22

Text and Web Mining

Objectives:

- Text and Web mining are the technologies analyzing less structured data having the goal to extract knowledge from text databases and from Web log files.
- The data warehouse provides the ideal structure for data mining and knowledge discovery, so the World Wide Web, with its lack of structure, provides the greatest technical challenge for those who would use AI and statistical methods to glean knowledge from data.
- To discover hidden patterns and relationship within the Web data, Web-mining task performs discovery of association rules, discovery of sequential patterns, and discovery of classification rules and data clusters.
- The two domains that pertain to Web mining are Web contents mining and Web usage mining.
- Web mining helps organization determine the lifetime value of customers and cross marketing strategies across product and identify population of potential customers for electronic commerce.
- The major strengths of the WeblogMiner are its scalability, interactivity, and the variety and flexibility of the analyses possible to perform.
- Web servers register a (Web) log entry for every single access they get in which they save the URL requested, the IP address from which the request originated, and a timestamp. To reduce the size of the log files to analyze Weblog analysis tools make assumptions in order to filter out some data like failed requests (i.e., errors) or page graphic requests, or to round off the log entries by combining similar requests.
- In multidimensional data cube, to reduce the data cube construction time, a bridge is built between the Web server and a data cube that would propagate access information directly to the aggregated cells of the data cube.

S. Sumathi and S.N. Sivanandam: *Text and Web Mining*, Studies in Computational Intelligence (SCI) **29**, 545–589 (2006)
www.springerlink.com

- Web Usage Mining is the "automatic discovery of user access patterns form Web servers". Examples of tools include Web SIFT, Web Utilization Miner, and Easy Miner.
- In biomedical literature, the first one is an information retrieval task, which retrieves useful MEDLINE documents for biological research with high precision, and the second is a text (data) mining task, which applies association rule mining and term extraction for mining MEDLINE.
- Science & Technology (S&T) text mining is the application of text mining to highly detailed technical material. There are over seventy text mining tools available on the Internet.

Abstract. An Overview on Web mining, WebLogMiner, and Web Usage Mining on Proxy Servers and Text Data Mining in Biomedical Literature Case Studies are described in this section.

The World Wide Web serves as a huge, widely distributed, global information service center for news, advertisements, consumer information, financial management, education, government, e-commerce, and many other information services. The Web also contains a rich and dynamic collection of hyperlink information and Web page access and usage information, providing rich sources for data mining.

Most previous studies of data mining have focused on structured data, such as relational, transactional, and data warehouse data. However, in reality, a substantial portion of the available information is stored in text databases (or document databases), which consist of large collections of documents from various sources, such as news articles, research papers, books, digital libraries, e-mail messages, and Web pages. Text databases are rapidly growing due to the increasing amount of information available in electronic forms, such as electronic publications, e-mail, CD-ROMs, and the World Wide Web (which can also be viewed as a huge, interconnected, dynamic text database).

Data mining and knowledge discovery has become a very useful technique in both research and everyday life. As more and more messages are published on the World Wide Web, the need for extracting important, useful information from the Web is increasing. Data mining in Web, or called Web mining becomes a very hot topic.

Data stored in most text databases are semistructured data in that they are neither completely unstructured nor completely structured. For example, a document may not only contain a few structured fields, such as title, authors, publication date, length, category, and so on, but also contain some largely unstructured text components, such as abstract and contents. There have been a great deal studies on the modeling and implementation of semistructured data in recent database research. Moreover, information retrieval techniques such as text indexing methods have been developed to handle unstructured documents.

Traditional information retrieval techniques become inadequate for the increasingly vast amounts of text data. Typically, only a small fraction of the many available documents will be relevant to a given individual or user. Without knowing what could be in the documents, it is difficult to formulate effective queries for analyzing and extracting useful information from the data. Users need tools to compare different documents, rank the importance and relevance of the documents, or find patterns and trends across multiple documents. Thus, text mining has become an increasingly popular and essential theme in data mining.

Web Usage Mining is an aspect of data mining that has received a lot of attention in recent years. An area that has received much less attention is the investigation of user behavior on proxy servers. Servers of Internet Service Providers (ISPs) log traffic from thousands of users to thousands of Web sites. No doubt that Web server administrators are interested in comparing the performance of their own site with those of competitors. Moreover, this kind of research can give a general overview of user behavior on the Internet or an overview of behavior within a specific sector.

Biological literature databases such as MEDLINE contain rich information required for biological research, currently there is great demand for extracting information from such databases. This section attempts to focus on two issues. The first one is an information retrieval task, which retrieves useful MEDLINE documents for biological research with high precision, and the second is a text (data) mining task, which applies association rule mining and term extraction for mining MEDLINE.

22.1 Data Mining and the Web

Text and Web mining are technologies analyzing less structured data having the goal to extract knowledge from text databases and from Web log files. Good examples of business-related text-mining applications in the domain of efficient handling with the textual information overload use visualization techniques, automatic building of onthologies (subject taxonomies), predicting financial trends from the text from news wires, information extraction from document databases, span filtering, document summarization and text categorization. Typical Web-mining application is in the area of customer profiling from the Web log files analyzing customer trace and historical information with the goals as increase sales of products at Web site. Internet opens wide possibilities to access the remote database. Due to considerable interest of the society to ecology and its vital activity it is reasonable to supply Internet with databases.

If the data warehouse provides the ideal structure for data mining and knowledge discovery, then the World Wide Web, with its lack of structure, provides the greatest technical challenge for those who would use AI and statistical methods to glean knowledge from data. Oren Etzioni in 1996 discussed these challenges as well as some techniques used in overcoming them in several deployed Web based systems. Etzioni in 1997 convincingly argues that the lack of structure that characterizes the Web is only apparent: Large portions of the Web are multilayered sites with data warehouse like structures (on-line catalogs of merchandise); other portion of the Web have very characteristics features (home pages); still other portions are partially labeled by HTML annotations like <title> as well as the linguistic and typographic conventions of files in natural language, postscript, latex, and all Web servers have a domain name that serves to partially limit what they might contain. These last points are most important for systems that learn about the Web by coming to understand common tags and the content associated with certain domain names. In short, Etzioni believes, as this writer certainly does that the Web is more a "gold mine" than a "quagmire."

In these systems, what Etzioni and his colleagues have deployed fall into three board classes that are enumerated as: (a) Resource discovery: locating unfamiliar documents and services on the Web – here the focus is on search: (b) Information extraction: "automatically extracting specific information from the Web resources" – here the focus is on understanding: and (c) Generalization: "uncovering general patterns at individual Web sites and across multiple sites" – here the focus is on learning. All three areas – search, understanding, and learning – are classical AI tasks. And, as for the last of these, learning, the Web-based systems discussed differ from those that merely interact with the *user* to learn his preferences and then search or act on his behalf; *these* systems all learn about the Web *itself* and do so by various methods, including interaction and experience. In that sense they are more genuinely intelligent.

22.1.1 Resource Discovery

Softbots are intelligent software agents – software robots – and one thing any robot wroth the name must be able to do in use tools. METACRAWLER (http://www.metacrawler.com) is a softbot that queries several Web search engines in parallel and produces the different syntax of different search engines and the need to query multiple search engines and prune the results. The main advantage of METACRAWLER is that it uses the multimillion document. Web indices and directories as *tools,* thus eliminating huge storage requirements and acting as essentially a gateway or interface with some intelligent processing. When run on one name, METACRAWLER gives almost number false positives, and also misses a great deal. (In contrast, METAFIND (http://www.metafind.com) – the favorite metasearch engine – and DOGpile (http://www. dogpile.com) are a good deal more comprehensive, but abound with false positives.

22.1.2 Information Extraction

NETBOT JANGO (http://www. Jango.com) will in response to a query visit manufacturer sites for product descriptions, magazine sites for product reviews and vendor sites for price quotes. "JANGO is intelligent enough to create accounts with passwords for vendors and recall them for later use and can also be instructed to make a purchase using the user's billing and shipping, information." As this is written, NETBOT, JANGO is in between versions, with number version available for testing.

22.1.3 Generalization

AHOY! (http://ahoy.cs.washington.edu: 6060) is a softbot that learns what a home page is and where to find them; it can locate home pages with both sensitivity and specificity, given at least some of: e-mail address, affiliation, country. It worked poorly on the test set – the e-address book – we used, failing not only on such typical cases as URLs beginning with ftp://, but also on such obvious cases as http://www.domain.com/-Name/when given name @ domain.com as the e-mail address, but number institutional affiliation. It worked best on home pages for those with academic affiliations. A disclaimer Etzioni in 1996 makes that: None of these systems (except AHOY! which is probably not good enough to market) is today exactly as described by Etzioni and his colleagues in the literature; they have been licensed to Internet companies and significantly changed by their proprietors.

22.2 An Overview on Web Mining

Web mining is the application of data mining technologies to huge Web data repositories. To discover hidden patterns and relationship within the Web data, Web mining task can be divided into 3 general categories that use these data mining technologies: discovery of association rules, discovery of sequential patterns, and discovery of classification rules and data clusters. Basically, there are two domains that pertain to Web mining Web contents mining and Web Usage Mining.

The World Wide Web is continuously growing and "collecting" all kinds of resources. Despite the anarchy in which it is growing, the Web is one of the biggest repositories ever built. As a confluence of data mining and world Web technologies, analyzing and exploring regularities using data mining in Web user behavior can improve systems performance and enhance the quality and delivery of the Internet information services to the end user. It can also help an organization to determine the lifetime value of customers and cross marketing strategies across product and identify population of potential customers for electronic commerce. Analysis of how users are accessing a site is critical for determining effective marketing strategies and optimizing the logical structure of the Web site. For selling advertisements on the World Wide Web, analyzing user access patterns helps in targeting ads to specific groups of users.

Characteristics of Web documents make it not easy for automated discovery of Web-based information:

- The lack of structure of information sources and unique characteristics of the client-server model in the World Wide Web, including
- Differences between the physical topology of Web repositories and user access paths;
- Difficulty in identification of unique users as well as user sessions or transactions.

However, even though the Web is dynamic and unordered, it provides many examples of semistructures as linguistic convention, semistructured documents (e.g., catalogs), Web directories, HTML tags (e.g., <title>) etc.

22.2.1 Taxonomy of Web Mining

Web Content Mining

Web content mining is the process of *extracting knowledge* and *information discovery* from sources across the World Wide Web. Web content mining uses two major approaches: agent-based and database. The goal is to provide structural information, categorize, filter, and interpret documents.

Agent-based Approach

Web agents are intelligent tools for information retrieval and extending data mining techniques to provide a higher level for organization for semistructured data available on the Web.

Intelligent search agents search for relevant information using domain characteristics and user profiles to organize and interpret the discovered information.

Information Filtering/Categorization

A number of Web agents (HyPursuit and Bookmark Organizer) use various information retrieval techniques and characteristics of open hypertext Web documents to automatically retrieve, filter, and categorize them.

Personalized Web agents

Web agents in this category learn user preferences and discover Web information sources based on these preferences, and those of other individuals with similar interests (using collaborative filtering).

22.2.2 Database Approach

Multilevel Databases

At higher level(s) metadata or generalizations are extracted from lower levels, which contain semistructured information stored in various Web repositories such as hypertext documents, and organized in structured collections, i.e., relational or object-oriented databases.

Web Query Systems

Many Web-based query systems and languages utilize standard query language such as SQL, structural information about Web documents, and even natural language processing for the queries that are used in World Wide Web searches.

Characteristics of Web Query Systems

Web Query Systems	**Characteristics**
W3QL	Combines structure queries, based on the organization of hypertext documents, and content queries, based on information retrieval techniques.
WebLog	Logic-based query language for restructuring extracts information from Web information sources.
Lorel and UnQL	Query heterogeneous and semistructured information on the Web using a labeled graph data model.
TSIMMIS	Extracts data from heterogeneous and semistructured information sources and correlates them to generate an integrated database representation of the extracted information.

Web Usage Mining

Web Usage mining, also known as Web log mining, is the process of *discovering interesting patterns* in Web access logs. Other sources of user information include *referrer logs,* which contain information about the referring pages for each page reference and user registration.

More sophisticated systems and techniques for discovery and analysis of patterns are now emerging. These tools can be placed into two main categories, as discussed below.

Pattern Discovery Tools

The emerging tools for user pattern discovery use sophisticated techniques from AI, data mining, psychology, and information theory, to mine to knowledge from collected data. For example, WEBMINER (Mobasher et al., 2000; Cooley et al., 1999) automatically discovers association rules and sequential patterns from server access logs.

Pattern Analysis Tools

Once access patterns have been discovered, analysts need the appropriate tools and techniques to understand, visualize, and interpret these patterns, such as Web Viz system (Pitkow and Bharat, 1994). Some tools have proposed using OLAP techniques such as data cubes for the purpose of simplifying the analysis of usage statistics from server access mechanism for querying the discovered knowledge.

22.2.3 Web Mining Tasks

- Data preprocessing for Web mining
- Mining patterns and discovering knowledge using association rule mining, time sequential mining, and classification and clustering mining
- Analysis of mined knowledge

Web Usage Mining Architecture

The architecture divides the Web usage mining process into two main parts. The first part includes the domain dependent of transforming the Web data into suitable transaction form. This includes preprocessing, transaction identification, and data integration components. The second part includes the largely domain independent application of generic data mining and pattern matching techniques as part of the system's data mining engine.

Data Pre-processing

Web usage data collected in various logs is at a very fine granularity. There is a need to group individual data collection events into groups, called *Web transactions,* before feeding it to the mining system. Clearly improved data quality can improve the quality of any analysis on it. Portions of Web usage data exist in sources as diverse as *Web server logs, referral logs, registration files,* and index server logs. Intelligent integration and correlation of information from these diverse sources can reveal usage information that may not be evident from any one of them. Hence, there is a need to preprocess the data to make it easier to mine to knowledge.

Pattern/Knowledge Mining

The key component of Web mining is the mining process itself. Web mining has adapted technique from the field of data mining, databases, and information retrieval as well as developing some techniques of its own, e.g., *path analysis.* Web usage mining studies reported to date have mined for *association rules, temporal sequences, clusters,* and *path expressions.* As Web continues to expand, there is a continual need to figure out new kinds of knowledge about user behavior that needs to be mined.

Mined Knowledge Analysis

The output of knowledge mining algorithms is often not in a form suitable for direct human consumption, and hence there is a need to develop tools that incorporate statistical methods, visualization, and human factors to help better understand the mined knowledge. Web mining, in particular, is the creation of intelligent tools that can assist in the interpretation of mined knowledge. Clearly, these tools need to have specific knowledge about the particular problem domain to do any more than filtering based on statistical attributes of the discovered rules or patterns.

22.2.4 Mining Interested Content from Web Document

The World Wide Web is a collection of electronic documents whose size is growing exponentially. This makes it difficult for users to find useful information. The Web provides an enormous resource for many informational topics but does not provide a good means to find the information relevant to specific user interests. The main problem is related to the difficulty of locating and tracking appropriate sources from the enormous number of unstructured documents.

Web Mining Environment (WME)

WME is capable of finding, extracting, and structuring information related to a particular domain from Web documents using general-purpose indices. This approach presents a tool for the automatic translation of a domain specification into a Web miner able to retrieve data in that domain on the Web. Further researches are now focusing on machine learning to improve extraction capability and automatic domain normalization.

To judge the performance, WME was compared to different search engineer such as Alta Vista, Yahoo, and MetaCrawler in calculating the recall measure and precision measure for the extraction task. As a result, WME returns a higher percentage of documents than Yahoo and MetaCrawler with well semistructured domains.

- The WME Architecture

The main phases needed by a WME to satisfy a user query are:

(i) Domain page retrieval
(ii) Retrieved domain page structuring
(iii) Information extraction from a structured page
(iv) User presentation of the results and local database population

The Web engine returns a first approximation of a set of Web pages that might contain the requested information. A WME uses these descriptions, by which Web page addresses are followed, and a set of *weighted string patterns* to sort the addresses in decreasing order with respect to a content-base interest rate. The *weighted string patterns* are defined by the WME designer that creates a set of string pattern templates and assigns to each of them a weight based on his knowledge of the domain. String patterns may also be created at search time by the WME by using user query values. WME designer also defines a threshold used by the WME to extract the most "interesting" pages from the ordered set. The second phase classifies and qualifies text segments in a page based on the structural characteristics of it. Once a page has been structurally analyzed the phase 3 extracts the requested information by first locating the text segments of interest and then extracting the text strings meeting the user query. In the last phase of entity values are presented to the user in the appropriate fields for the query-by-example interface. Moreover, they are inserted into the local database in order to provide fast retrieval to successive queries and to allow data analysis functions to derive useful information.

- Design of a WME Generator (WMEG)

The WMEG processes the Miner Attributes and instantiates the template source codes. The WME developer specifies all the parameters by using a *graphical interface.* These parameters form the *Miner Attributes,* and are the input for the *code generator.* External extraction functions, with some template files, are used as additional inputs.

Relational Fuzzy C-Maximal Density Estimator (RFC-MDE)

RFC-MDE is an approach for automatic discovery of user session profiles in Web log data. It is an extension of Maximal Density Estimator (MDE), which yields a robust estimate of the parameters without assuming that the contamination rate is known. A "user session" is defined as a temporally compact sequence of Web accesses by a user. A new distance measure is defined between two Web sessions that captures the organization of a Web site. The goal of RFC-MDE is to categorize these sessions.

Most data mining applications involve data that is corrupted with noise. Existing robust estimators such as, MDE, suffer from their strong dependence of a known or an assumed amount of noise, or equivalently an estimated scale value. RFC-MDE can perform fuzzy clustering to the sessions with an unknown amount of noise. The RFC-MDE algorithm was successfully used to cluster the sessions extracted from real server access logs into typical user session profiles, and even to identify the noisy sessions and profiles. As a by-product of the clustering processes, associations between different URL addresses on a given site can easily be inferred from the resulting robust profiles.

22.2.5 Mining Pattern from Web Transactions/Logs

Using Web log files, studies have been conduced on analyzing system performance, improving system design, understanding the nature of Web traffic, and understanding user reaction and motivation.

Discovery and analysis of various data relationships is essential in fully utilizing the valuable data gathered in daily transactions. A comprehensive analysis tool must be able to automatically discover such data relationships, including the correlation among Web pages, sequential patterns over time intervals, and classification of users according to their according to their access patterns. This technique must be able to discover relationships in very high traffic servers with very large access logs.

Open Market (e.g., Open Market Inc. 1996), Web Trend (e.g., Software Inc., 1995), and NetGenesis (net. Genesis, net. Analysis desktop., 1996) are Web analysis tools to determine the number of accesses to the server and the individual files within the organization's Web space, the times or time intervals of visits, and domain names and the URLs of users of the Web server. However, these tools are designed to deal with low-to-moderate traffic servers, and they usually provide little or number analysis of data relationships among the accessed files and directories within the Web space.

WEBMINER

WEBMINER is a framework for Web transaction mining to discover association rules and sequential patterns from data collected in World Wide Web transactions. This framework includes formal data transaction models for various Web mining tasks. It can be used in very high-traffic servers with very large access logs. In order to provide a greater degree of user control, a query language has been built on top of the basic engine. The user can specify the type of pattern to look for, and only those are retrieved. The system tends to be extended to incorporate components for clustering and the discovery of classification rules.

- Structure of access log data

According to the Common Log Format specified as part of the HTTP protocol, a log entry contains the client IP address, user id, access time, request method, and the URL of the page accessed, the protocol used for data transmission, an error code, and the number of bytes transmitted. A sample entry from a Web server access log is listed as following:

Moose.cs.umn.edu mobasher – [09/Aug/1996:09:55:50 – 05001] "GET/ ~suharyon/lisa.html HTTP/1.0" 200 654

There are a variety of files accessed as a result of a request by a client to view a particular Web page. Typical examples include:

<filename>.html: the HTML file for the requested page;
<filename>.gif.or <filename>.jpg: image files.
<filename>.map?<x,y>: file mapped to coordinates x and y of an image map file;
<program>.cgi?<arguments>: a server-side executable file.

- Data Cleansing

The primary objective of Web mining is to discover interesting patterns in accesses to various Web pages within the Web space associated with a particular server. According to the task, all irrelevant and redundant log entries such as, image file without a hyper link, image map files, and other multimedia support files can be removed. Data cleansing is performed by checking the suffix of the URL name. All the URL entries with filename suffixes such as, gif, jpeg, JPEG, jpg, JPG, and map are removed from the log. Then the data is formatted appropriately according to the application.

- Data and Transaction Model for Association Rules

Association rule mining is to discover all associations and correlation among data items where the presence of one set of items in a transaction implies (with a certain degree of confidence) the presence of other items. In the context of Web mining, this problem amounts to discovering the correlation among accesses to various files available on the server by a given client. In WEBMINER, each transaction is based on the set of all log entries belonging to the same client (IP address and user id), within a given maximum time gap (provided by the user).

- Data and Transaction Model for Sequential Patterns

Sequential mining is to find intertransaction patterns such that the presence of a set of items is followed by another item in the time-stamp ordered transaction set. Using the temporal characteristics of the data, another important kind of data dependency that can be discovered are similar time sequences. For example, we may be interested in finding common characteristics of all clients that visited a particular file within the time period $[\mathrm{t}_1,\ \mathrm{t}_2]$; we may be interested in a time interval (within a day, or within a week, etc.) in which a particular file is most accessed. In WEBMINER, each transaction is defined as a set of all the URL names and their access times for the same client within a user-specified maximum time gap.

- Clustering and Classification

Classification allows one to develop a profile for items belonging to a particular group according to their common attributes. This profile can then be used to classify new data items that are added to database.

In Web mining, classification techniques allow one to develop a profile for clients who access particular server files based on demographic information available on those clients. Much of this information on clients can be obtained by analyzing client requests and the information transmitted by the client browser, including the URL. The relationships can be discovered such as: clients who often access URL_1 tend to be from educational institutions; or clients who placed an on-line order in URL_2, tend to have previously visited the site for Company X, etc.

After obtaining profile information on clients and discovering classifications of data items (server files), it is obviously easy to cluster clients or data items that have similar characteristic together. It can facilitate the development and execution of future marketing strategies.

In a clustering and classification task, the definition of transaction depends on the process. If the process is defined to be clustering based on individual URL names, and *clustering transaction* can be defined to be the same as that of association rules with infinity as a max time gap (in the case of sequential patterns, max. time gap corresponds to the definition of window size).

In classification task, we add additional attributes provided by user to the attributes defined from the clustering task. These additional attributes come from user registration, on-line survey forms, and techniques such as "anonymous ticketing."

22.2.6 Web Access Pattern Tree (WAP tree)

WAP tree is developed for efficient mining of access patterns from pieces of logs. A Web access pattern is a sequential pattern in a large set of pieces of Web logs, which is pursued frequently by users. The novel data structure, WAP trees, stores highly compressed, critical information for access pattern mining and facilitates the development of novel algorithms for mining access patterns in large set of log pieces.

Comparing the scalabilities of WAP mine and GSP, the algorithm proposed in R. Srikant, R. Agrawal, ACM SIGMOD Canada, 1996, with threshold as well as the number of access sequence in the database, the result shows that WAP mine outperforms GSP in quite significant margin, and WAP mine has better scalability than GSP. The success of WAP tree and WAP mine can be credited to the compact structure of WAP tree and the novel *conditional search* strategies.

- Construction of WAP tree

The WAP tree is an effective data structure. It registers all count information for pattern mining, and frees the mining, and frees the mining process from counting candidates by pattern matching. The conditional search strategies narrow the search space efficiently and make best use of WAP-tree structure. It avoids the overwhelming problems of generating explosive candidates in *Apriori*-like algorithms.

It is easy to show that a WAP tree can be partitioned and structured in the form similar to B+ tree, and can be implemented evening pure SQL. Therefore, WAP tree as well as mining use WAP tree is highly scalable.

- WAP mine

Access patterns can be mined sequential pattern mining techniques. Almost all previously proposed methods for sequential pattern mining are based on a sequential pattern version of *Apriori heuristic.* The essential structure of the WAP min algorithm is as follows:

- Mining WAP from WAP tree

The WAP tree structure constructed by Algorithm 1 provides some interesting properties that facilitate mining Web access patterns. WAP tree uses *conditional search,* instead of searching all Web access patterns with same suffix, to mine all Web access patterns. As the suffix becomes longer, the remaining search space becomes smaller potentially.

WebLogMiner

WebLogMiner (Zaiane et al., 1998) is a knowledge discovery tool, which benefits from OLAP and data mining techniques, and multidimensional data cube, to interactively extract implicit knowledge from very large Web log files. Concrete examples using these techniques were given for time-series pattern analysis. The major strengths of this design are its scalability, interactivity, and the variety and flexibility of the analyses possible to perform.

WebLogMiner is not limited by the huge sizes of the log files continuously expanding, the type of data collected in the Web logs, and the techniques used to analyze this data. Despite these strengths, the discovery potential of such a design is still limited due to the current impoverished Web log files. The implementation of a WebLogMiner prototype is still processing. The plan to integrate the technology developed for data mining system DBMiner in WebLogMiner and use DBMiner's sparse data cube capabilities, and develop additional modules for time-series pattern analysis is continuing.

22.3 Text Mining

22.3.1 Definition

Text mining should not be confused with the better known Internet search engine tools or database management capabilities. Analogous to data mining, which extracts useful information from any type of data with large quantities, text mining is a procedure applied to large volumes of free unstructured text. After a traditional search for documents is completed, such as in format of full text, abstracts, or indexed terms, text mining explores the complex relationship among documents.

Science & Technology (S&T) text mining is the application of text mining to highly detailed technical material (Kostoff). There are three major components of S&T text mining.

(1) Information Retrieval, the foundational step of text mining. It is the extraction of relevant records from the source technical literatures or text databases for further processing.
(2) Information Processing, the extraction of patterns from the retrieved data obtained in the previous step. According to Kostoff, it has three components: bibliometrics, computational linguistics, and clustering techniques. This step typically provides ordering, classification, and quantification to the formerly unstructured material.
(3) Information Integration. It is the combination of the information processing computer output with the human cognitive processes.

22.3.2 S&T Text Mining Applications

There are several existing and potential text mining applications.

(1) Retrieving Documents

Text mining can be used to improve the comprehensiveness and relevance of information retrieved from databases. Related research focuses on different techniques for their comprehensiveness and S/N of records retrieved. Most high-quality methods use some type of iterative method with relevance feedback to modify the initial test query for increased comprehensiveness and precision of the records retrieved.

(2) Identify Infrastructure

Text mining can be used to identify the elements of the infrastructure of a technical discipline. These infrastructure elements are the authors, journals, organizations, and other group or facilities that contribute to the advancement and maintenance of the discipline. Additionally, text mining can provide their specific relationships to the total technical discipline or to subdiscipline areas.

(3) Identify Technical Themes/Relationships

Text mining can be used to identify technical themes, their inter-relationships, their relationships with the infrastructure and technical taxonomies through computational linguistics. By categorizing phrases and counting frequencies, S&T text mining can estimate adequacies and deficiencies of S&T in subtechnology areas.

(4) Discovery from Literature

There are different kinds of literature-based discovery: examining relationship between liked, overlapping literatures, and discovering relationships or promising opportunities that would not be found when read separately. Successful performance of literature discoveries can lead to identification of promising new technology opportunities and research directions, such as extrapolation of ideas from on discipline to a disparately related discipline.

(5) Technology Forecasting

In the process of retrieving and relating useful text data, text mining can also provide the time series for trend extrapolation. As an extension of the process, text mining can be used to identify state-of-the-art research & development (R&D) emphases and portend future development.

22.3.3 Text Mining Tools

There are over seventy text mining tools available on the Internet. A thorough documentation written by van Gemert lists 71 text mining tools with brief description of each of the company and its product. One of them is VantagePoint, which was developed by Search Technology, Inc., with the technology assistance of Dr. Alan Porter, the director of Technology Policy and Assessment Center (TPAC) at Georgia Tech.

Innovation Indicators

As mentioned before, bibliometrics is one of the text mining techniques to capture desired information. It uses counts of publications, patents, or citations to measure and interpret the advance of technologies. These counts can then be reasoned as innovation indicators of a certain technology.

Innovation indicators collect information on technology life cycle status, innovation contextual influence, and product market potential concepts. Examples of research and development (R&D) innovation indicators can be obtained by the following bibliometrics measures (TPAC).

(1) Fundamental Research—number of items in databases such as Science Citation Index
(2) Applied Research—number of items in databases such as Engineering Index
(3) Development—number of items in databases such as U.S. Patents
(4) Application—number of items in databases such as Newspaper Abstracts Daily
(5) Societal Impacts—issues raised in the Business and Popular Press abstracts

These innovation indicators can be defined along with the identification of technical themes, where co-occurrence of clustering analysis is required. Therefore, combined with text mining technique, innovation indicators can be generated as aids to show the maturity level of technology, similar in concept to Technology Readiness Levels (TRLs).

Innovation Flow Mapping

Innovation flow mapping is a technique to model the influences on or drivers of technology development in a graphical manner. It can be used as a brainstorming tool in the early stage of planning or examining the prospects for a technology and whether the institutions and the organizational capability exist to complete the development. Similar to an *Interrelationship Diagraph*, an innovation flow map consists of blocks of identified technologies/sources and cause/influence relationship arrows in between blocks. It can depict the location of research domain relative to each other and institutional interest and overlaps.

Competitive Technological Intelligence (CTI)

Competitive technological intelligence (CTI) is another use of text mining techniques. It grew aggressively in the 1990s when companies, universities, and government agencies were in need of knowing what capabilities others have of developing a particular technology. An essential part of CTI comes from topical searches in R&D databases of the text mining process.

CTI analysis can be done between market competitors or countries. An example by Porter and Detampel (1995) is a comparison of the patenting frequency in several areas related to multichip modules for the top two patenters, IBM and AT&T.

22.3.4 Text Data Mining

Data "mining" is not a very good metaphor for what people in the field actually do. Mining implies extracting precious nuggets of ore from otherwise worthless rock. If data mining really followed this metaphor, it would mean that people were discovering new facts within their databases/data warehouses. In practice, data mining applications tend to be (semi-)automated discovery of trends and patterns across very large datasets, usually for the purposes of decision making. But in the case of text, we can take the mining-for-nuggets metaphor seriously.

Text Data Mining vs. Information Retrieval

The goal of IR is to help users find documents that satisfy their information needs. Like looking for needles in a needle stack – the problem is not so much that the desired information is not known, but rather that it co-exists with many other valid pieces of information.

The fact that an IR system can return useful documents does not imply that a new discovery has been made: the information had to have already been known by the author of the text, otherwise she could not have written it down. Text data mining means the extraction of new, never-before encountered information.

How can text data mining help IR?

1. Text clustering to provide thematic overviews of text collections.
2. Automatically generating term associations to aid in query expansion
3. Using co-citation analysis to find general topics within a collection or identify central Web pages.

A second goal of data mining on the Web is to analyze the transactions run in a Web-based system, be it to optimize the system or to find information about the clients in a system.

TDM and Computational Linguistics

Empirical computational linguistics computes statistics over large text collections in order to discover useful patterns. These patterns are used to inform algorithms for natural language processing, such as

a) part-of-speech tagging
b) word sense disambiguation (e.g., prices, prescription, and patent are likely to co-occur with the medical sense of drug; while abuse, paraphernalia, and illicit are likely to occur with the illegal drug sense of the word – Church & Lieberman)
c) bilingual dictionary creation (Armstrong, 1994). See example at the end.

A classification of data mining and text data mining applications

	Finding patterns	**Finding nuggets**	**Finding nuggets**
		Novel	**Non-novel**
Nontextual data	Standard data mining	-	Database queries
Textual data	Computational linguistics	Real text data mining	Information retrieval

Real TDM

1. One body of work uses text **category labels** (associated with Reuters newswire) to find “unexpected patterns” among text articles (Feldman & Dagan, 1995). The main approach is to compare distributions of category assignments within subsets of the document collection. For instance, distributions of commodities in country C1 are compared against those of country C2 to see if interesting or unexpected trends can be found.
2. **On-line new event detection**.
   The input is a stream of stories in chronological order, and whose output is a yes/no decision for each story, made at the time the story arrives, indicating whether the story is the first reference to a newly occurring event. In other words, the system must first detect the first instance of what will become a series of reports on some important topic.
3. **Using text to form hypotheses about disease.**
   Experts can only read a small subset of what is published in their fields and are often unaware about developments in related fields. It should be possible to find useful linkages between information in related literatures, if the authors of those literatures rarely refer to (are unaware of) each other’s work (Swanson, 1987).

For example, when investigating causes of migraine headaches, he extracted various pieces of evidence from titles of articles in the biomedical literature, e.g.

- stress is associated with migraines
- stress can lead to loss of magnesium
- calcium channel blockers prevent some migraines
- magnesium is a natural calcium channel blocker
- spreading cortical depression (SCD) is implicated in some migraines
- high levels of magnesium inhibit SCD
- migraine patients have high platelet aggregability
- magnesium can suppress platelet aggregability

These clues suggest that magnesium deficiency may play a role in some kinds of migraine headache – a hypothesis that did not exist at the time Swanson found these links, but was subsequently supported by medical tests.

This approach has only partly been automated. There is of course a potential for combinatorial explosion of potentially valid links. Sophisticated algorithms are needed for helping with the pruning process, since a good pruning algorithm will want to take into account various kinds of semantic constraints. This is an area of investigation.

4. Using text to uncover social impact.
   Narin et al. 1977 found that the technology industry relies more heavily than ever on government-sponsored research results. They explored relationships among patent text (science references given on the first page) and the original published research literature (looked in the acknowledgements to see who funded the research).
5. The LINDI Project.
   This investigated how researchers can use large text collections in the discovery of new important information.

An important problem in molecular biology is automating the **discovery of the function of newly sequenced genes.**

Human genome researchers perform experiments in which they analyze the co-expression of tens of thousands of novel and known genes simultaneously.

A gene a co-expresses with gene b when both are found to be activated in the same cells at the same time with much more likelihood than chance (*mutual information?*). Given this huge collection of genetic information, the goal is to determine which of the novel genes are medically interesting, meaning that they are co-expressed with already understood genes that are known to be involved in disease.

22.4 Discovering Web Access Patterns and Trends by Applying Olap and Data Mining Technology on Web Logs – Case Study

As a confluence of data mining and WWW technologies, it is now possible to perform data on Web log records collected from the Internet Web page access history. The behavior of the Web page readers is imprinted in the Web server log files. Analyzing and exploring regularities in this behavior can improve system performance, enhance the quality and delivery of Internet information services to the end user, and identify population of potential customers for electronic commerce. Thus, by observing people using collections of data, data mining can bring considerable contribution to digital library designers. We have been developing the knowledge discovery tool, WebLogMiner, for mining Web server log files. This section presents the design of the WebLogMiner, reports the current progress, and outlines the future work in this direction. This case study is taken from Virtual-U Research Laboratory and Intelligent Database Systems Research Laboratory, Simon Fraser University, Canada.

Web servers register a (Web) log entry for every single access they get in which they save the URL requested, the IP address from which the request originated, and a timestamp. With the rapid progress of World Wide Web (WWW) technology, and the ever-growing popularity of the WWW, a huge number of Web access log records are being collected. Popular Web sites can see their Web log growing by hundreds of megabytes every day.

Condensing these colossal files of raw Web log data in order to retrieve significant and useful information is a nontrivial task. It is not easy to perform systematic analysis on such a huge amount of data and therefore, most institutions have not been able to make effective use of Web access history for server performance enhancement, system design improvement.

Using Web log files, studies have been conducted on analyzing system performance, improving system design, understanding the nature of Web traffic, and understanding user reaction and motivation. One innovative study has proposed adaptive sites: Web sites that improve themselves by learning from user access patterns. While it is encouraging and exciting to see the various potential applications of Web log file analysis, it is important to know that the success of such applications depends on what and how much valid and reliable knowledge one can discover from the large raw log data. Currently, there are more than 30 commercially available applications for Web log analysis and many more are available free on the Internet (The University of Illinois maintains a list of Web access analyzers on a Hyper News page accessible at http://union, ncsa.uiuc.edu/HyperNews/get/www/loganalyzers, html).

The most frequent reports predefined by Web log analysis tools are: a summary report of hits and bytes transferred, a list of top requested URLs, a list of top referrers, a list of the most common browsers used, hits per hour/day/week/month reports, hits per domain report, an error report, a directory tree report, etc. Despite the fact that some of the reports can be customized with some of these tools, the majority of the currently available Web log analysis tools have rigid predefined reports. Most if not all of these Web log analysis tools have limitations with regard to the size of the Web log files, whether it is physical in size or practical in time because of the low speed of the analysis. To reduce the size of the log files to analyze Web log analysis tools make assumptions in order to filter out some data like failed requests (i.e., errors) or page graphic requests, or to round off the log entries by combining similar requests. Different assumptions are made for each of the Web log analysis tools resulting in the prospect of different statistics with the same log file. It has been reported that the analysis of the same Web log with different Web log analysis tools ended up with different statistic results.

Overall, the current Web log analysis tools are still limited in their performance, the comprehensiveness and depth of their analyses, and the validity and reliability of their results. The recent progress and development of data mining and data warehousing has made available powerful data mining and data warehousing systems. Many successful data mining systems can handle very large data files like the Web log files. However, we have not seen a systematic study and development of data warehousing and mining systems for mining knowledge from Web access log records. Recent research and development of data mining technology have promoted some studies on efficient data mining for user access patterns in distributed systems, referred to as mining path traversal patterns. Understanding user access patterns in a Web site using these mining techniques not only helps improve Web system design, but also leads to wise marketing decisions (e.g., putting advertisements in proper places, classifying users, etc.). However, mining path traversal patterns is still in its infancy.

In this section we propose to use data mining and data warehousing techniques to analyze Web log records. Based on our experience on the development of relational database and data warehouse-based data mining system, DBMiner, by the Intelligent Database Systems Research Laboratory at Simon Fraser University, and on the development of a Web based collaborative teaching and learning environment, Virtual-U, by the Virtual-U Research Laboratory at Simon Fraser University, we jointly study the challenging issues on data mining in Web log databases, and propose the WebLogMiner system, which performs data mining on Web log records collected from Web page access history.

22.4.1 Design of a Web Log Miner

The most commonly used method to evaluate access to Web resources or user interest in resources is by counting page accesses or “hits.” As we will see, this is not sufficient and often not correct. Web server log files of current common Web servers contain insufficient data upon which thorough analysis can be performed. However, they contain useful data from which a well-designed data mining system can discover beneficial information.

Web server log files customarily contain: the domain name (or IP address) of the request; the user name of the user who generated the request (if applicable); the date and time of the request; the method of the request (GET or POST); the name of the file requested; the result of the request (success, failure, error, etc.); the size of the data sent back; the URL of the referring page; and the identification of the client agent.

A log entry is automatically added each time a request for a resource reaches the Web server. While this may reflect the actual use of the resources on a site, it does not record reader behaviors like frequent backtracking or frequent reloading of the same resource when the resource is cached by the browser or a proxy. A cache would store resources and hand them to a client requesting them without leaving a trace in the log files. Frequent backtracking and reload may suggest a deficient design of the site navigation, which can be very informative for a site designer; however, this cannot be measured solely from the server logs. Many have suggested other means of data gathering like client-site log files collected by the browser, or a Java Applet. While these techniques solve problems created by page backtracking and proxy caching, they necessitate the user's collaboration, which is not always available. Until the Web server log files are enriched with more collected data, our data mining process is solely based on the information currently gathered by the Web servers.

Researchers working on Web log analysis discredit the use of Web access counts as indicators of user interest or measure of the interestingness of a Web page. Access counts, when considered alone, can be misleading metrics. For example, if one must go through a sequence of documents to reach a desired document, all documents leading to the final one get their counters incremented even if the user is not interested in them at all. The access counters alone do not account for the user's ability to access the information and the appropriateness of the information to the user. Nonetheless, when access counts are used in conjunction with other metrics, they can help infer interesting findings.

Despite the impoverished state of the server logs, much useful information can be discovered when using data mining techniques. The date and time collected for each successive request can give interesting clues regarding the user interest by evaluating the time spent by users on each resource, and can allow time sequence analysis using different time values: minutes, hours, days, months, years, etc. The domain name collected can allow practical classification of the resources based on countries or type of domain (commercial, education, government, etc.). The sequence of requests can help predict next caching those resources, or by allowing clustering of resources in a site based on user motivation. Notwithstanding, the server logs cannot be used as recorded by the Web server and need to be filtered before data mining can be applied.

In the WebLogMiner project, the data collected in the Web logs goes through four stages. In the first stage, the data is filtered to remove irrelevant information and a relational database is created containing the meaningful remaining data. This database facilitates information extraction and data summarization based on individual attributes like user, resource, user's locality, day, etc. In the second stage, a data cube is constructed using the available dimensions. On-line analytical processing (OLAP) is used in the third stage to drill-down, roll-up, slice, and dice in the Web log data cube. Finally, in the fourth stage, data mining techniques are put to use with the data cube to predict, classify, and discover interesting correlations.

22.4.2 Database Construction from server log Files

Data cleaning and data transformation the data-filtering step is a typical step adopted by many Web log analysis tools. While typically Web log analysis tools may filter out requests for page graphics (as well as sound and video) in order to concentrate on data pertaining to actual page hits, we tend to keep these entries because we believe they can give us interesting clues regarding Web site structure, traffic performance, as well as user motivation. Moreover, one user action can generate multiple server requests, and some of them are requests for page media. Some of these requests are important to deduce the intended action of the user. Another typical cleaning process consists of elimination log entries generated by Web agents like Web spiders, indexes, link checkers, or other intelligent agents that prefetch pages for caching purposes. We chose not to screen out these requests generated by the Web agents. It is often interesting and useful to analyze Web agents' behavior on a site and compare the traffic generated by these automated agents with the rest of the traffic. The data filtering we adopted mainly transforms the data into a more meaningful representation. We tend to consider most of the data are relevant and eliminate a minimal amount of data.

There are two types of data cleaning and data transformation, one that does not necessitate knowledge about the resources at the site and one that does. Cleaning the date and time field of the log entry, for instance, does not need any knowledge about the site itself. The date and time field is simply restructured in a set of fields to specify the day, month, year, hour, minute, and second. Filtering out server requests that failed or transforming server error cods is also generic. Transforming IP address to domain names is independent from the site content as well. However, associating a server request or a set of server requests to an intended action or event clearly necessitates knowledge about the site structure. Moreover, different dynamically generated Web pages can be the result of a single script, thus, an identical server request. Knowledge about the parameters provided to the script to generate the dynamic page, or knowledge about the necessary sequence in the request history before a request for a script, can be essential in disambiguating a server request and associating it to an event.

Metadata provided by the site designers is required for the knowledge-based data cleaning and transformation. The metadata consists of a mapping table between a server requests (URL) with parameters, if available, or a sequence of requests (URLs) and an event with a representative URL. The transformation process replaces the request sequence by the representative URL and adds the event tag to the log entry.

After the cleaning and transformation of the Web log entries, the Web log is loaded into a relational database and new implicit data, like the time spent by event, is calculated. The time spent by event (or page) is approximated from the difference between the time the page for the current event is requested and the time the next page is requested with an upper-bound threshold for the case when the user does not come back to the same server. This notion of time spent is an approximation of the actual perusal duration since it intrinsically includes the time for network transfer, navigation inside the page, etc. It may seem a biased metric but can be very useful comparing pages with the same design.

22.4.3 Multidimensional Web log data cube

Construction and manipulation after the data has been cleaned and transformed, a multidimensional array structure, called a data cube, is built to aggregate the hit counts. The multidimensional data cube has numerous dimensions (i.e., generally more than 3), each dimension representing a field with all possible values described by attributes. For example, the dimension URL may have the attributes: server domain, directory, file name, and extension; or the dimension time may have the attributes: second, minute, hour, day, week, month, quarter, year. Attributes of a dimension may be related by partial order indicating a hierarchical relationship among the dimension attributes. Hierarchies are generally predefined, but in some cases partitioning the dimension in ranges automatically generates these hierarchies. For example the dimension file size can be partitioned into size ranges and later grouped into categories like tiny, small, medium, large, huge.

Examples of dimensions in the Web log data cube include the following. Notice that each dimension is defined on a concept hierarchy to facilitate generalization and specialization along the dimension.

URL of the resource, where the concept hierarchy used is defined on the server directory structure; type of resource, defined on a range hierarchy; size of the resource, defined on a range hierarchy; time at which the resource was requested, defined on a time hierarchy; time spent in page, defined on a range hierarchy of seconds; domain name from which the request originated, defined on a prebuilt domain hierarchy; agent that made the request, defined on a prebuilt hierarchy of known Web agents and browsers; user, defined on a prebuilt user hierarchy, server status, defined on an error code hierarchy.

The multidimensional structure of the data cube provides remarkable flexibility to manipulate the data and view it from different perspectives. The sum cells allow quick summarization at different levels of the concept hierarchies defined on the dimension attributes.

Building this Web log data cube allows the application of OLAP (On-Line Analytical Processing) operations, such as drill-down, roll-up, slice and dice, to view and analyze the Web log data from different angles, derive ratios, and compute measures across many dimensions.

The drill-down operation navigates from generalized data to more details, or specializes an attribute by stepping down the aggregation hierarchy. For example, presenting the number of hits grouped by day from the number of hits grouped by month is a drill-down along the hierarchy time. The roll-up is the reverse operation of the drill-down. It navigates from specific to general, or generalizes an attribute by climbing up the aggregation hierarchy.

For example, the aggregation of total requests from group-by organization by day to group-by country by day is a roll-up by summarization over the server domain hierarchy.

The slice operation defines a subcube by performing a selection on one dimension by selecting one or some values in a dimension. It is a literal cut of a slice (or slices) on the same dimension. For example, the selection domain = ".edu" on the dimension server domain, is a slice on the educational internet domain. The dice operation is a set of consecutive operations on several dimensions. It defines a subcube by performing selections on several dimensions. For example, a subcube can be derived by dicing the Web log data cube on four dimensions using the following clause, country = "Canada" and month = 11/97 and agent = "Mozila" and file type = "cgi." These OLAP operations assist in interactive and quick retrieval of 2D and 3D cross tables and chartable data from the Web log data cube, which allow quick querying and analysis of very large Web access history files.

22.4.4 Data mining on Web log data cube and Web log database

On-line analytical processing and the data cube structure offer analytical modeling capabilities, including a calculation engine for deriving various statistics, and a highly interactive and powerful data retrieval and analysis environment. It is possible to use this environment to discover implicit knowledge in the Web log database by implementing data mining techniques on the Web log data cube. The knowledge that can be discovered is represented in the form of rules, tables, charts, graphs, and other visual presentation forms for characterizing, comparing, associating, predicting, or classifying data from the Web access log.

These data mining functions are briefly explained as follows.

Data Characterization

This function characterizes data in the Web log. It consists of finding rules that summarize general characteristics of a set of user-defined data. The rules are generated from a generalized data cube produced using the Web log data cube and the OLAP operations. For example, the traffic on a Web server for a given type of media in a particular time of day can be summarized by a characteristic rule.

Class Comparison

Comparison plays the role of examining the Web log data to discover discriminant rules, which summarize the features that distinguish the data in the target class from that in the contrasting classes. For example, to compare requests from two different Web browsers (or two Web robots), a discriminant rule summarizes the features that discriminate one agent form the other, like time, file type, etc.

Association

This function mines association rules at multiple levels of abstraction. For example, one may discover the patterns that accesses different resources consistently occurring together, or accesses from a particular place occurring at regular times.

Prediction

Prediction involves predicting values or value distributions of an attribute of interest based on its relevance to other attributes. Both relevance analysis and predictive model construction need statistical analysis techniques. This helps prediction of possible values of missing data or the value distribution of certain attributes in a set of objects. For example, the access to a new resource on a given day can be predicted based on accesses to similar old resources on similar days, or the traffic for a given page can be predicted based on the distribution of traffic on other pages in the server directory.

Classification

Classification consists of building a model for each given class based upon features in the Web log data and generating classification rules from such models. The models are constructed by analyzing a training Web log data set whose class label is known. The classification rules can be used to develop a better understanding of each class in the Web log database, and perhaps restructure a Web site or customize answers to requests (i.e., quality of service) based on classes of requests.

Time-series Analysis

Time-series analysis is to analyze data collected along time sequences, periodicity, and so on. It may also involve attribute relevance analysis, model construction, classification, and prediction. Thus time-series analysis explores most of the techniques developed in the above (data mining functions) plus its own techniques for time-sequence search, similarity analysis, periodicity analysis, and so on. For example, time-series analysis of the Web log data may disclose the patterns and trends of Web page accesses in the last year and suggest the improvement of services of the Web server. Since most data mining functions other than time-series analysis share many commonalities with the work in traditional data mining systems, such as IBM Intelligent Miner (Quest), Silicon Graphics MineSet, DBMiner, this section does not discuss the application of these data mining functions to Web log mining in detail. The focus is on time-series analysis because Web log records are highly time related, and the goals of data mining with Web log records are largely aimed at mining time-related patterns.

The time-series analysis includes network traffic analysis, event sequences and user behavior pattern analysis, transition analysis, and trend analysis. With the availability of data cube technology, such analysis can be performed systematically in the sense that analysis can be performed on multiple dimensions and at multiple systematically in the sense that analysis can be performed on multiple dimensions and at multiple granularities. Moreover, there are major differences in time-series analysis of Web log mining in comparison with other traditional data mining processes.

We take trend analysis as an example to illustrate such a process. In the analysis of the trend of Web accessing in the Virtual-U environment, we would like to see how a user changes his/her Web navigation behavior and focuses his/her attention to interested topics. The analysis takes the following steps.

1. *Data/user selection.* Since a Virtual-U user accesses the Virtual-U Web pages regularly, those who access the Virtual-U Web page only occasionally for curiosity will not be included in the analysis. That is, access regularity will be taken as a standard (or threshold) to filter out scattered accesses.
2. *Cycle detection.* For Web accessing, a regular Virtual-U user usually starts a Virtual-U session, traverses a set of page with possible inputs to some pages, and then leaves the Virtual-U sessions for a while or for a long time before coming back to start another session. Thus the starting or restarting of a Virtual-U Web page, following by a sequence of other local Web page accesses forms a cycle. A data mining task needs to detect such cycles effectively for meaningful analysis. We have developed techniques on how to find such cycles and detect periodically efficiently using data cube structure and OLAP technique.
3. *Trend analysis.* With the accumulation of the discovered sequences and periods, analysis can be performed on them to discover patterns and trends with different interests. One kind of patterns, which can be easily discovered, is the repetition of similar Web page accesses. For the same set of Web pages, many users will access such pages repeatedly in a similar pattern. Number of accesses will be associated with each page node to register such access frequency.

Moreover, with data cube technology, one can roll-up the access history to get general Web page accessing statistics. Such access statistics form a group of Web access trees/graphs. These access trees demonstrate some clear trend along the time axis: taking the structural complexity of the access tree as a focus, one can easily see the trend is the gradually reduced structural complexity of the access tree, which shows more experienced users are more focused on specific topics reduced number of random searches to other scattered topics.

Notice with the data cube technology, such a trend can be demonstrated with different time granularities, such as by week, biweek, month, bimonth, quarter, etc. by simple clicking the button of time axis. However, the trend will be “discovered” by a human with the help of visualization tools instead of a fully automated process.

It is evident now that Web server should collect and enter more information in their logs. It is urgent to study specific needs in Web access pattern analysis and recommend a new structure for Web access logs. This new structure would significantly simplify the data cleaning and data transformation stage. The experience showed us that the data cleaning and data transformation step is not only crucial, but also is the most time consuming. A good data filtering process needs metadata provided by Web site designers.

Due to the important size and the ever-exploding nature of the Web log files, the construction of the multidimensional data cube necessary for the on-line analytical processing and knowledge discovery, is very demanding and time consuming. To reduce the data cube construction time, we plan to build a bridge between the Web server and a data cube that would propagate access information directly to the aggregated cells of the data cube. The incrementally updated Web log data cube would allow real time Web access analysis and data mining.

22.5 Web Usage Mining on Proxy Servers: A Case Study

Web Usage Mining is an aspect of data mining that has received a lot of attention in recent years. Commercial companies as well as academic researchers have developed an extensive array of tools that perform several data mining algorithms on log files coming from Web servers in order to identify user behavior on a particular Web site. Performing this kind of Investigation on the Web site can provide information that can be used to better accommodate the user’s needs. An area that has received much less attention is the investigation of user behavior on proxy servers. Servers of Internet Service Providers (ISPs) log traffic from thousands of users to thousands of Web sites. It was a Belgian ISP that showed interest in the subject and consequently provided data from one of their proxy servers for a thesis. This section is a summary of that thesis and lays emphasis on the attained results. The ISP chose to remain anonymous because of privacy issues.

The Internet is generally said to have become available to a large public around 1994-1995. Since that time a great number of companies have thrown themselves on this new medium. In the beginning many entrepreneurs saw great new opportunities to make money by setting up an Internet company. Later on, some of the so-called *brick-and-mortar* companies began to see a need to go on-line. Some of those even changed their business so drastically that not much of the original company was left. Every large company has spent a lot of effort and money to develop a well-established Web site. The ones that have not spent enough effort may find themselves faced with strategic disadvantages in years to come.

In order to have a successful Web site (and especially a successful e-commerce site) it is crucial to know the users of that site. This need has given rise to a whole new field in research, called Web Usage Mining. It is commonly seen as a subdivision of Web Mining, which implies that data mining techniques are applied to data from the World Wide Web. When the data under consideration emerges from Web servers log files, we enter the field out Web usage mining. It is therefore the “automatic discovery of user access patterns form Web servers.”

Because it is so important to know one’s customers in order to better suit their needs, companies are willing to spend money on the analysis of their log files. As a consequence, apart from tools that were developed by academic researchers, there is simultaneously a significant number of commercial tools that have been developed to meet these needs. Examples of academic tools include WebSIFT (Robert Cooley University of Minnesota, 1999) and Web Utilization Miner (Myra et al., 1998). An example of a commercial Web usage mining tool is EasyMiner, developed by MINEit Software Ltd. All of these tools are designed to understand the most common log file formats so that the process requires very little preprocessing. Unfortunately, when analyzing a log file from a Web server, one can only analyze browsing behavior on a single site. To perform research on a sector or even on general browsing behavior, the log file data of a proxy server is a lot more appropriate because of the many-to-many relationship between sites and users.

22.5.1 Aspects of Web Usage Mining

As in the other form of data mining, in the Web usage mining it is equally important to pay attention to a proper data collection, a thorough preprocessing phase, and the data mining techniques themselves.

22.5.2 Data Collection

Data for Web usage mining can be collected at several levels. We may be faced with data from a single user or a multitude of them on one hand and single site or a multitude of sites. Combining both factors offers four possibilities, as illustrated in Fig. 22.1.

	1 Site	**Multiple sites**
user	Java applets or Javascripts	Modified browser
Multiple users	Server level	Proxy server level

Fig. 22.1. Segments of Web traffic

Data about behavior of a single user on a single site can be collected by means of JavaScript or Java applets. Both methods require user participation in the sense that the user has to enable their functionality. An applet has the additional problem that it may take some time to load the first time. However, it has the advantage that it can capture all clicks, including pressing the back or reload buttons. A script can be loaded faster but cannot capture all clicks.

A modified browser is situated in the second segment. It can capture the behavior of a single user over all visited Web sites. Its advantages over Java applets and JavaScript are that it is much more versatile and will allow data collection about a single user over multiple Web sites. That is why this kind of data collection is used regularly by market research groups, e.g., Nielsen//Netratings, in order to collect information on how certain user groups behave on-line.

The third way of data collection is on the Web server level. These servers explicitly log all user behaviors in a more or less standardized fashion. It generates a chronological stream of requests that come from multiple users visiting a specific site. Since Web servers keep record of these requests anyhow, this information is readily available. Sometimes an analyst will use some additional information to better identify users, such as information from *cookies* or socio-demographic information about the users that may have been collected. This kind of data collection also has a number of drawbacks. Like JavaScripts, it cannot capture page views that were generated by pressing back or reload buttons. Apart from that, it also cannot log page views generated by a cache, either a local cache on the computer of the user, or a cache from an ISP's proxy server.

The fourth level of data collection logs behavior of multiple users visiting multiple Web sites. This kind of information can be found in log files origination from proxy servers. These servers are used by ISPs to give customers access to the World Wide Web. They also function as a cache server. This means that they will keep pages that were recently requested on this server and, if the same request is made by another user shortly after that, they will send the cached page to that user, instead of requesting it once more on the Web server were that page is located.

22.5.3 Preprocessing

Preprocessing is an aspect of data mining of which the importance should not be underestimated. If this phase is not performed adequately, it is not possible for the mining algorithms to provide reliable results.

22.5.4 Data Cleaning

First of all, irrelevant data should be removed to reduce the search space and to skew the result space. Since the intention is to identify user sessions and build up out of page views, not all hits in a log file are necessary. This is true for server logs as well as for proxy logs. A log file generates a hit for every requested file. Since an HTML page may consist of several files (text, pictures, sounds, several frames) it would be useful if we could keep only a single hit for each page view. To get an idea of user behavior, it is only necessary to keep track of the files that the user specifically requested. Very often, all hits with a suffix like .jpg, .gif, .wav, etc. are removed out of the log file. Even though this will also be done in the research that is described later on, it also has a drawback. Sometimes users specifically want to see a picture on a separate page. This page view will be deleted while it should not.

22.5.5 User and Session Identification

After the log file has been cleaned, the next step is to identify users. This very often poses a serious problem. If every computer in the world has its own unique IP-address, there would not be a problem. However, most ISPs make use of *dynamic IP addresses.* This means that every time a user logs on to the Internet, he will be given different addresses. This makes it impossible to distinguish returning users. As a consequences, it is usually simply assumed that every new IP address is the same. This occurs when the agent log shows a change in browser software or operating system. Some sites try to solve the problem of user identification through the use of cookies that contain an identification number. However, users very often delete cookies or disable their use, which makes that this technique is not always reliable either. Other sites try to identify users by asking them for a login and password. It is clear, however, that not every site can do this since it very often scares users away.

Assuming that users have been identified, the next step is to identify sessions. The goal of session identification is to divide the page accesses of each user into individual sessions. A rule of thumb that is commonly used is that when there is an interval of 30 minutes between two page views, the click stream should be divided in two sessions. This rule has been applied since of timeout of 25.5 minutes was established, based on empirical data. This is why, in the further research it is carried out with a timeout of 30 minutes. After users and sessions have been identified, the file under consideration will have additional fields that mention for each line the number of users and of the session.

22.5.6 Data Mining Techniques

For the actual pattern discovery in Web usage mining, mostly the same techniques are employed as in other forms of data mining. The most common ones will be briefly described.

Log file analysis

Even though this is not a data mining technique as such, it is probably the most widely used technique to obtain structured information out of server logs.

There are a large number of tools on the market that will accept the most common log file formats as an input to answer some basic questions that every Web site administrator has. It will provide information such as: the number of hits and page views, the number of unique and returning users, the average length of a page view, an overview of the browsers and operating systems that were used, an overview of keywords that were used in search engines and that led to the Web site, etc. Despite lacking in the depth of its analysis, this type of knowledge can be potentially useful for improving the system performance, enhancing the security of the system, facilitating the site modification task, and providing support for marketing decisions.

Association Rules

In Web usage mining, association rules are used to find out which pages are frequently visited together. In the particular research carried out in this work, they will be used to find out which Web sites and which sectors are frequently visited together. An association rule is usually presented in the following syntax:

$$\text{KeynetBe} \Leftarrow \text{VtmBe}\,\&\,\text{TvlBe}\,(15 : \ 2.788\%,\ 0.27)$$

This rule means that out of the 15 instances (representing 2.788% of the database) that visited the sites of www. vtm.be and www.tv1.be together, 27% also visited www.ketnet.be. The *support* is 15, the *confidence* 27%.

Sequential Patterns

This technique has some similarities with the association rules and is used in this section in addition to those association rules. The difference is that it takes the time dimension into account. The algorithm tries to find sequences in which a certain page (or Web site) usually comes before or after another page (or Web site). In other words, it "attempts to find inter-session patterns such that the presence of a set of items is followed by another item in a time-ordered set of sessions or episodes."

Clustering

In general, clustering is a process of creating a partition so that all the members of each set of the partition are similar according to some metric. In Web usage mining, we can narrow the definition to a *technique to group users in clusters based on their common characteristics.* Clustering algorithms learn in an unsupervised way. They discover their own classes and subsets of related objects in the training set. Then it has to find descriptions that describe each of these subjects.

Classification

Contrary to clustering, classification is a supervised way of learning. The database contains one or more attributes that denote the class of a tuple and these are known as *predicted attributes*, whereas the remaining attributes are called *predicting attributes.* A combination of the predicted attributes defines a class. In the Web domain one is interested in developing a profile of users belonging to a particular class or category. For example, 45% of users who visit two or more sites of television stations in a single session are younger than 21. The algorithms that perform classification include decision tree classifiers, Bayesian classifiers, k-nearest neighbor classifiers, etc.

22.5.7 E-metrics

E-metrics is based on statistics, which is a data mining technique. Therefore, it can be considered to be a Web usage mining method like any other. Moreover, they also try to gain insight into browsing behavior of users and performance of Web sites (NetGenesis- www.netgenesis.com).

E-metrics is measured with which Web sites can be evaluated. They can be compared with regular metrics and ratios as these are used in traditional industry, such as return on investment, net profit, market share, rentability, etc. As Web sites gain a more important position in companies, there emerges a need to evaluate these Web sites – that consume more and more money – and quantify their performance. The intention is to give indications of how well the Web site performs in order to investigate to what extent these measures change over time and how well they perform compared to those of competitors.

Two kinds of e-metrics can be identified, those that can be applied to every Web site and those that were designed for a specific kind of Web site, very often e-commerce sites.

General e-metrics

In this section we briefly describe a few general e-metrics. However, both *stickiness* and *average duration* are explained in more detail because of their importance in the undertaken research.

Stickiness

This is probably one of the most widely used e-metrics. It is a composite metric that indicates the effectiveness with which the content of the page or the Web site can keep the attention of the user. In general it is assumed that sticky sites are better than sites that are less sticky. A possible formula is as follows.

$$\boxed{\text{Stickiness} = \text{Frequency}^{*}\ \text{Duration}^{*}\ \text{Total site reach}}$$

Where

$$\boxed{\begin{aligned}\text{Frequency} &= \frac{\text{Number of visits in time period T}}{\text{Number of unique users who visited in T}}\\ \text{Duration} &= \frac{\text{Total amount of time spent viewing all pages}}{\text{Number of visits in time period T}}\end{aligned}}$$

Where

$$\text{Total site reach} = \frac{\text{Number of unique users who visited in T}}{\text{Total number of unique users}}$$

This formula can be reduced to:

$$\boxed{\text{Stickiness} = \frac{\text{Total amount of time spent viewing all pages}}{\text{Total number of unique users}}}$$

so that one does not need to have all the data for the complete formula to calculate stickiness. Usually stickiness is expressed in minutes per user.

Average duration

This is quite a simple e-metric with which several pages of a Web site (or complete Web sites) can be compared to each other. The metric expresses how long users view a certain page or site on average. In this work the following formula has been used:

$$\boxed{\text{Average duration} = \frac{\text{Total duration of site (or page) X}}{\text{Total number of page views}}}$$

It is impossible to suggest an ideal value for this metric. For entire sites, the value should usually be as high as possible. For individual pages, it depends on the nature of that page. A navigation page should have a low value, which means that users easily find their way, while content pages should have a higher value.

Specific e-metrics

Apart from a vast amount of possible general e-metrics, there are also a great number of metrics specifically for e-commerce sites. These include:

Personalization index: This expresses to what extent data that were asked from the user to fill in on a form, are used to offer a personalized service. This value should be greater than 0.75.

Acquisition cost: This divides promotion costs (in the form of banners) by the number of click-throughs, so that the marketing team can discover to what extent the marketing efforts are effective to acquire users.

Cost per conversion: This divides promotion costs by the number of sales. It is the number that marketing people use to determine the best investment of their promotional budget.

RMF analysis: This is a special analysis in which customers are evaluated on three aspects: Recency (when was the last time they purchased something?), Monetary Value (how much money has he spent on the site already?), and Frequency (how frequent does the user purchase a good on the site?). All users are then placed in this three-dimensional model to find possible user segments.

22.5.8 The Data

The data used for research purposes in this section was offered to us by a Belgian ISP that chose to remain anonymous because of privacy issues. The data used here come from a proxy server that handles request of users that have a broadband Internet connection.

The log file

To offer Internet access, this ISP makes use of six proxy servers, all located in Belgium. Every user is assigned to one of those. As long as this user does not change his installation settings, he will always log on to the same server. This means that the log history of every user can always be found on the same server. This is a clear advantage for data mining research.

Unfortunately, the ISP does not make use of fixed IP addresses. Like most other ISPs, it uses a pool of dynamic IP addresses, which makes it more complicated to identify users. Fortunately, on-the-fly IP addresses—meaning that another address may be used for each file request—are not used here, since it would make this research quite impossible.

At the end of a session, the user can explicitly release his IP address so that it can be used for another user. However, most people do not do this and turn off their computer without doing so. In this case, their address is automatically taken back after 30 minutes, which is convenient because it is also the time interval that will be used to distinguish sessions with the same IP address.

Luckily, there are no time zones in Belgium, because if this were so, it would have to be solved somehow. Also, the server in this research is not load balanced. This, too, would make matters more complicated. As mentioned before, IPs that change within sessions would make it very difficult, if not impossible, to conduct a research. All these problems could arise with other data sets. In the framework of this section, solutions for these problems have not been looked for. It may be an opportunity for future research.

Preprocessing

Before any actual data mining algorithms can be applied on the data, the data needs to be preprocessed so that it can serve as the input of several algorithms. As mentioned in the introduction, most tools that were designed for conventional Web Usage Mining on Web servers perform this preprocessing automatically. In this case, however, this part of the process will have to be executed manually. The advantage of this is that we have more control over the way it is done.

Data Cleaning

First of all, all files with an extension .jpg, .gif, .wav, etc. have been removed from the log file, an action that drastically reduced the size of the file. Secondly, all irrelevant fields were removed. Only three fields were retained: IP address, time stamp, and URL. Finally, only the DNS name in the URLs was kept, in the form of ww.websitename.com (other prefixes and suffixes are possible instead of www and com). This was done in order to facilitate the comparison of different lines in the database, and because the exact pages that were visited are irrelevant in this research. After this process of data cleaning, the original log file that comprised 600MB was reduced to 185MB. The file contains 1,714,813 hits.

User and session identification

The only data that is at disposal to identify users and sessions is the time stamp and the IP address. To start with, every IP address is seen as a different user. However, a different user can represent several sessions. Within the series of lines that are linked to a certain IP address, different sessions are identified by using a *time out* of thirty minutes. This means that if there is a time interval of more than thirty minutes between two consecutive clicks, two sessions will be identified. The problem that we face by doing this is that it is impossible to know whether it was the same person who simply did not touch his computer for half an hour, or effectively a different user that was assigned an IP address because the previous user turned off his computer. However, even if it was the same person, it is still useful to regard his click stream as two (or more) different sessions.

A problem here is that it is impossible to correctly identify the number of unique users, which is needed to calculate stickiness. However, the most important aspect of stickiness will be to compare the results of different sectors with each other. So, if the absolute values of stickiness are somewhat incorrect, that will be crucial. They should be seen relatively to one another. The file contained 7,287 different IP addresses and 10,548 sessions were identified.

22.6 Text Data Mining in Biomedical Literature by Combining with an Information Retrieval Approach – Case Study

This section attempts to focus on two issues. The first one is an information retrieval task, which retrieves useful MEDLINE documents for biological research with high precision, and the second is a text (data) mining task, which applies association rule mining and term extraction for mining MEDLINE. As a result, the obtained association rules may be used for consistency checking and error detection in annotation of MeSH terms in MEDLINE records. This case study is taken from T.N. Tran et al., Tokyo Institute of Technology, Japan.

A large part of the information required for biological research is currently recorded in free-text form such as MEDLINE abstracts. This information is important for many types of analysis, such as classification of proteins into functional groups, discovery of new functional relationship, extraction of protein interaction information, and so on. Recently there has been an increasing deal of research that attempts to extract the abundant information in biological literature databases. Since MEDLINE abstracts are recorded in free-text form, it is natural to apply natural language processing techniques into these kinds of task. To date, a great deal of research has attempted to use information extraction techniques to find gene–gene interactions, protein–protein interactions. In general, most of them are based on natural language processing techniques such as parsing, not from the viewpoint of mining the text sources to discover novel knowledge. The first goal of this research is to address this issue by applying association rules and term extraction for mining MEDLINE.

On the other hand, the task of "preprocessing," i.e., retrieving only MEDLINE documents, which contain relevant biological information prior to the information extraction or text mining from MEDLINE also, plays an important role in biological research. Current systems for accessing MEDLINE such as PubMed (http://www.ncbi.nlm.nih.gov/PubMed/) accept keyword-based queries to text sources and return documents that are hopefully relevant to the query. Since MEDLINE contains an enormous amount of papers and the current MEDLINE search engines is a keyword-based one, the number of returned documents is often large, and many of them in fact are nonrelevant. The approach to solve this is to make use of existing databases of organisms such as *S. cerevisiae* using supervised machine learning techniques.

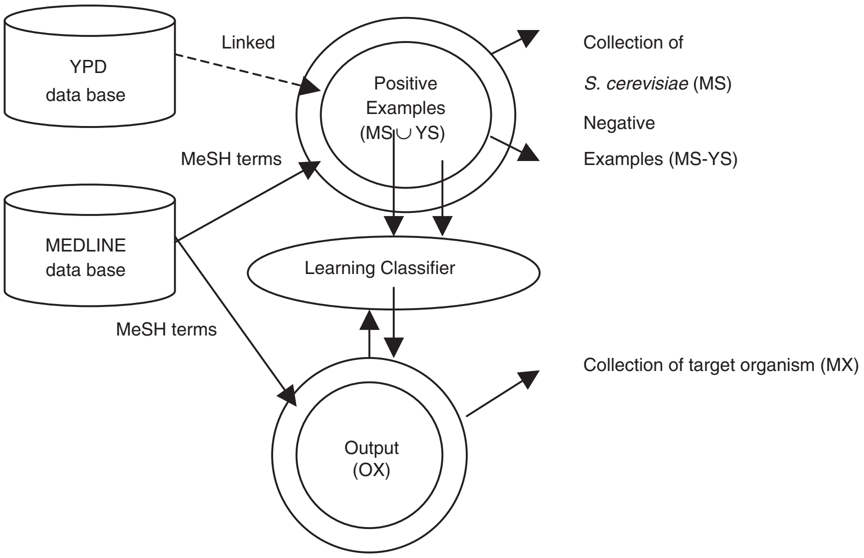

Fig. 22.2. Outline of the Information Retrieval task

22.6.1 Information Retrieval Task – Retrieve Relevant Documents by Making use of Existing Database

Figure 22.2 shows the illustration of the Information Retrieval task. In this figure, YPD database (standing for Yeast Protein Database – http://www.proteome.com/databases/index/html) is a biological database, which contains genetic functions and other characteristics of a kind of *Yeast* called *S. cerevisiae*. Given a certain organism X, the goal of this task is to retrieve its relevant documents, i.e., documents containing useful genetic information for biological research.

Let MX, MS be the sets of documents retrieved from MEDLINE by querying for the target organism X and *S. cerevisiae*, respectively (without any machine learning filtering) and YS be the set of documents found by querying for the YPD terms for *S. cerevisiae* (YS omitted in Figure 22.2 for the reason of simplification). The set of positive and negative examples then is collected as the intersection set and difference set of MS and YS, respectively. Given the training examples, OX is the output set of documents obtained by applying naïve Bayes classifier on MX.

22.6.2 Naïve Bayes Classifier

Naïve Bayes classifiers are among the most successful known algorithms for learning to classify text documents. A naïve Bayes classifier is constructed by using the training data to estimate the probability of each category given the document feature values of a new instance. The probability an instance d belongs to a class c_k is estimates by Bayes theorem as follows:

$$P(C = c_k|d) = \frac{P(d|C = c_k)P(C = c_k)}{P(d)}$$

Since $P(d/C = c_k)$ is often impractical to compute without simplifying assumptions, for the naïve Bayes classifier, it is assumed that the features $\mathrm{X}_1,\ \mathrm{X}_2, \ldots ..\mathrm{X}_\mathrm{n}$ are conditionally independent, given the category variable C. As a result:= $P\left(d|\,C = c_k\right) = \coprod\limits_{i} P\left(d_i\,|C = c_k\right)$

22.6.3 Experimental results of Information Retrieval task

Two kinds of yeast called *Pombe* and *Candida* have been used as the input organisms. To evaluate the validity of using the machine learning approach, the recall and precision before learning (corresponding to the collection of the input organism in Figure 22.2) and after learning (the output collection in Figure 22.2) have been calculated for both *Pombe* and *Candida.* A certain number of documents (50 in this experiment) in each of dataset are taken randomly, checked by hand whether they are relevant or not, then the number of relevant documents in each dataset is estimated. The recall and precision in each case can be calculated as follows:

$$Precision\ = \frac{\#documents\,match\,classifier\,and\,relevant}{\#documents\,match\,classifier}$$

$$recall\ = \frac{\#documents\,match\,classifier\,and\,relevant}{\#documents\,relevant}$$

22.6.4 Text Mining Task – Mining MEDLINE by Combining Term Extraction and Association Rule Mining

We have used the MEDLINE dataset collected from YPD database, which were previously described before as the input for the text-mining task. This dataset consists of 14572 abstracts in HTML form pertaining to *S. cerevisiae.* The text mining task from the collected dataset consists of two main modules: the **T**erm **E**xtraction module and the Association-Rule Generation module. The Term Extraction module itself includes the following stages:

- *XML translation* (the dataset was collected in HTML from): This stage translates the MEDLINE record from HTML form into an XML-like form, conducting some preprocessing dealing with punctuation.
- *Part-of-speech tagging*: Here, the rule-based Brill part-of-speech tagger was used for tagging the title and the abstract part.

- *Term Generation*: Sequences of tagged words are selected as potential term candidates on the basis of relevant morphosyntactic patterns (such as "Non Noun," "Noun Adjective Noun," "Adjective Noun," "Noun Preposition Noun," etc.). For example, "in vivo." "Saccharomyces cerevisiae" are terms extracted from this stage.
- *Stemming*: Stemming algorithm was used to find variations of the same word. Stemming transforms variations of the same word into a single one, reducing vocabulary size
- *Term Filtering*: In order to decrease the number of "bad terms," in the abstract part, only sentences containing verbs listed in the "verbs related to biological events."

After necessary terms have been generated from the Term Extraction module, the Association-Rule Generation module then applies the Apriori algorithm (Agrawal et al. 1994) using the set of generated terms to produce association rules (each line of the input file of Apriori-based program consists of every term extracted from a certain MEDLINE record in the dataset). The goal for text mining in this section is to find relations between the MeSH terms and the substances (location) in a MEDLINE record pertaining to *S. cerevisiae* and to discover the relations between other terms occurred in the title, abstract with the journal name, and so on.

22.6.5 Finding the Relations Between MeSH Terms and Substances

First, we just extracted terms in MeSH terms and Substances fields. We conducted this experiment in order to find the relations between MeSH terms and Substances. Figure 22.3 shows a list of ten rules among obtained rules. Here, those terms begin with "m_" stand for MeSH terms and those terms begin with "s_" stand for Substances. For example, the sixth rule in Figure 22.3 is translated as follows: The rule that in a MEDLINE record if Saccharomyces cerevisiae/genetics and Fungal Proteins/metabolism occur in the MeSH field then Fungal Proteins also occur in the Substances field of this MEDLINE record has a support of 13.5% (i.e., this rule is matched 1971 times in the total of 14572) and a confidence of 99.9%.

- *Freq Verb-Terms*: This variation extracts all terms and also the "frequently-occurring verbs" that occurred in the corresponding sentence (in the stemmed form).
- *None_Freq Verb-Terms*: This variation extracts all terms but does not extract the "frequently occurring verbs."

22.6.6 Finding the Relations Between Other Terms

Next, we attempt to find the relationship between other terms except those that occurred in MeSH terms and Substances fields. For example, association rules concerning the relationship between terms occurred in the title and

1: s_fungal_proteins ← m_fungal_proteins_genetics (26.3%/3829, 99.9%)
2: s_fungal_proteins ← proteins ← m_fungal-proteins_metabolism (21.1%/3070, 99.9%)
3: m_amino_acid_sequence ← m_sequence_homology_ _ amino_acid (13.2%/1922, 94.6%)
4: s_plasmids ← s_plasmids (10.5%/1689, 90.7%)
5: s_plasmids ← m_plasmids (10.6%/1540, 99.5%)
6: s_fungal_proteins ← m_saccharomyces_ cerevisiae_genetics
m_fungal_protein_metabolism (13.5%/1971, 99.9%)
7: m-saccharomyces_ceevisiae_genetics ←amino-acid-sequence
m_base_sequence (23.0%/3343, 80.6%)
8: m_amino_acid_proteins ← m_molecular_sequence_data
s_fungal_proteins (22.4%/3266, 75.8%)
9: s_fungal_proteins ← m_molecular_sequence_data
m_fungal_protein_genetics (16.6%/2417, 100.0%)
10: s_fungal_proteins ← m_molecular_sequence_data
m-fungal_proteins_metabolism (11.2%)/1632, 100.0%)

Fig. 22.3. Ten rules obtained by MeSHterms-Substances (*minimum support* = 0.1, *minimum confidence* = 0.75)

abstract part with the journal name may be interesting. We have used two variations for the reason of comparison as follows:

Figures 22.4 and 22.5 show the list of twenty rules among obtained rules by these two variations with the minimum confidence set to 0.75, respectively. For example, the 12th rule in Figure 22.5 implies that “the rule that in a MEDLINE record if iron sulfur protein occurs this MEDLINE document is published in J Biology chem. Journal has the support of 0.2% (i.e., this rule is matched 19 times in the total of 14572), and the confidence of 84.2%.” It can be seen that the relation between journal name and terms extracted from the title and the abstract has been discovered from this example.

It can be seen from Figures 22.4 and 22.5 that making use of terms can produce interesting rules that cannot be obtained using only single-words. The main difference between Freq Verb-Terms (Figure 22.4) and None_Freq Verb-Terms (Figure 22.5) is that the former is used for discovering the relations between “frequently-occurring verbs” and terms, while the latter is used for discovering relations among terms.

22.7 Related Work

There has been a great deal of research aimed at extraction information from biology texts. Despite the attractiveness of automatically extraction of useful information from biomedical text sources, previous works seem to be constrained from extracting factual assertions. Fukuda et al. attempt to identify protein names from biological papers. Andrade and Valencia also concentrate

```
1: have ← gever_protein (0.2%/25, 80.0%)
2: requir ← gwi4 (0.3%/36, 75.0%)
3: requir ← gdc42 (0.2%/30, 76.7%)
4: requir ← gre-rrna_process (0.3%/36, 77.8%)
5: requir ← groper (0.2%/31, 80.6%)
6: requir ← granslat_activat (0.2%/28, 75.0%)
7: requir ← guclear_fusion (0.2%/29, 75.9%)
8: requir ← gntegrity (0.3%/37, 75.7%)
9: suggest ← gapip (0.2%/27, 77.8%)
10: contain ← gondon (0.3%/32, 84.4%)
11: us ← grobe (0.6%/78, 87.2%)
12: us ← geast_two-hybrid_system (0.2%/27, 85.2%)
13: us ← polymeras_chain_reaction (0.2%/29, 89.7%)
14: yeast ← grf (0.9%/108, 76.9%)
15: yeast ← gpen_read_frame (0.4%/46, 91.3%)
16: identify ← gwo-hybrid_screen (0.3%/34, 94.1%)
17: indicat ← gdrl (0.2%/26, 76.9%)
18: interact ← gwo-hybrid_system (0.6%/76, 82.9%)
19: interact ← gwo-hybrid_screen (0.3%/34, 88.2%)
20: interact ← gwo-hybrid_assay (0.2%/27, 77.8%)
```

Fig. 22.4. First twenty rules obtained by Freq Verb-Terms (*minimum support* = 0.002, *minimum confidence* = 0.75)

on extraction of keywords, not mining factual assertions. There have been many approaches to the extraction of factual assertion using NLP techniques such as syntactic parsing.

Sekimizu et al. attempt to generate automatic database entries containing relations extracted from MEDLINE abstracts. Their approach is to parse, determine noun phrase, spot the frequently occurring verbs, and choose the most likely subject and object from the candidate NPs in the surrounding text. Rindflesch uses a stochastic part-of-speech tagger to generate an underspecified syntactic parse and then uses semantic and pragmatic information to construct its assertions. This system can only extract mentions of well-characterized genes, drugs cell types, and not the interactions among them. Thomas et al. use an existing Information Extraction system called SRI's Highlight for gathering data on protein interactions. Their work concentrates on finding realizations directly between proteins. Blaschke et al. attempt to generate functional relationship maps from abstracts; however, it requires a predefined list of all named entities and cannot handle syntactically complex sentences.

22.7.1 Future Work: For the Information Retrieval Task

Although using an existing database of *S. cerevisiae* is able to obtain a high precision for other yeasts and organisms, the recall value is still low, especially

1: protein ← oligomer (0.2%/19, 78.9%)
2: protein ← co-immunoprecipitated (0.1%/14, 78.6%)
3: gene ← differently _expression (0.1%/13, 76.9%
4: yeast ← orf (0.9%/108, 76.9%)
5: yeast ← open_read_frame (0.4%/46, 91.3%)
6: yeast ← kb_segment (0.2%/20, 100.0%)
7: yeast ← orf (0.1%/13, 84.6%)
8: yeast ← alyeast (0.1%/18, 94.4%)
9: yeast ← other_orf (0.1%/15, 100.0%)
10: sequ ← other_orf (0.1%/15, 86.7%)
11: essenty ← yeast_cell-viabl (0.1%13, 76.9%)
12: j_biol_chem. ← iron_sulfur_protein (0.2%19, 84.2%)
13: j_boil_chem. ← mitochondry_membran_system (0.1%/16, 75.0%)
14: open_read_frame ← codon (0.3%/32, 81.2%)
15. open_read-frame ← kb_segment (0.2%/20, 85.0%)
16: evidenc ← server_line (0.1%/13, 76.9%)
17: alpha ← a_specific_gene (0.2%/20, 75.0%)
18: nucleu ← nuclear_local_sequ (0.1%/13, 76.9%)
19: er ← secl2p (0.1%/14, 78.6%)
20: assembly ← mitochondry_membran_system (0/1%/16, 100.0%)

Fig. 22.5. First twenty rules obtained by None_Freq Verb_Terms *(minimum support* = 0.001, *minimum confidence* = 0.75)

for the yeasts that are remarkably from *S. cerevisiae.* Since yeasts such as *Candida* might have many unique attributes, we may improve the recall by feeding the documents checked by hand back to the classifier and conduct the learning process again. The negative training set has still contained many positive examples; so we need to reduce the noise by making use of the learning results.

22.7.2 For the Text Mining Task

By combining term extraction and association rule mining, we are able to obtain interesting rules such as the relations among journal names and terms, terms and terms. Particularly, the relations among MeSH terms and "Substances" may be useful for error detection in annotation of MeSH terms in MEDLINE records. However, the current algorithm treats extracted terms such as "cdc37_caryogamy_defect," "cdc37_in_mitosy," "cdc37_mutat" to be mutually independent. It may be necessary to construct semiautomatically term taxonomy, for instance users are able to choose only interesting rules or terms then feedback to the system.

22.7.3 Mutual Benefits between Two Tasks

Gaining mutual benefits between two tasks is also an important issue for future work. First, by applying text mining results, it should be noted that we can decrease the number of documents being “leaked” in the Information Retrieval task. As a result, it is possible to improve the recall. Conversely, since the current text mining algorithm creates many unnecessary rules (from the viewpoint of biological research), it is also possible to apply the information retrieval task first for filtering relevant documents, then apply to the text mining task to decrease the number of unnecessary rules obtained, and to improve the quality of the text mining task.

22.8 Summary

The World Wide Web is continuously growing and “collecting” all kinds of resources, text, multimedia applications, etc. Despite the anarchy in which it is growing, the Web is one of the biggest repositories ever built. Analyzing the Web access logs of different Web sites can help understand the user behavior and the Web structure, thereby improving the design of this colossal collection of resources. Therefore, it is important to build tools to analyze the Web access patterns.

Currently available Web log analysis tools report interesting statistics, but are limited by the huge sizes of the log files continuously expanding, the type of data collected in the Web logs, and the techniques used to analyze this data. Consequently, it is imperative to design a good Web log analysis tool that would overcome the current limitations of the Web log and recommend directives for new Web log standards that would help better analyze the Web access trends and discover useful knowledge from the access records.

We have outlined the design of the system WebLogMiner, which benefits from OLAP and data mining techniques, and multidimensional data cube, to interactively extract implicit knowledge from very large Web log files. Concrete examples using these techniques were given for time-series pattern analysis. The major strengths of this design are its scalability, interactively, and the variety and flexibility of the analyses possible to perform. Despite these strengths, the discovery potential of such a design is still limited due to the current impoverished Web log files.

This section has introduced two tasks concerning information extraction from biological literature databases such as MEDLINE. The first one is an information retrieval, which attempts to retrieve useful documents for biology research with high precision, and the second one is a text mining task which attempts to apply association rule mining and term extractors for mining MEDLINE. It can be seen that making use of the obtained results is useful for consistency checking and error detection in annotation of MeSH terms in MEDLINE records. In future work, combining these two tasks together may be essential to gain mutual benefits for both two tasks. This chapter has also revealed the concepts of text data mining and Web mining.

22.9 Review Questions

1. Give an overview and the taxonomy of Web mining.
2. What are the characteristics of Web mining systems?
3. Explain in detail Web mining environment.
4. Explain relational fuzzy C- Maximal density estimator.
5. Explain Web access pattern tree and Web log miner.
6. Briefly discuss the design methodology of a Web log miner.
7. How is database constructed from server log files?
8. Write a short note on multidimensional Web log data cube.
9. With a case study explain the Web usage mining on proxy servers.
10. What are the various data mining techniques used in Web data mining?
11. Explain in detail naïve Bayes classifier and the information retrieval task.
12. Explain text mining task – Mining MEDLINE.
13. What are the rules obtained by MeSH terms – substances?
14. State the rules obtained by None-Freq Verb-Terms.
15. What are the potential text mining applications and state some of the text mining tools?
16. Compare text mining and information retrieval.
17. Write a note on computational linguistics.