Log Mining to Improve the Performance of Site Search*

Gui-Rong Xue[1] Hua-Jun Zeng[2] Zheng Chen[2] Wei-Ying Ma[2] Chao-Jun Lu[1]

[1]Department of Computer Science and Technology
Shanghai Jiao-Tong University, Shanghai 200030, P. R. China
grxue@sjtu.edu.cn, cj-lu@cs.sjtu.edu.cn
[2]Microsoft Research Asia
49 Zhichun Road, Beijing 100080, P.R.China
{i-hjzeng, zhengc, wyma}@microsoft.com

ABSTRACT

In despite of the popularity of current search engines, people still suffer search failure and lots of non-relevant results when finding some specific information from a specific website. This is because the site search performance is not satisfying as the whole Web search. This paper analyzes the specialty of site search compared with traditional Web search, and the non-applicability of link-based re-ranking techniques such as HITS and PageRank. In this paper, we propose to use log mining to improve the site search performance. With the help of website taxonomy, a generalized association rule mining technique is applied to users' log to abstract the user's access patterns at different levels, and the mining results are then applied to re-ranking the retrieved pages. Our mining algorithm tackles the diversity problem of user's access behavior and mines out general patterns. The experimental results show that our proposed method outperforms keyword-based method by 15% and DirectHit by 13% respectively.

Keywords

Site Search, Log Mining, Generalized Association Rule, Taxonomy, Re-Ranking

1. INTRUDUCTION

Global search engines such as Google, AltaVista, and Lycos have been a great help for users to find desired information on the ever growing Web. Given clear and unambiguous queries, they can return just what you want most of the time. But this isn't always the cases. As pointed out in [9], people often pose unclear and general queries to Web search engines to find appropriate websites as good starting points. Once at the site, the user has a choice of following hyperlinks or using site search to get specific information they desired. Due to the low-efficiency of following hyperlinks, there is a tremendous need for site search techniques. As reported in a recent Forrester survey [14], website managers also consider search to be a critical factor of their sites' functionality.

Site search can be simply defined to be search functionality specific to one site. However,

* This work was performed in Microsoft Research Asia.

unlike the general search engines, site search engines are notoriously problematic at present. In [14], Forrester tested site search facilities of 50 websites, but none of them received satisfying result. For example, they often didn't find the content that best matched what the user wanted; they rarely put all the best content on the first page of results; they typically returned more irrelevant results than useful ones. What are the reasons of these failures?

First of all, most site search engines merely use "full text search" technology which retrieves a large amount of documents containing the same keywords inputted by the user, such as the Cha-Cha system [12] and the navigation system [13]. Due to the shortness of user's queries and poor ranking mechanism, it is a time consuming job for the users to go through the results to find out their really desired information.

Many techniques, which are very successful in Web search, seem directly applicable in site search, such as link analysis [6][23] and clickthru-based ranking [3]. But both of them couldn't work well for the following reasons. The link analysis techniques, such as HITS [6] and PageRank [23], use the Web link information to increase the rank of high referenced pages. However the link information within a website isn't so strong to reflect the page's reference number. It's only the editor's organization to the website. Thus the most important Web page need not to be the highest referenced page; high referenced pages are often home page, index pages and help pages which are not really wanted by users. The failure of applying link analysis to Web TREC datasets [2] also demonstrate that link analysis doesn't work for a sub-space of Web.

DirectHit [3] is a clickthru-based ranking method used in global search engines. According to a particular query, it utilizes the previous session logs of same query to return pages that most users visit. To get a statistical significant result, it is only applied to a small set of popular queries. Because of the lack of previous query sessions and the diversity of user's access patterns, DirectHit doesn't work for site search, as well.

In this paper, we proposed a novel re-ranking method based on site logs. Any website keeps a set of access logs, which record each browsing behavior of its users and the time, duration and URL. We can get from these logs each page's access frequency and the traversal patterns of information finding. They reflect Web pages' importance in users' point of view and associations among Web pages, which can be used to improve the performance of site search. Generally, the process of discovering useful patterns from Web logs is called *log mining* [21] (or *usage mining*).

Log mining includes straightforward statistics, such as page access frequency, as well as more sophisticated forms of analysis, such as association rule mining, sequential pattern mining, clustering, and etc. In this paper, we are particularly interested in association rule mining [22]. To our best knowledge, there is no effort devoted to improving the site-search performance by association rule. This paper will present such a method in detail.

A normal association rule mining algorithm may often fail to discover significant rules due to the data diversity problem. Comparing to normal association rule mining, generalized

association rule mining utilizes a predefined taxonomy, and extracts significant association rules at different abstract level of the taxonomy. We designed an algorithm which is based on the FP-growth algorithm [5] to mine the generalized association rules efficiently. After the association rules are generated and pruned, they are applied to re-rank the search results given a query.

Here we summarize the contributions of our work: (1) proposal of utilizing Web page access logs to improve site search, (2) an efficient tool that do log preprocessing, taxonomy generating, generalized association rule mining, rule pruning, and search result re-ranking, and (3) experiment of our method on Berkeley CS website, in which the result outperforms keyword-based method by 15% and DirectHit by 13% respectively.

The organization of the paper is as follows. In the next section, we review the related work. In Section 3, some basic terms and algorithms used in log mining are given. Then, in section 4, we mine the generalized association rules and integrate the result into site search. In Section 5, we present our experimental results related to this new method, and conclude in Section 6.

2. RELATED WORK

Because of the importance of the site search, many works have been done to improve the function and performance of the site search. In the mean time, association rule mining has also received much attention on various areas, such as recommendation system, collaborative filtering, site modification, etc. These two research topics were formerly developed independently. In this section, we discuss some conventional approaches of site search and applications of association rules mining.

Traditionally, most site search engines merely use "full text search" technology, which retrieves a large amount of documents containing the same keywords provided by the user. For example, the Cha-Cha system [12] is such kind system, but it does not present the results in a ranked list. As an alternative, it organizes Web search results in a way as to reflect the underlying structure of the intranet. In their approach, an “outline” or “table of contents” is created by first recording the shortest paths in hyperlinks from root pages to every page within the Web intranet. After the user issues a query, those shortest paths are dynamically combined to form a hierarchical outline of the context in which the search result occur. Therefore, Cha-Cha system‘s goal is to provide well-organized search results. It can’t save users’ time when the relevance of search result is poor.

M. Levence [13] creates a navigation system for semi-automation user navigation which builds trails of information, i.e. sequences of linked pages, which are relevant to the user query. The best it can do for this type of query is provide a good starting point for the user to initiate a navigation session. But the system does not further use the associated relation of the pages to help user navigating and re-rank the search result.

In the following, we briefly discuss the idea of association rules and its applications. Since its introduction in 1993, association rule mining has received a great deal of attention. Today the

mining of such rules is still one of the most popular pattern-discovery methods in data mining and log mining.

An example of association rule is:

- 60% of the users who visit the page *A* also visit the page *B*

where the percentage is refereed to as *confidence*.

Earlier works on applying association rule to log mining focused on many fields. They have been used to mine path traversal patterns and to facilitate the best design and organization of Web pages [20][10][11]. Some recommender systems [25] have been developed for recommending Web pages using the *Apriori* algorithm to mine association rules over users' navigation histories. In other cases, e.g., access prediction [26], pages pre-fetching [15], the association rule is also utilized to find users' access patterns.

3. PRELIMINARIES

In this section, we list some basic terms and algorithms used in log mining.

Association Rules: Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of *items*. An *association rule* is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$.

For example, suppose that user who accessed the page *a* and page *b* also tend to access the page *c*. In that case, the corresponding association rule is "$a \wedge b \Rightarrow c$". The antecedent of the rule *X* consists of *a* and *b*, and the consequent *Y* consists of *c*.

Confidence and Support: Let *D* be a set of transactions, where each transaction $T \in D$ is a set of items (itemset) such that $T \subseteq I$. We say that a rule $X \Rightarrow Y$ holds in transaction set *D* with *confidence c*% if *c*% of transactions in *D* that contain *X* also contain *Y*. We say that a rule $X \Rightarrow Y$ holds in transaction set *D* with *support s*% if *s*% of transactions in *D* contain both *X* and *Y*.

Returning to our example, suppose we find that in 90% of transactions in which user accessed the page A and page B, they also accessed C. Moreover, say that 5% of transactions include all three items. In that case, the confidence of the rule is 90% while its support is 5%.

Association Rule Mining Problem: Given a set of transactions *D*, the problem of mining association rules is to generate all association rules that have support *s*% at least as great as some user-specified minimum support $s_{\min}$% and confidence *c*% at least as great as some user-specified minimum confidence $c_{\min}$%.

Several algorithms have been presented in the literature [5][16][17] for finding all such association rules. Many of them are variations on the *Apriori* algorithm, which works in two phrases: (1) it finds all itemsets that have support above the minimum support; and (2) it uses these itemsets to generate all rules whose confidence are above the minimum confidence.

Taxonomy: By taxonomy, we mean "is-a hierarchy" where a node's descendents represent specializations of that node. Figure 1 shows an example of a taxonomy.

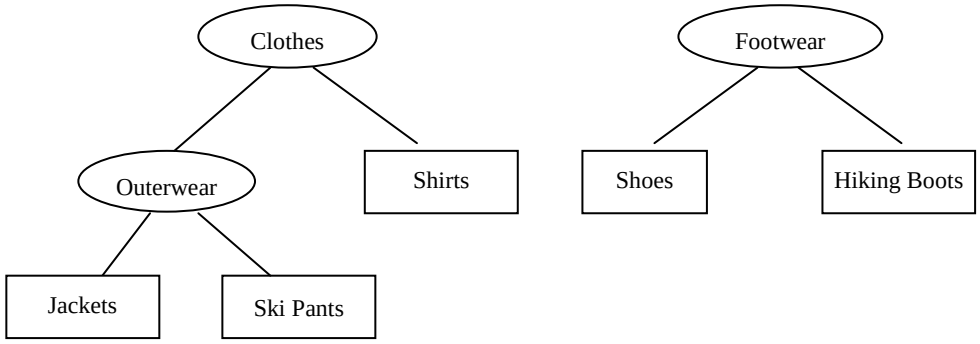

Figure 1: Taxonomy Structure

For example, taxonomy may indicate that Jackets is a kind of Outwear. In that case, the taxonomy would have Jackets as a descendent of Outwear.

Formally speaking, we model one or more taxonomies as a directed acyclic graph Γ on the items $I = \{i_1, i_2, \ldots, i_m\}$. Edges in Γ denote "is-a" relationships among items. Specifically, an edge from c up to p in Γ indicates that p is the parent of c and means that c is a particular kind of p (or in other words, that p is a generalization of c).

Generalized association rules: Generalized association rules [22] improve upon standard association rules by incorporating a taxonomy Γ. In particular, a generalized association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. And no item in Y is an ancestor of any item in X. The reason for the latter requirement is that any rule of the form "$x \Rightarrow$ ancestor(x)" is true with 100% confidence and consequently redundant.

Now, we say that a generalized association rule $X \Rightarrow Y$ holds in transaction set D with confidence c% if c% of transactions in D that contains X or a descendent of X, also contain Y or a descendent of Y. Moreover, itemset X has support s% in transaction set D if s% of transaction in D contains Z or a descendent of Z. Support for rules is defined as follows: $X \Rightarrow Y$ has support s% if the itemset $X \cup Y$ has support s%.

4. GENERALIZED ASSOCIATION RULE MINING FOR SITE SEARCH

We developed a general log mining tool to discover generalized association rules for site search. This tool is illustrated in Figure 2.

As can be seen from Figure 2, the tools use the log files and a taxonomy as input, and output a set of association rules. When a user pose a query, a full-text search engine first find all the Web pages that matches the query words, these Web pages are then re-ranked by mined association rules dynamically.

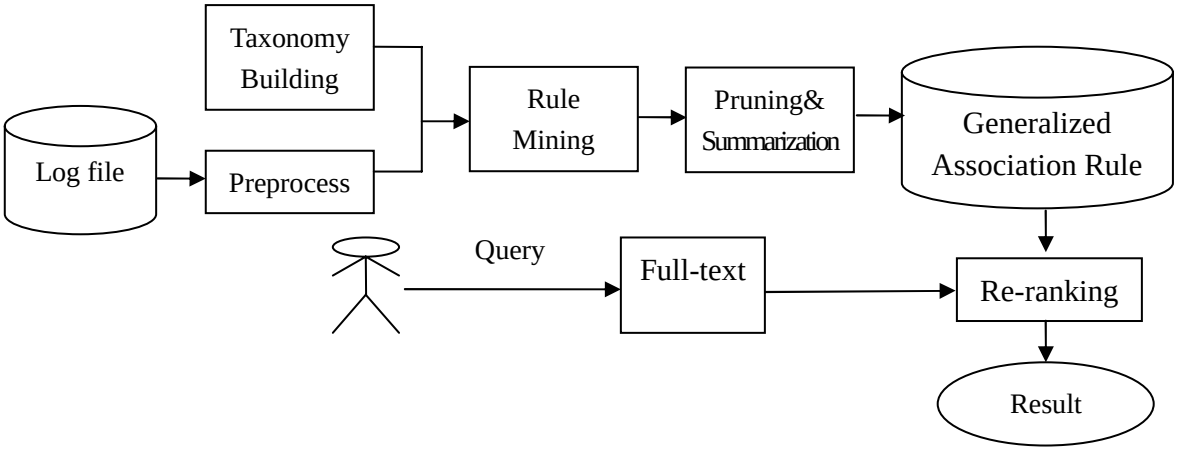

Figure 2 Model of site search using generalized association rule

Our tool mines generalized association rule instead of standard association rule to tackle the data diversity problems. Earlier works on association rule mining [1][4][7][21] only mine the relationship among distinct Web pages, which lead to three main problems:

- First, our statistics on a real Web log show that most of the pages have low hit rate. As can be seen from Figure 5, about 80% of the pages are visited less than 10 times. Therefore, if we mine the standard association rule, those pages are always ignored. However, in most of cases, these pages may contain latent information about the user's access patterns which can't be mined using standard association rule.
- Second, a website usually contains thousands, even millions of pages. It is not easy to find those users who access some common pages. This is because of the diversity of the users. Moreover, this also leads to the difficulty of finding the same access pattern.
- Third, there are some latent semantic topics in a website. For example, there exist two topics in Berkeley CS's website: AI and machine learning, each consisting of several pages. These two topics are frequently co-visited. However, the standard association rule is unable to find the relationships between these two topics.

To overcome the above problems, we can utilize existing taxonomies, which contain some semantic information about the website, such as content cluster, site directory. Traditional log mining algorithms only discover the leaf-level items in the taxonomy. While our method make full use of the taxonomies, which allows us to mine the association rules at different levels of abstraction. We wish access patterns may contain some interesting regularities at higher levels of abstraction. For example, the rule "$A \Rightarrow B$" may have insufficient support using standard association rule mining, but the rule "ancestor(A)$\Rightarrow B$" may pass the support requirement. That is because additional transactions may support "$A' \Rightarrow B$", where "ancestor (A) =ancestor (A')".

Our method is detailedly stated in the following sub-sections.

4.1. Preprocessing

The starting and critical point for successful log mining is data pre-processing. The required tasks are data cleaning, user identification, and session identification.

An entry in Web server log contains the timestamp of a traversal from a source to a target page, the IP address of the originating host, the type of access (GET or POST) and other data. Many entries are considered uninteresting for mining and are removed. The filtering is application dependant. However, in most cases accesses to images are filtered out.

Table 1 Example of Log file

#	IP Address	Time/Date	Protocol	Request URI	Result
1	24.5.193.7	11/1/2001 12:00:30 AM	GET/HTTP 1.0	/~demmel/cs267/tsp/doc/taskq/node15.html	200
2	159.226.21.3	11/1/2001 13:03:08 AM	GET/HTTP 1.0	/~eran	200
3	169.229.90.77	11/1/2001 14:00:32 AM	GET/HTTP 1.0	/~chema/papers/dns/dns_report/node2.html	200
4	216.126.153.59	11/1/2001 15:00:00 AM	GET/HTTP 1.0	/~culler/cs258-s99/slides/lec05/sld026.htm	200

The remaining entries must be grouped by the visitor that performed them. An investigation on such approaches can be found in [19]. We currently assume that consecutive accesses from the same host during a certain time interval come from the same user.

Once we assess the originator of each entry, we group consecutive entries to a user session or “transaction”. Different grouping criteria are modeled and compared in [19]. We support two criteria:

(1) A new session starts when the duration of the whole group of traversals exceeds a time threshold, similarly to [19].

(2) The elapsed time between two consecutive traversals exceeds a threshold.

After preprocessing, we can get a set of n pages, $P= \{p_1, p_2, \ldots, p_n\}$, and a set of m transactions, $T=\{t_1, t_2, \ldots, t_m\}$,where each $t_i \in T$ is a subset of P.

4.2. Site Taxonomy Building

By analyzing the Web pages, we found that pages are not randomly scattered. Many websites have a hierarchical organization of content, called page hierarchy. A page hierarchy is a partial order of Web pages, in which a leaf node represents a Web page corresponding to a file in the website. A non-leaf node in a page hierarchy represents a Web directory in the website.

We construct taxonomy Γ using the hierarchy above; the taxonomy is initialized with only the root which represents the top level of the website. For each URL in the URL list generated, if it does not exist in the taxonomy, a node for the page is created. Next we parse the URL and for each prefix, which is a directory, we create an ancestor node if it doesn’t exist in the

taxonomy. For example, for the URL http://www.cs.berkeley.edu/~jordan/courses/index.html, the node is itself, its first prefix http://www.cs.berkeley.edu/~jordan/courses/, and its second prefix http://www.cs.berkeley.edu/~jordan/, may be created and a link is added between itself and its first prefix, between its first prefix and its second prefix, and between its second prefix and the root. The Figure 3 is the result of the website http://www.cs.berkeley.edu:

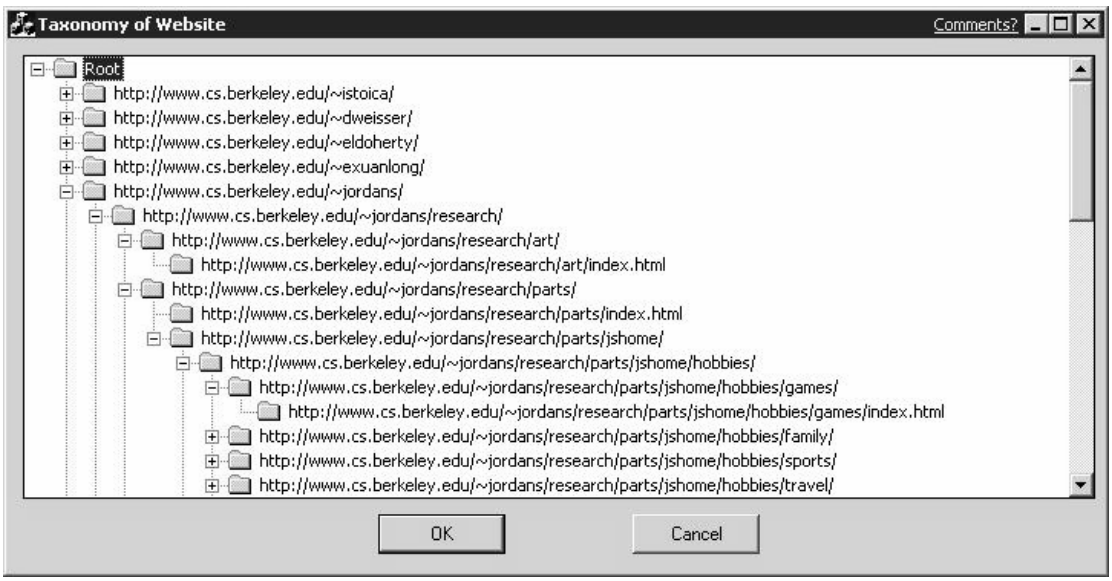

Figure 3. A website taxonomy

4.3. Mining Algorithm

To support taxonomies, one can use algorithms *apriori* [17] for mining standard association rules by considering "extended transaction" that contains not only the items in transactions but also their ancestors. To make this process efficient, certain optimizations are done to restrict the number of itemsets that need to be counted at various stages in the algorithm and the number of ancestors added to form extended transaction.

The performance of the algorithm *apriori* is poor, because there exist at least 10^5 items, and millions of candidate itemsets being created in multiple scans of the database. So we designed a generalized version of FP-growth algorithm [5].

The basic algorithm of generalized FP-growth has three steps: (1) generalized FP-tree construction, (2) generalized FP-tree generation and (3) optimization. The first step and the second step are same as [5], the difference is that the transaction is change to extended transaction which includes the parent nodes of original transaction nodes. The third step is to the optimization of algorithm according to the following property [22]:

> The support of an itemset X that contain both an item x and its ancestor x' will be the same as the support of the itemset X-$\{x'\}$.

After the frequent itemsets have been created, the algorithm generates the association rule as

described in [17];

4.4. Pruning and Summarization

We do not present all generalized association rules to the user but only those that we deem "interesting". This is particularly important since, in practice, we find that there are thousands of similar and/or redundant rules that would overwhelm the user otherwise.

Uninteresting Rules: Suppose we have two association rules: $X \Rightarrow Y$ and $X' \Rightarrow Y$, where X' is a parent of X. We define the rule $X \Rightarrow Y$ is *uninteresting*, iff

$$\text{Support}(X \Rightarrow Y) / \text{Support}(X' \Rightarrow Y) \approx \text{Support}(X) / \text{Support}(X')$$

#	Rule	Supp.
1	Outerwear $\Rightarrow$ Footwear	8
2	Jackets $\Rightarrow$ Footwear	4

Item	Supp.
Outerwear	2
Jackets	1

Figure 4 Examples of interesting and uninteresting rule

The proof is given by [22]. As the example shown in Figure 4, assuming we have the same taxonomy as in Figure 1, we do not consider rule 2 to be interesting since its support can be predicted based on rule 1. In other words, we can see from the right table that just 1/2 of amount of buying Outerwear are buying Jackets. Afterward, we can calculate the support of Jackets$\Rightarrow$Footwear to be just 1/2 of the support of Outerwear$\Rightarrow$Footwear. Thus the rule Jackets$\Rightarrow$Footwear is uninteresting.

Mutual Information Based Pruning: According to [24], if two points, x and y, have probabilities $P(x)$ and $P(y)$, then their mutual information, $I(x, y)$, is defined to be

$$I(x, y) = \log_2 \frac{P(x, y)}{P(x)P(y)}$$

Informally, mutual information compares the probability of observing x and y together (the joint probability) with the probabilities of observing x and y independently (chance). If there is a genuine association between x and y, then the joint probability $P(x, y)$ will be much larger than chance $P(x)P(y)$, and consequently $I(x, y) >> 0$. If there is no interesting relationship between x and y, then $P(x, y) \approx P(x)\,P(y)$, and thus, $I(x, y) \approx 0$. If x and y are in complementary distribution, then $P(x, y)$ will be much less than $P(x)\,P(y)$, forcing $I(x, y) << 0$. Suppose that there is a rule $X \Rightarrow Y$, if $\text{supp}(X \Rightarrow Y) \approx \text{supp}(X) \times \text{supp}(Y)$, then it's likely that X and Y are independent and rule $X \Rightarrow Y$ is uninteresting, so we prune it.

Summarization: The following Rules

$$A, B \Rightarrow C \qquad A, C \Rightarrow B \qquad B, C \Rightarrow A$$

can be defined as association hyperedges, i.e., sets of items that are strongly predictive w.r.t. each other. The selection criterion is as follows: Given an item set with enough support, all rules are checked which can be formed using this set with all items appearing in the rule. For example, for the item set {*A B C*}, the rules $AB \Rightarrow C$, $AC \Rightarrow B$ and $BC \Rightarrow A$ would be considered. If the confidence of each rule is greater than the minimal confidence, the item set is selected. The confidence of the itemset is the average of the confidence of all rules;

For example: two rules:

$$A \Rightarrow B \text{ (Supp.:16, Conf.: 93\%)}$$

$$B \Rightarrow A \text{ (Supp.:16, Conf.: 97\%)}$$

They are association hyperedges; we group them together and their confidences are the average of the confidences:

$$\{A, B\} \text{ (Supp.:16, Conf.: 95\%)}$$

After pruning and summarization, about 40% of the rule can be reduced.

4.5. Re-ranking

We propose a novel algorithm to re-rank the result using generalized association rule. In general, the pages in an association rule are accessed frequently together and mostly the content of the pages are relative, so we can use the association rule to improve the performance of site search.

Our algorithm is similar to DirectHit. Both of them make use of the previous users' query sessions. DirectHit algorithm uses the click popularity to improve the performance of the search. The higher frequency a Web page is visited by user, the more important the Web page is. Comparing to DirectHit, our algorithm utilizes the popular clicked pages of the same query and improves the associate pages' rank which is based on the association rules.

First we implement the DirectHit algorithm on a "full text search" engine. DirectHit uses the sessions of the users' queries and the pages which are relative to the users' queries.

1. Through the site search engine, the user inputs a query word *Q*, then the search engine returns a result set *D* with the score which is based on the similarity of the page *d* and the query *Q*.
2. The pages are re-ranked according to the similarity score and the click popularity;

$$Score(d) = \alpha \times Sim(d) + (1 - \alpha) \times Pop(d)$$

where *Sim* (*d*) is the similarity between the query *Q* and the page *d*, the *Pop* (*d*) is the click popularity of the page *d*.

Then we propose our algorithm using generalized association rules; likewise we need the sessions of the previous users' queries.

1. The user inputs a query word *Q*, and get a result set *D* with the score which is based on the similarity between the page and the query.
2. From the previous query sessions, we get the most popular 5 pages as a set *P* which is selected by previous users using the same query.
3. Then we get the rules whose antecedents contain the pages in *P* or the ancestors of the pages in *P*, and acquire the descendant of the rules as a set *R*.
4. According to *R*, we calculate the support and the confidence of each page *d* in the result set *D*, if the page *d* is in *R*, we calculate the support and the confidence of the page *d* directly, if a parent node of *d* is in *R*, we calculate the support and the confidence of the page *d* according to the ratio of the page to its parent.
5. The result is re-ranked according to the similarity, the support and the confidence;

$$Score(d) = \alpha \times Sim(d) + (1-\alpha) \times Supp(d) \times Conf(d)$$

where *Sim* (*d*) is the similarity between the query *Q* and the page *d*. *Supp* (d) and *Conf* (*d*) are the support and the confidence of the rule of the page *d* respectively.

5. EXPERIMENTAL RESULTS

In order to test the effectiveness of our proposed algorithm, we use the Berkeley CS website as the experiment. We downloaded one-month's log file from http://www.cs.berkeley.edu/logs. The log records users' visit information, in which one record is corresponded to one HTTP request for a Web object by a specific user. After preprocessing, only text pages are reserved in the final dataset, which contains 112,059 pages, 296,667 users and 1,474,389 visit records. Figure 5 shows a statistic of frequency distribution of the Web pages.

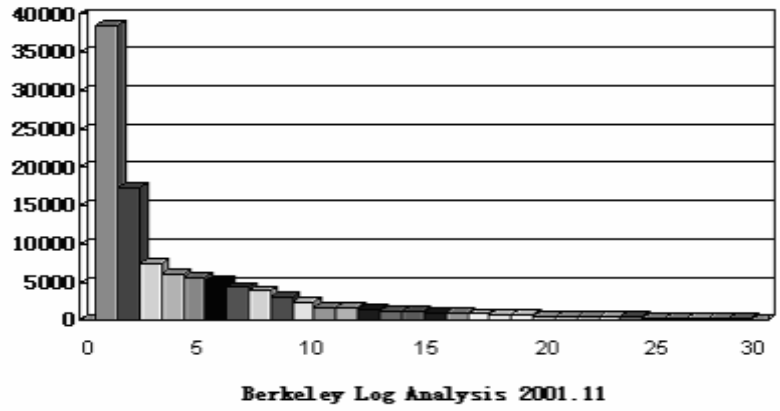

Figure 5 Frequency distribution of the pages
X axis: the hits number of a page, Y axis: the page numbers

In our experiment, we mine generalized association rules with confidence of at least 60% and

support of at least 10 times in transactions.

To compare generalized association rule mining with standard association rule mining, we also mine the association rule without taxonomy. Mining generalized association rules took about 19.7 minutes while mining standard association rules took about 11.4 minutes. The rule statistics are as Table 2.

Table 2 Rule statistics

Methods	Rules Mined	Pruning		Summary	Rules Left
		Uninteresting	MI		
Generalized	246538	3042	32204	130024	146280
Standard	101304	0	16537	49832	59848

Observe that although the number of rules mined by generalized association rules is significantly more than that using stand association rules, a greater percentage of those rules is eliminated during pruning.

After the pruning and summarization, we find something very interesting in the result. We find that the most items in the standard association rules have hyperlink relationships. But through the generalized association rule mining, we can discover not only the standard rule, but also high level relationship, which can not be explored by hyperlink structure. For example, In

Table 4, the rule 1 shows that *~adj*, *~bh* and *~ddgarcia* are the pages, which are pertaining to the staff who work on computer vision. And we can also discover the rules, which can find those pages being of the same interest and about the same courses. The following is the result of our experiments. And the remark in Figure 7 is the topic of the real website.

Table 3 Results of Association Rule

Association Rule	Supp.	Conf.
~qtluong/gallery/slides-ice/folio-ice2b.html ⇒ ~qtluong/gallery/index.html	5.0%	83%
~luca/cs170/project/ex1.dat ⇒ ~luca/cs170/index.html	3.2%	93%
~gnguyen/ns/navbtn.htm ~gnguyen/ns/outlinec.htm ⇒ ~gnguyen/ns/img001.htm	2.4%	90%

Table 4 Results of Generalized Association Rule

#	Generalized Association Rule	Supp.	Conf.	Remark
1	~adj ~christos people/faculty/homepages ~bh ⇒ ~ddgarcia	0.6%	98%	Computer vision
2	~eanders/pictures ~qtluong/landscapes/lf ~qtluong/photography/lf ⇒ ~jhauser/pictures/history	0.6%	87%	Picture
3	~harrison ~hilfingr ~yelick people/faculty/homepages⇒ ~jordan	0.6%	97%	Machine Learning
4	~yuhong ~yinli ~xiaoye ⇒ ~xia	0.3%	100.0%	Chinese Staff
5	~wilensky/cs188/lectures ~xuanlong/cs188⇒ ~wilensky/cs188/assignments	0.3%	95%	Same Course

The mining result is then used to improve the performance of the site search according to algorithms described in section 4.5.

We compare our algorithm with the pure text-based site search engineer, DirectHit algorithm, association rule and generalized association rule. We run the 4 algorithms on each of the query and evaluate the precision. Several volunteers are required to do some tests on our platform to pose the following ten queries and evaluate the relevance of searching results. We then computed precision at top 20 pages for each algorithm-query pair. Following is our queries: *hyperlink, EM, object oriented program, reinforcement learning, vision, Bayesian network, HMM, data mining, picture, Jordan*. The result is shown in Figure 6.

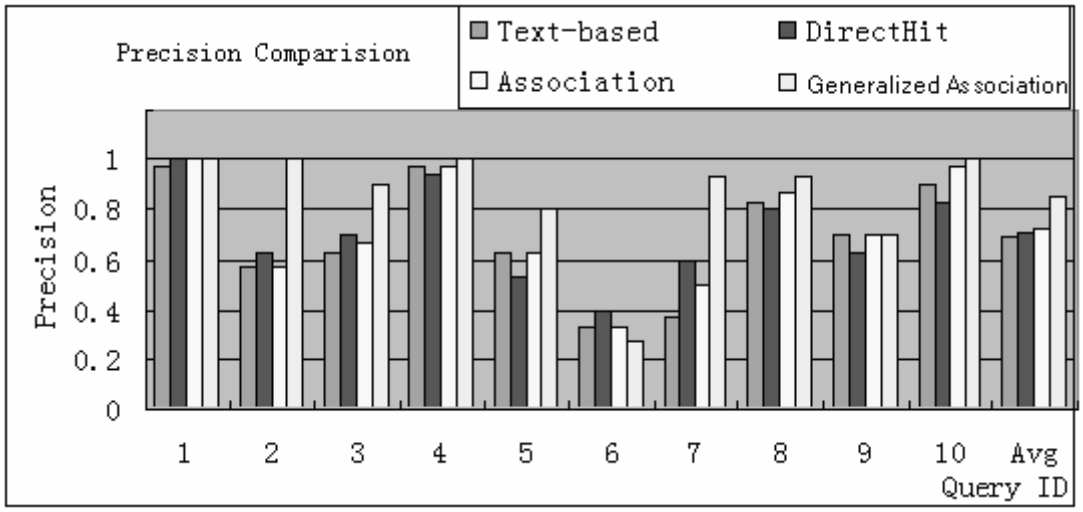

Figure 6 Precision comparison for 4 algorithms

In the experiment, we set the parameter α to be 0.7, The comparison of precision for 4 algorithm is

shown in Figure 6. The result labeled as Avg is the average of all of the above 10 queries. According to Avg, we found that our proposed algorithm outperform the “full text search” and DirectHit algorithm. The average improvement of precision over the “full text search” is 15% and 13% for DirectHit.

We can see from Figure 6 that, the association rule doesn’t improve the search result significantly. Based on our analysis, we find that few standard association rules contain the pages which are both in query result because of the diversity of the users’ access behaviors and complexity of the website. Using the generalized association rule, some associated topics can be discovered. In a topic, there are many pages which are similar in content, so the ratio of co-occurrence of the pages in the query result is increased. Hence it can improve the performance of site search. DirectHit incorporates the frequency of user’s access to compute the page score. The higher frequency a Web page is visited by user, the more important the Web page is. However, according to our statistics, in most of cases, the user only clicks the first 1-10 pages which are ranked by the page score. Therefore, DirectHit can not make significant improvement for the site search.

6. CONCLUSIONS AND FUTURE WORK

In this paper, the association rule mining for user access patterns has been discussed. We proposed a generalized association rule mining method, which utilizes a taxonomy of website to mine the different level association rules, also we proposed a method of using generalized association rule to improve the performance of site search. Our experiments show that the method is efficient and feasible for site search.

The construction of the taxonomy based on the URLs of the pages implies that the underlying page organization reflects the semantics of the pages. In case this cannot be assumed, the taxonomy should be constructed according to the semantics of the pages, e.g., using content-based hierarchical clustering or classification method.

The results of generalized log mining represent the user’s view to the website, and also reflect the user’s access pattern. A natural extension of our method is to combine the user’s access pattern and link analysis together to improve the performance of site search. According to the link space of the website is too sparse and link information is only the editor’s view to the organization of the website, our main line of the future research involves combining the users’ access pattern and link information together, and re-ranking the Web pages, and ultimately improving the performance of the Web search.

7. REFERENCES

[1] C.-H Yun and M.-S. Chen, Mining Web Transaction Patterns in an Electronic Commerce Environment, Proceedings of the 4th Pacific-Asia Conf. on Knowledge Discovery and Data Mining, April 18-20, 2000.

[2] D. Hawking, E. Voorhees, P. Bailey, and N. Craswell. Overview of TREC-8 Web Track. In Proceedings of TREC-8, pages 131–150, Gaithersburg MD, November 1999.

[3] DirectHit: http://www.directhit.com.

[4] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. Data Mining and Knowledge Discovery, 1(3): 259 - 289, November 1997.

[5] J. Han, Jian Pei, Yiwen Yin, Mining Frequent Patterns without Candidate Generation, 2000 ACM SIGMOD Intl. Conference on Management of Data.

[6] J. Kleinberg, Authoritative Sources in Hyperlinked Environment, in Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithm, 1998.

[7] J. Pei, J. Han, B. Mortazavi-Asl, and H. Zhu, Mining Access Pattern efficiently from web logs, Proceedings 2000 Pacific-Asia Conf. on Knowledge Discovery and Data Mining, April 2000.

[8] M. Chen, J. Park and P. S. Yu, Efficient Data Mining for Path Traversal Patterns, IEEE Trans. on Knowledge and Data Engineering, Vol. 10, No. 2, pp. 209-221, April 1998.

[9] M. Hearst, Next Generation Web Search: Setting Our Sites. IEEE Data Engineering Bulletin, Special issue on Next Generation Web Search, Luis Gravano(Ed.), Sep 2000

[10] M. Spiliopoulou, and C. Pohle, Data mining for measuring and improving the success of Web sites. Data Mining and Knowledge Discovery, 5:85–14. 2001

[11] M. Spiliopoulou, C. Pohle, and L. Faulstich, Improving the effectiveness of a Web site with Web usage mining. In Advances in Web Usage Analysis and User Profiling. Berlin: Springer, pp. 142–162, 2000.

[12] M.Chen, M.Hearst, J. Hong and J.Lin Cha-Cha: A System for Organizing Intranet Search Results, In Proceedings of the 2nd USITS, Boulder, CO, October 11-14, 1999.

[13] M.Levene, R.Wheeldon A Web Site Navigation Engine Proceedings 10th International WWW Conference, 2001

[14] P. Hagen, H. Manning, and Y. Paul. Must search stink? The Forrester report, Forrester, June 2000.

[15] Q. Yang, H. Hanning Zhang and I.Tianyi Li. Mining Web Logs for Prediction Models in WWW Caching and Prefetching . In The Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD'01, Industry Applications Track, August 26 - 29, 2001.

[16] R. Agrawal and R. Srikant, Fast Algorithms for Mining Association Rules, Proceedings of the 20th Int'l Conference on VLDB, Santiago, Chile, Sep. 1994.

[17] R. Agrawal, T. Imielinski, A. Swami, Mining Associations between Sets of Items in Massive Databases, Proceedings of the ACM-SIGMOD 1993 Int'l Conference on Management of Data, Washington D.C., May 1993

[18] R. Baeza-Yates and B.Ribeiro-Neto. Modern Information Retrieval. Addison-Wesley, 1999.

[19] R. Cooley, B. Mobasher, and J. Srivastava, Data Preparation for Mining World Wide Web Browsing Patterns, Knowledge and Information Systems V1(1). 1999.

[20] R. Cooley, P. Ning Tan, J. Srivastava, Discovery of Interesting Usage Patterns from Web Data , To appear in Springer-Verlag LNCS/LNAI series, 2000.

[21] R. Kosala,H. Blockeel, Web Mining Research: A Survey, ACM SIGKDD，July 2000.

[22] R. Srikant and R. Agrawal, Mining Generalized Association Rules, Proceedings of the 21st Int'l Conference on Very Large Databases, Zurich, Switzerland, Sep. 1995.

[23] S.Brin and L. Page, the Anatomy of a Large-Scale Hypertextual Web Search Engine, Proceedings 7th International WWW Conference, 1998

[24] T. Cover and J. Thomas. Elements of Information Theory. John Wiley & Sons, 1991.

[25] W. Lin, S. Alvarez, C. Ruiz Efficient Adaptive-Support Association Rule Mining for Recommender Systems, Data Mining and Knowledge Discovery, 6, 83–105, 2002

[26] Z. Albrecht and A. Nicholson, Predicting users' requests on the WWW, in J Kay (ed), CISM Courses and Lectures No. 407, International Centre for Mechanical Sciences, Proceedings of the Seventh International Conference on User Modeling (UM-99), Banff, Canada, 20-24 June, 1999.