International Journal of Information Sciences and Application.
ISSN 0974-2255 Volume 3, Number 2 (2011), pp. 65-78

http://www.irphouse.com

Web Mining: A Solution to Web Searching Inefficiencies

Nnebedum V. I.

Department of Electrical and Electronics Engineering
Federal University of Technology Owerri, Nigeria

Abstract

The Web is considered as the biggest database - estimated to be in tens of tetra bytes. Traditionally, information in the Web is assessed through a retrieval technology called Web search engines. Once a *query*, typically of *keywords*, is typed into a Web search engine, a list of *best-matched* Web pages appear. Nonetheless there are still some challenges on *precision* and *low recall* due to irrelevance of many of the search results. This paper tries to reveal these gaps as the performance of three popular search engines are evaluated. Most importantly, a solution was proffered by introducing the data mining approach in retrieving information from the Web. Web mining models designed provided solutions to the inefficiencies of Web searching.

Keywords: Data mining, Web mining, Text mining, Search engine, Web browser, Crawling, Indexing, HTML, Meta tag and Algorithm.

Introduction

The Web is a massive database, and millions of new Web pages are loaded into it every day, to spread information, to advertise services or even for entertainment [Diego, 2007]. All data in the Web are not stored in a single computer. The data are spread over many computers stationed over geographical areas.

To navigate and bring up needed page(s). Search engines need to *crawl*, *index* and *search* quickly billions of the pages, for millions of users. Not only that the database is massive, data in the Web are much less coherent and they changes more rapidly. These challenges are faced by search engines and this is why they end up retrieving "irrelevant" and or "unverified" information [Steven, 2004].

Web Search Inefficiencies: Investigations, Results and observations

The usefulness of a search engine traditionally depends on the relevance of the result

set it gives back. While there are millions of Web pages that include a particular word or phrase, some pages may be more *relevant*, *popular*, or *authoritative* than others. Obviously some search engines performs better that the others in *locating* and *ranking* these pages. In fairness, there are remarkable improvements in the search results nowadays as some search engine organisations are quickly including some new emerging information retrieval technologies not covered in this paper. This is evident in the evaluation conducted in this research, because same search by different search engines produces different search results. But in all, the challenges of *precision* and *low recall* leading to the retrieval of "irrelevant" information are still there.

Investigation

A practical experiment was conducted (17th January 2010, 15.00 hrs) to actually compare a search by three search engines (Google, Yahoo!Search and MSN-Bing). The assessment greatly depends on the *effectiveness and efficiency* of the search engine. This evaluation is based on *personal satisfaction* that is devoured of *any pre-conceptions* of which search engine is better.

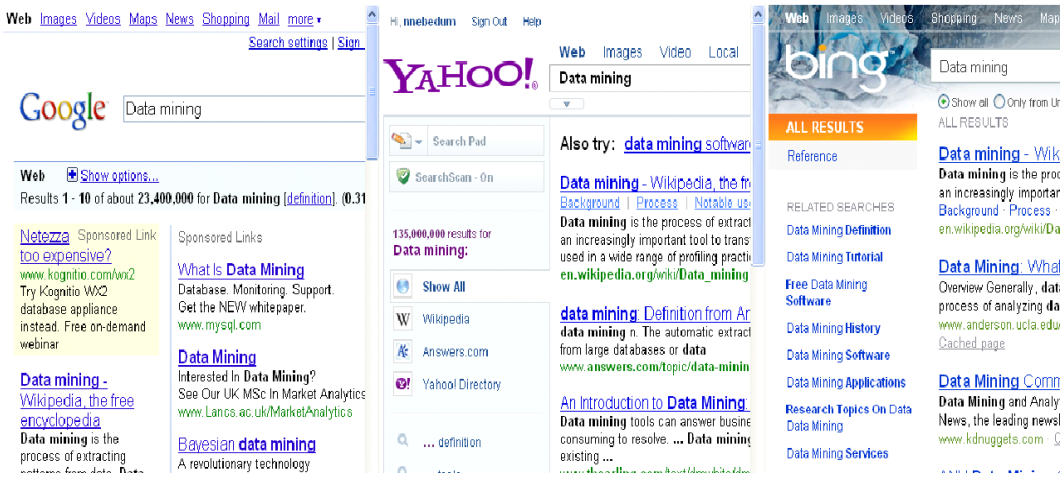

Figure 1: Evaluating three search engines.

Three *informational* keywords used are "Search engine", "Data mining" and "FUTO EEE Department". They are entered in each of the engines as shown in figure 1 above.

The first 50 pages fetched from each search engine were carefully evaluated:

- To ascertain the number of Web pages fetched by each search engine,
- To know how "relevance" the fetched pages are in relation the knowledge sought,
- To find out how the fetched pages are ranked (i.e. displayed according to importance) in 10s per view, and,
- To check the speed of fetch (though not recorded in the Table 1).

Results

Table 1: Search results on keywords on search engines.

Keyword = "Search Engine"

Search Engine	No. of Web Pages fetched	No. of pages relevant in				
		1st 10	2nd 10	3rd 10	4th 10	5th 10
Google	316,000,000	10	10	10	9	10
Yahoo! Search	1,530,000,000	10	10	10	10	10
MSN - Bing	253,000,000	10	9	10	9	9

Keyword = "Data mining"

Search Engine	No. of Web Pages fetched	No. of pages relevant in				
		1st 10	2nd 10	3rd 10	4th 10	5th 10
Google	23,600,000	10	10	9	9	9
Yahoo! Search	136,000,000	9	8	9	10	10
MSN - Bing	24,500,000	7	6	5	4	3

Keyword = "FUTO+EEE+Department" or "FUTO EEE Department"

Search Engine	No. of Web Pages fetched	No. of pages relevant				
		1st 10	2nd 10	3rd 10	4th 10	5th 10
Google	170	9	4	4	1	0
Yahoo! Search	165	5	2	0	0	0
MSN - Bing	2	2	0	0	0	0

Observations

The experiment shows that Google performed better in bringing out better quality (*high importance*) results, but poor in *crawling* the entire Web. A big advantage of Yahoo is that the crawl rate of its *spiders* is faster than Google and MSN-Bing. Yahoo still reads the keywords *meta tag*. MSN's Bing is the most susceptible to be used among the three.

General problems which search engines contend with are:

- Keeping the index fresh and complete,
- Identifying and removing malicious content and link,
- Identifying content of good quality,
- Detecting duplicate hosts and content, to avoid unnecessary crawling,
- Improving ranking to make it dependent on the person posing the query [Henzinger,2002].

The Solutions to Web Search Inefficiencies

Web mining provides solution to Web search inefficiencies. Web mining is the use of data mining technologies to automatically discover and extracting formation from the Web document and services [Eirinaki, 2003]. The automation feature of data mining tries to manage the *volume* and *personal judgment* issues faced in the Web searching technology.

In a typical search engine a query is first established, but in Web mining a query is not known. Web mining tries to find relations in the data that look like an interesting answer to the user's need and then tries to find the corresponding query for it. Web mining is interested in the **content** (text, image, records, etc), the **structure** (hyperlinks, tags, etc) and the **usage** (http logs, app server logs, etc) of the data. It works in a personalized way by creating a system which responds to user queries in a manner which is dependent on who the user is.

Mining the content of the Web discovers useful information from the content of a Web page. The type of the web content may consist of text, image, audio or video data in the web. The technologies that are normally used are Natural language processing (NLP) and Information retrieval (IR) [Witte, 2006]. The Web structure also contains vital and retrievable information about the user habit and mining it involves the uses *graph theory* to analyse the node and connection structure of a web site. The basic approach is to extracting patterns from hyperlinks in the web and the document structure. Both approaches uses the tree-like structure to analyse and describe the HTML (Hyper Text Markup Language) or XML (eXtensible Markup Language) tags within the web page [Bing, 2007], [Soumen, 2002].

Web usage mining technology is becoming more prominent than others. It analyses and discovers interesting patterns on how web user uses data on the web. The usage data captures when the user browses or makes transactions on the web site. The Web usage information is most times generated automatically by Web servers and collected in server log. Using some special tool that record user, it is possible to determine such information as the number of accesses to the server, the times or time intervals of visits as well as the domain names and the URLs of users of the Web server as shown in Table 2 [Roberto, 2002].

Table 2: A typical Web server log.

#	IP Address	Userid	Time	Method/ URL/ Protocol	Status	Size	Referrer	Agent
1	123.456.78.9	-	[25/Apr/1998:03:04:41 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.04 (Win95, I)
2	123.456.78.9	-	[25/Apr/1998:03:05:34 -0500]	"GET B.html HTTP/1.0"	200	2050	A.html	Mozilla/3.04 (Win95, I)
3	123.456.78.9	-	[25/Apr/1998:03:05:39 -0500]	"GET L.html HTTP/1.0"	200	4130	-	Mozilla/3.04 (Win95, I)
4	123.456.78.9	-	[25/Apr/1998:03:06:02 -0500]	"GET F.html HTTP/1.0"	200	5096	B.html	Mozilla/3.04 (Win95, I)
5	123.456.78.9	-	[25/Apr/1998:03:06:58 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
6	123.456.78.9	-	[25/Apr/1998:03:07:42 -0500]	"GET B.html HTTP/1.0"	200	2050	A.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)

The Web Mining Process

The Web mining process is similar to the data mining process. The only difference is that in data mining, the data is often and already collected and stored in a data warehouse, but in Web mining, data collection is a substantial task, especially for Web structure and content mining, which involves crawling a large number of target Web pages. However once the data is collected, same three-step process of *data pre-processing*, *Web data mining* and *post-processing* are applied.

The architecture of Web mining process (especially in Web usage mining) is divided into three main parts. The first part includes the domain dependent processes of transforming the Web data into suitable transaction form. This include, pre-processing (data cleaning, data integration and the log entries partitioning), transaction identification, and data integration components. The second part includes the domain independent application of generic data mining and pattern matching techniques as part of the system's data mining engine. Transaction data are formatted to conform to the data model of the appropriate data mining task such as **Association rule and Sequential patterns**, **Clustering and Classification rules**, as well as **Path analysis**.

The third part is the use of query mechanism that will allow the user (analyst) to provide more control over the discovery process - by specifying various constraints. Some of the common tools used in Web mining are OLAP/Visualization, Knowledge query mechanism and Intelligent agents. The architecture is well adopted by WEBMINER systems, detailed in [Cooley, 1997].

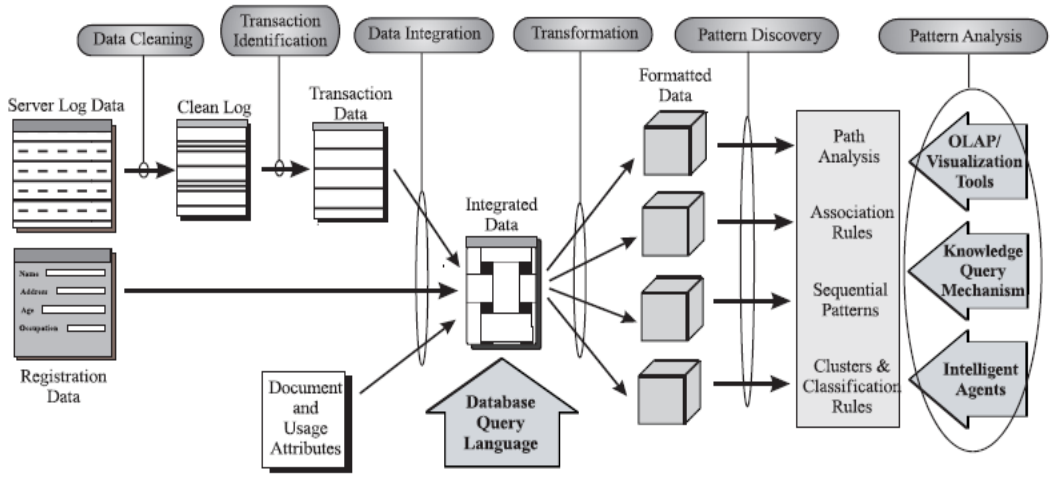

Figure 2: Web Usage Mining Architecture.

Building a Web Mining Model

The real task in mining the Web is embedded in the model developed as *rules* or *algorithms* to perform those tasks which Web search engines could not do. The algorithms that can, for instance, exploit user feedback, either from explicit user

evaluation or implicit Web logs, or predicting user need through informational, navigational, or transactional activities.

Two important data mining models that are commonly used in Web mining tasks, (especially in Web usage and content mining) are **Association rules** and **Sequential patterns**. The two models are proposed in this paper.

Association rule mining finds sets of data items that occur together frequently, while Sequential pattern mining finds sets of data items that occur together frequently in some sequences. Both rules are used in finding *regularities* in the Web data. For example, association rule mining (in Web usage mining) is used in finding *users' visit* patterns, and sequential pattern mining is used in finding *users' navigation* patterns. They are used in analyzing *clickstreams* in server logs and also in finding *language* or *linguistic* patterns from natural language texts.

More information on these can be got from **Bing Liu**'s book on "Web Data Mining" published online at http://www.cs.uic.edu/~liub/WebMiningBook.html

Association Rules

The objective of association rule is to find all co-occurrence relationships called associations among data items. Association rule are generally applied to databases of transactions where each transaction consists of a set of items. In such a framework, the problem is to discover all associations and correlations among data items where the presence of one set of items in a transaction implies (with a certain degree of confidence) the presence of other items. In Web usage mining, this problem amounts to discovering the correlations among references to various files available on the server by a given client. Each transaction captures a set of URLs address, for instance, as a client visits the server or the Web;

The problem of mining association rules can be stated as follows:

Let $I = \{i1, i2, \ldots, im\}$ be a set of **items**.

Let $T = (t1, t2, \ldots, tn)$ be a set of **transactions**(the database),

where each transaction ti is a set of items such that $t_i \subseteq I$.

Then an **association rule** is an implication of the form,

$$X \rightarrow Y$$

where $X \subset I$, $Y \subset I$, and $X \bigcap Y = \phi$. X (or Y) is a set of items, called an **itemset**.

A transaction $t_i \in T$ is said to **contain** an itemset X if X is a subset of ti (we also say that the itemset X **covers** ti).

The **support count** of X in T (denoted by $X.count$) is the number of transactions in T that contain X.

The strength of a rule is measured by its **support** and **confidence**.

The **support** of a rule, $X \rightarrow Y$, is the percentage of transactions in T that contains

$X \cup Y$, and can be seen as an estimate of the probability, Pr($X \cup Y$). The rule support thus determines how frequent the rule is applicable in the transaction set T.

Let n be the number of transactions in T. Then the support of the rule $X \rightarrow Y$ is computed as:

$$Support = \frac{(X \cup Y).count}{n}$$

The **confidence** of a rule, $X \rightarrow Y$ is the percentage of transactions in T that contain X also contain Y. It can be seen as an estimate of the conditional probability, Pr(Y | X). It is computed as:

$$Confidence = \frac{(X \cup Y).count}{X.count}$$

Confidence thus determines the **predictability** of the rule.

Most common approaches of implementing association rule discovery are based on the ***Apriori Algorithm***. Apriori algorithm finds group of items (*pageviews* appearing in the preprocessed log) occurring frequently together in many transactions (i.e., satisfying a client specified minimum support threshold). A typical Apriori Algorithm is shown in figure 3.

To ensure efficient itemset generation, the Apriori algorithm assumes that the items in I are sorted in **lexicographic order** (a total order) of {w[1], w[2], …, w[k]} to represent a k-itemset w consisting of items w[1], w[2],…, w[k], where w[1] < w[2] < … < w[k] according to the total order.

The procedure

Apriori algorithm listed in figure 3 generates all frequent itemsets F by making multiple passes over the data. First, it counts the supports of individual items (line 1) and determines whether each of them is frequent (line 2), then in each subsequent pass k, perform three steps:

1. The seed set of itemsets Fk-1 is found to be frequent in the (k-1)-th pass. It uses this seed set to generate **candidate itemsets** Ck (line 4). This is done using the candidate-gen() function.
2. From the transaction database the actual support of each candidate itemset c in Ck is counted (lines 5–10).
3. At the end of the pass, the candidate itemsets that are actually frequent will be determined (line 11).

The final output of the algorithm is the set F of all frequent itemsets (line 13).

The **candidate-gen() function** consists of two steps, - join step and pruning step.

Join step (lines 2–6) joins two frequent (k-1)itemsets to produce a candidate c (line 6). The two frequent itemsets f1 and f2 have exactly the same items except the

last one (lines 3–5), *c* is added to the set of candidates *Ck* (line 7).

Pruning step (lines 8–11 determines whether all the *k*-1 subsets of *c* are in *Fk*□1. If anyone of them is not in *Fk*-1, *c* cannot be frequent and is thus deleted from *Ck* [Agrawal, 1994].

```
Algorithm Apriori(T)
1   C1 ← init-pass(T);                          // the first pass over T
2   F1 ← {f | f ∈ C1, f.count/n ≥ minsup};      // n is the no. of transactions in T
3   for (k = 2; Fk–1 ≠ ∅; k++) do               // subsequent passes over T
4     Ck ← candidate-gen(Fk–1);
5     for each transaction t ∈ T do             // scan the data once
6       for each candidate c ∈ Ck do
7         if c is contained in t then
8           c.count++;
9       endfor
10    endfor
11    Fk ← {c ∈ Ck | c.count/n ≥ minsup}
12  endfor
13  return F ← ⋃k Fk;

Function candidate-gen(Fk–1)
1   Ck ← ∅;                                     // initialize the set of candidates
2   forall f1, f2 ∈ Fk–1                        // find all pairs of frequent itemsets
3     with f1 = {i1, … , ik–2, ik–1}            // that differ only in the last item
4     and f2 = {i1, … , ik–2, i'k–1}
5     and ik–1 < i'k–1 do                       // according to the lexicographic order
6       c ← {i1, …, ik–1, i'k–1};               // join the two itemsets f1 and f2
7       Ck ← Ck ∪ {c};                          // add the new itemset c to the candidates
8       for each (k–1)-subset s of c do
9         if (s ∉ Fk–1) then
10          delete c from Ck;                   // delete c from the candidates
11      endfor
12  endfor
13  return Ck;                                  // return the generated candidates
```

Figure 3: The Apriori algorithm for generating frequent itemsets.

Sequential Patterns

Discovering sequential patterns is to find inter-transaction patterns such that the presence of a set of items is followed by another item in the *time-stamp* ordered transaction set.

In Web server transaction logs (Table 2), a visit by a client is recorded over a period of time. The *time-stamp* associated with a transaction in this case will be a time

interval which is determined and attached to the transaction during the data cleaning or transaction identification processes. Sequential patterns in Web server access logs allows organizations to predict user visit patterns and helps in targeting advertising aimed at groups of users based on these patterns.

In classic sequential pattern mining, no rules are generated. It is, however, possible to define and generate many types of rules such as **sequential rules**, **label sequential rules** and **class sequential rules.**

A **sequential rule** is an implication of the form, $X \rightarrow Y$, where Y is a sequence and X is a **proper subsequence** of Y. That is X is a subsequence of Y and the length Y is greater than the length of X. The **support** of a sequential rule, $X \rightarrow Y$, in a sequence database S is the fraction of sequences in S that contain Y. The **confidence** of a sequential rule, $X \rightarrow Y$, in S is the proportion of sequences in S that contain X also contain Y.

One of the common algorithms for mining sequential patterns and also used in this paper is called **GSP algorithm** [Srikant, 1996].

In **GSP algorithm** *Fk* is used to store the set of all frequent *k*-sequences, and *Ck* to store the set of all candidate *k*-sequences.

```
Algorithm GSP(S)
1   C1 ← init-pass(S);                          // the first pass over S
2   F1 ← {⟨{f}⟩| f ∈ C1, f.count/n ≥ minsup};   // n is the number of sequences in S
3   for (k = 2; Fk−1 ≠ ∅; k++) do               // subsequent passes over S
4     Ck ← candidate-gen-SPM(Fk−1);
5     for each data sequence s ∈ S do           // scan the data once
6       for each candidate c ∈ Ck do
7         if c is contained in s then
8           c.count++;                          // increment the support count
9       endfor
10    endfor
11    Fk ← {c ∈ Ck | c.count/n ≥ minsup}
12  endfor
13  return ⋃k Fk;
```

Function candidate-gen(*Fk*-1)

1. **Join step. J**oin Fk-1 with Fk-1. A sequence s1 joins with s2 , if the subsequence obtained by dropping the first item of s1 is the same as the subsequence obtained by dropping the last item of s2. The candidate sequence generated by joining s1 with s2 is the sequence s1 extendedwith the last item in s2. There are two cases:
    - the added item forms a separate element if it was a separate element in s2,and is appended at the end of s1 in the merged sequence, and
    - the added item is part of the last element of s1 in the merged sequence otherwise.

When joining F1 with F1, we need to add the item in s2 both as part of an itemset and as a separate element.

2. **Prune step.** A candidate sequence is pruned if any one of its (k-1)-subsequences is infrequent (without minimum support).

Figure 4: The GSP Algorithm for generating sequential patterns

Application of the Web Mining Model

Abstract as the above models looks, they can be applied to real life data to achieve results. One example each from the association rule and sequence pattern will be used to demonstrate the application of the models

Association Rules

Example

Assuming URL1 (https:www.xyz.com/product/product1) and URL2 (https:www.xyz.com /product/product2) represents clients site visitation, one can use Association rule discovery techniques to analyze the correlations between the clients site visit to product Web pages such as:

40% of clients who accessed the Web page with URL1, also accessed URL2; or that 10% of clients accessed the Web page with URL1,URL2

The association rule for this site visit is represented as:

URL1→ URL2 [support = 10%, confidence = 40%].

Since such transaction databases usually contain extremely large amounts of data, current association rule discovery techniques try to prune the search space according to **support** for items under consideration.

Sequential Patterns

Example

From a server log (Table 2) it could also be found that:

30% of clients who visited URL1 (as in Association example), had done a search in Yahoo, within the past week on keyword w; or that,

60% of clients who placed an online order in URL2, also placed an online order in URL3 within 15 days.

Given the sequence database (Table 3), the sequential rules that could be generated is as follows:

<{1}{7}> →<{1}{3}{7, 8}> [sup = 2/5, conf = 2/3]

Data sequences 1, 2 and 3 in table 3 contain <{1}{7}>, and data sequences 1 and 2 contain <{1}{3}{7, 8}>

Table 3: An example of sequence database.

Sequence ID	Data Sequence
1	<{1}{3}{5}{7, 8, 9}>
2	<{1}{3}{6}{7, 8}>
3	<{1, 6}{7}>
4	<{1}{3}{5, 6}>
5	<{1}{3}{4}>

Analyzing the Web mined data

As in data mining process, the output of Web mining algorithms is often not in a form suitable for direct human consumption as can be seen in the two examples given above. In analyzing the Web mined data, other mechanisms or tools are applied, to help an analyst to better assimilate the knowledge. The tools involve statistical methods, visualization, and human factors.

Clearly, these tools need to have specific knowledge about the particular problem domain to do any more than filtering based on statistical attributes of the discovered rules or patterns. In Web mining, *intelligent agents* are developed based on discovered access patterns, the topology of the Web locality, and certain heuristics derived from user behavior models. Dr. Mohammadia in [Mobasher, 2001] discusses in details the foundation as well as the practical side of intelligent agents and their theory and applications for web data mining and information retrieval.

Other uses on Web Mining

Aside just correcting the inefficiencies of Web searching discussed in this paper, It is important to also state that Web mining can be used for several purposes in addition to finding new information or knowledge. It can be used for adaptive Web design, Web site reorganization, Web site personalization, and several performance improvements. Web site personalization is becoming popular in Web application nowadays and thus are discussed below.

Existing Achievements: Web Mining tools

Researches are going on in developing Web mining tools. Web mining is a huge, interdisciplinary and very dynamic scientific area, converging from several research communities such as database, information retrieval, and artificial intelligence especially from machine learning and natural language processing. Few achievements have been recorded, though not yet popular.

Common Web mining tools in the market are **WUM,** a sequence miner, whose primary purpose is to analyze the navigational behaviour of users visiting a Web site; **Modeler Web Mining, very** good in performing ad hoc predictive Web analysis; **Nihuo Web Log Analyzer**, excellent in logging files automatically created and

maintained by a web serve**; VISITaTOR** and **Angoss,** which uses mining, exploration, and predictive modeling of business data as critical importance in the development of strategies to accelerate revenues, reduce costs, and manage risk.

Recommendations for future Research Work

Amidst goodies brought by Web mining, there are still few challenges confronting the new field. Efforts are on ongoing by researcher to provide solutions to some of the challenges.

One big challenge in Web mining is that there is yet to be a general purpose and integrated Web mining systems as it is in Web searching. Web mining tools are still designed and used for special purpose. The results of the most of Web miming applications are still interpretable and useful to secluded users and analysts. The tri-nature (context, structure and usage mining) of Web mining is one of causes of its divergence application. Effort should be made towards having an integrated general purpose Web mining tool as in Web searching (e.g. Google, AltaVista, Yahoo!Search or MSN's Bing)

Secondly more research efforts are to be put in developing new techniques to handle noisy, uncertain, vague, and incomplete information [Crestani, 2003]. Though researches are ongoing on this areas, more efforts need to be put to create adaptive sites that would appear different to different users [Joshi, 1998], and developing some techniques such as fuzzy sets [Miyamoto, 1990] and logic for information fusion, text extraction, query language models, and document clustering. Other areas are to created are neural networks for document and term classification and clustering, and multimedia retrieval; genetic algorithms for document classification, image retrieval, relevance feedback, and query learning; probabilistic techniques for Ranking; rough sets and multivalued logics for document clustering; and bayesian networks for retrieval models, ranking, thesaurus construction, and relevance feedback.

And lastly, more efforts are suggested in designing Web system that exploits user feedback, either from explicit user evaluation or implicitly from Web logs. Implicit information given by the authors of Web pages in the form of several conventions used in HTML design is invaluable. With the technique of personalization, Web designers are focusing on intelligent Web server. The Web users prefer Web server, which is capable of learning their information needs and preferences. With the technique of learning user navigation patterns, the information providers would be glad to view the improvement of the effectiveness on their Web sites, which results in adapting the Web site design or by biasing the user's behaviour towards satisfying the goals of the site.

Conclusion

As the popularity of the World Wide Web continues to increase, there is a growing need to develop tools and techniques that will help improve its overall usefulness. Since one of the principal goals of the Web is to act as a *world-wide distributed*

information resource, Web searching techniques has made efforts, but more efforts should be made to develop techniques that will make it more useful in this regard.

Web mining is a very active research area. A paper like this can only scratch on the surface. I tried to include references to the most important works in the areas and hope to have provided a good starting point for explorations into this rapidly expanding and exciting research area.

References

[1] [Agrawal, 1994] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. In Proc.of the 20th Intl. Conf. on Very Large Data Bases (VLDB'94), pp. 487–499, 1994.

[2] [Bing, 2007] Bing Liu, "Web Data Mining: Exploring Hyperlinks, Contents and Usage Data", Springer, 2007

[3] [chau, 2009] May Y. chau; Web mining Technology and academic Librarianship. First Monday 1995-2009. http://www.firstmonday.org/issues/issue4_6/chau/index.html

[4] [Cooley, 2000] R. Cooley. Web Usage Mining: Discovery and Application of Interesting Patterns from Web data. PhD thesis, Dept. of Computer Science, University of Minnesota, May 2000

[5] [Crestani, 2003] F. Crestani, G. Pasi (Eds.), Handling vagueness, subjectivity and imprecision in information access, Information Processing and Management, 39(2) (2003).

[6] [Diego, 2007] Diego Puppin, A Search Engine Architecture Based on Collection Selection, Ph.D. Thesis Universit`a degli Studi di Pisa, 2007 http://hpc.isti.cnr.it/~diego/phd/tesi.pdf

[7] [Eirinaki, 2003] Eirinaki, M., Vazirgiannis, M. (2003) "Web Mining for Web Personalization", ACM Transactions on Internet Technology, Vol.3, No.1, February 2003

[8] [Henzinger, 2002] M. Henzinger, R. Motwani, C. Silverstein, Challenges in web search engines, SIGIR Forum 36 (2) (2002).

[9] [Hirai, 2000] Jun Hirai, Sriram Raghavan, Hector Garcia-Molina, and Andreas Paepcke. Webbase: A repository of Web pages. In Proceedings of the Nineth International World-Wide Web Conference, May 2000. [http://ilpubs.stanford.edu:8090/473/1/2000-51.pdf]

[10] [Hotho, 2008] Andreas Hotho, Gerd Stumme: Mining the World Wide Web - Methods, Applications, and Perspectives, http://kobra.bibliothek.uni-kassel.de/bitstream/urn:nbn:de:hebis:34-2008021320337/3/HothoStummeMiningWWW.pdf

[11] [Joshi, 1998] A Joshi, R Krishnapuram - Proc. Workshop in Data Mining and knowledge Discovery (1998) - Robust Fuzzy Clustering Methods to Support Web Mining - citeseer.comp.nus.edu.sg

[12] [Kdnuggets, 2009] KDnuggets - Software : Web Mining and Web Usage Mining http://www.kdnuggets.com/software/Web-mining.html

[13] [Kleinberg, 1999] Jon Kleinberg. Authoritative sources in a hyperlinked environment. Journal of the ACM, 46(5):604-632, November 1999. http://www.cs.cornell.edu/home/kleinber/auth.pdf

[14] [Miyamoto, 1990] S. Miyamoto, Fuzzy Sets in Information Retrieval and Cluster Analysis, Kluwer Academic Publishers, 1990.

[15] [Mobasher, 2000] Mobasher, B., Cooley, R. and Srivastava, J. (2000) "Automatic Personalization based on Web usage Mining" Communications of the ACM, Vol. 43, No.8, pp. 142-151

[16] [Mobasher, 2001] Mobasher, B., Dai, H., Kuo, T. and Nakagawa, M. (2001) "Effective Personalization Based on Association Rule Discover from Web Usage Data" In Proceedings of WIDM 2001, Atlanta, GA, USA, pp. 9-15

[17] [Masoud, 2004] Masoud, M, "Intelligent Agents for Data Mining and Information Retrieval"; Idea Group Publishing; ISBN 1591401941; Feb 2004.

[18] [Pierrakos, 2003] Pierrakos, D., Paliouras, G., Papatheodorou, C., Spyropoulos C. D. (2003) "Web usage mining as a tool for personalization: a survey", User modelling and user adapted interaction journal, Vol.13, Issue 4, pp. 311-372

[19] [Roberto, 2002] José Roberto de Freitas Boullosa, Geraldo Xexéo, An Architecture for Web Usage Mining http://citeseer.ist.psu.edu/cache/papers/cs2/205/http:zSzzSzwww.boullosa.orgzSzartigoszSzArch_Web_Mining_2002.pdf/an-architecture-for-Web.pdf

[20] [Sergey, 1998] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. In Proceedings of the Seventh International World-Wide Web Conference, 1998.

[21] [Srikant, 1996] Srikant R. and Agrawal, R:. Mining Sequential Patterns: Generalizations and Performance Improvements. In Proc. of the 5th Intl. Conf. Extending Database Technology (EDBT'96), pp. 3–17, 1996.

[22] [Soumen, 2002] Soumen Chakrabarti, "Mining the Web: Analysis of Hypertext and Semi Structured Data", Morgan Kaufmann, 2002

[23] [Steven, 2004] Steven Garcia: An Introduction To Search Engine Architecture. Proceedings of the Second Australian Undergraduate Students' Computing Conference, 2004 .http://www.cs.berkeley.edu/~benr/publications/auscc04/tutorials/garcia-auscc04.pdf

[24] [Vise, 2005] David Vise and Mark Malseed (2005). The Google Story, 37. ISBN ISBN 0-553-80457-X. http://www.thegooglestory.com/

[25] [W3C, 1999] Web Characterization Terminology & Definitions Sheet.

[26] http://www.w3.org/1999/05/WCA-terms/ . W3C Working Draft 24-May-1999.

[27] [Witte, 2006] Ren´e Witte (2006): Introduction to Text Mining—Tutorial at EDBT'06— http://www.edbt2006.de/edbt-share/IntroductionToTextMining.pdf