Link Analysis to Visualize a Web Graph

Chandresh Pratap Singh[1], Suman Saha[2], Suman Kumar Saurabh[3]

CSE Department, JUIT, Waknaghat
Solan, H.P 173234, India

Abstract

Day by day the web size is growing enormously because at each and every second data is uploaded and downloaded from the web. Some data have proper link to address its location over the web but some data do not provide its proper link that causes the un-accessing of the document. So, it is very difficult to find matches in data for known patterns of interest and also to discover new patterns of interest. This problem can be solved by Web Link analysis technique.web link analysis is a data analysis technique used to evaluate relationship (connections) between nodes. In a web graph, nodes represent web pages and edge represents hyperlinks. In this paper, we have proposed a technique that helps to visualize a web graph of a specific domain. It provides a better understandable view of a large web.

Keywords: *Web, Web Graph, Link analysis, Visualization.*

1. Introduction

A lot of data is uploaded and downloaded over the web daily that makes the size of the web bigger. So, the problem of accessing and analyzing of a specific data has become worse. That problem can be solved by the web link analysis. Our proposed technique helps to easily understand the web graph by visualizing the network of web that contains the nodes and edges. The nodes represent the web pages and the edges represent the hyperlinks. To access a specific data over the web, we must analyze the web links. The process of analyzing the web is known as web link analysis. It is a data analysis technique used to evaluate relationships between nodes [1]. In Previous research, it has been found that the World Wide Web is a scale-free network. It resembles a bow-tie structure and is composed of four components: the Central Core, IN, OUT, and tendrils and tubes. On the basis of statistical analysis, all these research provide insights to understand the global Web.

The hyperlinks of the web give it additional structure, the network of these links is a rich source of latent information. The graph induced by the hyperlinks between Web pages, known as Web graph. In a graph, nodes represent static html pages and hyperlinks represent edges, it can either be direct or undirected. Recent Estimates suggest that there are over billion nodes in the Web graph. The average node has roughly seven hyperlinks to other pages. However, almost all of this research uses the traditional graph representation of the Web with a vertex representing each page and a directed edge representing each hyperlink. To do this whole task we need to perform three steps: web crawling, link analysis and generate a visualizing web graph. The web crawling is the technique to crawl all the web pages over the web. There is web crawler used to do this task. A web crawler is (also known as a Web spider or Web robot) is a program or automated script which browses the World Wide Web in a methodical and automated manner [13] [12]. Secondly, we do the link analysis on the basis of retrieved hyperlinks by the web crawler. And thirdly, we generate the visualization of the web graph.

In this study, we are mainly focusing on web crawling, web Link network analysis and Web graph visualization techniques to map the local Web domains. We wish to know whether the characteristics of the global Web are applicable locally, that is within an organizational domain. Specifically, this study is to provide an exemplary approach to analyzing network properties, identifying topical communities, and visualizing the map of a local domain.

2. Preliminary Knowledge:

In this section, we formulate the background knowledge behind the web crawling, web link analysis and web graph visualization.

2.1 Web Crawling

Web crawlers are programs that exploit the graph structure of the Web to move or to visit from one page to another page. In their beginning such programs were also called wanderers, robots, spiders, and worms, words that are quite evocative of Web imagery. Web crawlers are designed to retrieve Web pages and add them or their

representations to local repository/databases. Web crawlers are mainly used to create a copy of all the visited pages for later processing by a search engine that will index the downloaded pages that will help in fast searches [14]. Web search engines work by storing information about many web pages, which they retrieve from the WWW itself. These pages are retrieved by a Web crawler (sometimes also known as a spider) — which is an automated Web browser that follows every link it sees. Search engines are programs that allow the user to enter key words that are used to search a database of web pages. Each search engine searches in a different way and searches different sites. That is why doing a search in one search engine will produce different results than doing the same search in another engine.

A web crawler uses a program, that given one or more seed URLs, downloads the web pages associated with these URLs, extracts any hyperlinks contained in them, and recursively continues to download the web pages identified by these hyperlinks.

A good crawler must have to follow all the given Properties [15]:

Robustness: The crawler must have the ability to handle spider traps.

Distribution: The crawler should have the ability to execute in a distributed fashion across multiple machines.

Scalability: The crawler architecture should permit scaling up the crawl rate by adding extra machines and bandwidth.

Performance and efficiency: The crawl system should make efficient use of various system resources including processor, storage and network bandwidth.

Quality: Given that a significant fraction of all web pages are of poor utility for serving user query needs, the crawler should be biased towards fetching “useful” pages first.

Freshness: In many applications, the crawler should operate in continuous mode: it should obtain fresh copies of previously fetched pages. A search engine crawler, for instance, can thus ensure that the search engine’s index contains a fairly current representation of each indexed web page. For such continuous crawling, a crawler should be able to crawl a page with a frequency that approximates the rate of change of that page.

Extensible: Crawlers should be designed to be extensible in many ways – to cope with new data formats, new fetch protocols, and so on. This demands that the crawler architecture should be modular.

A minimal web crawler follows a number of steps to extract and store all the hyper links from the web page over the web, indexed them and stores them into a repository for further processing. The web crawling steps are shown in the figure given below:

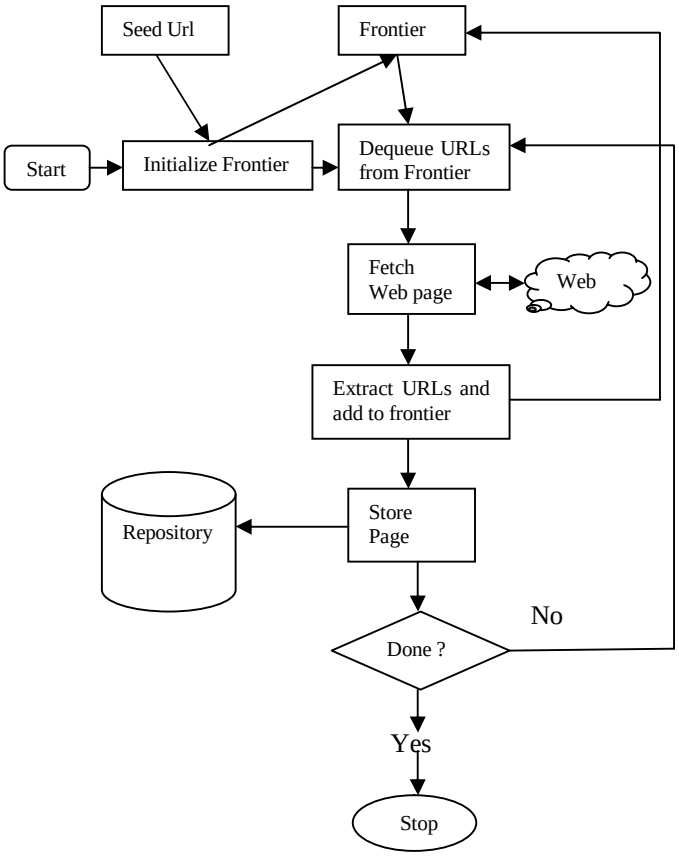

Fig. 1 Stepwise processing of a Crawler.

2.2 Web Link Analysis

Web Link analysis is the technique to evaluate relationships between nodes. There are multiple links that connected to multiple nodes. As there are billion number of nodes and links, So it becomes very difficult to analyze which links are useful for the user and which are not. To analyze links we have a number of link analysis algorithms which helps to rank the links on the basis of their importance [2].

In this section we will provide a description of the Link Analysis Ranking (LAR) algorithms currently used by most well-known search engines. we will describe the PageRank algorithm used by Google and the HITS algorithm used by Teoma.

A simple heuristic that can be viewed as the predecessor of all link analysis ranking algorithms is to rank the pages according to their popularity. The popularity of a page on

the web is measured by the number of pages that point to it. We refer to this algorithm as the InDegree algorithm, since it ranks pages according to their in-degree in the graph G. That is, for every node i, $a_i = |B(i)|$.

Page Rank Algorithm:

Page Rank algorithm's basic idea is to ignore the text on Web pages and other content, only consider the hyperlinks between pages, the Web as a huge directed graph G=(V,E), on behalf of the node v ∈ V a Web page, there are directed edges from node to node on behalf of a hyperlink, the node outdegree is the starting page, a hyperlink from the total number of infiltration degree refers to all hyperlinks pointing to the total number of nodes. Page Rank is defined as follows: the Web mapping as a directed graph, is a page pointing to the set of all pages, all pages that point to a collection of pages. For each node out of 0 S, = {set of all digraphs} N nodes, all other nodes, so that node S can have the Page Rank value of the uniform transfer to all other pages. The page rank specific iteration formula is as follows:

$$PR(j) = (1-d) + d\sum_{i \in B_i} \frac{PR(i)}{|F_i|}$$

Where, *d* is defined as users continue to click on the link random probability, it depends on the number of clicks, is set to between 0-1, usually set to 0.85. *d* higher the value, click on the link to the greater probability, therefore, the user stops clicking and random surfing to another page with a constant edge probability (1-d) said. No matter how the inbound links, surf to the probability of a page is always (1-d). Pagerank value determines the website of the random probability of access to this page; users click a link within the page the probability of the number of links on the page exclusively by the number of decisions. Thus, a page accessible through the probability of random surfing is connected to his other links on the page by clicking probabilities. In summary, we suggest that PageRank is a better indicator serving as a substitution of the number of citations for measuring papers' influence. On the one hand, it has high relevancy with the traditional citation measures and will hardly lead to perverse results. On the other hand, it could incorporate the importance of citing papers to a specific cited paper, therefore excavating several important papers that may suffer from low citations.

Page Rank's implementation process follows as: the page URL into a unique integer corresponding to each hyperlink with its integer ID to the index stored in the database, after pretreatment, the set of initial each page PR value is 0, through the above recursive algorithm, each page of the Page Rank value of repeated iterations until the results converge. The Page Rank algorithm, introduced by Page et al., precompiled a rank vector that provides a priori "importance" estimates for all of the pages on the Web. This vector is computed once, offline, and is independent of the search query. Page Rank algorithm, the weights for the contribution of external links is the average that is without considering the importance of the different links.

HITS (Hyperlink-Induced Topic Search):

HITS algorithm is the use of Hub / Authority approach the search method. The basic idea of HITS algorithm is: the Web page is divided into pages and Authorities Hub page. Hub pages is to provide a collection of web links pointing to the authority of the Web page itself may not be important, or that there is no point to it a few pages, but the Hub website has provided a link to a topic to the site are most important collection of links. Generally a good point too many good authorities Hub Web page, a good authority page is a page from the many good points Hub Web pages. The Hub and Authorities to strengthen relations between web pages can be used for authoritative Web pages and Web structure and resources found in the automatic discovery.

HITS algorithm is as follows: the query q submitted to match the traditional keyword-based search engine, search engine returns many pages, pages from which to take the former as the root set of n (root set), with s said. S by adding to the page referenced by s web pages and references to s expansion into a larger set of T. T in the calculation of every page of the Authority and hub weights of the weight, which is a recursive process.

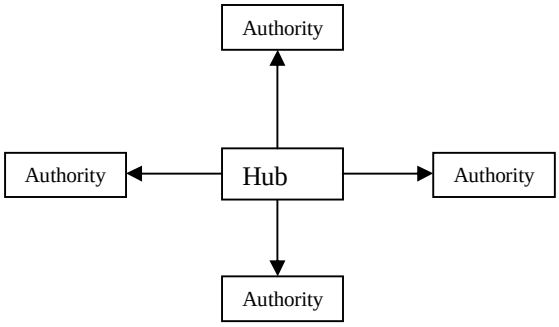

Fig. 2 A Hub Graph.

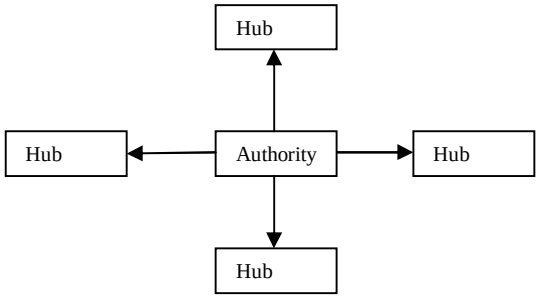

Fig. 3 An Authority graph

2.2 Web Graph

The pages and hyperlinks of the World-Wide Web may be viewed as nodes and edges in a directed graph. This graph has about a billion nodes today, several billion links, and appears to grow exponentially with time. A web graph is a graph of the physical layer with routers, computers etc as nodes and physical connections as edges [4]. In a web graph nodes represents the web pages and edges represents the hyperlinks. Edges can be directed or undirected.

There are some basic statistics of a web graph:

- Size and connectivity of the graph.
- Number of connected components.
- Distributed pages per site.
- Distribution of incoming and outgoing connections per site.
- Average and maximum length of the shortest path between any two vertices(diameter).

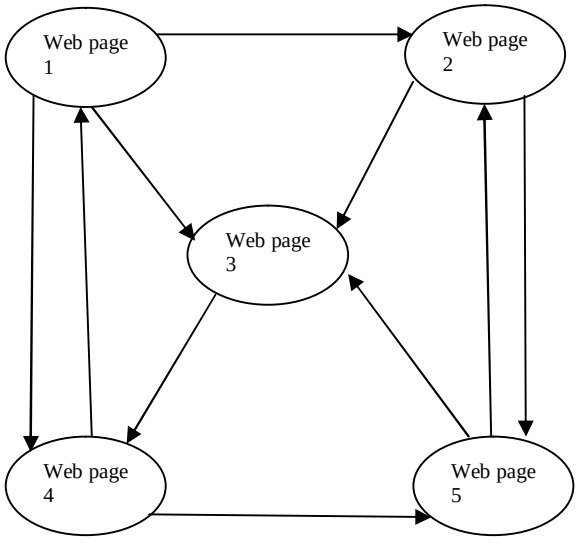

Fig. 4 A Sample of Web Graph.

The connectivity of a graph follows a power law distribution. In a power law distribution, it is observed that over several orders of magnitude with an exponent g, is in the 1.6-1.9 range [3]. The Distribution of number of connections per node follows a power law distribution. In a Study at Notre Dame University reported that,

g = 2.45 for out degree distribution
g = 2.1 for in degree distribution

3. Crawling the Local Domain:

Using crawler we will crawl the local web domain. The working of a minimal crawler follows the following steps to extract all the hyperlinks from the web page. The steps are given below:

Step 1: Firstly the crawler will select a Seed URL or URLs from the frontier.
Step 2: If the Frontier queue is empty, then enter any Seed URL into the frontier queue.
Step 3. If there is URL present in the queue, then it pick up the URL by the rule of First-in-First-out.
Step 4. Fetch the web-page corresponding to that URL from the World Wide Web.
Step 5: Use a parser to extract all the hyperlinks from that web page.
Step 6: Add all the Hyperlinks retrieved into the frontier queue.
Step 7: Add all the visited web pages into the repository.

The above algorithm helps us to retrieve total 61 number of .php files.

Node 1 http://juit.ac.in/index.php
Node 2 http://juit.ac.in/University/university.php
Node 3 http://juit.ac.in/University/university.php
Node 4 http://juit.ac.in/University/genesis.php
Node 5 http://juit.ac.in/University/governors.php
Node 6 http://juit.ac.in/University/supervisory.php
Node 7 http://juit.ac.in/University/management.php
Node 8 http://juit.ac.in/University/visitors.php
Node 9 http://juit.ac.in/University/linkages.php
Node 10 http://juit.ac.in/University/archi.php
Node 11 http://juit.ac.in/University/coo.php
Node 12 http://juit.ac.in/University/vc.php
Node 13 http://juit.ac.in/University/brig.php
Node 14 http://juit.ac.in/University/staff_admin.php
Node 15 http://juit.ac.in/University/staff_accounts.php
Node 16 http://juit.ac.in/University/server.php
Node 17 http://juit.ac.in/Department/Department.php
Node 18 http://juit.ac.in/Department/Electronics/ece.php
Node 19 http://juit.ac.in/Department/CSE/cse.php
Node 20 http://juit.ac.in/Department/bio/bio.php

Node 21 http://juit.ac.in/Department/civil/civil.php
Node 22 http://juit.ac.in/Department/physics/physics.php
Node 23 http://juit.ac.in/Department/maths/maths.php
Node 24 http://juit.ac.in/Department/pd/pd.php
Node 25 http://juit.ac.in/Department/pharmacy/pharma _cy . php
Node 26 http://juit.ac.in/lrc/library.php
Node 27 http://juit.ac.in/lrc/about_library.php
Node 28 http://juit.ac.in/lrc/collection.php
Node 29 http://juit.ac.in/lrc/newarrival.php
Node 30 http://juit.ac.in/lrc/services.php
Node 31 http://juit.ac.in/lrc/SIAM-e-Book-Collection.php
Node 32 http://juit.ac.in/lrc/workinghours.php
Node 33 http://juit.ac.in/lrc/staff.php
Node 34 http://juit.ac.in/lrc/lrc_help.php
Node 35 http://juit.ac.in/lrc/recommend_book.php
Node 36 http://juit.ac.in/lrc/Learning_Resource_Center .php
Node 37 http://juit.ac.in/lrc/iBIRA.php
Node 38 http://juit.ac.in/lrc/ContactUs.php
Node 39 http://juit.ac.in/tnp/tnp.php
Node 40 http://juit.ac.in/tnp/tnp.php
Node 41 http://juit.ac.in/Campus/jyc.php
Node 42 http://juit.ac.in/photogallery.php
Node 43 http://juit.ac.in/ieee/ IEEE.php
Node 44 http://juit.ac.in/Contact/Contact.php
Node 45 http://juit.ac.in/Contact/address.php
Node 46 http://juit.ac.in/Contact/location.php
Node 47 http://juit.ac.in/Contact/web.php
Node 48 http://juit.ac.in/attachments/aicteaccreditation .php
Node 49 http://juit.ac.in/attachments/Ad.php
Node 50 http://juit.ac.in/attachments/Biospectrum2010 .php
Node 51 http://juit.ac.in/attachments/efy1.php
Node 52 http://juit.ac.in/virtualcampus/index.php
Node 53 http://juit.ac.in/attachments/WilliamWebster_ Scholarship.php
Node54 http://www.juit.ac.in/Department/pharmacy /admission_flyer.php
Node 55 http://www.juit.ac.in/Department/bio/ summer_training.php
Node 55 http://juit.ac.in/attachments/MOUJUIT.php
Node 56 http://www.juit.ac.in
Node 57 http://www.juit.ac.in/University/university.php
Node 58 http://www.juit.ac.in/Departments/Electronics /ece.php
Node 59 http://www.juit.ac.in/lrc/library.php
Node 60 http://www.juit.ac.in/tnp/tnp.php
Node 61 http://www.juit.ac.in/Campus/jyc.php

Objects	No. of Directed Links	rank
University	16	0.100
Department	9	0.050
Library	9	0.05
Contacts	4	0.03

Fig. 5 Table shows the objects and its corresponding nodes with rank.

4. Visualizing the Domain Map:

In Fig. 6, A Web Graph of site: http://juit.ac.in is shown. In this Web Graph, total 61 nodes are used. Nodes are the total number of web Pages present in the site: juit.ac.in. this graph is formed on the basis of retrieved links by the Web Crawler. We have used java based graph algorithm platform to design this graph [9]. Following are the data types used in it:

```
class Vertex {
int identity;
int predecessorNode;
int x;
int y;
Color color;
int hopDist;
int inDegree;
int outDegree;
double distance;

 class Edge {
int fromNode;
int toNode;
double weight;
double flow;
Color color;

class Graph {
protected int vertexSerialNo;
 ListArray vertexList;
 ListArray2 edgeList;
boolean directed;

}
```

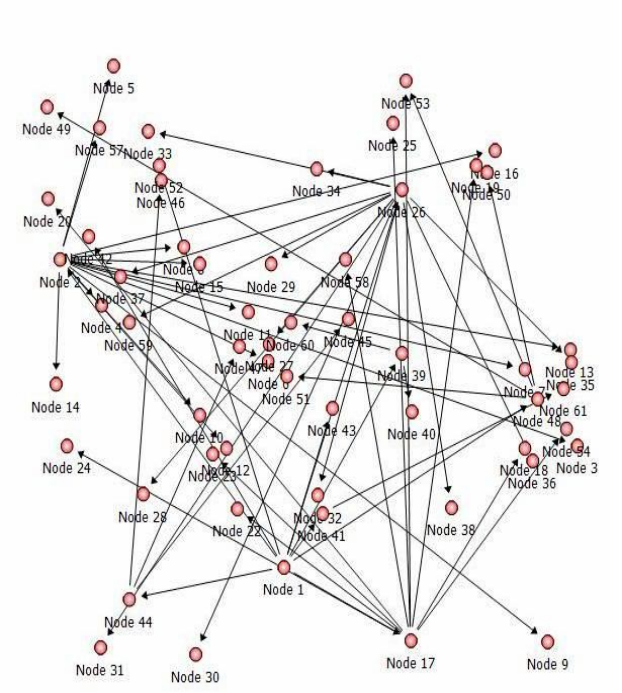

Fig. 6 A Web Graph of domain http://juit.ac.in.

In Fig: 7, A matrix diagram is given. It shows a N*N adjacency matrix between the nodes. The points in the figure are the nodes that formed on the basis of the retrieved links.

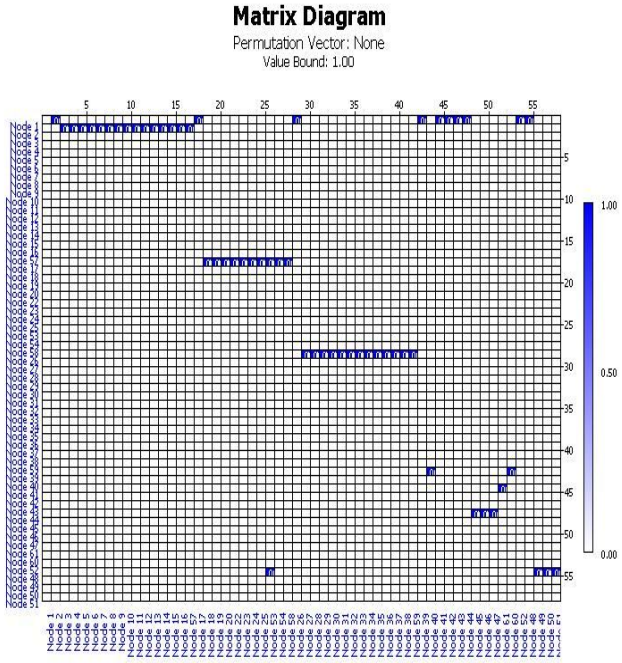

Fig. 7 Adjacency Matrix of The Web Graph

5. Conclusions:

In this paper, we use various web link analysis techniques to rank the retrieves hyperlinks. On the basis of retrieved hyperlinks we used our proposed technique to visualize the web graph of a local domain. we analyzed a local Web graph by heuristically examining its network properties. The results were consistent with previous works on the Web, and proposed Using co-citation analysis and visualization techniques, this paper offered a way to investigate a local Web structure and to visualize its domain map that is more intuitively understandable.

References

[1] Kolda, T.G.; Bader, B.W.;Kenny, J.P; "Higher-order web link analysis using multilinear algebra", Data Mining, Fifth international conference on Digital Object Identifier, Publication year: 2005.

[2] Sheu, P.; Yu, H.; Ramamoorthy, C.; Joshi, A.; Zadeh, L.; "Link Analysis in Web Mining: Techniques and Applications", Semantic Computing Digital Object Identifier Page(s): 69 – 86, Copyright Year: 2010.

[3] Lai, Wei; Huang, Xiaodi; "From graph data extraction to graph layout: Web information visualization", Information Sciences and Interaction Sciences (ICIS), 2010 3rd International Conference on Digital Object Identifier, Publication Year: 2010 , Page(s): 224 – 229.

[4] Sriram Raghavan, Garcia-Molina, H., "Representing Web Graphs", Data Engineering, 2003. Proceedings. 19th International Conference on Digital Object Identifier, Publication Year: 2003, Page(s): 405 - 416.

[5] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, D.Sivakumar, Andrew S. Tomkins, Eli Upfal, "The Web as a Graph".

[6] Lise Getoor, Christopher P. diehl, "Link Mining: a survey".

[7] Daniel M. Dunlavy and Tamara G. Kolda, "Temporal Link Prediction Using Matrix and Tensor Factorizations", Sandia National Laboratories.

[8] Ben Taskar, Ming-Fai Wong, Pieter Abbeel, Daphne Koller, "Link Prediction in Relational data".

[9] Ding-Yi Chen, Tyng-Ruey Chuang and Shi-Chun Tsai, "JGAP: a Java-based graph algorithms platform", Software Practice and Experience Softw. Pract. Exper.2001; **31**:615–635.

[10] Danny Dunlavy, Tammy Kolda, Philip Kegelmeyer, "Tensor Decompositions for analyzing Multi-Link Graphs", SIAM Parallel Processing for Scientific Computing March 13, 2008.

[11] Brett W. Bader & Tamara G. Kolda, "Tensor Decompositions, the MATLAB Tensor Toolbox, and Applications to Data Analysis", Sandia National Laboratories.

[12] Carlos Castillo , "Effective Web Crawling" Castillo.pdf.

[13] "Web Crawling and Indexes", Cambridge University Press, April 2009.

[14] Junghoo Chao, "Crawling the Web: Discovery and Maintenance of Large-scale Web Data".
[15] Brian Pinkertan, "Web Crawler: Finding what people want.

Authors:

Chandresh Pratap Singh received his B.Tech degree in Information Technology from Krishna Institute of Engineering & Technology (U.P.T.U), Ghaziabad, U.P, India, in 2009. He is currently pursuing M.TECH degree in Computer Science & Engineering at Jaypee University of Information Technology, Waknaghat, Solan, H.P, India. His research interest includes Data mining, Web Mining and Image processing.

Suman Saha working as a Senior Lecturer in JUIT, Waknaghat, Solan, H.P, India. He received his M.Tech degree in Computer Science from Indian Statistical Institute, Kolkata, India, in 2004. He is currently doing his Ph.D (Thesis to be submitted) in computer Science from Indian Statistical Institute, Kolkata, in 2010. He has been awarded Senior Research Fellowship by the Department of Science Tehnology (DST) Government of India, in 2004. His research interests include Web Intelligence, Web Graph, Semantic Web Services, Web Crawling, Tensor Decomposition, Cloud Computing and Grid Computing.