Hierarchical Web Structure Mining

Wookey LEE
†Faculty of Computer Science, Sungkyul University, 147-2 Anyang, Kyongkido, 430-742 Korea

Abstract The World Wide Web is nearing omnipresence. The explosively growing number of Web contents including digitalized manuals, emails, pictures, multimedia, and Web services require a distinct and elaborate structural framework that can provide a navigational surrogate for clients as well as for servers. In this paper, we exploited Web mining methods based on link oriented similarity measures. We introduce the Web structure mining concept and the corresponding issues. And the Web spamming, the artificial manipulation working on the Web structures is discussed from which a solution scheme based on singular value decomposition method is derived to proof the Web structure spamming.

Keyword Web mining, Similarity Measure, SVD, Web spamming, Web structure mining

1. Introduction

The World Wide Web is a collection of Web sites and its Web contents. The Web evolves continuously and changes dynamically since new Web sites are born and the old ones disappear simultaneously, and contents of those Web sites are updated at any times. While the Web contains vast amount of information and provides an access to it at any places and any times, that is a prize beyond our reach without efficient searching tools for the Web. Efficient searching for Web contents becomes more important than ever before as the Web evolves and users increase explosively. Portal sites with search engines are popular and commonly used tools for searching Web contents at this time, although some promising efforts are continued for more efficient and effective use of the Web such as semantic Web [1,3].

Most of portal sites have their search engines, which are used to find relevant Web contents for users' search queries. For efficient responses to users' queries, many portal sites have their index databases in which a collection of pointers to positions in Web pages of occurrences of indexed words. Search engines find relevant Web contents by seeking indexed words related to the query strings given by users in the index databases. Portal sites update their index databases using special-purpose Web clients, called as spiders, or search crawlers. Spiders send HTTP requests to a set of target Web sites to fetch Web pages of those sites. Since a Web site has its homepage and Web contents linked each other, search crawlers fetch the homepage of the Web site first and then obtain other Web contents by traversing referenced links within Web pages. Commonly used techniques to traversing referenced links are breadth first search and depth first search. Regardless of which techniques are used to traverse Web contents of a Web site, it is necessary to avoid traversing a Web content that has already been visited and fetched since cycles or loops among links within Web pages may result in ineffective and/or inefficient collection of Web contents with search crawlers [14,19].

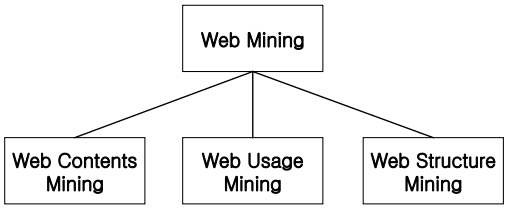

Figure 1. Classification of Web Mining

In searching the WWW, there are two fundamental problems; the first problem is that Web search engines only allow for low precision criteria, which may generate many irrelevant search results. And the second pitfall is the low amount of recall, which is due to the inability to index all the information available on the Web. These two problems are similar to the problem of traditional information retrieval techniques. The Web contents mining approach has basically been researched upon traditional information retrieval techniques. The contents based Web mining is very weak for contents manipulations. One of the intentional, sometimes vicious manipulations of web databases is a deliberately manipulated web page. The web manipulation refers to web page contents and hyperlinks that are created with

the intention of misleading search engines. Traditional search engines based on the information retrieval techniques are well known to weaknesses for manipulations on body, title, meta tag, anchor text, and URL so that the added keywords can be invisible to persons through ingenious use of color representations, but can mislead the search engines [7].

Web Usage Mining has mainly focused on the analysis of usage patterns recorded in the Web usage log of Web servers. Web Usage Mining is the application of data mining techniques to large Web data repositories in order to produce results that can be used in the Web design tasks [1]. Commonly used Web Usage Mining algorithms are association rule generation, sequential pattern generation, and clustering. Association rule generation techniques discover the correlations between items found in a database of transactions. The problem of discovering sequential patterns is that of finding inter transaction patterns by analyzing Web Usage log to determine temporal relationships among data items. In the context of Web Usage Mining a transaction is a group of Web page accesses, which is not easy problem to identify an item being a single page access.

The Web structure mining technique is widely put to use and is expected to minimize these two problems [15]. However, the results of many Web search engines using Web mining techniques are equally hard to assay since search engines usually return huge lists of URLs, most of which can be judged almost irrelevant to the query [10]. In identifying the reason for this problem we can look to the inaccuracy of the Web mining algorithm on one hand, and Web pages that are deliberately composed to spam the search engine, on the other. We can divide Web mining into three areas of interest based on which part of the Web one wishes to mine; they are, Web content mining, Web structure mining, and Web usage mining. Like traditional Information retrieval techniques, Web content mining alerts the discovery of useful information on the basis of match percentages gathered by scanning Web contents, related data, and uploaded documents [15]. Yahoo, DMOZ, and many other Web search engines use this type of algorithm. However, there are two main reasons why traditional information retrieval (IR) techniques may not be effective enough in ranking query results.

In this study, we focus on finding hop constrained spanning tree for a Web site with the objective of connecting Web contents closely that are relatively important to each other. We use PageRank to measure relative importance of Web contents to other Web contents linked to the Web content. Note that PageRank is a link weight that can be interpreted as the degree of relative importance of a Web content to other Web contents linked directly to the Web content in a Web site. Lee and Geller [18] use spanning tree for structuring the Web for more effective use of the Web.

2. Graph Theoretic Approach

2.1 Web Objects

In this paper, we view the World-Wide Web as a hierarchy of Web objects with a schema represented in figure 2. The WWW is viewed as a set of Web sites, and a Web site as a set of Web pages with arcs and content elements. We focused on the Web site that is modeled as a directed graph with Web nodes and Web arcs, where the Web nodes correspond to HTML files with page contents, and the Web arcs correspond to hyperlinks interconnecting the Web pages.

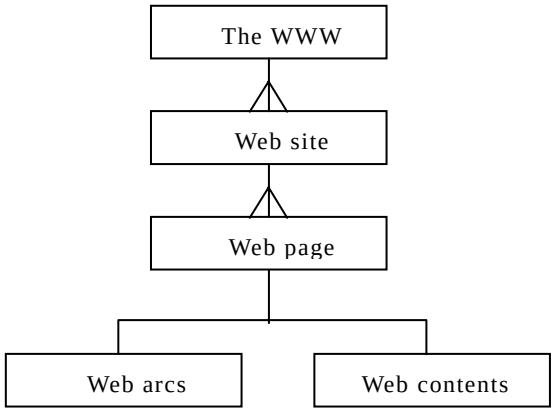

Figure 2. The schema of the World-Wide Web

Notice that the schema of the World-Wide Web, that is based on an Entity Relationship Diagram description, but some optional details are simplified. A box represents an entity, a line a relationship. A triple line represents cardinality (one-to-many). The plain line at the bottom represents a Generalization-Specialization.

The WWW can be viewed formally as digraph with Web nodes and arcs, where the Web nodes correspond to HTML files having page contents and the arcs correspond to hypertext links interconnected with the Web pages. The Web-as-a-graph approach can be a starting point to generate a structure of the WWW that can be used for

Web site designers, search engines, Web crawling crawlers, and Web marketers and analysts [4, 14].

Formally the Web directed graph $G = (N, A)$ can be represented with an arc function $x_{ij} : N^k \rightarrow \{0, 1\}, \forall i, j \in N$ consists of a finite Web node set N, a finite Web arc set A of ordered pairs of Web nodes, and the Web arc elements (i, j) respectively, where $i, j \in N = \{0, 1, 2, 3, \ldots, n\text{-}1\}$, and $n = |N|$ the cardinality of Web pages. There is a mapping for the nodes corresponding to Web pages and the arcs to Uniform Resource Identifiers [4, 5].

Since a Web site consists of a homepage (that can be accessed by its domain name) and many other Web contents linked each other (that can be located with its corresponding URLs or by clicking links within its Web pages), Web contents (including the homepage) of a Web site can be represented as a tree consisting of a set of nodes and associated arcs [5,18]. Here, a Web content corresponds to a node, while links within the Web site among Web contents become directed arcs of the trees. If we transform Web contents of a Web site into a corresponding tree, we can find a set of paths through which any Web page of the Web site can be accessed from the homepage and all other pages within the Web site. This set of paths can be obtained from finding a spanning tree for the Web site. Since a spanning tree does not have any cycles or loops among nodes, search crawlers can avoid revisiting Web contents that has already been visited to fetch the contents if the search crawlers send HTTP request messages according to paths obtained by the spanning tree.

Occasionally, search crawlers do not fetch all of Web contents of a Web site according to policies of portal sites that use the search crawlers. For example, Yahoo fetches only some base pages of Web sites not fetching all individual Web pages, while AltaVista and Google gather most of Web pages within Web sites. Since gathering Web contents with search crawlers dispersed among huge number of Web sites is time-consuming tasks, portal sites may be needed to set their search crawlers to traverse a Web site down until to only preconfigured number of links from the homepage to reduce the time for gathering Web contents [16]. In this study, we call this preconfigured number of links as hop limit. Hop limit of a Web content can be interpreted as the number of clicks needed to arrive at the Web content from the homepage of a Web site. Administrators (or owners) of portal sites and Web sites can set hop limits. Portal sites can use hop limit to let search crawlers know the scope of search space for a Web site. On the other hand, a Web site can use the hop limit to make a hop constrained spanning tree that gives paths through which any Web content within the site can be accessed within limited clicks from the homepage of the site without falling into cycles or loops [19].

2.2 Web structure mining

Web structure mining tries to discover the model underlying the link structures of the Web. The model is based on the topology of the hyperlink with or without the link description. This model can be used to categorize the Web pages and is useful to generate information such as similarity and relationships between Web sites [2]. And the link structure of the Web contains important implied information, and can help in filtering or ranking Web pages. In particular, a link from page A to page B can be considered a recommendation of page B by the author of A. Some new algorithms have been proposed that exploit this link structure—not only for keyword searching, but other tasks like automatically building a Yahoo-like hierarchy or identifying communities on the Web. The qualitative performance of these algorithms is generally better than the IR algorithms since they make use of more information than just the contents of the pages. While it is indeed possible to influence the link structure of the Web locally, it is quite hard to do so at a global level. So link analysis algorithms that work at a global level possess relatively robust defenses against spamming [7,17].

There are two major link-based search algorithms, HITS (Hypertext Induced Topic Search) and PageRank. The basic idea of the HITS algorithm is to identify a small sub-graph of the Web and apply link analysis on this sub-graph to locate the authorities and hubs for the given query. The sub-graph that is chosen depends on the user query. The selections of a small sub-graph (typically a few thousand pages), not only focus the link analysis on the most relevant part of the Web, but also reduce the amount of work for the next phase. The main weaknesses of HITS are known to non-uniqueness and nil-weighting [8]. THESUS suggested a domain based PageRank algorithm, but its limitation depends on the usefulness of the ontology and the thesaurus that the

system tries to include semantics among Web documents.

Google, which among search engines is ranked in the first place, uses the PageRank algorithm. The basic idea of PageRank is that, if source page u has a link to target page v, then the author of source page u is implicitly conferring some importance to page v. Let N_u be the out-degree of page u and let *Rank*(*p*) represents the importance of page p. Then, the link(u, v) confers a certain number of units of rank to v. This simple idea leads to the following iterative fix-point computation that yields the rank vector over all of the pages on the Web. If *n* is the number of pages, assign all pages the initial value 1/*n*. Let B_v represent the set of pages pointing to *v*. For each iteration, links between Web pages propagate the ranks as follows

$$\forall v, Rank^{(i+1)}(v) = \sum_{w \in B_v} Rank^{(i)}(u) / N_u \qquad (1)$$

We continue the iterations until the rank is stabilized to within some defined threshold. The final vector contains the PageRank vector over the Web. This vector is computed only once after each crawl of the Web; the values can then be used to influence the ranking of search results [11,13]. Guaranteeing the rank vector to converge, PageRank algorithm uses the following equation with a damping factor (d):

$$Rank^{(i+1)}(u) = (1-d)E + d(\sum_{i \in B_v} Rank^{(i)}(v) / N_v)$$
$$where, E = \left[\frac{1}{n}\right]_{n \times 1} \qquad (2)$$

In Google, we usually set the value of the damping factor to 0.85 [10]. And we can see that the PageRank vector converges either slowly or quickly in relation to the magnitude of the damping factor.

2.3 Web structure mining Issues

The Web-as-a-graph, however, has weaknesses such as Unreachable paths, Circuits, Repetitive cycles. The unreachable path is that a Web page sometimes can not be accessed from the usual path or a hub mode. The circuit as a cycle is that a client visits the same page again and again periodically. In order to generate a Web structure, the circuits and repetitive cycles should be detected and removed, without which a Web client may be lost in Cyber space through the complex cycles [1,19] or may inevitably gather information with swallow depths from the root node [9].

Why are we interested in the hierarchical structure? One reason is that a Web site consists of a home page (e.g., default.html) that can be the root node from which a hierarchical structure can correspondingly be derived. The other reason is that the hierarchical structure can conceive very simple data structure so that the crawling performance to search the Web on the fly can be highly enhanced.

The typical hierarchical examples are breadth first search and depth first search with which Web catalogues, site maps, and usage mining can be pertained [9,19]. The breadth first approach, including *backlink* count method [9,10,14], has some advantages so that an 'important' Web page can be accessed within relatively fewer steps from its root node. It can statistically minimize the total number of depths. In the Web environment, however, the tree by breadth first approach may result extremely flat so that almost all the pages stick to the root node. On the other hand, the depth first approach is popularly adopted for practitioners to exploit a structure with a stack data format [6, 18]. In structuring the Web, the approach is not appropriate because it may result in a long series of Web pages. It means that the series of pages entail mouse clicks, so that as much time consumption to access each page is required. The worst thing on the derived structures by these two approaches is that no measures or values, even no semantics can be endowed.

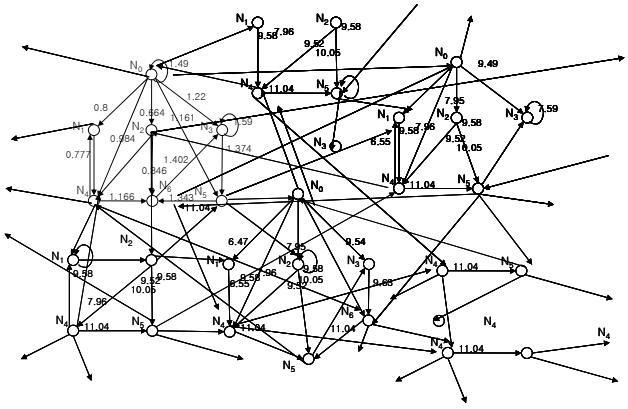

Figure 3. The WWW with nodes and arcs

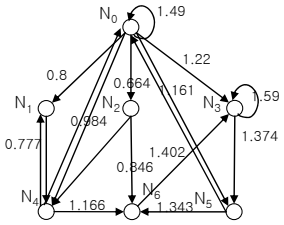

Figure 4. Problem domain

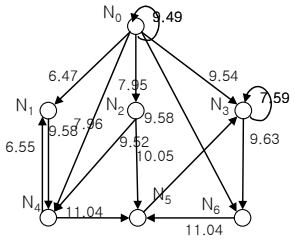

Figure 5. Notifying the root

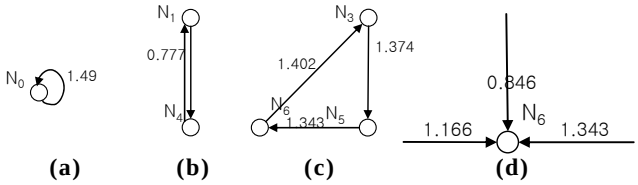

Figure 6. Degenerated Circuits as (a) self-circuit, (b) two nodes, (c) more than three nodes, and (d) tree constraints.

Even with the Link Measure, the semantic structuring algorithm [18] may generate wrong solutions such that each node has only one parent, but it is not a hierarchical structure anymore. See Figure 6 (a) through (c), in that the degenerate case, the more this path may be followed, the more the weight total will increase, which can be called “white hole.” If the weights are negative, it can be called a “black hole.” It is a cycle, and there are so many cycles in the Web environment. Therefore a cycle proof algorithm should be required to generate a structure.

On the other hand, there is another problem called “repetitive cycles” in the Web graphs. The repetitive cycle is that the identical cycles derived from the same cycles to have the order of the nodes appearing different permutations. For example, the repetitive cycles appear as Figure 5: $N_1 \rightarrow N_2 \rightarrow N_3$, $N_2 \rightarrow N_3 \rightarrow N_1$, $N_3 \rightarrow N_1 \rightarrow N_2$, or generally (b) $N_1 \rightarrow N_2 \rightarrow \ldots N_n$, $N_2 \rightarrow N_3 \rightarrow \ldots N_n \rightarrow N_1$, etc. The repetitive cycles can make search performance drastically low, because the system has to remember all visited nodes and to analyze whether the node sequences are identical with permutations. We solved the problem with a polynomial time algorithm [17].

3. Evaluation for Web Structure Mining

According to the PageRank algorithm, the example Web structures given figure 2 through 4 can be calculated as following table 1.

Table 1. *PageRank* values for Figure 5

N0	N1	N2	N3	N4	N5	N6
1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.9797	0.7247	0.4414	0.4414	1.7164	1.8297	0.8664
1.2267	1.0250	0.5387	0.5387	1.3424	1.6019	0.7263
1.0335	0.9334	0.5531	0.5531	1.6533	1.4914	0.7820
1.1318	0.9752	0.5068	0.5068	1.5352	1.6019	0.7418
1.0947	0.9719	0.5369	0.5369	1.5812	1.5255	0.7523
1.1029	0.9693	0.5213	0.5213	1.5756	1.5593	0.7495

The system starts at a URL that may be given by the user or via a search engine such as Yahoo, Lycos, Google, *etc*. Once a Web site (starting at index.html) has been transferred, our steps are applied. The system calculates the necessary weights from the user query and the page vectors, and generates a hierarchical abstraction of the Web site. The resulting hyperlink structure has the added advantage of human-understandable labels (in the form of the page names) and a uniform granularity of detail, both of which are lacking in clustering steps.

The problem of finding a tree structure of a Web site from a directed graph is *n log n*, for there must exist a Web page having the highest weight within a Web site [19]. If the problem domain were enlarged to an Intranet or the whole Web, then the time complexities would be exponential or NP-hard respectively [9, 19].

When specifying a manipulation page in terms of context based PageRank algorithm, the criteria to determine which pages that the rank value indicates can be decided by the SVD (Singular Value Decomposition) method [13,17]. The SVD decomposes the transition matrix as *U*, *V*, and *S* matrix as following equation (3). This method has an advantage that can analyze the matrix within a predetermined error range with giving arbitrary values, and a disadvantage that it is not applicable to nonsingular matrix even though it is unrealistically rare case.

$$M = USV \qquad (3)$$

Where U: m*m orthogonal matrix with left singular vectors of M, V: n*n orthogonal matrix with left singular vectors of M, and S: m*m diagonal matrix with positive singular value of *M*, for $\sigma_i = \sqrt{eigen\ value\ of\ MM^T}$.

We can derive the rank value from the Frobenius norm as following equation (4) that analyze the matrix *M* and the approximated matrix M_k that gives lower ranking selecting *k* maximum values and replacing the other

values 0's [17].

We normalized weight values and made stochastic transition matrices *U*, *S*, and *V* of the SDV assessment measure described in section 3. With this transition matrix, we can calculate each page's rank value. Figure 7 shows the change of the method value in the each iteration by the SDV. We can find that the weight value converges on certain value of the assessment measure and can get eigenvalues in matrix *S* as follows.

U=

−0.836	−0.026	0.265	−0.29	0.354	0.1452	−0.01
−0.114	−0.005	0.538	0.446	−0.31	0.0318	0.635
−0.094	−0.004	0.442	0.365	−0.26	−0.04	−0.77
−0.5	−0.182	−0.65	0.38	−0.39	−0.027	0.003
−0.115	0.7287	−0.01	−0.39	−0.55	0.0675	3E−04
−0.119	0.0963	0.045	−0.04	0.061	−0.983	0.05
−0.028	0.6527	−0.15	0.538	0.506	0.0678	−0.01

S=

2.59	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.63	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.18	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.72	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.35	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.14	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00

V=

−0.062	0.9126	−0.16	0.334	0.161	0.0316	0.007
−0.299	0.3452	0.175	−0.75	−0.39	0.2061	0.065
−0.219	−0.002	0.153	−0.27	0.686	−0.325	0.527
−0.546	−0.096	−0.14	−0.1	0.407	0.1166	−0.7
−0.38	−0.011	0.818	0.409	−0.14	0.0056	0.002
−0.016	0.11	0.003	−0.09	−0.25	−0.913	−0.29
−0.645	−0.163	−0.48	0.252	−0.33	−0.067	0.386

Figure 7. SVD Results

The result and the error terms deleting the last singular value from matrix *S* is 0.002 that is about relatively 0.09% of total errors [17]. And if we delete a page, then the singular value is 0.137 that is about relatively 2.1% of total errors. With this we can see that the node is included discarding criteria, which is stochastically significant level so that the information loss by discriminating the node 5 is about 5.6%. This represents the same result with the method. The assessment by SVD, however, represents an exceptional result that our method detects the normal node 2 is also included. Thus we can say that it shows the transition matrix converges low weight result to the unimportant pages, but it is not always a manipulation page.

With this we can see that the node is included discarding criteria so that the information loss by discriminating the node can be is done. The assessment by SVD, however, represents an exceptional result that our method detects the normal node is also included. Thus we can say that it shows the transition matrix converges low weight result to the unimportant pages, but it is not always a manipulation page.

In the example, web page which is suspected as a manipulation page has the smallest rank value but in this complex example, we can not make the page have the smallest value with which we can reduce the rank value significantly.

5 Conclusions and Future Works

The Web mining is classified 3 cases among which the Web structure mining is the most promising so that the Web spamming can be much harder than other approaches. At first, we introduced the structure mining concept and the corresponding issues. After that we discussed on the Web spamming, the artificial manipulation working on the Web structures. Actually, the Web manipulation refers to hyperlinked pages on the web that are created with the intention of misleading search engines. It is one of the most significant problems in the web search engine that can generate the best output to the user's submitted query and can effectively avoid the intentionally biased web pages. We discovered that the intentionally biased web page was exploiting the limitations of the PageRank based search engine's algorithm. In order to solve the problem originating from link based manipulation, we modified the PageRank algorithm to the filtering algorithm that incorporates the similarity between link contexts and hypertext information that can be generalized to the context based measure. The SVD has been adopted to reduce matrix dimensions or to utilize possibly to derive a hidden semantics in the keyword by document matrix. We, however, extend the SVD as an assessment measure to detect the rank-manipulated pages. It can be measured by the traditional transition matrix method as well as the SVD method; so that the method reduced about 17% amount of the rank that is minimum 209.4% higher than normal (not manipulated) web page changes. Using this proposed approach, the chance of manipulated web pages getting high ranks than deserved can be detected, and we

$$\|M - M_k\|_F = \min_{\text{rank}(B)\le k} \|M - B\|_F = \sqrt{\sigma_{k+1}^2 + \sigma_{k+2}^2} \quad (4)$$

$$\text{low rank approximation error} = \frac{\|M_k\|_F}{\|M\|_F}$$

can reinforce search accuracy significantly. In this work, we proposed the method that makes a virtual the most

significant problem to be addressed was that the Web search engine's result was hard to integrate in a manner consistent with the user's submitted query. And we discovered the reason: the spamming page was exploiting the limitations of the PageRank based search engine's algorithm. Then, we defined the spamming page and divided it into two cases, spamming by use of the contents of a Web page and spamming via the Web page links. In order to solve the problem originating from link based spamming, we proposed the modified PageRank algorithm using the similarity between link contexts and target pages [17].

Using this proposed algorithm, the possibility of spamming pages, like those in the Google bombing, getting high rankings will be reduced, and further, we can buttress search accuracy by also increasing the ability to consider the semantics of each page and subsequent links to the existing PageRank algorithm.

References

[1] Robert Cooley, Bamshad Mobasher, Jaideep Srivastava: Data Preparation for Mining World Wide Web Browsing Patterns. Knowledge and Information Systems, 1(1): pp.5-32, 1999

[2] Eiron, N. McCurley, K., and Tomlin, J. Ranking the web frontier. Proceedings of the international conference on World Wide Web, (WWW'04). Pp.309-318, 2004

[3] Etzioni, O., Cafarella, M., Downey, D., Popescu, A., Shaked, T., Soderland, S., Weld, D., Yates, A. Methods for Domain-Independent Information Extraction from the Web: An Experimental Comparison. AAAI, pp.391-398, 2004

[4] Faloutsos, M., Faloutsos, P., Faloutsos, C. On Power-law Relationships of the Internet Topology. In SIGCOMM '99. pp.251-262. 1999

[5] Garofalakis, J., Kappos, P. and Mourloukos, D. Web Site Optimization Using Page Popularity, *IEEE Internet Computing*, 3(4): 22-29, 1999

[6] Glover, E. J., Tsioutsiouliklis, C., Lawrence, S., Pennock, D., and Flake, G. Using Web Structure for Classifying and Describing Web Pages. WWW (2002), 562-569

[7] Gyongyi, Z., Garcia-Molina, H., Pedersen, J. Combating Web Spam with TrustRank. VLDB, pp.576-587, 2004

[8] Haveliwala, T. Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search, IEEE TKDE. 15(4), pp.784-796, 2003

[9] Henzinger, M. R., Heydon, A., Mitzenmacher, M. and Najork, M. On Near-uniform URL Sampling, Computer Networks, 33(1) (2000), 295-308

[10] Hou, J., Zhang, Y. Effective Finding Relevant Web Pages from Linkage Information. IEEE TKDE 15(4) (2003), 940-951

[11] Kleinberg, J. M. Authoritative sources in a hyperlinked environment, Journal of the ACM, Vol.46 (5). (Sept. 1999). 604-632,

[12] Lau, T., Etzione, O., and Weld D. S. Privacy Interfaces for Information Management, *CACM* 42(10): 89-94, Oct. 1999

[13] Lerman, K., Getoor, L., Minton, S., and Knoblock, C. Using the Structure of Web Sites for Automatic Segmentation of Tables. SIGMOD (2004) 119-130

[14] Pandurangan, G., Raghavan, P. and Upfal, E. Using PageRank to Characterize Web Structure. COCOON (2002), 330-339

[15] J. Srivastava, Robert Cooley: Web Business Intelligence: Mining the Web for Actionable Knowledge. INFORMS Journal on Computing 15(2): 191-207 (2003)

[16] Toyota, M. and Kitsuregawa, M. A system for Visualizing and Analyzing the Evolution of the Web with a Time Series of Graphs, In Hypertext, pp. 151-160, 2005

[17] Wookey Lee: Discriminating Biased Web Manipulations in Terms of Link Oriented Measures. In ISCIS 2005: 585-594

[18] Wookey, L., Geller, J. Semantic Hierarchical Abstraction of Web Site Structures for Web Searchers, Journal of Research and Practice in Information Technology, 36(1), pp.71-82, 2004

[19] Wookey Lee, Seung Kim, Sukho Kang: Structuring Web Sites Using Linear Programming. EC-Web 2004: 328-337