Queensland University of Technology
Brisbane Australia

This may be the author's version of a work that was submitted/accepted for publication in the following source:

Nayak, Richi
(2008)
Data mining in web services discovery and monitoring.
International Journal of Web Services Research, *5*(1), pp. 63-81.

This file was downloaded from: https://eprints.qut.edu.au/224830/

© Consult author(s) regarding copyright matters

This work is covered by copyright. Unless the document is being made available under a Creative Commons Licence, you must assume that re-use is limited to personal use and that permission from the copyright owner must be obtained for all other uses. If the document is available under a Creative Commons License (or other specified license) then refer to the Licence for details of permitted re-use. It is a condition of access that users recognise and abide by the legal requirements associated with these rights. If you believe that this work infringes copyright please provide details by email to qut.copyright@qut.edu.au

Notice: *Please note that this document may not be the Version of Record (i.e. published version) of the work. Author manuscript versions (as Submitted for peer review or as Accepted for publication after peer review) can be identified by an absence of publisher branding and/or typeset appearance. If there is any doubt, please refer to the published source.*

QUT Digital Repository:
http://eprints.qut.edu.au/

Nayak, Richi (2008) ***Data mining in web services discovery and monitoring***. International Journal of Web Services Research, 5(1). pp. 62-80.

© **Copyright 2008 Idea Group Publishing / Information Science Publishing**

International Journal of Web Services Research , Vol.X, No.X, 200X

Data Mining in Web Services Discovery and Monitoring

Richi Nayak
School of Information Systems
Queensland University of Technology
Brisbane, QLD 4001, Australia
r.nayak@qut.edu.au

ABSTRACT:

The business needs, the availability of huge volumes of data and the continuous evolution in Web services functions derive the need of application of data mining in the Web service domain. This paper recommends several data mining applications that can leverage problems concerned with the discovery and monitoring of Web services. This paper then presents a case study on applying the clustering data mining technique to the Web service usage data to improve the Web service discovery process. This paper also discusses the challenges that arise when applying data mining to Web services usage data and abstract information.

KEY WORDS:
Web Services, Data Mining, Knowledge Discovery, Service Discovery

INTRODUCTION

Simplicity and ubiquity are factors that brought upon the success of the Web. Once solely a repository that provides access to information for human interpretation and use, the Web has evolved to become a repository of software components (Alesco & Smith, 2004; Clark 2002; Newcomer, 2002). These service oriented components, or Web services, are emerging as the key enabling technology for today's e-commerce and business-to-business applications, and are transforming the Web into a distributed computation and application framework. The implication of this transformation is that the volume of data generated on the Web is increasing exponentially. Whilst human users have generated huge volumes of data from using the Web and warranted a myriad of research in the data mining community (Kosala & Blockeel, 2000), the volume of data generated by machines using and providing Web services would certainly make the former dwarf in size. Since Web servers store the access request information in Web server access logs, all interactions between Web services are recorded, similar to the human-server interactions that are prevalent on the Web now. Web server access logs recording site accesses have been a rich source for discovering the browsing behavior of site visitors in Web mining. In a similar manner, logs that record Web services accesses can be mined for Web services usage patterns by organizations. With the huge volume of accesses, the amounts of logs that can be collected make a feasible source for data mining. Data mining (DM) methods search for distinct patterns and trends that exist in data but are 'hidden' among the vast amount of data (Han & Kamber, 2001).

Since Web service is a relatively new technology, there are a lot of areas where improvements can be made to complement the current state of the art. Not only these improvements enhance the technical workings of Web services, they can also generate new business opportunities. As will be shown in this paper, some of these improvements can be made possible through data mining. For example, as the number of web services increases, it becomes increasingly important to provide a scalable infrastructure of registries that allows both developers and end-users to perform service discovery.

Given the competitiveness of the current business market place, the need to exploit sources of data for business intelligence is greater than ever before. The ability for a business to obtain previously unknown knowledge about itself and its customers is a valuable asset. Business needs, the availability of huge volumes of data, and in particular, the possible Web services improvement areas form the premises for the application of data mining using the Web services data. The goal of this paper is thus to present how Web services data can be utilized in data mining and the kinds of benefits that can be gained from such operations.

This paper discusses the web services in the context of data mining. It first discusses the basics of data mining, and then recommends a set of data mining applications that can leverage problems concerned with the discovery and monitoring of Web services. A case study is presented that demonstrates the improvement in the Web discovery with a proposed clustering data mining method on using the Web query logs. Some of the challenges that arise while mining Web services data are also presented. In the end, this paper looks at the current technologies in use for web service discovery and log mining.

DATA MINING AND ITS OPEARTIONS

Data mining is the search for distinct patterns and trends that exist in datasets but are 'hidden' among the vast amount of data. A data mining task includes problem identification, data pre-processing, data modeling and pattern evaluation. With the high costs associated with data mining operations, it is essential for businesses to know whether their investments are worthwhile. This requires the analyst to have an understanding of the application domain, the relevant prior knowledge, the data that is available for mining, and the goals of the end user. Once the objective of data mining task is identified, the next step is to prepare the data for mining. This pre-processing step involves basic operations such as the removal of noise or outliers, handling inconsistencies and missing values to ensure the quality of the data set, and transforming it into data structures specific to the data mining task. Traditionally used for extracting information from data within databases, recent researches in data mining are geared towards Web data, including Web server logs (Cooley, Mobasher, & Srivastava, 1997; Wang, Huang, Wu, & Zhang, 1999; Nayak, 2002; Nayak & Seow, 2004). The preprocessing step is particularly relevant to this study as its assists to determine how the Web services data can be mined. Data modeling is next used to infer rules from the pre-processed data or build model that fits best on this data set. The data modeling step refers to the application of a DM technique or a combination of techniques for identifying patterns from the derived data set. The kinds of patterns that can be mined depend on the data mining operation as listed in table 1.

Predictive modeling solves a problem by looking at the past experiences with known answers, and then projecting to new cases based on essential characteristics about the data. This process enables the prediction of the unknown value of a variable given known values of other variables. The classification predictive modeling is used to predict discrete nominal values, whereas value prediction, or regression, is used to predict continuous values. Clustering or segmentation solves a problem by identifying objects with similar characteristics in a data set, and thus forming the groups of similar objects. The link analysis operation establishes association among objects by finding items that imply the presence of other items in the given data set. A specialization of link analysis is similar time sequence discovery that identifies similar occurrences, or similar sequences of occurrences, in sets of time-series data. This analysis explores the temporal relationship between sets of time-series data by matching all data sequences that a similar to a given sequence, or by matching all sequences that are similar to one another. Deviation analysis is concerned with identifying anomalies, unusual activities or events within a data set, which are indicated by the presence of outliers.

Mining Operation	Goal	Approaches
Predictive Modelling (Classification, Value prediction)	To predict future needs based on previous data	Decision tree, Neural networks, Bayesian, linear and non-linear regression
Clustering or Segmentation	To partition data into segments	Demographic, Neural networks
Link Analysis (Association, Sequential, Similar time discovery)	To establish association among items	Counting occurrences of items such as Apriori Algorithms
Deviation Analysis	To detect any anomalies, unusual activities	Statistics and visualisation techniques

Table 1: Various DM Operations and Approaches

The final step involves the interpretation of derived models or rules. Operations such as filtering, restructuring and data visualization are utilized to represent the extracted knowledge in a meaningful and easily understandable manner to users.

APPLICATION OF DATA MINING IN WEB SERVICES

The benefits of the data mining applications can be seen as falling into two broad categories – those that deliver business value and those that have technical value (Nayak & Tong, 2004). Applications with business value are those that can be used by management to assist in making strategic decisions or by human resource in maximising staffing levels while minimising costs. Applications with technical value are those that can be used by technical staff in devising new services or in services that their organization can use.

Business Applications

The development in business analytics using data mining has accelerated tremendously over the past few years. With the availability of operational data stored in transactional systems, coupled with advances in efficient data mining algorithms and increased power in processors, data mining operations have become viable solutions for deriving business intelligence. Data mining can be used to provide insights on the planning of Web services deployment via “Web services cost and savings prediction”, and on how costs on staffing for the monitoring of services can be optimised via “performance monitoring”.

Web Services Cost and Savings Prediction

Chen (2003) performed an analysis on factors that are affecting the adoption of Web services. Financial considerations were identified as major cause. The enabling technologies of Web services are only concerned with the technicalities and do not have any provision for the planning of costs and savings. Unless businesses have an idea of how much they should spend on Web services, and on the foreseeable savings, they will be indecisive on the adoption of Web services. A possible solution is the cost and performance model for Web services investment (Larsen & Bloniarz, 2000). The model helps businesses to assess the costs in terms of the functionalities required and to increase in target performance. For each assessment area, three levels of service (modest, moderate, and elaborate) are defined in order to identify the most profitable option. The model, however, does not take into consideration that many businesses may not have any experience in Web services and thus find the estimation difficult in the first place. Furthermore, it does not provide measures on the effectiveness at which a business can deploy Web services. For example, if a business can find out that other similar projects cost less but yield the same returns, then it may consider outsourcing the project.

Businesses can learn from the experiences of similar organizations and get a good approximation of these values from the data collected by research firms such as Nemertes (Johnson, 2003). If *predictive mining* models representing Web services deployment costs can be built considering the previous data, then businesses intending to adopt Web services can make use of these models both for estimating the cost of their deployment, as well as deciding whether outsourcing them is more feasible than developing them in-house. *Value prediction* is suitable in this instance to model the investment versus return functions for the prediction of figures for costs and savings. *Regression techniques* (Han & Kamber, 2001) derive the predicted continuous values obtained from functions that best fit the case. For predicting the costs, the input data required consists of, for each deployment, the number of staff member involved, the time it took, and the complexity of the deployment. The complexity of the deployment can be quantified in terms of the lines of code used in the programs, and the annual revenue from the operations that the deployment oversees. The costs of the proposed deployment can be predicted based on these parameters. Once the costs are known, prospective savings can be predicted. Using inputs such as the cost of the deployment, and the original and new cost of the operation, savings can be determined. Having determined the costs and the savings that can be gained, businesses can identify the size of Web services deployment that is best suited for them and turn the discovered insights into action.

Performance Monitoring

Strategic placement of human resource plays a crucial role in the effective monitoring of performance and handling of events. In today's business environment where resources are stretched to minimise costs, adequate resources especially personnel with the required expertise may not be available to handle all reported incidents. This leads to the need to prioritise tasks. A service being used by many clients at the time when a problem occurs should have a higher priority than a service being used by few clients at the same time. By knowing the usage pattern of services, training programs on groups of services with *similar usage patterns* can be developed. This allows staff monitoring the services at certain times to have a more in depth knowledge of particular services. As an example, if a Web service uses an Oracle database to support its back-end operations, then presence of the Oracle database administrator at times when the service is at peak usage would be desirable. Such knowledge is especially valuable to companies that do not need permanent staff with some specialist knowledge, but whose service is required only under certain circumstances.

To identify Web services with similar usage patterns, similar time sequence analysis can be used (Peng et al., 2000). The input for such an operation is time-series data recording the number of clients using a particular service at any moment in time. Although such data is not normally collected explicitly, it is implicitly recorded in the web server access logs. The steps in generating this time-series data from web server logs are as follows:

1. Select from the web server log all entries related to the offered Web services by extracting all entires containing the web service's URL in the URL field.

2. Group the entries by Web services and client IP addresses, and then order the entries by time. This gives a set of a client's interaction with a web service.
3. Calculate the time between each interaction to determine separate client sessions with the web service. A client session is one 'use' of the web service. The duration of a client session for different services varies depending on the nature of the service. Setting the threshold of session boundaries thus requires the knowledge about the individual services.
4. For each service, count the number of clients using it at specified time intervals. This can then be used to construct the time- series graph for each service.

Algorithms for approximate subsequence matching in time-series now can be applied to find Web services that have similar usage patterns. These patterns can then be used to help in the design of roster schedules that optimise staffing levels and skill requirements while minimising the number of employees that need to be present.

Technical Applications

This section discusses applications which are more technically oriented. Although the applications will ultimately produce business benefits in some way, the immediate analysis of the results are targeted towards technical staff that implement Web services.

Service Innovation

The Web services market has gathered much momentum, however, the number and choice of services that are on offer is limited - a fact which spawned the creation of the web site WebServicesIdea.com, which is tied to the renowned Web services broker Salcentral.com. For businesses, this lack of innovation in Web services represents business opportunities. For technical staff, this limits their ability to take advantage of the reusability concept that is central to the Web services architecture. It is important for service providers to establish themselves in the market by offering a range of quality services. The set of queries used by potential clients to find suitable Web services is a rich source for finding clues about what the clients want. If an unusual search term is used with other common search terms in the queries, and that the search terms are all related, then it is a good indication that there is a demand for a new service. The unusual search term may represent a new concept, or a specialisation of a general service currently being offered. For example, SMS (Short Message Service) sends text messages to mobile phones while a more recent technology MMS (Multimedia Message Service) sends multimedia messages. SMS is a frequently used search term but MMS is not. As the technology becomes more prevalent, demand for MMS Web services will emerge and the appearance of MMS in query data will be evidence of this.

The simplest approach in discovering uncommon search terms is by *deviation analysis* (Devore, 1995). Having counted the frequencies of the search terms appearing, simple measures such as median, quartiles and inter-quartile range (IQR) can be calculated. Then using the common heuristic that outliers fall at least 1.5 * IQR above the third quartile or below the first quartile, the unusual search terms can be identified. An alternative measure is the use of support to count the number of times the term appeared in total terms. If a search term has very low support, then it can be classified as an outlier. Given that the demand for different services varies, applying these measures to the raw frequency count will produce biased results towards less popular services, producing many false positives. This is best illustrated using the following example. A popular service is searched 1000 times using a common search term Q_1 and 10 times using an uncommon search term Q_2. A very specific service aimed at a niche market is searched 7 times using Q_3 and 3 times using Q_4, both of which are common for the service. When search terms for all the services are taken into account, and statistics is applied to the data, Q_2, Q_3 and Q_4 will be identified as uncommon search terms. However, Q_3 and Q_4 are false positives because they represent 70% and 30% of searches for the service. On the other hand, although Q_2 has 10 occurrences, it is only 1% of all searches for the popular service. Clearly, Q_2 is the outlier that should be identified in this case.

The solution to this is to group searches into search areas and then find the outliers for each search area. This can be done by:

1. Grouping the queries into search sessions
2. Joining all search sessions that are similar to form search areas
3. Form search term pools for each search area.
4. Within each search term pool, apply statistics to find the uncommon search terms that suggest demands for a new web service.

Service Recommendation

Recommender systems have been studied extensively in data mining (Mobasher, 2005). By analysing the behaviour and preferences of past and current customers, businesses are able to predict the characteristics and needs of new customers. In e-business cases, the goods or services recommended are typically consumable and the cost of a replacement in case of a wrong choice by the customer is small. The same principle can be applied in Web services for recommending services that may be of interest to Web service clients. With the current Web service framework, providers and requesters must choose the names and description of services very precisely when using the UDDI. For example, a service named “SMS” may not be returned from the query “Mobile Messaging Service” submitted by a user. The user while performing the search may not be aware of this other search term, and thus will fail to retrieve the service.

With the expected increase of Web Service use, it is beneficial for the UDDI or Web services framework to be able to provide recommendations of appropriate Web Services of interest. However, the situation is quite different in recommending Web services. Firstly, the selection of a Web service that a business will use as part of its business process cannot just be based on preference. The suitability of a service also depends on various resources required by the service such as the interfaces, functionality and security offered by the service, as well as the cost. A service fitting the aims is a matter of whether it satisfies the list of strict criteria. Secondly, the role of the chosen service in the business can also be important. If not chosen carefully, the cost of replacing the service is not limited only to searching for a new service, but also the cost of reconfiguring the systems and the loss in staff productivity due to the system downtime that incur. Web services providers can recommend services to clients based on the services that other similar clients have used in the past. This is because similar clients are likely to have similar service needs. A systematic use of a single data mining operation or in combinations can achieve this. The following case study details a proposed method of improved Web service discovery with the use of clustering data mining approach.

A CASE STUDY – Improved Web Service Discovery

Searching for Web services using Web services search engines such as Salcentral and the UDDI is limited to keyword matching on names, locations, business, buildings and tModels (unique identifiers for reusable concepts). A user wishing to locate a particular service can do so by specifying a partial service name. With the UDDI, the user can also search by service provider name or browse the registry by the various taxonomic schemes it supports, such as NAICS and

UNSPSC (UDDI.org, 2005). However, since users are usually interested in the functionalities of services, they would typically begin searching by service name. The other attributes, such as the provider, will only be secondary for discriminating services.

The matching of a query to a set of services is based on the names of the services. If the query does not contain at least one exact word as the service name, the service is not returned. Since Web services names are generally very short consisting of no more than five words, it is essential that search terms are precise and contain the words in the desired service's name. The implication of matching services at the string level only is that synonymous queries do not return the same set of services, even though the services have similar functionality. Queries that are different at the string level, but are similar semantically, retrieve result sets that are disjoints. This is a major drawback when searching for Web services because the users are ultimately concerned with the functionality of the services. The keyword-based method of service discovery suffers from low recall, where results containing synonyms or concepts at a higher or lower level of abstraction to describe the same service are not returned. For example, a service named "car" may not be returned from the query "automobile" submitted by a user, even they are obviously the same at the conceptual level.

This problem can be approached in two ways, either by returning an expanded set of results to the user, or by suggesting the user with other relevant search terms. The Web search engines make use of ontology to return an expanded set including subclasses, superclasses and "sibling" classes of the concept entered by the user (Lim et al., 2004). Although this approach improves recall by returning a very large result set that are related to the search term, it introduces the poor precision problem where many entries are of no interest to the user. Many researchers in the information retrieval field have also proposed different approaches of enhancing semantic similarly for expanding query terms with synonyms, hyponyms, hypernyms, latent semantic space (Varelas et al., 2005; Voorhees, 1995). They showed only some minor improvements for short queries and no improvement (even the degradation) for long queries. The lack of semantics has led to the development of OWL-S and DAML-based languages for service description in WSDL, where service capability matching is based on the inputs, outputs, preconditions and effects, and ontology are used to encode relationship between concepts (McIlraith, Son, & Zeng, 2001; Paolucci et al., 2002). However, with the current state of Web services, we are still a long way from automatic service matching. For now, the manual discovery of services will have to suffice and effort is needed to improve its efficiency.

This proposed method of improved Web service discovery (summarized as in Table 2) employs the second approach that is to suggest the user with other related search terms based on what other users had used in similar queries by using the clustering technique. Whilst previous Web search engine approaches capture the intra-query relationships by clustering queries on a per query basis (Beeferman et al., 2000; Wen et al., 2002), they omit the inter-query relationships that exist between queries submitted by a user in one search session. When a user cannot find a Web service with the required functionality, the user will submit other search terms that are synonymous to the initial query. This is based on the assumption that a user's need of a service does not change even when it cannot be satisfied by any service from the set of results. Therefore, the set of search terms used consecutively in a search session form a query trail and represent a user's service need. *Clustering* based on search sessions instead of individual queries can leverage the problem by taking advantage of user judgment implied in the query and Web server logs to provide the semantic links between keywords. If queries are clustered independent of other queries used in the same search session, the resulting clusters will have search terms that are highly similar at the string level and so will not be very useful. On the other hand, if grouping is performed based on search sessions, then query terms such as “SMS” and “Mobile Messaging Service” or “cars” and “automobile” would likely to be clustered together.

Step 1: Perform data Pre-processing
- **1.1** Extract the related entries from the user query log and the web server log.
- **1.2** Generate the query-service descriptions matching by consolidating entries in both logs.
- **1.3** Form the search session by grouping the entries by clients IP addresses and Web services.

Step 2: Perform search session similarity
- **2.1** Calculate the Jaccard similarity coefficient independently for search terms and service descriptions between each pair of Web services
- **2.2** Form a similarity matrix by aggregating the similarity coefficients for both components.

Step 3: Perform clustering
- **3.1** Apply hierarchal clustering to form the groups from this similarity matrix.
- **3.2** Store these clusters in the UDDI register to allow the improved search.

Table 2: The proposed method of Improved Web Service Discovery

Search session	IP address	Query	Service Descriptions
1	123.456.789 123.456.789	q1 q2	ID001, ID003 ID007, ID012, ID005
2	234.567.890 234.567.890 234.567.890	q1 q3 q4	ID002, ID003 ID007 ID008, ID012
3	345.678.901 345.678.901 345.678.901	q4 q5 q6	ID008 ID020, ID025 ID028, ID030

Table 3: Queries grouped into search sessions

Step 1: Data pre-processing

The data required for clustering the session queries come from both the user query log and the web server log. The query log contains the search terms together with some identification of the queries submitted, such as the IP address of the client and the timestamp. The entries required from the web server log are those that correspond to the entries in the query log, as well as the service description pages that follow each query. The data from these two sources require to be consolidated for *query – service descriptions matching*. Each entry in the query log forms a one-to-many relationship with entries in the Web server log. These relationships can be captured by matching the query recorded in the query log with the subsequent service descriptions viewed by the user recorded in the web server log. The resulting structure is: query {service descriptions} where query is the search term used, and the set of service descriptions correspond to those the user viewed and is a subset of the results returned from the query.

The next step is to group the above structure into *search sessions* (an example is shown in Table 3). A search session is defined as a set of queries submitted in sequence by a user to locate a particular service. It is similar to a transaction or server session in Web mining (Cooley et al., 1997; Srivastava et al., 2000), and thus it’s identification is done using the same approach. A set of queries are identified by using the IP address recorded in the Web server log together with the timestamps. The IP address identifies the user, while the timestamp differentiates multiple search sessions by the same user. Entries with the same IP address are grouped together and then sorted by time to obtain all searches performed by the same user. A threshold on the amount of time between query submissions is then be set to distinguish the different sessions.

Step 2: Search session similarity

Similarity between a pair of search sessions is an indication of their mutual relevance. For example, search sessions containing many of the same keywords and same service descriptions would presumably be relevant to each other. Search session similarity can be calculated based on the similarity of the set of search terms used and the set of service descriptions viewed between two search sessions. The similarity between two different search sessions X and Y is defined as:

$$\textbf{similarity (X,Y)} = \textbf{similarity}_{\textbf{search term}}\textbf{(X, Y)} + \textbf{similarity}_{\textbf{service descriptions}}\textbf{(X, Y)}$$

The Jaccard coefficient (Han & Kamber, 2001) is used to calculate the similarity using the search terms and service descriptions sets between X and Y. This measure calculates similarity based on the attributes that are present in both data instances being compared. Attributes that do not describe either of the data instances are regarded as unimportant. The Jaccard coefficient suits well in this situation as it is only practical to compare search sessions based on their contents and not all the possible keywords and service descriptions that exist in the entire search engine. Keywords and service descriptions that are absent in search sessions being compared are therefore insignificant. Each search session is first transformed into two binary vectors, with one representing the queries and the other the service descriptions viewed. The value 1 shows the presence of a keyword or service description. The length of these vectors corresponds to the total number of distinct queries and the total number of distinct descriptions viewed in the search session respectively. The search term similarity and service description viewed similarity are therefore given as:

$$\textbf{similarity}_{\textbf{search term}}\textbf{(X, Y)} = (\mathbf{T}_{XY}) / (\mathbf{T}_{X} + \mathbf{T}_{Y} + \mathbf{T}_{XY})$$
$$\textbf{similarity}_{\textbf{service descriptions}}\textbf{(X, Y)} = (\mathbf{D}_{XY}) / (\mathbf{D}_{X} + \mathbf{D}_{Y} + \mathbf{D}_{XY})$$

T_{XY} is the number of common search terms used in X and Y, T_X is the number of search terms used in X only, and T_Y is the number of search terms used in Y only. $D_{XY,}$ D_X and D_{XY} have the same meaning for the second equation but in terms of the service descriptions viewed. As an example, consider a comparison of the first two search sessions in table 3, with X and Y representing the IP addresses 123.456.789 and 234.567.890 respectively.

Query vector		**q1**	**q2**	**q3**	**q4**
	X =	**1**	**1**	**0**	**0**
	Y =	**1**	**0**	**1**	**1**

Service description vector		**ID001**	**ID002**	**ID003**	**ID005**	**ID007**	**ID008**	**ID012**
	X =	**1**	**0**	**1**	**1**	**1**	**0**	**1**
	Y =	**0**	**1**	**1**	**0**	**1**	**1**	**1**

$$\text{similarity}_{\text{search term}}\text{(X, Y)} = 0.25$$
$$\text{similarity}_{\text{service descriptions}}\text{(X, Y)} = 0.429$$
$$\text{similarity (123.456.789, 234.567.890)} = 0.679$$

Having obtained the similarity measure between the search sessions, a *similarity* matrix is constructed for use as input for clustering. A clustering algorithm usually requires a similarity matrix to determine the groups within the search sessions. A similarity matrix is an *m* by *m* matrix containing all the pair wise similarities between the search sessions being considered. If x_i and x_j are the i_{th} and j_{th} search sessions, respectively, then the entry at the i_{th} row and j_{th} column of the similarity matrix is the similarity value, $similarity_{ij}$, between x_i and x_j (Jain et al., 1999). Table 4 shows a similarity matrix for the search sessions listed in table 3, where X, Y and Z are distinct search sessions.

	X	Y	Z
X	**1.000**	**0.679**	**0.000**
Y	**0.679**	**1.000**	**0.311**
Z	**0.000**	**0.311**	**1.000**

Table 4: A similarity matrix for search sessions in Table 3.

Step 3: Clustering of search sessions

The hierarchical agglomerative clustering method, which is often used in information retrieval for grouping similar documents (Kosala & Blockeel, 2000) is used. This method uses a bottom-up strategy that starts by placing each data instance in its own cluster, and then successively merge clusters together until a stopping criterion is satisfied (Karypis et al., 1999). The reasons to use this method are manyfold. Firstly, similarity of search sessions is based on the number of common queries and service descriptions they share. This means that search sessions are assigned to the same cluster if they have many queries and service descriptions common. The type of clusters desired is therefore globular in nature. This algorithm has been proven powerful at discovering arbitrarily shaped clusters. Secondly, the algorithm must be resistant to noise and outliers. Since users can submit any arbitrary query in a search session that may not be related to other queries in the same session, outliers may be present. The algorithm uses a k-nearest neighbour graph in the partitioning phase that ensures to reduce the effects of noise and outliers. Thirdly, because the volume of query data can be very large, the algorithm should be scalable.

After the clusters are formed, the support for each of the search terms in each cluster is counted and then assigned weights. The weights are used to predict a user's service need by suggesting search terms from the cluster with the largest weight for the user's search term. Depending on the size and number of search terms that make up the clusters, the suggested terms can either be all search terms within the cluster, or be limited to those from the most similar search sessions.

Experimental evaluation

This experiment demonstrates the effectiveness of the search session clustering based on cross-referencing between the queries submitted and the service descriptions viewed. In order to facilitate the evaluation of the similarity measure, the synthetic data represent disjoint clusters among the queries and among the descriptions. The data is generated based on three clusters of queries and three clusters of service descriptions. This was done to mimic the real data where similar queries lead to similar service descriptions viewed. Table 5 shows a sample (sub-set) data set containing the processed search sessions. The keyword and service description vectors were clustered separately, and then the combined similarity matrix was clustered. The three clustering solutions are then compared. The results (Table 6) imply that the search sessions clustered using the combined measure are more compact than those of the single similarity measures. Higher internal similarity of a cluster shows that the search terms within the cluster share a closer affinity. Lower external similarity of a cluster shows that the search terms with other clusters share a lesser affinity. This is essential in the suggestion of search terms as users are interested in suggestions that are highly similar to those submitted.

IP address	Query	Service Description Viewed			
123.456.789	AB	53	54		
123.456.789	AC	52	55	56	
138.256.980	AI	53	55	56	
138.256.980	BB	67	68	69	74
140.214.100	BC	68	70	76	
146.210.012	CD	81	82		

Table 5. Sample of the input dataset

Cluster		Keyword only	Service description only	Combining Keyword and service descriptions
0	**Avg. Internal Similarity**	0.211	0.281	0.675
	Avg. External Similarity	0.111	0.050	0.013
1	**Avg. Internal Similarity**	0.637	0.772	0.838
	Avg. External Similarity	0.127	0.156	0.100
2	**Avg. Internal Similarity**	0.182	0.256	0.535
	Avg. External Similarity	0.088	0.126	0.064

Table 6. Similarity based on keywords, service descriptions viewed and combining both

Upon the generation of clusters, each term in the cluster is weighted according to their support in the cluster and ranked. Now when a user inserts a query term, the terms in the cluster that are found in higher ranking are returned as the expended set of queries in order to improve the service discovery. A test was conducted for evaluating the effect of recommending the similar query search terms according to the session similarity in locating a particular service. Table 7 presents a subset of results with and without the use of expanded query terms. Initially the web services were located using the original query term (s). The query term set was then expanded if matched with a cluster. In this experiment, we used the clusters generated by combining both keywords and service descriptions similarity to match a term(s) in the query. Results show that the proposed Web service discovery method was successful in locating more relevant Web services by expending the search terms with similar terms according to the session similarity. The last row of the Table 7 shows that the 'q14' query term does not exist in any cluster. Based on the analysis of several (25) queries we conclude that the expanded query terms set returned the results with higher recall than with the original query terms. The recall measures the degree of how well a returned Web service satisfy the need requested by a query. It is the percentage of relevant Web services returned with respect to the total number of relevant Web services to a query. For example, the search program created for testing purposes returns 1 match for the term “cars”. However if the search term is changed to “automobile” or “vehicle”, no match is found without including the recommended terms with clustering results, it is a general knowledge that “cars” is also known as “automobile” or “vehicle”. This problem is solved by using the terms appearing in the same clusters. Recall of the search results is increased by 47% but the precision is reduced by 10%. This is due to the fact that results included irrelevant results due to the wide coverage with similar terms in clusters. However this overall precision figure (reduction by 10%) is better than the precision achieved when the simple ontology methods (such as synonyms etc) (reduction by 23%) are used for improving the search results. Overall, the expansion of query terms with the terms as found in the best matched clusters improves the total number of relevant Web services returned by making the search wider.

User	Query Term	Web Services Returned	Matching Cluster	Expanded Query Terms	Web services Returned (with expanded set)
A	q1	1, 100	Cluster 1	q1, q5, q7, q9, q23	1, 3, 5, 100
B	q2, q4	2, 56, 67	Cluster 2	q2, q4, q8, q11, q12, q19	2, 15, 25, 56, 67
C	q3, q6	4, 9, 13	Cluster 3	q3, q6, q10, q15, q13, q18	4, 9, 13, 69, 84, 97
D	q14	89	N/A	N/A	N/A

Table 7. Web services Discovery Improvement by using the similar query terms

The above approach utilized clustering to improve Web service discovery. However, other data mining methods or a combinations of methods could also be used in improved Web service discovery. The predictive mining techniques can be used to build rules on service subscriptions based on the usage of services by the clients. Service providers have information such as the line of business, size of business and what services their clients use, they can use these as inputs for predictive modeling. Inputs such as the interfaces, functionality and security offered by the service, as well as the cost, and other resources required by the service can also be considered to improve the result. Classification techniques such as *decision trees* (Quinlan, 1999) can be used to build rules on service subscriptions. Since the only information service providers have about clients are those for billing purposes, the number of attributes available is small. Consequently, the structure of the resulting decision tree will be relatively simple and easily comprehensible to a human analyst. To further enhance the success rate of recommendations, service providers can find dissociations among the services they offer. Dissociations (Teng, 2002) capture negative relationships between services with rules such as the use of service X and Y implies that it is unlikely service Z will also be used, even though service X and Z are often used. That is, $X \Rightarrow Z$; $X \wedge Y \Rightarrow \neg Z$. By incorporating these dissociations in the recommendation process, specific recommendations can be made.

Association mining can also be used to analyze the Web server access logs that record all interactions between web services and users. With the huge volume of web services accesses, these logs can be used for identifying Web services with similar usage patterns. The tasks would be: the selection of all entries related to the offered Web services from the web server log; extracting a set of a client’s interaction with a web service; calculating client sessions with the Web service, application of the *link analysis algorithm* (Agrawal & Srikant; 1994) to find Web services with similar usage patterns, associate the new client with a usage pattern, and finally, recommend a set of similar services after being identified with a usage patterns.

CHALLENGES IN PERFORMING DATA MINING to WEB SERVICES DATA

Data fusion and data collection

The extraction-transform-load (ETL) process that precedes data mining operations is typically complex and costly (Han & Kamber, 2001). With businesses spanning their operations across the entire globe, as well as having multiple servers that provide mirrored services, the Web server access logs from Web servers located at different sites must be consolidated to facilitate data mining. Therefore there is a need for controlled and reliable data collection. Because of the huge volume of data that have to be consolidated, a scheme must be in place to co-ordinate the transfer and storage of the data, while keeping the associated costs down. The scheme should minimise network traffic so that it does not interfere with the normal operations of the service provider. By compressing the data before transport, and scheduling transfers during off-peak hours, the impact of data transfer and the volume that need to be transported can be greatly reduced.

Additionally, with the growing popularity of Web services, services are not just being used in performing simple functions such as accepting registration etc. Web services are increasingly being used as the automated solution generation enabling integration of various tasks. To accomplish a task, a Web service needs to consolidate many operations and manage interactions between systems. Consequently, the abstract information of Web services in communicating various operations such as service interfaces, operations, messages, service/bindings elements and process instance execution logs containing event traces about transaction services during their execution are required to be considered along with the Web service usage logs and Web query logs at the user request level. For example, to find some useful information about a Web service on tour operation, many other services (or system to system interactions) such as money payment, location finder etc are required to be coupled with the process of mining. The Web service usage logs should solemnly be used as sole input data to improve the process of Web service discovery or other suggested applications in the paper. The challenge lies in the integration of data in various forms but representing the information leading to achieve the same goal.

Moreover, research has shown that the use of non functional attributes, such as Quality of Service (QoS) and Cost of Service (CoS) descriptors improve the web service discovery, ranking and matching (Hung & Lee, 2003). As the QoS characteristics of a web service may comprise of service availability, efficiency, accessibility, reliability, performance, scalability and security. Due to the dynamic nature of such QoS and CoS descriptors, however, the current technologies for publishing and finding web services (WSDL and UDDI) lack support for these. New frameworks, such as OWL-S and WSMO should be used to collect information for these non functional attributes.

Computing resources

A data mining task is usually resources-consuming. At the same time, web services are often large-scale distributed applications. Asynchronous operations of data mining and Web clients request should be considered, otherwise, a client application may be undesirably blocked until the service results are returned. With the data mining functionality, the Web services should be flexible enough to run on any server reliably and accurately appropriate to client volumes.

Analysis results interpretation

Results from DM operations often require specialist domain knowledge to obtain the full benefits of the discovered knowledge. To maximise the end user's understanding, the outputs of data mining must be translated to a language or visualisation appropriate for the user's need. For example, in the search term suggestion application, the end users are service requesters trying to locate a particular service. Instead of graphical clusters showing which queries are similar, the terms present in the clusters should be shown to them. On the other hand, in the performance monitoring application, the users would find it useful to see the similar sub-sequences in the time-series represented graphically when determining services with similar usage patterns. Furthermore, the results should be easy to comprehend and display only the details that are relevant. One approach is to show only the important attributes so that users are not distracted by the less important details (Kohavi et al., 2005). By removing the complexity, users can focus their efforts on gaining insight from the result.

Data reliability

Mining for Web services usage from Web server logs may not produce accurate results reflecting the real usage. This is due to wrong reflection of service usage in the collected data. For example, implementations of firewalls often involve the masking of internal IP addresses by substituting this with the IP address of the server that connects the network to the Internet. When a Web service client accesses a Web service, the Web server of the Web service provider logs the communication from the service client. If a firewall is used at the client side, then multiple accesses to the same service from different clients within the network may be recorded as one client – that of the server. This in effect masks the service usage in terms of number of clients using the service. Additionally, Web services execution is a stateless operation. This makes the recording of the event in the Web service usage log a difficult task. Appropriate solutions should be used to deal with this issue such as simulating a stateful session by including a session ID in the web services strings.

Proprietary nature of data

Data is a valuable asset for businesses and so is not normally released into the public domain. Many data mining tasks require data to be collected from multiple sources such as query logs collected by Web services search engines and the cost of Web services deployments. Unless there are mutual agreements between the parties, obtaining the data, and in sufficient volume, may be a problem. The quality of the input data is a key factor in determining the quality of the final model. Reputable research companies such as IDC may provide an answer to this. With the resources to arrange the necessary legal agreements, these research firms can perform data mining on data collected from multiple businesses to discover knowledge that cannot be gained from other ways. This will be beneficial to all parties while preserving the privacy of the businesses involved, as it is the nature of data mining to produce results that are generalisations of the input data.

Privacy and security

Although as mentioned above that data mining produces generalisations of data, it does have the problem that some sensitive information can be inferred. This is called the inference problem and arises when users submit queries and deduces sensitive information from the legitimate response they receive (Thuraisingham, 2005). This can be a hindrance when collecting data from many businesses, as well as a problem when mining Web server access logs that record the services used by certain clients. Data integrity and confidentiality could be compromised. Therefore, even if data can be collected, measures must be in place to ensure that businesses contributing to data mining such as the service recommendation application are not disadvantaged.

EXISTING WORK

We do not know of any work that has investigated the detailed use of data mining in Web services as us. There are some researchers who have applied the concepts of data mining into service discovery and Web service log mining. Some of them are discussed here.

Web service discovery

Researchers have worked in the direction of addressing the shortcoming of UDDI by finding relationships between search terms and service descriptions, and of WSDL to represent semantics while describing service. Sajjanhar et al (2004) apply the regression function called singular value decomposition to discover semantic relationships on services for matching best services. Their preliminary results show a significant increase in correct matching between service descriptions and the search terms after application of their algorithm with IBM UDDI. The matched results are not merely based on the number of matched keywords within the service descriptions. The algorithm evaluates the keyword global weights within the SVD procedure and aggregates services containing the highest global weight words to find semantic matched services.

Wang and Stroulia (2003) developed a method for assigning a value of similarity to WSDL documents. They use vector-space and WordNet to analyse the semantic of the identifiers of the WSDL documents in order to compare the structures of their operations, messages and types, and determine the similarity among two WSDL documents. This helps to support an automatic process to localize Web services by distinguishing among the services that can potentially be used and that are irrelevant to a given situation. Dong et al (2004) build a Web-service search engine (Woogle) to support the similarity search for Web services along with key-word searching with utilizing clustering and association mining. Starting with a keyword search, a user can drill down to a particular Web service operation. However, when unsatisfied, instead of modifying the keywords, user can query for Web service operations according to the most similar and semantically associated keywords suggested by the engine using the data mining techniques.

Some preliminary works have also been conducted to employ semantic languages and techniques in the description of Web services resources. Web Ontology Language for Services (OWL-S) and DARPA Agent Semantic Markup Language for Web Services (DAML-S) are high-level ontology at the application level meant to answer the ‘what’ and ‘why’ questions about a Web Service (Alesso & Smith 2004). Ontology-based semantic Web describes its properties and capabilities so that: (1) software can automatically determine its purpose thus automating service discovery; and (2) software can verify and monitor service properties thus automating service monitoring. As the future of Web Services greatly depends on their ability to automatically identify the Web resources and execute them for achieving the intended goals of user, OWL-S and DAML-S can achieve what UDDI cannot.

Gruninger et al (2005) propose the process models to be described as first order ontology, and then automate the searching and composition of Web services. Mandell and MacIlraith (2003) present a technology for the customized and dynamic localization of Web services using the Business Process Language for Web Service (BPWS4J) with the semantic discovery service. It provides semantic translation of services to match the user requirements. Soyaden and Singh (2004) develop a repository of Web services that extends the UDDI current search model. The repository in the form of ontology of attributes (based on DAML) provides a wide variety of operations such as the publication of services, costs of services and services selection based on their functionality. Li et al (2005) propose an approach of DAML based ontology use in e-commerce service search. The ontology positioning above WSDL relates service description of a WSDL document to descriptions of other WSDL documents. Benatallah et al (2005) propose a matching algorithm that takes as input the requirements to be met by the Web services and an ontology of services based on logic descriptions, and recommends the services that best comply with the given requirements.

Web service log mining

Gombots et al (2006) apply “WSIM - Web Services Interaction Mining” to analyse the log data (interactions between Web service consumers and providers). They identify three levels of abstraction with respect to WSIM: the operation level, the interaction level and the workflow level. On the Web service operation level, only one single Web service and its internal behaviour is examined by analysing a given log output of the Web service. On the Web services interaction level, one Web service and its “direct neighbours” Web services (that the examined service interacts with) are examined. This analysis reveals interesting facts about a Web service’s interaction partners, such as critical dependencies. On the Web service workflow level, the large-scale interactions and collaborations of Web services which together form an entire workflow is examined. This details the execution of the entire process, e.g., what is the general sequence of execution of various operations.

Malek et al (2004) show the impact of data mining in detecting security attacks which could cripple Web services or compromise confidential information. They determine the relevance of different log records and define the attack signature with the use of sequential pattern mining. Then they discover the highly compact decision rules from the intrusion patterns for pattern searching that helps to describe some safeguard against the attacks.

CONCLUSION

Recent advances in storage technology make data collection a very cheap exercise. Often data is collected for recording indicating that a transaction has taken place. Beyond that there is little use for the data. This is especially the case in Web services, where the generation and collection of transaction data is an automated process. However, hidden in these data are patterns and trends that are unknown to its keeper and may provide insights about the current situation. For businesses, these can represent new opportunities to provide better services to customers and to improve the operations of the business. For other fields, they may represent unknown facts whose discovery prompts for investigations for the reasons behind them, and as a result build on the body of knowledge in the field. Data mining has the capability to take advantage of the collected data and extract previously unknown knowledge from them.

Driven by factors such as the availability of huge volume of data, and the increasing need for businesses to obtain intelligence on their customers and internal operations, data mining technologies have found successful applications in many areas. Since Web services data are generated and collected automatically as well as being in digital form, the quality of the data is good compared to those collected manually. The amount of data generated by Web activity and collected automatically by servers enable businesses to extract strategic insights about the activity on their sites, as well as about the characteristics of their customers. With the emergence of Web services, the next step in the mining of electronic commerce data will naturally turn from Web sites to Web services.

In this research, a number of data mining applications that make use of Web services data have been proposed. In particular, they focus on applications that would directly facilitate and improve the use of Web services. The usage range from delivering business value that can be used by management for strategic decision making, to providing technical benefits that target specialist end users. In particular, we focus on the use of data mining techniques in Web services discovery. The proposed search term suggestion for improved Web service discovery is based on the idea of finding search sessions that are similar to the one by a user to locate a particular service, and then suggest words used in those sessions. The usage data at the user request level is only being considered, the process instance execution logs and abstract information of Web services are not being utilized in this process. The experimental results on a data set reveal that the search sessions clustered using the combined measure of keywords and search session viewed are more compact than those of the single similarity measures. This indicates that the suggestion of search terms to users is more relevant when given on the basis of considering query sessions instead of considering one single query.

Recent work on conceptualising Web services with data mining techniques is also included emphasizing that the data mining techniques will play an important role in web service discovery and monitoring. Each suggested application requires performing further testing for validation and usability. We also plan to improve the search term suggestion method by combining the clustering based on session similarity with the query term expansion based on ontology (such as synonyms, hyponym etc). In the future work, we will study whether the method combining these two methods outperforms the individual ones. As data mining applications built on web services become more popular, there will be a growing need for further research to develop data mining algorithms which can scale the large, distributed data sets as well as deal with the concurrent, synchronous and real-time responses delivered for user requests. Most importantly, the data mining benefits should result in cost-saving to the businesses rather than incurring an extra cost to deploy data mining resources.

ACKNOWLEGMENT

Sincere thanks go to Cindy Tong for conducting experiments and analysis, also to students in ITB239, 2005 semester 1 who conducted a survey on Web services and data mining usage.

REFERENCES

DAML-S http://www.daml.org/services/

OWL-S http://www.w3.org/Submission/OWL-S

Agrawal, R., & Srikant, R. (1994). *Fast Algorithms for Mining Association Rules*. IBM Research Report RJ9839, IBM Almaden Research Center.

Alesso, P. & Smith, C. (2004) Developing the Next Generation Web Services - Semantic Web Services, Pub: A K Peters Ltd.

Beeferman, D., & Berger, A. (2000). *Agglomerative clustering of a search engine query log.* The Sixth ACM SIGKDD, Boston, Massachusetts, pages 407--416.

Benatallah, B., Hacid, M., Leger, Alain., Rey, C.,& Toumani, F. (2005) On Automating Web Service Discovery. VLDB Journal. 14 (1):84-96, Springer. (2005)

Gruninger M, Hull R, McIlraith S. (2005) A First – Order Ontology for Semantic Web Services. W3C Workshop on Frameworks for Semantics in Web Services, June 2005, Innsbruck

Chen, M. (2003). Factors affecting the adoption and diffusion of XML and Web services standards for E-business systems. *Int. J. of Human-Computer Studies, 58*, 259-279.

Clark, D. (2002). Next-generation Web services. *IEEE Internet Computing, 6*(2), 12-14.

Devore, J. L. (1995) Probability and statistics for engineering and sciences, Duxbury Press.

Dong, X, Halevy, J. Madhavan, E. Nemes, & Zhang. (2004) Similarity Search for Web Services. In the Thirtieth International Conference on Very Large Data Bases VLDB, Toronto, Canada, August 31 - September 3 2004

Gombotz R., & Dustdar S. (2006) On Web Services Workflow Mining. BPI Workshop, co-located at BPM, Nancy, France, Springer LNCS 3812 pp. 216–228, 2006.

Han, J., & Kamber, M. (2001). *Data Mining: Concepts and Techniques.* San Francisco: Morgan Kaufmann.

Hung P and Li H (2003) Web services discovery based on the trade-off between quality and cost of service: a token-based approach, ACM SIGecom Exchanges, Volume 4 , Issue 2, 2003, Pages: 21 - 31.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data Clustering: A Review. *ACM Computing Surveys (CSUR), 31*(3), 264-323.

Johnson, J. T. (2003). *State of the Web services world*. Retrieved 17 April, 2003, from the http://www.computerworld.com.au/index.php?secid=1398720840&id=622609517

Karypis, G., Han, E. H., & Kumar, V. (1999). Chameleon: A hierarchical clustering algorithm using dynamic modeling. *Computer, 32*(8), 68-75.

Kohavi, R., Mason, L., Parekh R., & Zheng Z (2005). Lessons and Challenges from Mining Retail E-Commerce Data. *Machine Learning, Vol 57, No 1-2, PP 83 -113.*

Kosala R and Blockeel H. (2000). Web Mining Research: A Survey. ACM SIGKDD Explorations, Vol 2, Issue 1, pp 1-15.

Larsen, K. R. T., & Bloniarz, P. A. (2000). A cost and performance model for Web service investment. *Communications of the ACM, 43,* 109-116.

Li L., Yang, Y. & Wu B. (2005) Ontology-Based Matchmaking in e-Marketplace with Web Services. APWeb 2005: Web Technologies Research and Development -7th Asia-Pacific Web Conference, Shanghai, China, March 29 - April 1, 2005. Proceedings 620-631.

Lim S-Y, Song M-H and Lee S-J (2004) The construction of domain ontology and its application to document retrieval. ADVIS 2004, LNCS 3261, pp 117-127, 2004.

Malek M & Haramantzis F. (2004) “Data Mining Techniques for Security of Web Services”, Proc. International Conference on E-Business and Telecommunication Networks (ICETE-2004), August 25-28, 2004, Setubal, Portugal

Mandell, D. & McIlraith, S. (2003) A Bottom Up Approach to Automating Web Services Discovery, Customization, and Semantic Translation. In *The Proceedings of the Twelfth International World Wide Web Conference Workshop on E-Services and the Semantic Web (ESSW '03)*. Budapest, 2003.

McIlraith, S. A., Son, T. C., & Zeng, H. (2001). Semantic Web services. *IEEE Intelligent Systems,16*(2), 46-53.

Mobasher, B. (2005) Web Usage Mining and Personalization. *In Practical Handbook of Internet Computing* Munindar P. Singh (ed.), CRC Press, 2005

Nayak, R. (2002) Data Mining for Web-Enabled Electronic Business Applications, in *Architectural Issues of Web-Enabled Electronic Business*", ed: S. Nanshi, Chapter 8, pp 128-139, Pub: Idea Group Publishers.

Nayak, R., and Seow, L. (2004) Knowledge Discovery in Mobile Business Data, in *Wireless Communication and Mobile Commerce*, editor: S. Nanshi, Chapter 6, pp 117-139, Pub: Idea Group Publishers.

Nayak, R., & Tong, C. (2004). "Applications of Data Mining in web services", *Proceedings of the 5th International Conferences on Web Information Systems*. Brisbane, Australia, 22- 26 Nov, pp. 199-205.

Newcomer, E. (2002). *Understanding Web services: XML, WSDL, SOAP, and UDDI*. Boston: Addison-Wesley.

Paolucci, M., Kawamura, T., Payne, T. R., & Sycara, K. (2002). *Semantic Matching of Web Services Capabilities.* In I. Horrocks and J. Handler, editors, 1st Int. Semantic Web Conference (ISWC), Sardinia, Italia, pages 333--347. Springer Verlag, 2002

Peng, C. S., H. Wang, S. R. Zhang, & D. S. Patker (2000). Landmarks: A new model for similarity-based patterns querying in time-series databases. In Proceedings of the 16 International Conference of Data Engineering (ICDE), San Diego, CA, February, pp 33-42, 2000.

Quinlan R. (1999) Simplifying decision trees. Int. J. Hum.-Comput. Stud. 51(2): 497-510

Sajjanhar A, Jingyu H., & Yanchun Z. (2004) High Availability with clusters of Web Services. APWeb 2004, LNCS 3007, Pg 644–653, 2004

Soydan, A. & Singh, M. (2004) A DAML – Based Repository for QoS – Aware Semantic Web Service Selection. IEEE International Conference on Web Services. (2004)

Srivastava, J., Cooley, R., Deshpande, M., & Tan, P.-N. (2000). Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations, 1*(2), 12-23.

Teng, C. M. (2002). Learning from Dissociations. *The 4th International Conference on Data Warehousing and Knowledge Discovery DaWaK* 2002, Aix-en-Provence, France.

Thuraisingham, b (2005). Privacy-preserving data mining: Developments and directions, *Journal of Database Management*, Vol. 16, No. 1, pp 75 - 87.

Varelas, G, Voutsakis,V. Raftopoulou, P., Petrakis, E. G. and E. E. Milios. (2005) Semantic similarity methods in wordnet and their application to information retrieval on the web. In *WIDM '05: Proceedings of the 7th annual ACM international workshop on Web information and data management*, pages 10–16, New York, NY, USA, 2005. ACM Press. 35

Voorhees, E. M. Query expansion using lexical-semantic relations.(1994) In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information* *retrieval*, pages 61–69. Springer-Verlag New York, Inc., Dublin, Ireland, 1994. 15, 31, 35

Wang, Y., & Stroulia, E. (2003) Semantic Structure Matching for Assessing Web – Service Similarity. First International Conference on Service Oriented Computing. LNCS 2910, pages 194--207. Springer, 2003.

Wen, J.-R., Nie, J.-Y., & Zhang, H.-J. (2002). Query clustering using user logs. *ACM Transactions on Information Systems (TOIS)*, *20*(1), 59-81.

ABOUT THE AUTHOR

Richi Nayak is a senior lecturer in the School of Information System, Queensland University of Technology, Brisbane, Australia. Her research interests are data mining and Web intelligence. She has published over 50 papers in journals, conference proceedings and books. She has developed various innovative mining techniques and applications related to XML, Software Engineering, e-commerce, m-commerce and Web services.