts and categorization. In A. F. Healy & R. W. Proctor (Eds.), *Comprehensive handbook of psychology, Volume 4: Experimental psychology*. New York: Wiley.

Graesser, A. C., & Black, J. B. (Eds.). (1985). *The psychology of questions*. Hillsdale, NJ: Lawrence Erlbaum Associates.

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

Harman, D. (1991). How effective is suffixing? *Journal of the American Society for Information Science*, *42*(1), 7–15.

Hayes, P. J., Andersen, P. M., Nirenburg, I. B., & Schmandt, L. M. (1990). TCS: A shell for content-based text categorization. In *Proceedings of CAIA-90, 6th IEEE Conference on Artificial Intelligence Applications* (pp. 320–326).

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

Hovy, E., Gerber, L., Hermjakob, U., Junk, M., & Lin, C. (2000). Question answering in Webclopedia. In *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of ECML'98, 10th European Conference on Machine Learning* (pp. 137–142).

Joachims, T. (1999). Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, & A. Smola (Eds.), *Advances in Kernel methods—Support vector learning*. Cambridge, MA: MIT Press.

Joachims, T. (2002). *Learning to classify text using support vector machines*. Dordrecht: Kluwer Academic Publishers.

Jones, W. P. (1986). On the applied use of human memory models: The memory extender personal filing system. *International Journal of Man-Machine Studies*, *25*, 191–228.

Kaufmann, L., & Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis*. New York: Wiley.

Kwok, C., Etzioni, O., & Weld, D. (2001). Scaling question answering to the Web. In *Proceedings of WWW'10*.

Laham, D. (1997). Latent semantic analysis approaches to categorization. In M. G. Shafto & P. Langley (Eds.), *Proceedings of the 19th Annual Meeting of the Cognitive Science Society* (p. 979).

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

Landauer, T. K. (2002). On the computational basis of learning and cognition: Arguments from LSA. In N. Ross (Ed.), *The psychology of learning and motivation* (Vol. 41, pp. 43–84).

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*. Hillsdale, NJ: Lawrence Erlbaum Associates.

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., & Foltz, P. W. (1998). Learning human-like knowledge by singular-value decomposition: A progress report. In M. I. Jordon, M. J. Kearns, & S. A. Solla (Eds.), *Advances in neural information processing systems* (Vol. 10, pp. 45–51).

Landauer, T. K., Laham, D., & Foltz, P. W. (2000). The intelligent essay assessor. *IEEE Intelligent Systems*, *15*(5), 27–31.

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. In M. G. Shafto & P. Langley (Eds.), *Proceedings of the 19th Annual Meeting of the Cognitive Science Society* (pp. 412–417).

Larkey, L. S. (1998). Automatic essay grading using text categorization techniques. In *Proceedings of the 21st Annual International Conference on Research and Development in Information Retrieval (SIGIR 98)* (pp. 90–95).

Lewis, D. D. (2001). Applying support vector machines to the TREC-2001 batch filtering and routing tasks. In E. M. Voorhees & D. K. Harman (Eds.), *The Tenth Text REtrieval Conference (TREC-2001)* (pp. 286–292).

Littman, M. L., Dumais, S. T., & Landauer, T. K. (1998). Automatic cross-language information retrieval using latent semantic indexing. In G. Grefenstette (Ed.), *Cross-language information retrieval*. Dordrecht: Kluwer Academic Publishers.

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

McKeown, M. G., Beck, I. L., Sinatra, G. M., & Loxterman, J. A. (1992). The contribution of prior knowledge and coherent text to comprehension. *Reading Research Quarterly*, *27*, 79–93.

McNamara, D. S., Kintsch, E., Songer, N. B., & Kintsch, W. (1996). Are good texts always better? Text coherence, background knowledge, and levels of understanding in learning from text. *Cognition and Instruction*, *14*, 1–43.

Miller, G. A. (1995). WordNet: A lexical database. *Communication of the ACM*, *38*(11), 39–41. WordNet is available online at: http://www.cogsci.princeton.edu/∼wn/.

Osgood, C. E., Suci, G. J., & Tannenbaum, P. H. (1957). *The measurement of meaning*. Champaign, IL: University of Illinois Press.

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

Paik, W., Liddy, E., D., Yu, E., & McKenna, M. (1996). Categorizing and standardizing proper nouns for efficient information retrieval. In B. Bogureav & J. Pustejovsky (Eds.), *Corpus processing for lexical acquisition*. Cambridge, MA: MIT Press.

Pasca, M. A., & Harabagiu, S. M. (2001). High-performance question/answering. In *Proceedings of SIGIR'2001, 24th ACM International Conference on Research and Development in Information Retrieval* (pp. 366–374).

Platt, J. (1998). Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. Burges, & A. Smola (Eds.), *Advances in Kernel methods—Support vector learning*. Cambridge, MA: MIT Press.

Prager, J., Brown, E., Coden A., & Radev, D. (2000). Question answering by predictive annotation. In *Proceedings of SIGIR'2000, 23rd ACM International Conference on Research and Development in Information Retrieval* (pp. 184–191).

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

Reed, S. K. (1972). Pattern recognition and categorization. *Cognitive Psychology*, *3*, 382–407.

Ross, B. H., & Murphy, G. L. (1999). Food for thought: Cross-classification and category organization in a complex real-world domain. *Cognitive Psychology*, *38*, 495–553.

Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A Bayesian approach to filtering junk e-mail. In *Proceedings of AAAI'98 Workshop on Learning for Text Categorization*.

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

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

Schütze, H., Hull, D. A., & Pedersen, J. O. (1995). A comparison of classifiers and document representations for the routing problem. In *Proceedings of SIGIR'95, 18th ACM International Conference on Research and Development in Information Retrieval* (pp. 229–237).

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Survey*, *34*(1), 1–47.

Stata, R., Bharat, K., & Maghhout, F. (2000). The term vector database: Fast access to indexing terms for web pages. In *Proceedings of WWW9* (pp. 247–256).

Swets, J. A. (1963). Information retrieval systems. *Science*, *141*(3577), 245–250.

Tarr, D., & Borko, H. (1974). Factors influencing inter-indexer consistency. In *Proceedings of the ASIS 37th Annual Meeting* (pp. 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.

Turney, P. D. (2001). Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. *Lecture Notes in Computer Science*, *2167*, 491–502.

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

Vapnik, V. (1995). *The nature of statistical learning theory*. New York: Springer.

Voorhees, E., & Harman, D. (Eds.). (2000). *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*. NIST Special Publication 500-249.

Voorhees, E., & Harman, D. (Eds.). (2001). *Proceedings of the Tenth Text REtrieval Conference (TREC-2001)*. NIST Special Publication 500-250.

Willett, P. (1988). Recent trends in hierarchic document clustering: A critical review. *Information Processing and Management*, *24*(5), 577–598.

Wolfe, M. B., Schreiner, M. E., Rehder, B., Laham, D., Foltz, P., Kintsch, W., & Landauer, T. K. (1998). Learning from text: Matching readers and texts by latent semantic analysis. *Discourse Processes*, *25*, 309–336.

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'99* (pp. 42–49).

Yang, Y., Slattery, S., & Ghani, R. (2002). A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, *18*(2/3), 219–241.

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