CHAPTER 6

Web Mining: Machine Learning for Web Applications

Hsinchun Chen and Michael Chau
University of Arizona

Introduction

With more than two billion pages created by millions of Web page authors and organizations, the World Wide Web is a tremendously rich knowledge base. The knowledge comes not only from the content of the pages themselves, but also from the unique characteristics of the Web, such as its hyperlink structure and its diversity of content and languages. Analysis of these characteristics often reveals interesting patterns and new knowledge. Such knowledge can be used to improve users' efficiency and effectiveness in searching for information on the Web, and also for applications unrelated to the Web, such as support for decision making or business management.

The Web's size and its unstructured and dynamic content, as well as its multilingual nature, make the extraction of useful knowledge a challenging research problem. Furthermore, the Web generates a large amount of data in other formats that contain valuable information. For example, Web server logs' information about user access patterns can be used for information personalization or improving Web page design.

Machine learning techniques represent one possible approach to addressing the problem. Artificial intelligence and machine learning techniques have been applied in many important applications in both

scientific and business domains, and data mining research has become a significant subfield in this area. Machine learning techniques also have been used in information retrieval (IR) and text mining applications. The various activities and efforts in this area are referred to as *Web mining*. The term Web mining was coined by Etzioni (1996) to denote the use of data mining techniques to automatically discover Web documents and services, extract information from Web resources, and uncover general patterns on the Web. Over the years, Web mining research has been extended to cover the use of data mining and similar techniques to discover resources, patterns, and knowledge from the Web and Web-related data (such as Web usage data or Web server logs). In this chapter, we have adopted a broad definition that considers Web mining to be "the discovery and analysis of useful information from the World Wide Web" (Cooley, Mobasher, & Srivastava, 1997, p. 558).

Web mining research overlaps substantially with other areas, including data mining, text mining, information retrieval, and Web retrieval. A possible classification of research in these areas is shown in Table 6.1. The classification is based on two aspects: the purpose and the data sources. *Retrieval* research focuses on retrieving relevant, existing data or documents from a large database or document repository, while *mining* research focuses on discovering new information or knowledge in the data. For example, data retrieval techniques are mainly concerned with improving the speed of retrieving data from a database, whereas data mining techniques analyze the data and try to identify interesting patterns. It should be noted, however, that the distinction between information retrieval and text mining is not clear. Many applications, such as text classification and text clustering, are often considered both information retrieval and text mining (e.g., Voorhees & Harman, 1998; Trybula, 1999). In fact, almost all text mining techniques have been investigated by the information retrieval community, notably the Text REtrieval Conference (TREC). Because information retrieval research has the primary goals of indexing and searching, we consider areas such as document clustering to be an instance of text mining techniques that is also part of the retrieval process. Similarly, Web retrieval and Web mining share many similarities. Web document clustering has been studied both in the context of Web retrieval and of Web mining. On the other hand, however, Web mining is not simply the application of information

Table 6.1 A classification of retrieval and mining techniques and applications

		Data/information sources		
		Any data	Textual data	Web-related data
Purpose	Retrieving known data or documents efficiently and effectively	Data Retrieval	Information Retrieval	Web Retrieval
	Finding new patterns or knowledge previously unknown	Data Mining	Text Mining	Web Mining

retrieval and text mining techniques to Web pages; it also involves non-textual data such as Web server logs and other transaction-based data. From this point of view, Web retrieval and Web mining are considered overlapping areas, in which the main criterion for classification is the specific purpose of the application.

It is also interesting to note that, although Web mining relies heavily on data mining and text mining techniques, not all techniques applied to Web mining are based on data mining or text mining. Some techniques, such as Web link structure analysis, are unique to Web mining. In general, it is reasonable to consider Web mining as a subfield of data mining, but not a subfield of text mining, because some Web data are not textual (e.g., Web log data).

As can be seen, Web mining research is at the intersection of several established research areas, including information retrieval, Web retrieval, machine learning, databases, data mining, and text mining. Most previous research has viewed Web mining from a database or data mining perspective (e.g., Chakrabarti, 2000; Cooley et al., 1997; Han & Chang, 2002). On the other hand, research in machine learning and information retrieval has also played a very important role in Web mining research. Machine learning is the basis for most data mining and text mining techniques, and information retrieval research has largely influenced the research directions of Web mining applications. In this chapter, we review the field from the perspectives of machine learning and information retrieval. The review emphasizes machine learning and traditional information retrieval techniques and how they have been applied in Web mining systems.

We begin with an overview of machine learning research and different paradigms in the field. We also review some methods commonly used for evaluating machine learning systems. The next section describes how machine learning algorithms were used in traditional information retrieval systems in the "pre-Web" era. We then review the field of Web mining and discuss how machine learning has been used in different Web mining applications. In the last section we conclude our review and suggest some future research directions.

Machine Learning: An Overview

Since the invention of the first computer in the 1940s, researchers have been attempting to create knowledgeable, educable, and intelligent computers. Many knowledge-based systems have been built for applications such as medical diagnosis, engineering troubleshooting, and business decision making (Hayes-Roth & Jacobstein, 1994). However, most of these systems have been designed to acquire knowledge manually from human experts, which can be both time-consuming and labor intensive. Machine learning algorithms have been developed to alleviate these problems by acquiring knowledge automatically from examples or source data. Simon (1983) emphasizes that machine learning is *any* process by which a system improves its performance. Similarly, Mitchell (1997, p. 2) defines machine learning as the study of "any computer algorithm that improves its performance at some tasks through experience." Machine learning algorithms can be classified as supervised or unsupervised learning. In supervised learning, training examples consist of input/output pair patterns. The goal of the learning algorithm is to predict the output values of new examples, based on their input values. In unsupervised learning, training examples contain only the input patterns and no explicit target output is associated with each input. The learning algorithm needs to generalize from the input patterns to discover the output values.

Machine Learning Paradigms

Many machine learning systems have been developed over the past decades. Langley and Simon (1995) identified five major areas of machine learning research, namely neural networks, case-based

learning, genetic algorithms, rule induction, and analytic learning. Chen (1995) identified three classes of machine learning techniques: symbolic learning, neural networks, and evolution-based algorithms. Drawing on these two classifications and a review of the field, we have adopted a similar framework and have identified the following five major paradigms: (1) probabilistic models, (2) symbolic learning and rule induction, (3) neural networks, (4) evolution-based models, and (5) analytic learning and fuzzy logic.

Probabilistic Models

The use of probabilistic models was one of the earliest attempts to perform machine learning, of which the most popular example is the Bayesian method. Originating in pattern recognition research (Duda & Hart, 1973), this method was often used to classify different objects into predefined classes based on a set of features. A Bayesian model stores the probability of each class, the probability of each feature, and the probability of each feature given each class, based on the training data. When a new instance is encountered, it can be classified according to these probabilities (Langley, Iba, & Thompson, 1992). A variation of the Bayesian model, called the naïve Bayesian model, assumes that all features are mutually independent within each class. Because of its simplicity, the naïve Bayesian model has been widely used in various applications in different domains (Fisher, 1987; Kononenko, 1993).

Symbolic Learning and Rule Induction

Symbolic learning can be classified according to the underlying learning strategy, such as rote learning, learning by instruction, learning by analogy, learning from examples, and learning from discovery (Carbonell, Michalski, & Mitchell, 1983; Cohen & Feigenbaum, 1982). Among these, learning from examples appears to be the most promising symbolic learning technique for knowledge discovery and data mining. It is implemented by applying an algorithm that attempts to induce the general concept description, which best describes the different classes of the training examples. Numerous algorithms have been developed, each using one or more techniques to identify patterns that are helpful in generating a concept description. Quinlan's ID3 decision-tree building algorithm

(Quinlan, 1983), and variations such as C4.5 (Quinlan, 1993), have become some of the most widely used symbolic learning techniques. Given a set of objects, ID3 produces a decision tree that attempts to classify all the objects correctly. At each step, the algorithm finds the attribute that best divides the objects into the different classes by minimizing entropy (information uncertainty). After all objects have been classified, or all attributes have been used, the results can be represented by a decision tree or a set of production rules.

Neural Networks

Artificial neural networks attempt to achieve human-like performance by modeling the human nervous system. A neural network is a graph of many active nodes (neurons), which are connected to each other by weighted links (synapses). Although knowledge is represented by symbolic descriptions such as decision tree and production rules in symbolic learning, knowledge is learned and remembered by a network of interconnected neurons, weighted synapses, and threshold logic units (Lippmann, 1987; Rumelhart, Hinton, & McClelland, 1986). Based on training examples, learning algorithms can be used to adjust the connection weights in the network so that it can predict or classify unknown examples correctly. Activation algorithms over the nodes can then be used to retrieve concepts and knowledge from the network (Belew, 1989; Chen & Ng, 1995; Kwok, 1989).

Many different types of neural networks have been developed, among which the feedforward/backpropagation model is the most widely used. Backpropagation networks are fully connected, layered, feed-forward networks in which activations flow from the input layer through the hidden layer and then to the output layer (Rumelhart, Hinton, & Williams, 1986). The network usually starts with a set of random weights and adjusts its weights according to each learning example. Each learning example is passed through the network to activate the nodes. The network's actual output is then compared with the target output and the error estimates are propagated back to the hidden and input layers. The network updates its weights incrementally according to these error estimates until the network stabilizes. Other popular neural network models include Kohonen's self-organizing map and the Hopfield network. Self-organizing maps have been widely used in unsupervised learning,

clustering, and pattern recognition (Kohonen, 1995); Hopfield networks have been used mostly in search and optimization applications (Hopfield, 1982).

Evolution-Based Algorithms

Another class of machine learning algorithms consists of evolution-based algorithms that rely on analogies with natural processes and the Darwinian notion of survival of the fittest. Fogel (1994) identifies three categories of evolution-based algorithms: genetic algorithms, evolution strategies, and evolutionary programming. Genetic algorithms have proved popular and have been successfully applied to various optimization problems. They are based on genetic principles (Goldberg, 1989; Michalewicz, 1992). A population of individuals in which each individual represents a potential solution is first initiated. This population undergoes a set of genetic operations known as crossover and mutation. Crossover is a high-level process that aims at exploitation, and mutation is a unary process that aims at exploration. Individuals strive for survival based on a selection scheme that is biased toward selecting fitter individuals (individuals that represent better solutions). The selected individuals form the next generation and the process continues. After a number of generations, the program converges and the optimum solution is represented by the best individual.

Analytic Learning

Analytic learning represents knowledge as logical rules and performs reasoning on these rules to search for proofs. Proofs can be compiled into more complex rules to solve problems with a small number of searches required. For example, Samuelson and Rayner (1991) used analytic learning to represent grammatical rules that improve the speed of a parsing system.

Although traditional analytic learning systems depend on hard computing rules, usually no clear distinction exists between values and classes in the real world. To address this problem, fuzzy systems and fuzzy logic have been proposed. Fuzzy systems allow the values of "false" or "true" to operate over the range of real numbers from zero to

one (Zedah, 1965). Fuzziness accommodates imprecision and approximate reasoning.

Hybrid Approaches

As Langley and Simon (1995, p. 56) have pointed out, the reasons for differentiating the paradigms are "more historical than scientific." The boundaries between the different paradigms are usually unclear, and many systems combine different approaches. For example, fuzzy logic has been applied to rule induction and genetic algorithms (e.g., Mendes, Voznika, Freitas, & Nievola, 2001), genetic algorithms have been combined with neural networks (e.g., Maniezzo, 1994), and because the neural network approach has a close resemblance to the probabilistic and fuzzy logic models, they can be easily combined (e.g., Paass, 1990).

Evaluation Methodologies

The accuracy of a learning system needs to be evaluated before it can be useful, and the limited availability of data often makes estimating accuracy a difficult task. A bad testing method could give a result of zero percent accuracy for a system with an estimated accuracy of 33 percent (Kohavi, 1995). Therefore, choosing a good methodology is very important to the evaluation of machine learning systems.

Several popular evaluation methods are in use, including holdout sampling, cross validation, leave-one-out, and bootstrap sampling (Efron & Tibshirani, 1993; Stone, 1974). In the holdout method, the data are divided into a training set and a testing set. Usually two-thirds of the data are assigned to the training set and one-third to the testing set. After the system is trained by the training data, it needs to predict the output value of each instance in the testing set. These values are then compared with the real output values to determine accuracy.

In cross-validation, the data set is randomly divided into a number of subsets of roughly equal size. Ten-fold cross validation, in which the data set is divided into ten subsets, is most commonly used. The system is then trained and tested for ten iterations, and in each iteration nine subsets of data are used as training data and the remaining set as testing data. In rotation, each subset of data serves as the testing set in one iteration. The accuracy of the system is the average accuracy over the ten

iterations. Leave-one-out is the extreme case of cross-validation, where the original data are split into n subsets, where n is the number of observations in the original data. The system is trained and tested for n iterations, in each of which $n-1$ instances are used for training and the remaining instance is used for testing.

In the bootstrap method, n independent random samples are taken from the original data set of size n. Because the samples are taken with replacement, the number of unique instances will be less than n. These samples are then used as the training set for the learning system, and the remaining data that have not been sampled are used to test the system (Efron & Tibshirani, 1993).

Each of these methods has strengths and weaknesses. Several studies have compared them in terms of accuracy. Holdout sampling is the easiest to implement, but a major problem is that the training and testing set are not independent. This method also does not make efficient use of data because as many as one-third of the data are not used to train the system (Kohavi, 1995). Leave-one-out provides an almost unbiased estimate, but it is computationally expensive and its estimations have very high variances, especially for small data sets (Efron, 1983; Jain, Dubes, & Chen, 1987). Breiman and Spector (1992) and Kohavi (1995) conducted independent experiments to compare the performance of several different methods, and the results of both experiments showed ten-fold cross validation to be the best method for model selection.

Machine Learning for Information Retrieval: Pre-Web

Learning techniques had been applied in information retrieval applications long before the emergence of the Web. In their *ARIST* chapter, Cunningham, Kitten, and Litten (1999) provided an extensive review of applications of machine learning techniques in IR. In this section, we briefly survey some of the research in this area, covering the use of machine learning in information extraction, relevance feedback, information filtering, text classification, and text clustering.

Information extraction is one area in which machine learning is applied in IR, by means of techniques designed to identify useful information from text documents automatically. Named-entity extraction is

one of the most widely studied sub-fields. It refers to the automatic identification from text documents of the names of entities of interest, such as persons (e.g., "John Doe"), locations (e.g., "Washington, D.C."), and organizations (e.g., "National Science Foundation"). It also includes the identification of other patterns, such as dates, times, number expressions, dollar amounts, e-mail addresses, and Web addresses (URLs). The Message Understanding Conference (MUC) series has been the primary forum where researchers in this area meet and compare the performance of their entity extraction systems (Chinchor, 1998). Machine learning is one of the major approaches. Machine-learning-based entity extraction systems rely on algorithms rather than human-created rules to extract knowledge or identify patterns from texts. Examples of machine learning algorithms include neural networks, decision trees (Baluja, Mittal, & Sukthankar, 1999), hidden Markov model (Miller, Crystal, Fox, Ramshaw, Schwartz, Stone, et al., 1998), and entropy maximization (Borthwick, Sterline, Agichtein, & Grishman, 1998). Instead of relying on a single approach, most existing information extraction systems combine machine learning with other approaches (such as a rule-based or statistical approach). Many systems using a combined approach were evaluated at the MUC-7 conference. The best systems were able to achieve over 90 percent in both precision and recall rates in extracting persons, locations, organizations, dates, times, currencies, and percentages from a collection of *New York Times* news articles (Chinchor, 1998).

Relevance feedback is a well known method used in IR systems to help users conduct searches iteratively and reformulate search queries based on evaluation of previously retrieved documents (Ide, 1971; Rocchio, 1971). The main assumption is that documents relevant to a particular query are represented by a set of similar keywords (Salton, 1989). After a user rates the relevance of a set of retrieved documents, the query can be reformulated by adding terms from the relevant documents and subtracting terms from the irrelevant documents. It has been shown that a single iteration of relevance feedback can significantly improve search precision and recall (Salton, 1989). Probabilistic techniques have been applied to relevance feedback by estimating the probability of relevance of a given document to a user. Using relevance feedback, a model can learn the common characteristics of a set of relevant documents in order to estimate the probability of relevance for the remaining documents in

a collection (Fuhr & Buckley, 1991; Fuhr & Pfeifer, 1994). Various machine learning algorithms, such as genetic algorithms, ID3, and simulated annealing, have been used in relevance feedback applications (Chen, Shankaranarayanan, Iyer, & She, 1998; Kraft, Petry, Buckles, & Sadasivan, 1995, 1997).

Information filtering and *recommendation* techniques also apply user evaluation to improve IR system performance. The main difference is that, although relevance feedback helps users reformulate their search queries, information filtering techniques try to learn about users' interests from their evaluations and actions and then to use this information to analyze new documents. Information filtering systems are usually designed to alleviate the problem of information overload in IR systems. The NewsWeeder system allows users to give an article a rating from one to five. After a user has rated a sufficient number of articles, the system learns the user's interests from these examples and identifies Usenet news articles that the system predicts will be interesting to the user (Lang, 1995). Decision trees also have been used for news-article filtering (Green & Edwards, 1996). Another approach is collaborative filtering or recommender systems, in which collaboration is achieved as the system allows users to help one another perform filtering by recording their reactions to documents they read (Goldberg, Nichols, Oki, & Terry, 1992). One example is the GroupLens system, which performs collaborative filtering on Usenet news articles (Konstan, Miller, Maltz, Herlocker, Gordon, & Riedl, 1997). GroupLens recommends articles that may be of interest to a user based on the preferences of other users who have demonstrated similar interests. Many personalization and collaborative systems have been implemented as software agents to help users (Maes, 1994).

Text classification and *text clustering* studies have been reported extensively in the traditional IR literature. Text classification is the classification of textual documents into predefined categories (supervised learning), and text clustering groups documents into categories defined dynamically, based on their similarities (unsupervised learning). Although their usefulness continues to be debated (Hearst & Pedersen, 1996; Voorhees, 1985; Wu, Fuller, & Wilkinson, 2001), the use of classification and clustering is based on the cluster hypothesis: "closely associated documents tend to be relevant to the same requests"

(van Rijsbergen, 1979, p. 30). Machine learning is the basis of most text classification and clustering applications. Text classification has been extensively reported at the Association for Computing Machinery's (ACM) Special Interest Group on Information Retrieval (SIGIR) conferences and evaluated on standard test beds. For example, the naïve Bayesian method has been widely used (e.g., Koller & Sahami, 1997; Lewis & Ringuette, 1994; McCallum, Nigam, Rennie, & Seymore, 1999). Using the joint probabilities of words and categories calculated by considering all documents, this method estimates the probability that a document belongs to a given category. Documents with a probability above a certain threshold are considered relevant. The *k*-nearest neighbor method is another widely used approach to text classification. For a given document, the *k* neighbors that are most similar to a given document are first identified (Iwayama & Tokunaga, 1995; Masand, Linoff, & Waltz, 1992). The categories of these neighbors are then used to categorize the given document. A threshold is used for each category. Neural network programs have also been applied to text classification, usually employing the feedforward/backpropagation neural network model (Lam & Lee, 1999; Ng, Goh, & Low, 1997; Wiener, Pedersen, & Weigend, 1995). Term frequencies, or tf*idf scores (term frequency multiplied by inverse document frequency), of the terms are used to form a vector (Salton, 1989), which can be used as the input to the network. Using learning examples, the network will be trained to predict the category of a document. Another new technique used in text classification is support vector machine (SVM), a statistical method that tries to find a hyperplane that best separates two classes (Vapnik, 1995, 1998). Joachims first applied SVM to text classification (Joachims, 1998). SVM achieved the best performance on the Reuters-21578 data set for document classification (Yang & Liu, 1999).

As with text classification, text clustering tries to place documents into different categories based on their similarities. However, in text clustering no predefined categories are set; all categories are dynamically defined. Two types of clustering algorithms are generally used, namely hierarchical clustering and non-hierarchical clustering. The *k*-nearest neighbor method and Ward's algorithm (Ward, 1963) are the most widely used hierarchical clustering methods. Willet (1988) has provided an excellent review of hierarchical agglomerative clustering algorithms for

document retrieval. For non-hierarchical clustering, one of the most common approaches is the K-means algorithm. It uses the technique of local optimization, in which a neighborhood of other partitions is defined for each partition. The algorithm starts with an initial set of clusters, examines each document, searches through the set of clusters, and moves to that cluster for which the distance between the document and the centroid is smallest. The centroid position is recalculated every time a document is added. The algorithm stops when all documents have been grouped into the final required number of clusters (Rocchio, 1966). The Single-Pass method (Hill, 1968) is also widely used. However, its performance depends on the order of the input vectors and it tends to produce large clusters (Rasmussen, 1992). Suffix Tree Clustering, a linear time clustering algorithm that identifies phrases common to groups of documents, is another incremental clustering technique (Zamir & Etzioni, 1998). Kraft, Bordogna, and Pasi (1999) and Chen, Mikulic, and Kraft. (2000) also have proposed an approach to applying fuzzy clustering to information retrieval systems.

Another classification method much used in recent years is the neural network approach. For example, Kohonen's self-organizing map (SOM), a type of neural network that produces a two-dimensional grid representation for n-dimensional features, has been widely applied in IR (Kohonen, 1995; Lin, Soergel, & Marchionini, 1991; Orwig, Chen, & Nunamaker, 1997). The self-organizing map can be either multi-layered or single-layered. First, the input nodes, output nodes, and connection weights are initialized. Each element is then represented by a vector of N terms and is presented to the system. The distance d_j between the input and each output node j is computed. A winning node with minimum d_j is then selected. After the network stabilizes, the top phrase from each node is selected as the label, and adjacent nodes with the same label are combined to form clusters.

Web Mining

Web mining research can be divided into three categories: Web content mining, Web structure mining, and Web usage mining (Kosala & Blockeel, 2000). Web content mining refers to the discovery of useful information from Web content, including text, images, audio, and video.

Web content mining research includes resource discovery from the Web (e.g., Chakrabarti, van den Berg, & Dom, 1999; Cho, Garcia-Molina, & Page, 1998), document categorization and clustering (e.g., Zamir & Etzioni, 1999; Kohonen, Kaski, Lagus, Salojärvi, Honkela, Paatero, et al., 2000), and information extraction from Web pages (e.g., Hurst, 2001). Web structure mining studies potential models underlying the link structures of the Web. It usually involves the analysis of in-links and out-links, and has been used for search engine result ranking and other Web applications (e.g., Brin & Page, 1998; Kleinberg, 1998). Web usage mining focuses on using data mining techniques to analyze search or other activity logs to find interesting patterns. One of the main applications of Web usage mining is to develop user profiles (e.g., Armstrong, Freitag, Joachims, & Mitchell, 1995; Wasfi, 1999).

Several major challenges apply to Web mining research. First, most Web documents are in HTML (HyperText Markup Language) format and contain many markup tags, mainly used for formatting. Although Web mining applications must parse HTML documents to deal with these markup tags, the tags can also provide additional information about the document. For example, a bold typeface markup (<b>) may indicate that a term is more important than other terms, which appear in normal typeface. Such formatting cues have been widely used to determine the relevance of terms (Arasu, Cho, Garcia-Molina, Paepcke, & Raghavan, 2001).

Second, traditional IR systems often contain structured and well-written documents (e.g., news articles, research papers, metadata), but this is not the case on the Web. Web documents are much more diverse in terms of length, structure, and writing style, and many Web pages contain grammatical and spelling errors. Web pages are also diverse in terms of language and subject matter; one can find almost any language and any topic on the Web. In addition, the Web has many different types of content, including: text, image, audio, video, and executable. Numerous formats feature: HTML; Extensible Markup Language (XML); Portable Document Format (PDF); Microsoft Word; Moving Picture Experts group, audio layer 3 (mp3); Waveform audio file (wav); RealAudio (ra); and Audio Video Interleaved (avi) animation file, to name just a few. Web applications have to deal with these different formats and retrieve the desired information.

Third, although most documents in traditional IR systems tend to remain static over time, Web pages are much more dynamic; they can be updated every day, every hour, or even every minute. Some Web pages do not in fact have a static form; they are dynamically generated on request, with content varying according to the user and the time of the request. This makes it much more difficult for retrieval systems such as search engines to generate an up-to-date search index of the Web.

Another characteristic of the Web, perhaps the most important one, is the hyperlink structure. Web pages are hyperlinked to each other; it is through hyperlinking that a Web page author "cites" other Web pages. Intuitively, the author of a Web page places a link to another Web page if he or she believes that it contains a relevant topic or is of good quality (Kleinberg, 1998). Anchor text, the underlined, clickable text of an outgoing link in a Web page, also provides a good description of the target page because it represents how other people linking to the page actually describe it. Several studies have tried to make use of anchor text or the adjacent text to predict the content of the target page (Amitay, 1998; Rennie & McCallum, 1999).

Lastly, the Web is larger than traditional data sources or document collections by orders of magnitude. The number of indexable Web pages exceeds two billion, and has been estimated to be growing at a rate of roughly one million pages per day (Lawrence & Giles, 1999; Lyman & Varian, 2000). Collecting, indexing, and analyzing these documents presents a great challenge. Similarly, the population of Web users is much larger than that of traditional information systems. Collaboration among users is more feasible because of the availability of a large user base, but it can also be more difficult because of the heterogeneity of the user base.

In the next section, we review how machine learning techniques for traditional IR systems have been improved and adapted for Web mining applications, based on the characteristics of the Web. Significant work has been undertaken both in academia and industry. However, because most commercial applications do not disclose technical or algorithmic details, our review will focus largely on academic research.

Web Content Mining

Web content mining is mainly based on research in information retrieval and text mining, such as information extraction, text classification and clustering, and information visualization. However, it also includes some new applications, such as Web resource discovery. Some important Web content mining techniques and applications are reviewed in this subsection.

Text Mining for Web Documents

As discussed earlier, text mining is often considered a sub-field of data mining and refers to the extraction of knowledge from text documents (Chen, 2001; Hearst, 1999). Because the majority of documents on the Web are text documents, text mining for Web documents can be considered a sub-field of Web mining, or, more specifically, Web content mining. Information extraction, text classification, and text clustering are examples of text-mining applications that have been applied to Web documents.

Although information extraction techniques have been applied to plain text documents, extracting information from HTML Web pages can present a quite different problem. As has been mentioned, HTML documents contain many markup tags that can identify useful information. However, Web pages are also comparatively unstructured. Instead of a document consisting of paragraphs, a Web page can be a document composed of a sidebar with navigation links, tables with textual and numerical data, capitalized sentences, and repetitive words. The range of formats and structures is very diverse across the Web. If a system could parse and understand such structures, it would effectively acquire additional information for each piece of text. For example, a set of links with a heading "Link to my friends' homepages" may indicate a set of people's names and corresponding personal home page links. The header row of a table can also provide additional information about the text in the table cells. On the other hand, if these tags are not processed correctly but simply stripped off, the document may become much noisier.

Chang and Lui (2001) used a PAT tree to construct automatically a set of rules for information extraction. The system, called IEPAD (Information Extraction Based on Pattern Discovery), reads an input

Web page and looks for repetitive HTML markup patterns. After unwanted patterns have been filtered out, each pattern is used to form an extraction rule in regular expression. IEPAD has been tested in an experiment to extract search results from different search engines and achieved a high retrieval rate and accuracy. Wang and Hu (2002) used both decision tree and SVM to learn the patterns of table layouts in HTML documents. Layout features, content type, and word group features are combined and used as a document's features. Experimental results show that both decision tree and SVM can detect tables in HTML documents with high accuracy. Borodogna and Pasi (2001) proposed a fuzzy indexing model that allows users to retrieve sections of structured documents such as HTML and XML. Doorenbos, Etzioni, and Weld (1997) also have applied machine learning in the ShopBot system to extract product information from Web pages. Some commercial applications also extract useful information from Web pages. For instance, FlipDog (http://www.flipdog.com), developed by the Whizbang! Labs (http://www.inxight.com/whizbang), crawls the Web to identify job openings on employer Web sites. Lencom Software (http://www.lencom.com) also developed several products that can extract e-mail addresses and image information from the Web.

Although information extraction analyzes individual Web pages, text classification and text clustering analyze a set of Web pages. Again, Web pages consist mostly of HTML documents and are often noisier and less structured than traditional documents such as news articles and academic abstracts. In some applications the HTML tags are simply stripped from the Web documents and traditional algorithms are then applied to perform text classification and clustering. However, some useful characteristics of Web page design would be ignored. For example, Web page hyperlinks would be lost, but "Home," "Click here," and "Contact us," would be included as a document's features. This creates a unique problem for performing text classification and clustering of Web documents because the format of HTML documents and the structure of the Web provide additional information for analysis. For example, text from neighboring documents has been used in an attempt to improve classification performance. However, experimental results show that this method does not improve performance because, often, too many neighbor terms and too many cross-linkages occur between different classes

(Chakrabarti, Dom, & Indyk, 1998; Yang, Slattery, & Ghani, 2002). Use of other information from neighboring documents has been proposed, including the predicted category of neighbors (Chakrabarti et al., 1998; Oh, Myaeng, & Lee, 2000), the anchor text pointing to a document (Furnkranz, 1999), and the outgoing links to other documents (Joachims, Chistianini, & Shawe-Taylor, 2001). It has been shown that using such additional information improves classification results.

Likewise, text clustering algorithms have been applied to Web applications. In the Grouper system, Zamir and Etzioni (1998, 1999) applied the Suffix-Tree Clustering algorithm described earlier to the search results of the HuskySearch system. The self-organizing map (SOM) technique also has been applied to Web applications. Chen and colleagues (Chen, Fan, Chau, & Zeng, 2001; Chen, Chau, & Zeng, 2002) used a combination of noun phrasing and SOM to cluster the search results of search agents that collect Web pages by meta-searching popular search engines or performing a breadth-first search on particular Web sites. He, Zha, Ding, and Simon (2002) use a combination of content, hyperlink structure, and co-citation analysis in Web document clustering. Two Web pages are considered similar if they have similar content, they point to a similar set of pages, or many other pages point to both of them.

The large volume of documents available on the Web makes it an excellent resource for linguistic studies. The digital library project groups of the University of California at Berkeley and Stanford University analyzed 88 million Web pages and calculated the document frequency of the 113 million terms found in those pages (University of California Berkeley. Digital Library Project, 2002). Roussinov and Zhao (2003) use the Web as a resource for finding phrases with high co-occurrences. Another example is the Strand system (Resnik, 1999), which attempts to identify bilingual parallel corpora on the Web.

Intelligent Web Spiders

Web spiders, also known as crawlers, wanderers, or Webbots, have been defined as "software programs that traverse the World Wide Web information space by following hypertext links and retrieving Web documents by standard HTTP protocol" (Cheong, 1996, p. 82). Since the early days of the Web, spiders have been widely used to build the underlying

databases of search engines (e.g., Pinkerton, 1994), perform personal searches (e.g., Chau, Zeng, & Chen, 2001), archive particular Web sites or even the whole Web (e.g., Kahle, 1997), or collect Web statistics (e.g., Broder, Kumar, Maghoul, Raghavan, Rajagopalan, Stata, et al., 2000). Chau and Chen (2003) provide a review of Web spider research.

Although most spiders use simple algorithms such as breadth-first search (e.g., Najork & Wiener, 2001), some use more advanced algorithms. These spiders are very useful for Web resource discovery. For example, the Itsy Bitsy Spider searches the Web using a best-first search and a genetic algorithm approach (Chen, Chung, Ramsey, & Yang, 1998). Each URL is modeled as an individual in the initial population. Crossover is defined as extracting the URLs that are pointed to by multiple starting URLs. Mutation is modeled by retrieving random URLs from Yahoo!. Because the genetic algorithm approach is an optimization process, it is well-suited to finding the best Web pages according to particular criteria. Webnaut is another spider that uses a genetic algorithm (Zacharis & Panayiotopoulos, 2001). Other advanced search algorithms have been used in personal spiders. Yang, Yen, and Chen (2000) applied hybrid simulated annealing in a personal spider application. Focused Crawler located Web pages relevant to a predefined set of topics based on example pages provided by the user (Chakrabarti, van den Berg, & Dom, 1999). It determined the relevance of each page using a naïve Bayesian model and the analysis of the link structures among the Web pages collected using the HITS algorithm (discussed in more detail in the section on Web structure mining). These values are used to judge which URL links to follow. Another similar system, Context Focused Crawler, also uses a naïve Bayesian classifier to guide the search process (Diligenti, Coetzee, Lawrence, Giles, & Gori, 2000).

Chau and Chen (in press) apply the Hopfield Net spreading activation to collect Web pages in particular domains. Each Web page is represented as a node in the network and hyperlinks are represented simply as links between the nodes. Each node is assigned an activation score, which is a weighted sum of a content and link scores. The content score is calculated by comparing the content of the page with a domain-specific lexicon, and the link score is based on the number of outgoing links in a page. Each node also inherits the scores from its parent

nodes. Nodes are then activated in parallel and activation values from different sources are combined for each individual node until the activation scores of nodes on the network reach a stable state (convergence). Relevance feedback also has been applied in spiders (Balabanovic & Shoham, 1995; Vrettos & Stafylopoatis, 2001). These spiders determine the next URL to visit based on the user's ratings of the relevance of the Web pages returned.

Multilingual Web Mining

The number of non-English documents on the Web continues to grow—more than 30 percent of Web pages are in a language other than English. In order to extract non-English knowledge from the Web, Web mining systems have to deal with issues in language-specific text processing. One might think that this would not be a problem because the base algorithms behind most machine learning systems are language-independent. Most algorithms, such as text classification and clustering, need only a set of features (a vector of keywords) for the learning process. However, the algorithms usually depend on some phrase segmentation and extraction programs to generate a set of features or keywords to represent Web documents. Many existing extraction programs, especially those employing a linguistic approach (e.g., Church, 1988), are language-dependent and work only with English texts. In order to perform analysis on non-English documents, Web mining systems must use the corresponding phrase extraction program for each language. Other learning algorithms, such as information extraction and entity extraction, also have to be tailored for different languages.

Some segmentation and extraction programs are language-independent. These programs usually employ a statistical or a machine learning approach. For example, the mutual-information-based PAT-Tree algorithm is a language-independent technique for key phrase extraction and has been tested on Chinese documents (Chien, 1997; Ong & Chen, 1999). Similarly, Church and Yamamoto (2001) use suffix arrays to perform phrase extraction. Because these programs do not rely on specific linguistic rules, they can be easily modified to work with different languages.

Web Visualization

Because it is often difficult to extract useful content from the Web, visualization tools have been used to help users maintain a "big picture" of a set of retrieval results from search engines, particular Web sites, a subset of the Web, or even the whole Web. Various techniques have been developed in the past decade. For example, many systems visualize the Web as a tree structure based on the outgoing links of a set of starting nodes (e.g., Huang, Eades, & Cohen, 1998). The best-known example of this approach is the hyperbolic tree developed by Xerox PARC (Lamping & Rao, 1996), which employs the "focus+context" technique to show Web sites as a tree structure using a hyperbolic view. Users can focus on the document they are looking at and maintain an overview of the context at the same time. A map is another metaphor widely used for Web visualization. The ET-Map provides a visualization of the manually cataloged Entertainment hierarchy of Yahoo! as a two-dimensional map (Chen, Schuffles, & Orwig, 1996). Some 110,000 Web pages are clustered into labeled regions based on the self-organizing map approach, in which larger regions represent more important topics, and regions close to each other represent topics that are similar (Lin, Chen, & Nunmaker, 2000). The WEBSOM system also utilizes the SOM algorithm to cluster over a million Usenet newsgroup documents (Kohonen, 1995; Lagus, Honkela, Kaski, & Kohonen, 1999). Other examples of Web visualization include WebQuery, which uses a bullseye's view to visualize Web search results based on link structure (Carrière & Kazman, 1997), WebPath, which visualizes a user's trail as he or she browses the Web (Frécon & Smith, 1998), and three-dimensional models such as Natto View (Shiozawa & Matsushita, 1997) and Narcissus (Hendley, Drew, Wood, & Beale, 1995). Dodge and Kitchin (2001) provide a comprehensive review of cybermaps generated since the inception of the Internet.

In these visualization systems, machine learning techniques are often used to determine how Web pages should be placed in the 2-D or 3-D space. One example is the SOM algorithm described in the section on pre-Web IR (Chen et al., 1996). Web pages are represented as vectors of keywords and used to train the network that contains a two-dimensional grid of output nodes. The distance between the input and each output node is then computed and the node with the least distance is selected.

After the network is trained through repeated presentation of all inputs, the documents are submitted to the trained network and each region is labeled by a phrase, the key concept that best represents the cluster of documents in that region. Multidimensional scaling (MDS) is another method that can position documents on a map. It tries to map high dimensionality (e.g., document vectors) to low dimensionality (usually 2D) by solving a minimization problem (Cox & Cox, 1994). It has been tested with document mapping and the results are encouraging (McQuaid, Ong, Chen, & Nunamaker, 1999).

The Semantic Web

A recent significant extension of the Web is the Semantic Web (Berners-Lee, Hendler, & Lassila, 2001), which seeks to add metadata to describe data and information, based on such standards as RDF (Resource Description Framework) and XML. The idea is that Web documents will no longer be unstructured text; they will be labeled with meaning that can be understood by computers. Machine learning can play three important roles in the Semantic Web. First, machine learning can be used to automatically create the markup or metadata for existing unstructured textual documents on the Web. It is very difficult and time-consuming for Web page authors to generate Web pages manually, according to the Semantic Web representation. To address this problem, information extraction techniques, such as entity extraction, can be applied to automate or semi-automate tasks such as identifying entities in Web pages and generating the corresponding XML tags. Second, machine learning techniques can be used to create, merge, update, and maintain ontologies. Ontology, the explicit representation of knowledge combined with domain theories, is one of the key elements in the Semantic Web (Berners-Lee et al., 2001; Fensel & Musen, 2001). Maedche and Staab (2001) propose a framework for knowledge acquisition using machine learning. In that framework, machine learning techniques, such as association rule mining or clustering, are used to extract knowledge from Web documents in order to create new ontologies or improve existing ones. Third, machine learning can understand and perform reasoning on the metadata provided by the Semantic Web in order to extract knowledge from the Web more effectively. The documents in the Semantic Web are much more precise, more structured, and less

"noisy" than the general, syntactic Web. The Semantic Web also provides context and background information for analyzing Web pages. It is believed that the Semantic Web can greatly improve the performance of Web mining systems (Berendt, Hotho, & Stumme, 2002).

Web Structure Mining

In recent years, Web link structure has been widely used to infer important information about Web pages. Web structure mining has been largely influenced by research in social network analysis and citation analysis (bibliometrics). Citations (linkages) among Web pages are usually indicators of high relevance or good quality. We use the term *in-links* to indicate the hyperlinks pointing to a page and the term *out-links* to indicate the hyperlinks found in a page. Usually, the larger the number of in-links, the more useful a page is considered to be. The rationale is that a page referenced by many people is likely to be more important than a page that is seldom referenced. As in citation analysis, an often-cited article is presumed to be better than one that is never cited. In addition, it is reasonable to give a link from an authoritative source (such as Yahoo!) a higher weight than a link from an unimportant personal home page.

By analyzing the pages containing a URL, we can also obtain the anchor text that describes it. Anchor text shows how other Web page authors annotate a page and can be useful in predicting the content of the target page. Several algorithms have been developed to address this issue.

Among various Web-structure mining algorithms, PageRank and HITS (Hyperlinked Induced Topic Search) are the two most widely used. The PageRank algorithm is computed by weighting each in-link to a page proportionally to the quality of the page containing the in-link (Brin & Page, 1998). The qualities of these referring pages also are determined by PageRank. Thus, the PageRank of a page p is calculated recursively as follows:

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

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

A Web page has a high PageRank score if it is linked from many other pages, and the scores will be even higher if these referring pages are also good pages (pages that have high PageRank scores). It is also interesting to note that the PageRank algorithm follows a random walk model—the PageRank of a page is proportional to the probability that a random surfer clicking on random links will arrive at that page.

Kleinberg (1998) proposed the HITS algorithm, which is similar to PageRank. In the HITS algorithm, *authority* pages are defined as high-quality pages related to a particular topic or search query. *Hub* pages are those that are not necessarily authorities themselves but provide pointers to other authority pages. A page to which many others point should be a good authority, and a page that points to many others should be a good hub. Based on this intuition, two scores are calculated in the HITS algorithm for each Web page: an authority score and a hub score, which are calculated as follows:

$$AuthorityScore(p) = \sum_{\substack{all\ q\ linking \\ to\ p}} (HubScore(q))$$

$$HubScore(p) = \sum_{\substack{all\ r\ linking \\ to\ p}} (AuthorityScore(r))$$

In other words, a page with a high authority score is one pointed to by many good hubs, and a page with a high hub score is one that points to many good authorities.

Following the success of the PageRank and HITS algorithms, other similar algorithms also have been proposed. Examples include the Stochastic Approach to Link-Structure Analysis (SALSA) algorithm (Lempel & Moran, 2001) and the Probabilistic HITS (PHITS) algorithm (Cohn & Chang, 2000). Web structure mining techniques are often used to enhance the performance of Web applications. For instance, PageRank has been shown to be very effective for ranking search results in the commercial search engine *Google* (http://www.google.com) (Brin & Page, 1998). It also has been used as a measure to guide search engine spiders, where URLs with higher PageRank are visited first (Cho et al., 1998). The HITS algorithm also has been used in various Web applications. One example is the Clever search engine (Chakrabarti, Dom, Kumar, Raghavan, Rajogopalan, Tomkins, et al., 1999), which achieves

a higher user evaluation than the manually compiled directory of Yahoo!. Bharat and Henzinger (1998) have added several extensions to the basic HITS algorithm, such as modifying how much a node influences its neighbors based on a relevance score. One of the major drawbacks shared by most Web structure analysis algorithms is their high computational requirement, because the scores often have to be calculated iteratively (Haveliwala, 1999; Kleinberg, 1998).

Another application of Web structure mining is to understand the structure of the Web as a whole. Broder et al. (2000) analyzed the graph structure of a collection of 200 million Web pages and 1.5 billion links. Their results suggest that the core of the Web is a strongly connected component and that the Web's graph structure is shaped like a bow tie. The strongly connected component (SCC) comprises around 28 percent of the Web. Another group that consists of 21 percent of Web pages is called IN, in which every Web page contains a direct path to the SCC. Another 21 percent of Web pages are in the group OUT. For every page in OUT, a direct path from SCC links to it. Twenty-two percent of Web pages are in the group TENDRILS, which consists of pages hanging off IN and OUT but without a direct path to SCC. The remaining Web pages, accounting for around 8 percent of the Web, are isolated components that are not connected to the other four groups.

Web Usage Mining

Web servers, proxies, and client applications can quite easily capture data about Web usage. Web server logs contain information about every visit to the pages hosted on a server. Some of the useful information includes what files have been requested from the server, when they were requested, the Internet Protocol (IP) address of the request, the error code, the number of bytes sent to the user, and the type of browser used. Web servers can also capture referrer logs, which show the page from which a visitor makes the next request. Client-side applications, such as Web browsers or personal agents, can also be designed to monitor and record a user's actions. By performing analysis on Web usage data (sometimes referred to as *clickstream analysis*), Web mining systems can discover useful knowledge about a system's usage characteristics and the users' interests. This knowledge has various applications, such as personalization and collaboration in Web-based systems, marketing,

Web site design, Web site evaluation, and decision support (Chen & Cooper, 2001; Marchionini, 2002).

Pattern Discovery and Analysis

One of the major goals of Web usage mining is to reveal interesting trends and patterns. Such patterns and statistics can often provide important knowledge about a company's customers or the users of a system. Srivastava, Cooley, Despande, and Tan (2000) provided a framework for Web usage mining, consisting of three major steps: preprocessing, pattern discovery, and pattern analysis. As in other data mining applications, preprocessing involves data cleansing. However, one of the major challenges faced by Web usage mining applications is that Web server log data are anonymous, making it difficult to identify users and user sessions from the data. Techniques like Web cookies and user registration have been used in some applications, but each method has its shortcomings (Pitkow, 1997). In pattern discovery and analysis, generic machine learning and data mining techniques, such as association rule mining, classification, and clustering, can often be applied. For instance, Yan, Jacobsen, Garcia-Molina, and Dayal (1996) performed clustering on Web log data to identify users who have accessed similar Web pages.

Web usage mining has been used for various purposes. For example, Buchner and Mulvenna (1998) proposed a knowledge discovery process for mining marketing intelligence from Web data. Data such as Web traffic patterns also can be extracted from Web usage logs in order to improve the performance of a Web site (Cohen, Krishnamurthy, & Rexford, 1998). Many commercial products have been developed to support analysis and mining of Web site usage and Web log data. Examples of these applications include WebTrends developed by NetIQ (http://www.netiq.com/webtrends), WebAnalyst by Megaputer (http://www.megaputer.com/products/wa), NetTracker by Sane Solutions (http://www.sane.com/products/NetTracker), and NetGenesis by CustomerCentric (http://www.customercentricsolutions.com/content/solutions/ent_web_analytics.cfm). Although most Web usage analysis applications focus on single Web sites, the advertising company DoubleClick (http://www.doubleclick.com), selling and administrating two billion online advertisements per day, collects gigabytes of clickstream data across different Web sites.

Search engine transaction logs also provide valuable knowledge about user behavior in Web searching. Various analyses have been performed on the transaction logs of the Excite search engine (http://www.excite.com) (Jansen, Spink, & Saracevic, 2000; Spink & Xu, 2000; Spink, Wolfram, Jansen, & Saracevic, 2001). Silverstein, Henzinger, Marais, and Moricz (1999) also conducted a study of 153 million unique search queries collected from the AltaVista search engine (http://www.altavista.com). Some of the interesting findings from these analyses include the set of most popular words used by the public in Web search queries, the average length of a search query, the use of Boolean operators in queries, and the average number of result pages viewed by users. Such information is particularly useful to researchers trying to reach a better understanding of users' Web searching and information-seeking behaviors and hoping to improve the design of Web search systems.

Personalization and Collaboration

In addition to the research in Web spiders discussed earlier, other agent techniques have been used in Web applications. Many of these aim to provide personalized information and services to users. Web usage data provide an excellent way to learn about users' interest (Srivastava et al., 2000). WebWatcher (Armstrong et al., 1995) and Letizia (Lieberman, 1995) are two early examples. In WebWatcher, a user specifies the information needs, and the traversal links of the user are captured. These data are then used to generate recommendations for the user based on simple learning algorithms. The Letizia system tries to learn the user's interests on the fly, employing heuristics based on the user's actions such as following a link or saving a document. The system explores neighboring Web pages of potential interest using a best-first search algorithm.

The exponential growth of the Web has greatly increased the amount of usage data in server logs. Web logs usually consist of usage data for more than one user. Web usage mining can help identify users who have accessed similar Web pages. The patterns that emerge can be applied in collaborative Web searching and collaborative filtering. In the Fab system, Web pages are recommended to users based on the Web pages visited by other users having similar interests (Balabanovic & Shoham, 1997). Similarly, Amazon.com (http://www.amazon.com)

uses collaborative filtering to recommend books to potential customers based on the preferences of other customers having similar interests or purchasing histories. Huang, Chung, Ong, and Chen (2002) used Hopfield Net to model user interests and product profiles in an online bookstore in Taiwan. Spreading activation and association rule mining are used to search the network in order to provide recommendations to users.

Conclusions and Future Directions

The Web has become the world's largest knowledge repository. Extracting knowledge from the Web efficiently and effectively is becoming increasingly important for a variety of reasons. We have reviewed research on how machine learning techniques can be applied to Web mining. It should be noted, however, that a major limitation of Web mining research has been the difficulty of creating suitable test collections that can be reused by researchers. A test collection is important because it allows researchers to compare different algorithms using a standard test-bed under the same conditions, without being affected by such factors as Web page changes or network traffic variations. Because of the enormity of the Web, a significant amount of data has to be included in a test collection in order to create a reasonable, representative subset. It is also difficult to collect Web usage data across different sites because most server log data and the data collected by companies such as DoubleClick are proprietary. One effort to address this issue is the Web Track in the TREC community, which has created a test collection with 18.5 million Web pages, amounting to 100 gigabytes of data (Hawking, Voorhees, Craswell, & Bailey, 1999).

Most current Web mining applications reviewed in this chapter only scratch the surface of the Web's "knowledge mine." Web mining activities are still in their early stages and should continue to develop as the Web evolves. One future research direction for Web mining is multimedia data mining. In addition to textual documents like HTML, MS Word Document, PDF, and plain text files, a large number of multimedia documents are contained on the Web, such as images, audios, and videos. Although textual documents are comparatively easy to index, retrieve,

and analyze, operations on multimedia files are much more difficult to perform; and with multimedia content on the Web growing rapidly, Web mining has become a challenging problem. Various machine learning techniques have been employed to address this issue. Predictably, research in pattern recognition and image analysis has been adapted for study of multimedia documents on the Web, such as video (Christel, Cubilo, Gunaratne, Jerome, O, & Solanki, 2002; Wactlar, Christel, Gong, & Hauptmann, 1999; see also Smeaton's chapter in this volume) and music (McPherson & Bainbridge, 2001). Relevant text that describes a multimedia file, such as the "alt" text (alternative text), anchor text, HTML headings, table headings, image and video captions, and descriptions, also have been used for analyzing multimedia documents (Rowe, 2002). However, these techniques are currently used primarily for information retrieval on the Web, rather than for Web mining. As a picture is worth a thousand words, we believe that Web mining applications should not ignore the knowledge embedded in multimedia data.

In addition to being content-diverse, the Web has become more international and multi-cultural. Non-English Web content has experienced strong growth over the past few years, and both globalization and e-commerce have stimulated extensive multilingual content. Current research in multilingual analysis includes Web page translations, such as the AltaVista Babel Fish (http://babelfish.altavista.com), and cross-language information retrieval in which a search query is entered in one language to retrieve Web pages in another. As with multimedia content, these techniques are often used only for information retrieval. Future Web mining applications should attempt to extract and infer knowledge from a set of multilingual documents.

Another important area is the Wireless Web. Although it is likely that the majority of Web content will continue to be traditional Web pages such as HTML documents, more and more documents on the Web will be written in formats designed for handheld devices such as PDAs (Personal Digital Assistants) and cellular phones. WML (Wireless Markup Language) and HDML (Handheld Device Markup Language) are examples of such formats. The wireless portion of the Web is also quite different from the traditional Web. The information contained in the Wireless Web is often more concise, more location-specific, and more

time-critical. In addition, because of the nature of wireless devices, usage patterns for the Wireless Web are also quite different from those of the traditional Web. It would be interesting to apply Web mining techniques to the Wireless Web and to use such techniques to improve wireless information delivery by methods such as information personalization.

The hidden Web, also known as the invisible Web or deep Web, has given rise to another issue facing Web mining research. The hidden Web refers to documents on the Web that are dynamic and not accessible by general search engines; most search engine spiders can access only the publicly indexable Web (or the visible Web). Most documents in the hidden Web, including pages hidden behind search forms, specialized databases, and dynamically generated Web pages, are not accessible by general Web mining applications. If, as has been estimated, the hidden Web is 400 to 550 times larger than the visible Web (Lyman & Varian, 2000), extracting information and knowledge from it constitutes a major challenge for search engines as well as Web mining applications.

As discussed earlier, the Semantic Web provides considerable prospects for Web mining research. However, the Semantic Web is not without its weaknesses, the major one being that it depends on Web authors for its success. If Web page authors do not see benefits for themselves in migrating to the Semantic Web, they will be reluctant to provide metadata markup in their Web pages. Because the Semantic Web is still in its infancy, Web-mining researchers should pay close attention to its development and see how it affects Web-mining applications as it matures.

The Web has become the largest knowledge base ever to have existed. However, without appropriate knowledge representation and knowledge discovery algorithms, it is just like a human being with extraordinary memory but no ability to think and reason. We believe that research in machine learning and Web mining are promising as well as challenging, and both fields will help produce applications that can more effectively and efficiently utilize the Web of knowledge for humankind.

References

Amitay, E. (1998). Using common hypertext links to identify the best phrasal description of target Web documents. *Proceedings of the ACM SIGIR'98 Post-Conference Workshop on Hypertext Information Retrieval for the Web.* Retrieved February 20, 2003, from mq.edu.au/~einat/publicat...sigir_98.ps

Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A., & Raghavan, S. (2001). Searching the Web. *ACM Transactions on Internet Technology, 1*(1), 2–43.

Armstrong, R., Freitag, D., Joachims, T., & Mitchell, T. (1995). WebWatcher: A learning apprentice for the World Wide Web. *Proceedings of the AAAI-95 Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments,* 6–12.

Balabanovic, M., & Shoham, Y. (1995). Learning information retrieval agents: Experiment with Web browsing. *Proceedings of the AAAI-95 Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, 13–18.

Balabanovic, M., & Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Communications of the ACM, 40*(3), 66–72.

Baluja, S., Mittal, V., & Sukthankar, R. (1999). Applying machine learning for high performance named-entity extraction. *Proceedings of the Conference of the Pacific Association for Computational Linguistics, 1999*, 365–378.

Belew, R. K. (1989). Adaptive information retrieval: Using a connectionist representation to retrieve and learn about documents. *Proceedings of the 12th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 11–20.

Berendt, B., Hotho, A., & Stumme, G. (2002) Towards Semantic Web mining. *Proceedings of the First International Semantic Web Conference*, 264–278.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American, 284*(5), 35–43.

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

Borodogna, G., & Pasi, G. (2001). A user-adaptive indexing model of structured documents. *Proceedings of the 10th IEEE International Conference on Fuzzy Systems, 2,* 984–989.

Borthwick, A., Sterling, J., Agichtein, E., & Grishman, R. (1998). NYU: Description of the MENE named entity system as used in MUC-7. *Proceedings of the Seventh Message Understanding Conference (MUC-7).* Retrieved February 20, 2003, from http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_proceedings/nyu_st_paper.pdf

Breiman, L., & Spector, P. (1992). Submodel selection and evaluation in regression: The x-random case. *International Statistical Review, 60*(3), 291–319.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Proceedings of the 7th World Wide Web Conference.* Retrieved February 20, 2003, from http://www7.scu.edu.au/programme/fullpapers/1921/com1921.htm

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., & Wiener, J. (2000). Graph structure in the Web. *Proceedings of the 9th International World Wide Web Conference*. Retrieved February 20, 2003, from http://www9.org/w9cdrom/160/160.html

Buchner, A., & Mulvenna, M. D. (1998). Discovering Internet marketing intelligence through online analytical Web usage mining. *SIGMOD Record, 27*(4), 54–61.

Carbonell, J. G., Michalski, R. S., & Mitchell, T. M. (1983). An overview of machine learning. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach* (pp. 3–23). Palo Alto, CA: Tioga.

Carrière, J., & Kazman R. (1997). WebQuery: Searching and visualizing the Web through connectivity. *Proceedings of the 6th World Wide Web Conference*, 107–117.

Chakrabarti, S. (2000). Data mining for hypertext: A tutorial survey. *SIGKDD Explorations, 1*(1), 1–11.

Chakrabarti, S., Dom, B., & Indyk, P. (1998). Enhanced hypertext categorization using hyperlink. *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, 307–318.

Chakrabarti, S., Dom, B., Kumar, S. R., Raghavan, P., Rajagopalan, S., Tomkins, A., Gibson, D., & Kleinberg, J. (1999). Mining the Web's link structure. *IEEE Computer, 32*(8), 60–67.

Chakrabarti, S., van den Berg, M., & Dom, B. (1999). Focused crawling: A new approach to topic-specific Web resource discovery. *Proceedings of the 8th International World Wide Web Conference*. Retrieved February 20, 2003, from http://www8.org/w8-papers/5a-search-query/crawling/index.html

Chang, C. H., & Lui, S. C. (2001). IEPAD: Information extraction based on pattern discovery. *Proceedings of the 10th World Wide Web Conference*. Retrieved February 20, 2003, from http://www10.org/cdrom/papers/223/index.html

Chau, M., & Chen, H. (2003). Personalized and focused Web spiders. In N. Zhong, J. Liu, Y. Yao (Eds.), *Web intelligence* (pp. 197–217). New York: Springer-Verlag.

Chau, M., & Chen, H. (in press). Creating vertical search engines using spreading activation. *IEEE Computer*.

Chau, M., Zeng, D., & Chen, H. (2001). Personalized spiders for Web search and analysis. *Proceedings of the 1st ACM-IEEE Joint Conference on Digital Libraries*, 79–87.

Chen, H. (1995). Machine learning for information retrieval: Neural networks, symbolic learning, and genetic algorithms. *Journal of the American Society for Information Science, 46*, 194–216.

Chen, H. (2001). *Knowledge management systems: A text mining perspective*. Tucson, AZ: University of Arizona. Retrieved February 20, 2003, from http://ai.bpa.arizona.edu

Chen, H., Chau, M., & Zeng, D. (2002). CI spider: A tool for competitive intelligence on the Web. *Decision Support Systems, 34*(1), 1–17.

Chen, H., Chung, Y. Ramsey, M., & Yang, C. (1998). A smart itsy-bitsy spider for the Web. *Journal of the American Society for Information Science, 49*, 604–618.

Chen, H., Fan, H., Chau, M., & Zeng, D. (2001). MetaSpider: Meta-searching and categorization on the Web. *Journal of the American Society for Information and Science and Technology, 52*, 1134–1147.

Chen, H., & Ng, T. (1995). An algorithmic approach to concept exploration in a large knowledge network (automatic thesaurus consultation): Symbolic brand and bound search vs. connectionist Hopfield net activation. *Journal of the American Society for Information Science, 46*, 348–369.

Chen, H., Schuffels, C., & Orwig, R. (1996). Internet categorization and search: A machine learning approach. *Journal of Visual Communication and Image Representation, 7*(1), 88–102.

Chen, H., Shankaranarayanan, G., Iyer, A., & She, L. (1998). A machine learning approach to inductive query by examples: An experiment using relevance feedback, ID3, genetic algorithms, and simulated annealing. *Journal of the American Society for Information Science, 49*, 693–705.

Chen, H. M., & Cooper, M. D. (2001). Using clustering techniques to detect usage patterns in a Web-based information system. *Journal of the American Society for Information Science and Technology, 52*, 888–904.

Chen, J., Mikulcic, A., & Kraft, D. H. (2000). An integrated approach to information retrieval with fuzzy clustering and fuzzy inferencing. In O. Pons, M. A. Vila, and J. Kacprzyk (Eds.), *Knowledge management in fuzzy databases* (pp. 247–260). Heidelberg, Germany: Physica-Verlag.

Cheong, F. C. (1996). *Internet agents: Spiders, wanderers, brokers, and bots.* Indianapolis, IN: New Riders Publishing.

Chien, L. F. (1997). PAT-tree-based adaptive keyphrase extraction for intelligent Chinese information retrieval. *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 50–58.

Chinchor, N. A. (1998). Overview of MUC-7/MET-2. *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. Retrieved February 20, 2003, from http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_proceedings/overview.html

Cho, J., Garcia-Molina, H., & Page, L. (1998). Efficient crawling through URL ordering. *Proceedings of the 7th World Wide Web Conference*. Retrieved February 20, 2003, from http://www7.scu.edu.au/programme/fullpapers/1919/com1919.htm

Christel, M. G., Cubilo, P., Gunaratne, J., Jerome, W., O, E.-J., & Solanki, S. (2002). Evaluating a digital video library Web interface. *Proceedings of the 2nd ACM-IEEE Joint Conference on Digital Libraries*, 389.

Church, K. (1988). A stochastic parts program and noun phrase parser for unrestricted text. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 136–143.

Church, K., & Yamamoto, M. (2001). Using suffix arrays to compute term frequency and document frequency for all substrings in a corpus. *Computational Linguistics, 27*(1), 1–30.

Cohen, E., Krishnamurthy, B., & Rexford, J. (1998). Improving end-to-end performance of the Web using server volumes and proxy filters. *Proceedings of the ACM SIGCOMM '98 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 241–253.

Cohen, P. R., & Feigenbaum, E. A. (1982). *The handbook of artificial intelligence* (Vol. 3). Reading, MA: Addison-Wesley.

Cohn, D., & Chang, H. (2000) Learning to probabilistically identify authoritative documents. *Proceedings of the 17th International Conference on Machine Learning*. Retrieved February 20, 2003, from http://citeseer.nj.nec.com/cache/papers/cs/18471/http:zSzzSzwww.cs.cmu.eduzSz~cohnzSzpaperszSzphits.pdf/cohn00learning.pdf

Cooley, R., Mobasher, B., & Srivastava, J. (1997). Web mining: information and pattern discovery on the World Wide Web. *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence*, 558–567.

Cox, T. F., & Cox, M. A. A. (1994). *Multidimensional scaling*: London: Chapman & Hall.

Cunningham, S. J., Witten, I. H., & Littin, J. (1999). Applications of machine learning in information retrieval. *Annual Review of Information Science and Technology, 34,* 341–384.

Diligenti, M., Coetzee, F., Lawrence, S., Giles, C. L., & Gori, M. (2000). Focused crawling using context graphs. *Proceedings of the 26th International Conference on Very Large Databases*, 527–534.

Dodge, M., & Kitchin, R. (2001). *Atlas of cyberspace*. Reading, MA: Addison-Wesley.

Doorenbos, R. B., Etzioni, O., & Weld, D. S. (1997). A scalable comparison-shopping agent for the World Wide Web. *Proceedings of the First International Conference on Autonomous Agents,* 39–48.

Duda, R., & Hart, P. (1973). *Pattern classification and scene analysis*. New York: Wiley.

Efron, B. (1983). Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association, 78*(382), 316–330.

Efron, B., & Tibshirani, R. (1993). *An introduction to the bootstrap*. London: Chapman & Hall.

Etzioni, O. (1996). The World Wide Web: Quagmire or gold mine. *Communications of the ACM, 39*(11), 65–68.

Fensel, D., & Musen, M. A. (2001). The Semantic Web: A brain for humankind. *IEEE Intelligent Systems, 16*(2), 24–25.

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning, 2*, 139–172.

Fogel, D. B. (1994). An introduction to simulated evolutionary optimization. *IEEE Transactions on Neural Networks, 5,* 3–14.

Frécon, E., & Smith, G. (1998). WebPath: A three-dimensional Web history. *Proceedings of the IEEE Symposium on Information Visualization*, 3–10.

Fuhr, N., & Buckley, C. (1991). A Probabilistic Learning Approach for Document Indexing. *ACM Transactions on Information Systems*, 9, 223–248.

Fuhr, N., & Pfeifer, U. (1994). Probabilistic information retrieval as a combination of abstraction, inductive learning, and probabilistic assumption. *ACM Transactions on Information Systems, 12*(1), 92–115.

Furnkranz, J. (1999). Exploiting structural information for text categorization on the WWW. *Proceedings of the 3rd Symposium on Intelligent Data Analysis (IDA'99)*, 487–497.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley.

Goldberg, D., Nichols, D., Oki, B., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM, 35*(12), 61–69.

Green, C. L., & Edwards, P. (1996). Using machine learning to enhance software tools for Internet information management. *Proceedings of the AAAI-96 Workshop on Internet-Based Information Systems*, 48–55.

Han, J., & Chang, K. C. (2002). Data mining for Web intelligence. *IEEE Computer, 35*(11), 64–70.

Haveliwala, T. H. (1999). *Efficient computation of PageRank* (Stanford University Technical Report, 1999). Retrieved January 10, 2003, from http://dbpubs.stanford.edu:8090/pub/1999-31

Hawking, D., Voorhees, E., Craswell, N., & Bailey, P. (1999). Overview of the TREC-8 Web track. *Proceedings of the Eighth Text Retrieval Conference (TREC-8)*, 1–24.

Hayes-Roth, F., & Jacobstein, N. (1994). The state of knowledge-based systems. *Communications of the ACM, 37*(3), 27–39.

He, X., Zha, H., Ding, C., & Simon, H. (2002). Web document clustering using hyperlink structures. *Computational Statistics and Data Analysis, 41*,19–45.

Hearst, M. (1999). Untangling text data mining. *Proceedings of ACL'99: The 37th Annual Meeting of the Association for Computational Linguistics*. Retrieved February 20, 2003, from http://acl.ldc.upenn.edu/P/P99/P99-1001.pdf

Hearst, M. A., & Pederson, J. O. (1996). Reexamining the cluster hypothesis: Scatter/gather on retrieval results. *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 76–84.

Hendley, R. J., Drew, N. S., Wood, A., & Beale, R. (1995). Narcissus: Visualizing information. *Proceedings of the 1995 Information Visualization Symposium*, 90–96.

Hill, D. R. (1968). A vector clustering technique. In K. Samuelson (Ed.), *Mechanized information storage, retrieval and dissemination* (pp. 225–234). Amsterdam: North-Holland.

Hopfield, J. J. (1982). Neural network and physical systems with collective computational abilities. *Proceedings of the National Academy of Science, 79*(4), 2554–2558.

Huang, M. L., Eades, P., & Cohen, R. F. (1998). WebOFDAV: Navigating and visualizing the Web on-line with animated context swapping. *Proceedings of the 7th World Wide Web Conference*. Retrieved February 20, 2003, from http://www7.scu.edu.au/programme/posters/1865/com1865.htm

Huang, Z., Chung, W., Ong, T. H., & Chen, H. (2002). A graph-based recommender system for digital library. *Proceedings of the 2nd ACM-IEEE Joint Conference on Digital Libraries*, 65–73.

Hurst, M. (2001). Layout and language: Challenges for table understanding on the Web. *Proceedings of the 1st International Workshop on Web Document Analysis*, 27–30.

Ide, E. (1971). New Experiments in Relevance Feedback. In G. Salton (Ed.), *The SMART retrieval system: Experiments in automatic document processing* (pp. 337–354). Englewood Cliffs, NJ: Prentice-Hall.

Iwayama, M., & Tokunaga, T. (1995). Cluster-based text categorization: A comparison of category search strategies. *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 273–281.

Jain, A. K., Dubes, R. C., & Chen, C. (1987). Bootstrap techniques for error estimation. *IEEE Transactions on Pattern Analysis and Machine Learning, 9*(5), 628–633.

Jansen, B. J., Spink, A., & Saracevic, T. (2000). Real life, real users and real needs: A study and analysis of users queries on the Web. *Information Processing & Management, 36*(2), 207–227.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Proceedings of the 10th European Conference on Machine Learning*, 137–142.

Joachims, T., Chistianini, N., & Shawe-Taylor, J. (2001). Composite kernels for hypertext categorization. *Proceedings of the 18th International Conference on Machine Learning*, 250–227.

Kahle, B. (1997, March). Preserving the Internet. *Scientific American, 276*(6), 82–83.

Kleinberg, J. (1998). Authoritative sources in a hyperlinked environment. *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, 668–677.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1137–1143.

Kohonen, T. (1995). *Self-organizing maps.* Berlin, Germany: Springer-Verlag.

Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V., & Saarela, A. (2000). Self organization of a massive document collection. *IEEE Transactions on Neural Networks, 11*(3), 574–585.

Koller, D., & Sahami, M. (1997). Hierarchically classifying documents using very few words. *Proceedings of the 14th International Conference on Machine Learning*, 170–178.

Kononenko, I. (1993). Inductive and Bayesian learning in medical diagnosis. *Applied Artificial Intelligence, 7*, 317–337.

Konstan, J. A., Miller, B., Maltz, D., Herlocker, J., Gordon, L., & Riedl, J. (1997). GroupLens: Applying collaborative filtering to Usenet news. *Communications of the ACM, 40*(3), 77–87.

Kosala, R., & Blockeel, H. (2000). Web Mining Research: A Survey. *ACM SIGKDD Explorations, 2*(1), 1–15.

Kraft, D. H., Bordogna, G., & Pasi, G. (1999). Fuzzy set techniques in information retrieval. In J. C. Bezdek, D. Didier, and H. Prade (Eds.), *Fuzzy sets in approximate reasoning and information systems* (pp. 469–510). Norwell, MA: Kluwer Academic.

Kraft, D. H., Petry, F. E., Buckles, B. P., & Sadasivan, T. (1995). Applying genetic algorithms to information retrieval systems via relevance feedback. In P. Bosc & J. Kacprzyk (Eds.), *Fuzziness in database management systems* (pp. 330–344). Heidelberg, Germany: Physica-Verlag.

Kraft, D. H., Petry, F. E., Buckles, B. P., & Sadasivan, T. (1997). Genetic algorithms for query optimization in information retrieval: Relevance feedback. In E. Sanchez, T. Shibata, & L. A. Zadeh (Eds.), *Genetic algorithms and fuzzy logic systems* (pp. 155–173). Singapore: World Scientific.

Kwok, K. L. (1989). A neural network for probabilistic information retrieval. *Proceedings of the 12th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 21–30.

Lagus, K., Honkela, T., Kaski, S., & Kohonen, T. (1999). WEBSOM for textual data mining. *Artificial Intelligence Review, 13*(5/6), 345–364.

Lam, S. L. Y., & Lee, D. L. (1999). Feature reduction for neural network based text categorization. *Proceedings of the 6th International Conference on Database Systems for Advanced Applications*, 195.

Lamping, J., & Rao, R. (1996). Visualizing large trees using the hyperbolic browser. *Proceedings of the ACM CHI '96 Conference on Human Factors in Computing Systems*, 388–389.

Lang, K. (1995). NewsWeeder: Learning to filter netnews. *Proceedings of the 12th International Conference on Machine Learning*, 331–339.

Langley, P., Iba, W., & Thompson, K. (1992). An analysis of Bayesian classifiers. *Proceedings of the 10th National Conference on Artificial Intelligence*, 223–228.

Langley, P., & Simon, H. (1995). Applications of machine learning and rule induction. *Communications of the ACM, 38*(11), 55–64.

Lawrence, S., & Giles, C. L. (1999). Accessibility of information on the Web. *Nature, 400*, 107–109.

Lempel, R., & Moran, S. (2001). SALSA: The stochastic approach for link-structure analysis. *ACM Transactions on Information Systems, 19*(2), 131–160.

Lewis, D. D., & Ringuette, M. (1994). Comparison of two learning algorithms for text categorization. *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, 81–92.

Lieberman, H. (1995). Letizia: An agent that assists Web browsing. *Proceedings of the 1995 International Joint Conference on Artificial Intelligence*, 924–929.

Lin, C., Chen, H., & Nunamaker, J. F. (2000). Verifying the proximity hypothesis for self-organizing maps. *Journal of Management Information Systems, 16*(3), 57–70.

Lin, X., Soergel, D., & Marchionini, G. (1991). A self-organizing semantic map for information retrieval. *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 262–269.

Lippmann, R. P. (1987). An introduction to computing with neural networks. *IEEE Acoustics Speech and Signal Processing Magazine, 4*, 4–22.

Lyman, P., & Varian, H. R. (2000). How much information? Retrieved January 10, 2003, from University of California, School of Information Management and Systems Web site: http://www.sims.berkeley.edu/how-much-info

Maedche, A., & Staab, S. (2001). Ontology learning for the Semantic Web. *IEEE Intelligent Systems*, *16*(2), 72–79.

Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM, 37*(7), 31–40.

Maniezzo V. (1994). Genetic evolution of the topology and weight distribution of neural networks. *IEEE Transactions on Neural Networks, 5*(1), 39–53.

Marchionini, G. (2002). Co-evolution of user and organizational interfaces: A longitudinal case study of WWW dissemination of national statistics. *Journal of the American Society for Information Science and Technology, 53*, 1192–1209.

Masand, B., Linoff, G., & Waltz, D. (1992). Classifying news stories using memory based reasoning. *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 59–64.

McCallum, A., Nigam, K., Rennie, J., & Seymore, K. (1999). A machine learning approach to building domain-specific search engines. *Proceedings of the International Joint Conference on Artificial Intelligence*, 662–667.

McPherson, J., & Bainbridge, D. (2001). Usage of the MELDEX digital music library. *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval.* Retrieved February 20, 2003, from http://ismir2001.indiana.edu/posters/mcpherson.pdf

McQuaid, M., Ong, T. H., Chen, H., & Nunamaker, J. F. (1999). Multidimensional scaling for group memory visualization. *Decision Support Systems*, *27*(1–2), 163–176.

Mendes, R. R. F., Voznika, F. B., Freitas, A. A., & Nievola, J. C. (2001). Discovering fuzzy classification rules with genetic programming and co-evolution. *Principles of Data Mining and Knowledge Discovery, Lecture Notes in Artificial Intelligence, 2168*, 314–325.

Michalewicz, Z. (1992). *Genetic algorithms + data structures = evolution programs*. Berlin, Germany: Springer-Verlag.

Miller, S., Crystal, M., Fox, H., Ramshaw, L., Schwartz, R., Stone, R., Weischedel, R., & the Annotation Group (1998). BBN: Description of the SIFT system as used for MUC-7. *Proceedings of the Seventh Message Understanding Conference (MUC-7).* Retrieved February 20, 2003, from http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_proceedings/bbn_muc7.pdf

Mitchell, T. (1997). *Machine learning*. New York: McGraw Hill.

Najork, M., & Wiener, J. L. (2001). Breadth-first search crawling yields high-quality pages. *Proceedings of the Tenth Internal World Wide Web Conference*, 114–118.

Ng, H. T., Goh, W. B., & Low, K. L. (1997). Feature selection, perceptron learning, and a usability case study for text categorization. *Proceedings of the 20th*

Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 67–73.

Oh, H. J., Myaeng, S. H., & Lee, M. H. (2000). A practical hypertext categorization method using links and incrementally available class information. *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 264–271.

Ong, T., & Chen, H. (1999). Updateable PAT-Tree approach to Chinese key phrase extraction using mutual information: A linguistic foundation for knowledge management. *Proceedings of the Second Asian Digital Library Conference*, 63–84.

Orwig, R., Chen, H., & Nunamaker, J. F. (1997). A graphical self-organizing approach to classifying electronic meeting output. *Journal of the American Society for Information Science, 48*, 157–170.

Paass, G. (1990). Probabilistic reasoning and probabilistic neural networks. *Proceedings of the 3rd International Conference on Information Processing and Management of Uncertainty*, 6–8.

Pinkerton, B. (1994). Finding what people want: Experiences with the Webcrawler. *Proceedings of the 2nd International World Wide Web Conference*. Retrieved February 20, 2003, from http://archive.ncsa.uiuc.edu/SDG/IT94/Proceedings/Searching/pinkerton/WebCrawler.html

Pitkow, J. (1997). In search of reliable usage data on the WWW. *Proceedings of the 6th International World Wide Web Conference*, 451–463.

Quinlan, J. R. (1983). Learning efficient classification procedures and their application to chess end games. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach* (pp. 463–482). Palo Alto, CA: Tioga.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Los Altos, CA: Morgan Kaufmann.

Rasmussen, E. (1992). *Clustering algorithms*. Englewood Cliffs, NJ: Prentice Hall.

Rennie, J., & McCallum, A. K. (1999). Using reinforcement learning to spider the Web efficiently. *Proceedings of the 16th International Conference on Machine Learning (ICML-99)*, 335–343.

Resnik, P. (1999). Mining the Web for bilingual text. *Proceedings of the 34th Annual Meeting of the Association of Computational Linguistics*, College Park. Retrieved February 20, 2003, from http://acl.ldc.upenn.edu/P/P99/P99-1068.pdf

Rocchio, J. J. (1966). *Document retrieval systems: Optimization and evaluation*. Unpublished doctoral dissertation, Harvard University.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.), *The SMART Retrieval System—Experiments In Automatic Document Processing* (pp. 337–354). Englewood Cliffs, NJ: Prentice-Hall.

Roussinov, D., & Zhao, L. (2003). Automatic discovery of similarity relationships through Web mining. *Decision Support Systems, 35*(1), 149–166.

Rowe, N. (2002). A high-recall self-improving Web crawler that finds images using captions. *IEEE Intelligent Systems, 17*(4), 8–14.

Rumelhart, D. E., Hinton, G. E., & McClelland, J. L. (1986). A general framework for parallel distributed processing. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed processing* (pp. 45–76). Cambridge, MA: The MIT Press.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed processing* (pp. 318–362). Cambridge, MA: MIT Press.

Salton, G. (1989). *Automatic text processing.* Reading, MA: Addison-Wesley.

Samuelson, C., & Rayner, M. (1991). Quantitative evaluation of explanation-based learning as an optimization tool for a large-scale natural language system. *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, 609–615.

Shiozawa, H., & Matsushita Y. (1997). WWW visualization giving meanings to interactive manipulations. *Proceedings of HCI International '97*, 791–794.

Silverstein, C., Henzinger, M., Marais, H., & Moricz, M. (1999). Analysis of a very large Web search engine query log. *ACM SIGIR Forum, 33*(1), 6–12.

Simon, H. A. (1983). Why Should Machine Learn? In R. S. Michalski, J. Carbonell, & T. M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach* (pp. 25–38). Palo Alto, CA: Tioga Press.

Spink, A., Wolfram, D., Jansen, B. J., & Saracevic, T. (2001). Searching the Web: The public and their queries. *Journal of the American Society for Information Science and Technology, 52*, 226–234.

Spink, A., & Xu, J. (2000, online). Selected results from a large study of Web searching: The Excite study. *Information Research, 6*(1). Retrieved January 4, 2003, from http://InformationR.net/ir/6-1/paper90.html

Srivastava, J., Cooley, R., Deshpande, M., & Tan, P. N. (2000). Web usage mining: Discovery and applications of Web usage patterns from Web data. *ACM SIGKDD Explorations, 1*(2), 12–23.

Stone, M. (1974). Cross-validation choices and assessment of statistical predictions. *Journal of the Royal Statistical Society, 36*, 111–147.

Trybula, W. (1999). Text mining. *Annual Review of Information Science and Technology, 34*, 385–419.

University of California Berkeley. Digital Library Project. (2002). Web term document frequency and rank. Retrieved January 10, 2003, from the University of California, Berkeley, Digital Library Project Web site: http://elib.cs.berkeley.edu/docfreq

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

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

Vapnik, V. (1998). *Statistical learning theory*: Chichester, UK: Wiley.

Voorhees, E. M. (1985). The cluster hypothesis revisited. *Proceedings of the 12th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 188–196.

Voorhees, E., & Harman, D. (1998). Overview of the sixth Text REtrieval Conference (TREC-6). *Proceedings of the Sixth Text Retrieval Conference (TREC-6)*, 1–24.

Vrettos, S., & Stafylopatis, A. (2001). A fuzzy rule-based agent for Web retrieval-filtering. *Proceedings of the 1st Asia-Pacific Conference on Web Intelligence*, 448–453.

Wactlar, H. D., Christel, M. G., Gong, Y., & Hauptmann, A. G. (1999). Lessons learned from the creation and deployment of a terabyte digital video library. *IEEE Computer, 32*(2), 66–73.

Wang, Y., & Hu, J. (2002). A machine learning based approach for table detection on the Web. *Proceedings of the 11th World Wide Web Conference*. Retrieved February 20, 2003, from http://www2002.org/CDROM/refereed/199

Ward, J. (1963). Hierarchical grouping to optimize an objection function. *Journal of the American Statistical Association, 58,* 236–244.

Wasfi, A. M. A. (1999). Collecting user access patterns for building user profiles and collaborative filtering. *Proceedings of the 1999 International Conference on Intelligent User Interfaces*, 57–64.

Wiener, E., Pedersen, J. O., & Weigend, A. S. (1995). A neural network approach to topic spotting. *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, 317–332.

Willett, P. (1988). Recent trends in hierarchical document clustering: A critical review. *Information Processing & Management, 24*, 577–597.

Wu, M., Fuller, M., & Wilkinson, R. (2001). Using clustering and classification approaches in interactive retrieval. *Information Processing & Management, 37*, 459–484.

Yan, T., Jacobsen, J., Garcia-Molina, H., & Dayal, U. (1996). From user access patterns to dynamic hypertext linkage. *Proceedings of the 5th World Wide Web Conference*. Retrieved February 20, 2003, from http://www5conf.inria.fr/fich_html/slides/papers/PS3/P8/all.htm

Yang, C. C., Yen, J., & Chen, H. (2000). Intelligent Internet searching agent based on hybrid simulated annealing. *Decision Support Systems, 28*, 269–277.

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 42–49.

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

Zacharis, Z. N., & Panayiotopoulos, T. (2001). Web search using a genetic algorithm. *IEEE Internet Computing, 5*(2), 18–26.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control, 8*, 338–353.

Zamir, O., & Etzioni, O. (1998). Web document clustering: A feasibility demonstration. *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 46–54.

Zamir, O., & Etzioni, O. (1999). Grouper: A dynamic clustering interface to Web search results. *Proceedings of the 8th World Wide Web Conference*. Retrieved February 20, 2003, from http://www8.org/w8-papers/3a-search-query/dynamic/dynamic.html