Department of Computing

Link Analysis Algorithms to Handle Hanging and Spam Pages

Ravi Kumar Patchmuthu

This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University

June 2014

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature : ……

(RAVI KUMAR PATCHMUTHU)

Date : 18th June 2014

Abstract

In the Web, when a page does not have any forward or outgoing links, then that page can be called as hanging page/dangling page/zero-out link page/dead end page. Most of the ranking algorithms used by search engines just ignore the hanging pages. However, hanging pages cannot be just ignored because they may have relevant and useful information like .pdf, .ppt, video and other attachment files. Hanging pages are one of the hidden problems in link structure based ranking algorithms because:

- They do not propagate the rank scores to other pages (important function of link structure based ranking algorithms).
- They can be compromised by spammers to induce link spam in the Web.
- They can affect Website optimization and the performance of link structure based ranking algorithms.

A detailed literature survey on link structure based ranking algorithms, hanging pages and their effect on Web information retrieval was conducted. Different link structure based ranking algorithms were explored and compared. Also, literature review on Web spam and in particular link spam was conducted. Finally, a detail literature survey on Web site optimization was conducted. The following are the research objectives:

- Handling Hanging Pages (HP) in the link structure based ranking algorithms. PageRank is used as the base algorithm throughout this research.

- Developing Hanging Relevancy Algorithm (HRA) to produce fair and relevant ranking results by including only the relevant hanging pages.
- Developing Link Spam Detection (LSD) algorithm to analyse and detect the effect of hanging pages in link spam contribution.
- Developing techniques and methods to improve Web Site Optimization (WSO) by studying the effect of hanging pages in Search Engine Optimization (SEO).

The following are the methodologies used to meet the research objectives:

- Web Graph is implemented where nodes are treated as Web pages and edges between nodes are treated as hyperlinks.
- PageRank algorithm is simulated, matrix interface as well as graph interface is created and the ranks of Web pages are computed.
- Experiments are conducted to show the effects of hanging pages and methods are proposed to handle hanging pages in ranking Web pages.
- Hanging Relevancy Algorithm (HRA) is implemented and only the relevant hanging pages are included in the rank computation to reduce the complexity.
- Link Spam Detection (LSD) algorithm is implemented to detect link spam contributed by hanging pages.
- Experiments are conducted to show the effect of hanging pages in Search Engine Optimization (SEO) and factors are proposed to build optimized Web sites.

PageRank algorithm was implemented and experiments were carried out to show its convergence. Three publicly available datasets, WEBSPAM-UK2006, WEBSPAM-

UK2007 and EU2010 were used for the experiments, apart from live data from the World Wide Web. Several experiments were conducted to show the effects of hanging pages on Web page ranking. Methods were proposed to include all the hanging pages in PageRank computation and to compare them with PageRank algorithm. The proposed methods were slower than PageRank algorithm but produced more relevant results. The experiments showed the percentage of hanging pages in the following datasets: WEBSPAM-UK2006 - 21.35%, WEBSPAM-UK2007 - 43.11%, EU2010 - 54.21% and the Curtin University (Sarawak) Website - 35.57%. The study showed that the hanging pages are keep increasing in the Web.

Hanging Relevancy Algorithm (HRA) is implemented and it produced more relevant results with less computation time compared to including all the hanging pages in the computation. The experiment also showed that the ranks of certain relevant hanging pages were increased by four, signifying that these pages deserved a better ranking.

Experiments were carried out using live Web data to prove the contribution of link spam by hanging pages. The results showed that the rank of the target page after link spam had increased by two and the order had also improved. This proved that hanging pages contributed to spam and the proposed method had detected link spam contributed by the hanging pages. Experiments were done to determine the On-Site and Off-Site ranking factors by taking www.curtin.edu.my as a sample Website. The link analysis experiment showed that 90% of the sample Website's back links are external links and only 10% are internal back links. 80% of the sample Website's links are followed back links and only 20% of them are no-followed back links. The experiments showed different ranking factors and also suggested factors to improve the ranking of the particular Website.

The research study has therefore, helped to improve the rankings of relevant hanging pages and reduce the link spam contributed by hanging pages in the Search Engine Result Pages of link structure based ranking algorithms.

Acknowledgement

This thesis would not have been possible without the guidance and help of several individuals, who in one way or another, contributed or extended their valuable assistance in the preparation and completion of this research study.

First, I would like to thank my Supervisor, Dr. Zhuquan Zang (Associate Professor, HOD, Department of Electrical and Computer Engineering), for providing me with all the administrative support and guidance throughout this research journey.

Next, I would like to express my utmost gratitude to my Co-Supervisor/Associate Supervisor, Dr. Ashutosh Kumar Singh (Ex. Associate Professor, Department of Electrical and Computer Engineering, Curtin University, (Miri Campus), currently, Professor, Department of Computer Application, NIT, Kurukshetra, India) to whom I was reporting during this research journey. I have enjoyed his energetic and enthusiastic thought provoking ideas, constructive comments and valuable advices during the research discussions; much of this is reflected in my research and in the thesis as a whole. I am really grateful for Dr. Singh's constant support, guidance and encouragement, and deeply thankful and proud of him as my supervisor. The enthusiasm and dedication he has shown towards research, has been commendable and particularly motivational for me, during the tough stages of this research endeavour.

I would like to acknowledge the support given by the Chairperson of the Doctoral Thesis Committee, Dr. Chua Han Bing (Associate Professor, HOD, Chemical Engineering) and the ex-chairperson Dr. K. Shenbaga (ex-Dean, Research and

Development, Curtin University). Both of them have provided guidance, encouragement and support and shared tremendous knowledge with me.

I am also obliged to acknowledge Professor Dr. Michael Cloke (Dean, SOE), Professor Dr. Marcus Man Kon Lee (Associate Dean, Research Training), Associate Professor Dr. Michael Kobina Danquah (Associate Dean, Research and Development) and Nur Afnizan Johan (Tom) for all the support given to me during the course of this research study.

I would also like to extend my sincere gratitude to my research colleague, Alex Goh Kwang Leng for participating in the research discussions and contributing to the research study as a whole. I wish him good luck for his journey towards his doctoral study. I like to acknowledge Dr. Anand Mohan, (Director, NIT, Kurukshetra, India) for his expert advice on this research study.

My appreciation also goes to Mdm. Sheila Gopinath for proof reading and editing this thesis, Billy Lau Pik Lik for the initial contribution to this research and Dr. Herbert Raj, my brother-in-law, for correcting few chapters. I extend my thanks to others as well, who have helped me either directly or indirectly in this research study.

My sincere gratitude goes to Jefri Bolkiah College of Engineering, Kuala Belait, and Ministry of Education, Brunei for supporting me in this research study.

Special thanks go to my family, especially to my wife Jelciana, for her constant support and motivation to complete this thesis. I would also like to acknowledge my son Arun and daughter Cynthea for their love and support, as well as my father and my in-laws for their telephonic support and motivation.

Last of all but most importantly; I would like to thank Almighty God for providing me with such great strength and opportunity to complete this thesis.

Publications

JOURNAL PAPERS

1. **Ravi Kumar Patchmuthu**, Ashutosh Kumar Singh and Anand Mohan. “Efficient Methodologies to Overcome the effects of Hanging Pages in Website Optimization”, International Journal of Web Engineering and Technology. (Scopus, Under Review).

2. **Ravi Kumar Patchmuthu**, Goh Kwang Leng, Ashutosh Kumar Singh and Anand Mohan. “Efficient Methodologies to determine the Relevancy of Hanging Pages with Stability Analysis”, Cybernetics and Systems: An International Journal. (**ISI**, Under Review).

3. **Ravi Kumar Patchmuthu**, Ashutosh Kumar Singh and Anand Mohan. “A New Algorithm for Detection of Link Spam Contributed by Zero-Out-Link Pages”, Turkish Journal of Electrical Engineering and Computer Sciences. (Accepted and waiting for publication, **ISI, IF: 0.58**).

4. **Ravi Kumar Patchmuthu**, Goh Kwang Leng and Ashutosh Kumar Singh. 2013. "Application of Markov Chain in the PageRank Algorithm”, Pertanika Journal of Science & Technology (JST). 21(2):541-554, July 2013 (**SCOPUS**).

5. Ashutosh Kumar Singh, **Ravi Kumar Patchmuthu** and Goh Kwang Leng. 2011. "Efficient Methodologies to Handle Hanging Pages Using Virtual Node", Cybernetics and Systems: An International Journal, 42 (8): 621-635, December 2011 (**ISI, IF: 1.182**).

6. **Ravi Kumar Patchmuthu** and Ashutosh Kumar Singh. 2010. "Web Structure Mining: Exploring Hyperlinks and Algorithms for Information

Retrieval", American Journal of Applied Sciences 7(6): 840-845, 2010 (**SCOPUS**).

7. Ashutosh Kumar Singh and **Ravi Kumar Patchmuthu.** 2009. "A Comparative Study of Page Ranking Algorithms for Information Retrieval", World Academy of Science, Engineering and Technology, 3 (4): 760-771.

REFERRED CONFERENCE PAPERS

8. **Ravi Kumar Patchmuthu**, Ashutosh Kumar Singh, and Anand Mohan. "Efficient Methodologies to Optimize Website for Link Structure Based Search Engines", In Proceedings of 2013 International Conference on Green Computing, Communication and Conservation of Energy, (ICGCE 2013), Dec. 12-14, 2013, Chennai, India, IEEE ICGCE (ISBN: 978-1-4673-6126-2/13/) 721-726.

9. Ashutosh Kumar Singh, **Ravi Kumar Patchmuthu** and Goh Kwang Leng. "Solving Dangling Relevancy Using Genetic Algorithm", In Proceedings of 2012 International Conference on Uncertainty Reasoning and Knowledge Engineering, Aug. 14-15, 2012, Jakarta, Indonesia, IEEE CPMG, NJ (ISBN: 978-1-4673-1459-6) 9-12.

10. **Ravi Kumar Patchmuthu**, Goh Kwang Leng and Ashutosh Kumar Singh. "Application of Markov Chain in the PageRank Algorithm", In proceedings of the 3rd International Conference On Science and Engineering, 8-9 Nov. 2011, Curtin University, Miri, Malaysia.

11. Ashutosh Kumar Singh, **Ravi Kumar Patchmuthu** and Goh Kwang Leng. "Efficient Algorithm for Handling Dangling Pages using a Hypothetical Node", In proceedings of the 6th IEEE International Conference on Digital Content, Multimedia Technology and its Applications (IDC2010) 2010, Aug. 16-18, Seoul, Korea.

12. **Ravi Kumar Patchmuthu** and Ashutosh Kumar Singh. "Web Structure Mining: Exploring Hyperlinks and Algorithms for Information Retrieval", In

proceedings of the 2nd International Conference on Science and Engineering, 24-25, Nov. 2009, Curtin University, Miri, Malaysia.

Table of Contents

List of Tables

List of Figures

Abbreviations

ARPAnet	Advanced Research Project Agency Network
AT	Anchor Text
BR	Bounce Rate
CERN	European Organisation for Nuclear Research
CTR	Click Through Rate
DG	Directed Graph
DT	Dwell Time
HITS	Hyperlink-Induced Topic Search
HP	Hanging Pages
HR	Host Rank
HRA	Hanging Relevancy Algorithm
HTML	HyperText Markup Language
IPR	Inverse PageRank
IR	Information Retrieval
IRP	Irrelevant Pages
LSD	Link Spam Detection
ME	Maximum Edges
NLP	Natural Language Processing
NPC	Number of Times a Page is Clicked
NPD	Number of Times a Page is Displayed
OPIC	Online Page Importance Computing
OSI	Open Systems Interconnection
PP	Proposed Probability Matrix
PR	PageRank
RP	Relevant Pages
RR	Relevancy Rule
QT	Query Term
SALSA	Stochastic Approach for Link-Structure Analysis
SCC	Strongly Connected Component
SEO	Search Engine Optimisation
SERPs	Search Engine Result Pages
TCP/IP	Transmission Control Protocol/Internet Protocol
TR	TrustRank
UG	Undirected Graph
VRY	Very Relevant Pages
WCM	Web Content Mining

WG	Web Graph
WIR	Web Information Retrieval
WPR	Weighted PageRank
WRP	Weak Relevant Pages
WSM	Web Structure Mining
WSO	Web Site Optimisation
WUM	Web Usage Mining
WWW	World Wide Web

Glossary

A	Adjacency matrix
A_p	Authority Weight
AT	Anchor Text
$B_{(p)}$	Set of Referrer pages
d	damping factor
de_i	Degree of vertex
d_s	DirichletRank Score
e	Edge
E	$n \times n$ matrix of all ones
G	Graph
G_h	Host Graph
$G(V, E)$	Direct Graph with V as set of vertices and E as edges
$G_h(V_h, E_h)$	Host Graph with V_h as set of Host vertices and E_h as Host edges
$G_w(V_w, E_w)$	Web graph with V_w as set of Web pages and E_w as hyperlink between Web pages
H_p	Hub Weight
id	in-degree
$I_{(p)}$	Set of Reference pages
JP	Jump Probability matrix
od	out-degree
P	Probability matrix
PP	Proposed Probability matrix
PV	Probability matrix with Virtual node
QT	Query Term
T	Irreducible closed subset
v	Eigenvector
VN	Virtual node
V_{one}	Total number of visitors viewing one page

V_{total}	Total entries to a page
W^{in}	Weight values of Incoming links
W^{out}	Weight values of Outgoing links
W_p	Website with number of pages *p*
W_p^{in}	Incoming links to Website *W*
W_p^{out}	Outgoing links from Website *W*
W_p^{hp}	Hanging pages going out from Website *W*

Chapter 1 Introduction

1.1 OVERVIEW

According to a recent survey conducted by Netcraft, an Internet service company, there are currently 958,919,789 Web sites in the World Wide Web (2014). Another report from Factshunt, an Internet service company, states that as of December 2013, there are 14.3 trillion live Internet pages (2013). It is evident, therefore, that the number of Websites is approaching the one trillion mark, and the number of Web pages is increasingly difficult to count. Factshunt, further states that in 2013, the average number of searches in the Google search engine was 149.16 billion per month, and the total number of searches was 2.0827 trillion (2013). Information retrieval from this many trillion pages is a mammoth task. Search engines and their ranking algorithms, thus play a very important role in extracting relevant information from the World Wide Web (WWW); however, searching for relevant information in this huge Web is a challenging task due to the non-standard structure of the Web, complex styles of different Web data, the exponential growth, dynamic nature of the Web and the unfair treatment of relevant hanging pages by the search engines.

The following are the background information related to this thesis. World Wide Web (WWW) is used as a medium in this research to collect data to analyse the link structure. This research uses the concepts of Information Retrieval (IR) to retrieve data from Internet. One of the Web mining techniques, Web Structure Mining (WSM) is applied in this research to retrieve the data using link structure analysis. This thesis utilizes the PageRank algorithm of Google search engine as the base ranking algorithm throughout this research. Another, related information in this study is Web spam which is also described here. Finally, the background information on Search Engine Optimization (SEO) is covered here.

World Wide Web

Today's Internet and the WWW are an extension of the Galactic Network and the packet switching concept, developed by J.C.R. Licklider and Leonard Kleinrock

(both of Massachusetts Institute of Technology - MIT), in the early 1960's (Leiner et al. 2009). The Advanced Research Project Agency Network (ARPAnet) was the first product of their research along with other researchers. After the introduction of the Transmission Control Protocol/Internet Protocol (TCP/IP) and the Open Systems Interconnection (OSI) in the early 1980s, the Internetworking concept evolved into the Internet in the mid-1980s. The Internet today is the result of the hard work of so many researchers and technologists who cover areas like technological evolution, operations and management of global and complex networks, and the social and commercial aspects of the information infrastructure.

In 1989, WWW was developed by Tim Berners Lee (Gillies and Cailliau 2000) of the European Organisation for Nuclear Research (CERN), and other researchers from organisations using distributed computing and Internet (Seymour, Frontsvog and Kumar 2011). V. Cerf (Stanford University) developed protocols and structure for the Internet in 1973 (Bing 2007). HyperText Markup Language (HTML) started in late 1991 and it became the standard markup language in 1995. There are many versions of HTML and the current version is HTML5 which was introduced in early 2008. There was also a need for Web browsers so that users could access, retrieve documents and perform other tasks on the Internet. The first Web browser, Mosaic was developed in 1992, followed by Netscape Navigator, Internet Explorer 1, OmniWeb, Chrome, etc. While some like Netscape Navigator, OmniWeb have been phased out, others like Internet Explorer, Mozilla Firefox, Opera and Safari are still in use.

Information Retrieval (IR)

Information Retrieval (IR) is a wide, often loosely-defined term which deals with the representation, storing, organization of and access to information items (Baeza-Yates and Ribeiro-Neto 1999). IR deals with two types of retrieval: Traditional IR and Web IR (Langville and Meyer 2006a). The Traditional IR, pre-existing the Web, is a search within a smaller, more controlled and non-linked collections environment. Examples of Traditional IR are searching for a book in a library's collection of books or searching for a movie title in a movie collections media. A Web Information Retrieval (WIR) is a search for everything within the world's largest linked document

collection, i.e. WWW.

The representation and organisation of the information items should provide the user with easy access to the information in which he or she is interested. Generally, IR is about document retrieval, and emphasises the document as the basic unit. These document collections are non-linked, generally static, and organized and categorized by the librarians, journal editors, catalogue editors, etc. They can be stored in physical form such as books, journals, and artwork, as well as in electronics format like microfiche, DVDs and Web pages. Previously the search mechanisms were manual but now most of them are computerized. These computerized mechanisms are referred to as search engines, and are introduced later in this chapter.

Traditional IR collection uses three basic computer-aided search techniques: Boolean, Vector Space and Probabilistic models (Baeza-Yates and Ribeiro-Neto 1999). These search models developed in the 1960s, have grown, meshed and morphed into new search models. There are thousands of search engines in the Web, all of which use one of the three basic search techniques mentioned above. There is also a fourth search technique model in the Traditional IR called meta-search engines, which combines three basic models.

Precision, Recall and *Freshness* are the three important parameters used to measure the performance of search engines. *Precision* is the ratio of the number of relevant documents retrieved against the total number of documents retrieved, while *Recall* refers to the ratio of the number of relevant documents retrieved against the total number of relevant documents in the collection; the higher the precision and recall the better the search engine. *Freshness* relates to how fast, fresh and new contents can be retrieved by search engines. A general architecture of an IR system is given in Figure 1.1 (Bing 2007).

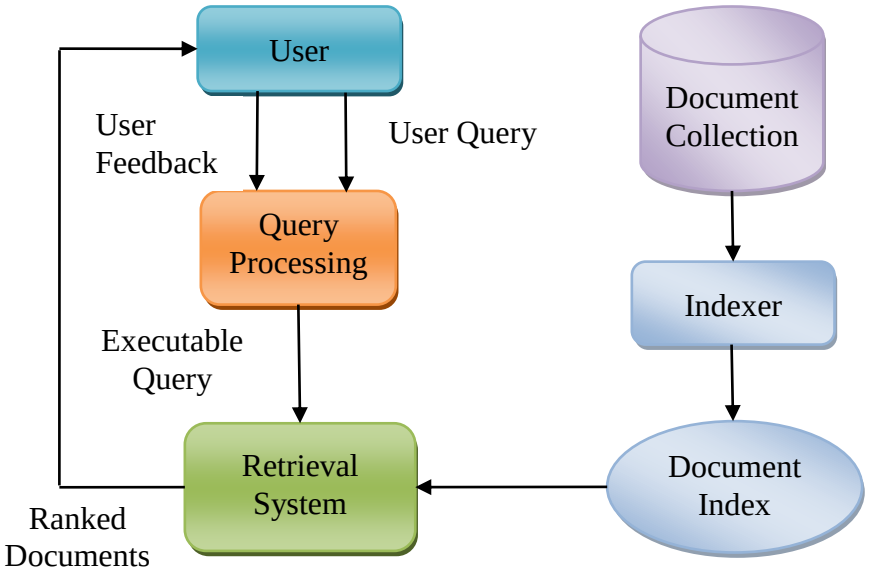


Figure 1.1: General IR System Architecture

The General IR system architecture consists of a query processing module, retrieval system, document collector or a text database, an indexer module, document index and of course a user, to issue a query. Here, a user looking for information issues a query to the retrieval system through the query processing module. The retrieval module then uses the document index to retrieve those documents that are relevant to the query terms, computes relevancy scores for them, ranks the retrieved documents according to the scores, and subsequently presents them to the user. The indexer helps the efficient retrieval of the documents by indexing them.

Overview of Web Mining

The WWW is a huge, explosive, diverse, dynamic and mostly unstructured data repository, which supplies an incredible amount of information, and also raises the complexity of dealing with the information from the different perspectives of information seekers, Web service providers and business analysts. The users want to have effective search tools to find the relevant information easily and precisely from the Web, since most of the existing tools provide information much of which may not be relevant to the user queries. Web service providers would like to find ways to predict the behaviour of users, personalize information to reduce the traffic load and design the Web site suited for various user groups. Business analysts require tools to study the needs of common users and consumers. All of them expect tools or techniques to help them satisfy their demands and solve their problems encountered

on the Web. Therefore, Web mining has become an active and popular research field, since it helps to retrieve relevant information from the Web.

Web mining is the use of data mining techniques to automatically discover and extract information from the Web. According to Kosala and Blockeel (2000), Web mining consists of the following tasks:

- *Resource finding*: retrieving intended Web documents
- *Information selection and pre-processing*: automatically selecting and pre-processing specific information from retrieved Web resources
- *Generalization:* automatically discovering general patterns at individual Web sites as well as across multiple sites
- *Analysis*: validating and/or interpreting of the mined patterns

Web mining is more complex than WIR, because apart from IR, Web mining includes generalization and analysis (Baeza-Yates 2003).

Resource finding is the process of retrieving the data, that is either online or offline from the electronic newsgroups, newsletters, newswire, libraries and HTML documents that are available as text sources on the Web. Information selection and pre-processing involves selecting the HTML documents and transforming them by removing HTML tags, stop words, stemming etc.

Generalization is the process of discovering general patterns at individual Web sites as well as across multiple sites. Analysis refers to the validation and/or interpretation of the mined patterns. Humans play an important role in the information or knowledge discovery process on the Web, since it is an interactive medium. This is especially important for validation and/or interpretation.

The Web is very large, in the order of terabytes, and is still growing rapidly. It is a huge and effective source for data mining and warehousing with many organisations, individuals and societies using this facility to provide their public information.

Moreover, the Web page contents are much more complex than any other traditional text documents. Today, Web pages lack a standard structure and they contain more complex styles than standardized formats.

Due to the rapid growth and unstructured format of Web, conducting a search has become difficult. In addition to its amazing growth, the Web is dynamic, i.e. the information is updated frequently (Bing 2007). News, stocks and markets, e-commerce sites, company advertisements and web service centres update their pages regularly. The WWW serves a broad diversity of user communities. Web users may have different backgrounds, interests and usage purposes. Due to these reasons, even though the Web is a large repository of information, very often only a small portion of the relevant information is available to the Web user. To summarise, the following characteristics of Web make IR challenging and demanding (da Gomes Jr. and Gong 2005):

- Web is huge.
- Web pages are semi-structured.
- Web information tends to be diverse in meaning.
- Web is dynamic in nature.

These challenges have led to the development of solutions like Database (DB), Information Retrieval (IR), Natural Language Processing (NLP), and Machine Learning along with Web mining for effective IR from Web.

Web Mining Categories

There are three areas of Web mining according to Web data usage utilised as input in the data mining process, namely, Web Content Mining (WCM), Web Usage Mining (WUM) and Web Structure Mining (WSM). WCM is concerned with information retrieval of from the WWW into a more structured form, and indexing the information to retrieve it quickly. WUM is the process of identifying the browsing patterns by analysing the user's navigational behaviour. WSM discovers the model underlying the link structures of the Web pages, catalogues them and generates

information such as the similarity and relationship between them, by taking advantage of their hyperlink topology.

The Web classification (Cooley, Mobasher and Srivastava 1997) is shown in Figure 1.2. Even though there are three Web mining areas, the differences between them are narrowing because they are all interconnected. WCM and WSM are basically used to extract knowledge from the WWW. Web content is concerned with the retrieval of information from WWW into more structured forms. WSM helps to retrieve more relevant information by analysing the link structure. Most researchers now focus on a combination of the three Web mining areas to produce better findings.

Web Content Mining (WCM)

WCM is the process of extracting useful information from Web documents that may consist of text, images, audio, video or structured records like tables and lists. Mining can be applied to the Web documents as well as the result pages produced from a search engine. Two approaches to content mining are the agent based and database approach. The agent based approach concentrates on searching for relevant information, using the characteristics of a particular domain to interpret and organize the collected information. The database approach is used for retrieving the semi-structure data from the Web. WCM has roots in IR and NLP.

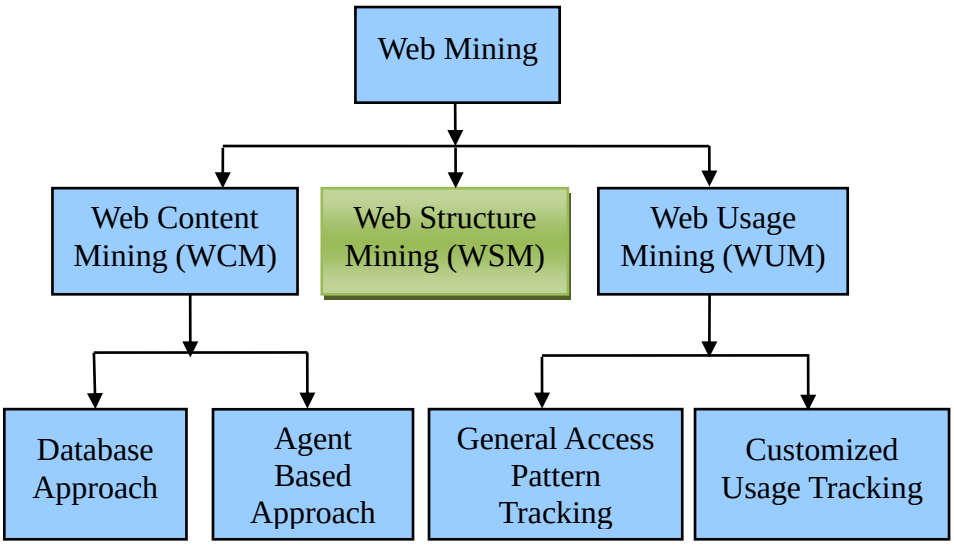


Figure 1.2: Web Classification

Web Usage Mining (WUM)

WUM is the process of extracting useful information from the secondary data derived from the interactions of the user, while surfing on the Web. It extracts data stored in server access logs, referrer logs, agent logs, client-side cookies, user profile and meta data. Analysing such data can help organisations study customers' Web browsing patterns, to facilitate e-commerce specific processing such as customised promotional campaigns, marketing decisions for better strategy and for designing a better Website (Chang et al. 2001).

Web Structure Mining (WSM)

The goal of WSM is to generate the structural summary about the Web site and Web page. It tries to discover the link structure of the hyperlinks at the inter-document level. Based on the topology of the hyperlinks, WSM categorise the Web pages and generates information like similarity and relationship between different Web sites. This type of mining can be performed at the document level (intra-page) or at the hyperlink level (inter-page). In this type of mining, the link structure is represented as a graph, in which Web documents are the nodes and the hyperlinks are the directed edges of the graph. Useful information can be mined by processing the relationship between nodes and edges. The research in this thesis is based on WSM, and analyses the link structure of the Web and the link structure based ranking algorithms.

Search Engines

As the Internet grew, retrieving the relevant information became more difficult. Researchers quickly realized the need for a tool or application to retrieve information from the Internet, and they developed the search engine. These engines are used to download, index and rank Web pages according to keywords, and present them to the user in the form of Search Engine Result Pages (SERPs). Search engines can also be called as Web index servers, and they are the most visited Websites by Internet users (Chang et al. 2001).

The Internet and search engines have changed the life style of digital users. According to Zhang et al., the growth of the Internet follows Moore's law and they theoretically predicted that the Internet doubles every 5.32 years (2008). A recent report from Internetlivestats, an Internet survey company, states that there are

currently 2.9 billion Internet users in the world, and the Internet penetration is only 39% (2014). Hence, there is still a lot of room for the Internet penetration to increase. As the Internet grows, search engines ought to do a lot of work in producing relevant information.

The very first application or tool used for searching in the Internet was, Archie, in 1990 (Seymour, Frontsvog and Kumar 2011). There were many search tools developed in the early 1990s, but the real full text crawler-based search engine is the WebCrawler, introduced in 1994. It was one of the first popularly used search engines and laid the foundation for all the modern search engines. This was followed by Lycos in 1994, and Magellan, AltaVista, Excite, Inktomi, SAPO, Yahoo!, Dogpile, Ask Jeeves in the ensuing years. Many of them like the WebCrawler and Lycos are still active, but a few of them were acquired by AltaVista and Inktomi, which in turn were purchased by Yahoo!. In 1998, Google and MSN, joined the search family, and they became popular due to their superior search technology.

Netmarketshare's latest statistics show the following breakdown in terms of the search engine market share. Google has the largest share at 69.55%, followed by Baidu (Chinese Search Engine) at 16.77%, Yahoo at 6.53%, Bing at 6.18%, AOL at 0.26%, Ask.com at 0.14%, Excite at 0.01% and others at 0.56% (2014). Google is a link structure based search engine which controls the majority of the search engine market share**,** due to its mathematically proven PageRank algorithm and other ranking factors like trust, social and user metrics**.**

The following are a few important search engine categories:

- Crawler-based search engines (Google, ask.com)
- Directory-based search engines (Yahoo!, dmoz.org)
- Hybrid search engines (Google and Yahoo!)
- Meta Search engines (Metacrawler and Dogpile)
- Specialty search engines (Yahoo Shopping, Froogle, Bizrate, Pricegrabber etc.)

These search engines download, index and store hundreds of millions of Web pages continuously, and answer tens of millions of queries every day. Therefore, the Web mining and ranking mechanism has become very important for effective information retrieval. The sample architecture (Duhan, Sharma and Bhatia 2009) of a Web search engine is shown in Figure 1.3.

There are three important components in a search engine. They are Crawler, Indexer and Ranking module. The Crawler is also called a Robot or Spider that traverses the Web and downloads the Web pages. These pages are sent to an indexing module, which parses them and builds the index based on the keywords in those pages. An alphabetical index is generally maintained using the keywords. When a user types a query using keywords on the interface of a search engine, the query processor component matches the query keywords with the index and returns the URLs of the pages to the user. But before presenting the pages to the user, the ranking modules rank all the selected keywords, and present the most relevant pages at the top and less relevant ones at the bottom. This makes the search results navigation easier for users.

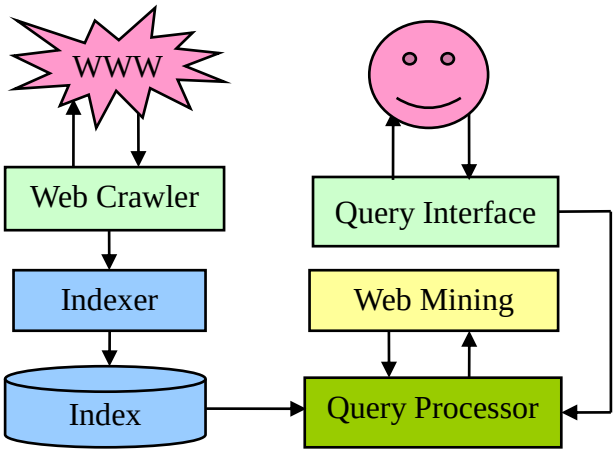


Figure 1.3: Sample Architecture of a Search Engine

Web Spam

Spam can intrude in any information system like e-mail, Web, social, blog or any review forum (Spirin and Han 2011).Web spamming is an activity on the Web where by spammers try to deceive the search engine ranking algorithms, and try to gain a better ranking in the SERPs (Perkins 2001). There are two categories of Web spam

techniques (Gyongyi and Garcia-Molina 2005a). These are *boosting techniques* and *hiding techniques* and their classification is shown as follows in Figure 1.4.

Boosting techniques refer to methods that achieve high relevance or importance for one page; *hiding techniques* refer to methods that do not influence the ranking of search engine but assist boosting techniques. One example is to manipulate the colour scheme of the anchor text. *Boosting techniques* can be further classified into *term spamming* (also called as *content spamming*) and *link spamming*, while *hiding techniques* can be classified into *content hiding*, *cloaking* and *redirection* as shown in Figure 1.4.

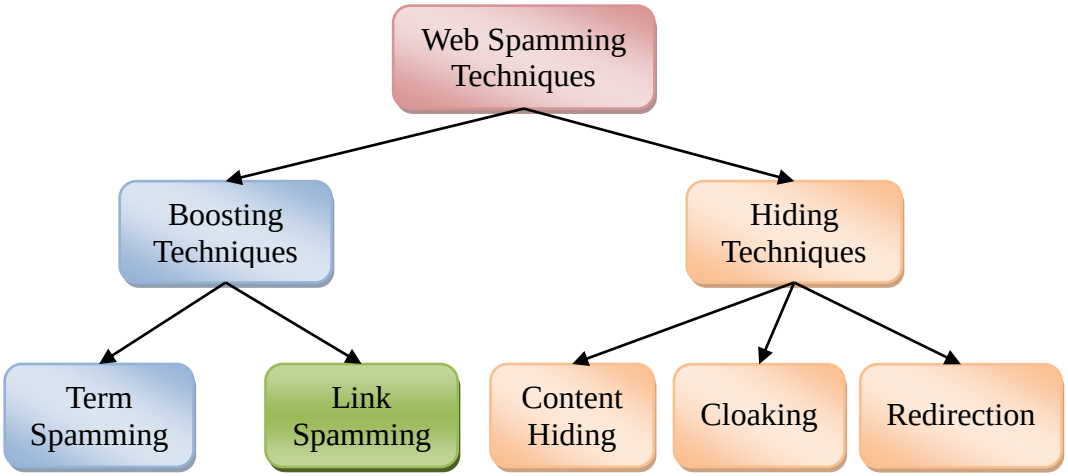


Figure 1.4: Web Spam Techniques Classification

Term spamming or *content spamming* refers to changes in the content of the Web pages, like inserting a large number of keywords in different places of the page (Becchetti et al. 2008; Davison 2000; Drost and Scheffer 2005). In *term spamming*, if excessive keywords are used in the body of the page then it can be called *body spam*; if excessive keywords are used in the title, then it can be called *title spam* and so on.

Link spamming refers to changes in the link structure of the Web sites, by creating *link farms* (Baeza-Yates, Castillo and L´opez 2005; Zhang et al. 2004; Becchetti et al. 2008). A *link farm* is a thickly connected set of pages, created especially for the purpose of deceiving a link structure based ranking algorithm. One of the objectives of this research is to detect link spam formed by Hanging Pages (HP).

Content hiding refers to spam terms or links in a Web page that are invisible to the user, and especially designed for search engines. *Cloaking* refers to giving the Web user different content from what a search engine sees (Wu and Davison 2005). *Redirection* is the process of redirecting the browser to another Uniform Resource Locator (URL), as soon as the page is loaded.

There are many researchers (Castillo et al. 2006; Gyongyi and Garcia-Molina 2005a; Becchetti et al. 2008) who have studied Web spam and developed methods to detect it. They are discussed in Chapter 5.

***Website Optimisation* (WSO)**

Website Optimisation (WSO) or Search Engine Optimisation (SEO) is a process of making a Website friendly and easy to navigate for users as well as search engine robots, so that the Website will get more traffic and its rank will improve in an organic way in the SERPs (Kumar, Singh, and Mohan 2013). Even though SERPs produces hundreds of pages for a particular query, the users only look into the first two or three pages, thus resulting in huge competition among the commercial companies to appear in the top of the SERPs. This competition leads the Web masters or WSO professionals to use Black Hat techniques in WSO. Black Hat is a Web optimisation term referring to illegal spam techniques that are used to achieve higher than deserved rankings in SERPs. On the other hand, White Hat techniques use good and legitimate methods to achieve better rankings for Websites. Web masters or Web developers should therefore, use White Hat techniques to develop Websites from scratch to achieve better ranking.

Basics of WWW and search engine are covered in this chapter because this thesis used crawler to download data from WWW for experiments and helps to improve Search Engine Result Pages (SERPs). Information Retrieval (*IR*) is also described here because this thesis is based on the concept of the *IR*. This thesis uses the concept of Web structure mining hence Web mining is described in this chapter. Web spam is explored in this chapter because this thesis helps to combat link spam that occurs due to hanging pages. Finally, WSO is introduced in this chapter because this thesis studies the effect of hanging pages in WSO.

1.2 PROBLEM STATEMENT

Hanging pages are one of the hidden problems in link structure based ranking algorithms because:

- They do not propagate the rank scores to other pages (important function of link structure based ranking algorithms).
- They can be compromised by spammers to induce link spam in the Web.
- They can affect Website optimization and the performance of link structure based ranking algorithms.

In the Web, when a page does not have any forward or outgoing links, then that page can be called as hanging page/dangling page/zero-out link page/dead end page. For uniformity and consistency reason, 'hanging page' is being used throughout this thesis.

1.3 RESEARCH MOTIVATION

One of the problems with the current search on the Internet is that the hanging pages are not included in the ranking process and the relevant hanging pages do not reflect the correct ranking order in the SERPs.

Hanging pages can be manipulated by spammers to form link spam in the link structure based ranking algorithms. Also hanging pages can affect the optimization of Web sites. All the above problems have raised the need to handle the hanging pages and develop an efficient algorithm to solve the problems of hanging pages.

Motivation for this study is to achieve deserved ranking for the relevant hanging pages and to handle the spam induced by hanging pages. This will be useful for the Web users who are looking more relevant information and the researchers working in this area.

1.4 RESEARCH GOALS

The main objectives of this research study are four fold. The first one is the handling of Hanging Pages (HP) in the link structure based ranking algorithms. PageRank is used as the base algorithm throughout this research. A detailed study on different hanging pages and how they affect the rank of neighbouring pages is done. Various methodologies are proposed to handle hanging pages in the link structure based ranking algorithms, especially for the PageRank algorithm.

The second objective is to study the relevancy of hanging pages using the Hanging Relevancy Algorithm (HRA) to produce fair and relevant ranking results. Experiments are conducted with the dataset and the results are compared with the original PageRank algorithm.

The third objective is to analyse the effect of hanging pages in link spam contribution and develops the Link Spam Detection (LSD) algorithm. Methods are proposed to detect the link spam contributed by hanging pages.

Finally, the fourth objective is to study the effect of hanging pages in Search Engine Optimization (SEO) and propose methods to improve Web Site Optimization (WSO). This thesis is organized as follows:

1.5 RESEARCH METHODOLOGY

For the above mentioned research goals, the following methodologies are adopted.

- Web Graph is implemented where nodes are treated as Web pages and edges between nodes are treated as hyperlinks.
- PageRank algorithm is simulated, matrix interface as well as graph interface is created and the ranks of Web pages are computed.
- Experiments are conducted to show the effects of hanging pages and methods are proposed to handle hanging pages in ranking Web pages.
- Hanging Relevancy Algorithm (HRA) is implemented and only the relevant hanging pages are included in the rank computation to reduce the complexity.

- Link spam detection (LSD) algorithm is implemented to detect link spam contributed by hanging pages.
- Experiments are conducted to show the effect of hanging pages in Search Engine Optimization (SEO) and factors are proposed to build optimized Web sites.

1.6 THESIS ORGANISATION

Chapter 1 introduces the background information related to this thesis especially, WWW in general, Information Retrieval (IR) in the Web, Web mining, Web spam and Website Optimization (WSO). It also introduces the problem statement, research motivation, research goals and the methodologies used in this research.

In Chapter 2, a detailed literature survey on the related research is provided. First, a comparative study of link structure based ranking algorithms and in particular, PageRank (PR) algorithm (Brin and Page 1998) used by the Google Search engine is done. Hanging pages are introduced and the related work on hanging pages is described. Thereafter, Web spam and the related work on Web spam are described. Next, Website optimization (WSO) is introduced; their challenges are explored and the stages of WSO are described in detail. Preliminaries and the mathematical definitions used in this research are also described in this chapter. After that, three large publicly available datasets – WEBSPAM-UK2006, WEBSPAM-UK2007 and EU2010 are introduced and the percentage of hanging and non-hanging pages are computed and provided. The parameters' settings for all the algorithms are presented. This chapter concludes with a simulation of PageRank program and Weighted PageRank (WPR) (Xing and Ghorbani 2004) and the results are compared.

Chapter 3introduces the problems of hanging pages in link structure based ranking algorithms and propose two methods (Method 1 and Method 2) to handle hanging pages. The PageRank algorithm is modified according to the proposed methodologies, experiments carried out using the dataset and the results compared with the original PageRank algorithm. Methods 1 and 2 produced fair ranking results by producing a decent rank for the hanging pages when compared with the PageRank algorithm, but then both Methods 1 and 2 took more iteration to converge. Method 1

took 36 iterations to converge and Method 2 took 95 iterations to converge. On handling hanging pages, Method 1 performed better than the standard PageRank algorithm and Method 2 by producing fair and decent ranks for hanging pages.

Chapter 4 introduces the Hanging Relevancy Algorithm (HRA) to determine the relevancy of hanging pages in the link structure based ranking algorithms. As more and more meaningful hanging pages keep increasing in the Web, their relevancy has to be determined according to keywords or query terms to make the SERPs fair and relevant. Exclusion of these pages in ranking calculation can give biased/inconsistent results. On the other hand, inclusion of these pages will reduce the speed significantly. However most of the IR ranking algorithms exclude the hanging pages. But there are relevant and important hanging pages on the Web and they cannot be just ignored. In the proposed methodology, Anchor Text (AT) is used to determine the relevancy of hanging pages against keywords or query terms and stability analysis is done to show the rank results are consistent before and after altering the link structure. PageRanks are first computed without the hanging relevancy function and then with hanging relevancy function. The latter produced fair and relevant results compared with the former. This method compromises between complexity and relevancy. It has slow down the ranking process due to the query dependent approach, but it produces fair ranking results by including only the relevant hanging pages.

Chapter 5 proposes Link Spam Detection (LSD) algorithm to detect the link spam contributed by hanging pages. Link spammers are constantly seeking new methods and strategies to deceive the search engine ranking algorithms. Search engines need to come out with new methods and approaches to challenge the link spammers to maintain the integrity of the ranking algorithms. Here, a target page is selected randomly and link spam is induced. PageRank program is applied to the induced link spam structure and the ranks are computed. The experiment showed that there was a considerable improvement in the PageRank of the induced link spam structure. Also methods are proposed to detect link spam using eigenvectors and eigenvalues. Another important finding in this study is the significant role played by the hanging pages in forming irreducible closed subsets. Experiments were done using live data from the Internet. One of the top 10 Websites, Amazon.com is selected and the pages

downloaded using a Crawler program developed in MATLAB. PageRank program is applied to the pages before and after link spam. The experiments clearly show that hanging pages do contribute to link spam. Also, second eigenvector and eigenvalues are computed for the downloaded pages using the Markov analysis. The second eigenvector has detected the link spam contributed by hanging pages in the form of irreducible closed subset.

Chapter 6 explores the problems of hanging pages in optimizing a Website. Hanging pages can affect the WSO process, especially for the link structure based search engine ranking algorithms like PageRank, HITS and SALSA. This chapter first analyses the effect of hanging pages in Website optimization. Next, this chapter suggests methods to improve the ranking of Web pages through analysis and simulation. Experiments are done using live Internet data. Programs are created to crawl the Web and the pages are downloaded. Experiments are conducted on the downloaded pages to produce *back link* and *broken link* analysis. Comparisons of *followed* and *no-followed* links are carried out and On-Site and Off-Site ranking factors computed for the Curtin University (Sarawak) Web site. Also, methods are suggested to improve the On-Site and Off-Site ranking factors including hanging pages.

Finally, Chapter 7 summarises the results of this research study, discusses the implications and concludes with a few recommendations for future research.

Chapter 2 Literature Review

2.1 INTRODUCTION

The review of previous studies in the area of Link structure based ranking algorithms, hanging pages, Web spam and Search Engine Optimisation are presented here. With the rapid growth of WWW and the users' demand for knowledge, it has become more difficult to manage information on the WWW and satisfy user needs. Users are looking for better IR techniques and tools to locate, filter and extract the necessary information. Most of them use IR tools like search engines to find information from the WWW. Generally, many Web users do not see beyond the top few pages of the search results (Broder 2002; Jansen et al. 1998; Silverstein et al. 1999). Therefore, search engines need to produce the relevant results within the top few pages, or they will decline in popularity. According to Borodin et al., Web users are not only looking for relevant information also but also for *authoritative* sources, i.e. trusted sources of correct and authentic information, like getting the information direct from the home page of a company (2005). Hence, in current Web searches, there is a shift from *relevance* to *authoritativeness*, and the main task of the search engine ranking algorithms have also shifted to finding and ranking the more authoritative Web documents.

With the above shift from relevancy to authoritativeness, the link structure of the Web plays a very important role in sourcing for authoritative documents. Through the hyperlink structure, the Web offers a rich context of information. Here, a link from page *a* to *b* denotes an endorsement for the quality of page *b*. Therefore, the Web can be imagined as a network of recommendations which contains information about the authoritativeness of the pages. Based on this concept, Kleinberg (1999a) and Brin and Page (1998) introduced the HITS and PageRank link analysis algorithms, where hyperlink structures are used to rank Web pages.

The HITS algorithm collects the Web pages using the query dependent method, while the PageRank algorithm collects Web pages using the query independent method.

While, the former became popular in the research field due to its methodology, the latter became popular in the research as well as commercial areas due to its efficiency. Soon after the success of the HITS and PageRank algorithms, researchers developed many derivatives of both algorithms. The HITS and PageRank algorithms and their important derivatives are described in the next section.

Important link structure based ranking algorithms are introduced and compared here, particularly the PageRank (PR) algorithm (Brin and Page 1998) used by the Google Search engine. Apart from PageRank, other link structure based ranking algorithms are discussed and compared. These include the Weighted PageRank (WPR) (Xing and Ghorbani 2004), Hyperlink-Induced Topic Search (HITS) (Kleinberg 1999a), Stochastic Approach for Link Structure Analysis Algorithm (SALSA) (Lempel and Moran 2001), DistanceRank (Zareh Bidoki and Yazdani 2008) and DirichletRank algorithms (Wang et al. 2008). Ranks are calculated for PageRank and Weighted PageRank algorithms for a given hyperlink structure. For the purposes of this research study, a PageRank program was developed to analyse the properties of the link structure based ranking algorithms, especially the PageRank algorithm.

2.2 LINK STRUCTURE BASED RANKING ALGORITHMS

With the increasing number of Web pages and users on the Web, the number of queries submitted to the search engines are also increasing rapidly. Therefore, search engines needs to be more efficient. Web mining techniques are employed by search engines to extract relevant documents from the Web database and provide the necessary information to the users. The search engines become very successful and popular if they use efficient ranking mechanisms. The Google search engine is very successful because of its PageRank algorithm. Such algorithms are used by the search engines to present the search results by considering the relevance, importance and content score; they also use Web mining techniques to order the search results according to the user interest. Some ranking algorithms depend only on the link structure of the documents, i.e. their popularity scores (WSM), whereas others look for the actual content in the documents (WCM).Some, however, use a combination of both i.e. they use the document content as well as the link structure to assign a rank value for a given document (Singh and Kumar 2009; Kumar and Singh 2010). If the search results are not displayed according to the user interest, then the search engine

will lose its popularity. Thus, the ranking algorithms have become very important. Some of the popular link structure based ranking algorithms are discussed in the following section.

2.2.1 *Citation Analysis*

Link analysis is similar to social networks and citation analysis. The citation analysis was developed in information science as a tool to identify core sets of articles, authors, or journals of a particular field of study. The "Impact factor" developed by Eugene Garfield, is used to measure the importance of a publication(Garfield 1972).It takes into account the number of citations received by a publication, and is proportional to the total number of citations a publications has. This measurement treats all the references equally. Important references which are regularly referred to, however, would be given additional weight. Pinski and Narin (1976) proposed a model to overcome this problem called "influence weights", where the weight of each publication is equal to the sum of its citations, scaled by the importance of these citations. The influence weight (W) of the i^{th} unit is given in Equation 2.1.

$$W_i = \sum_{k=1}^{n} \frac{W_k C_{ki}}{S_i} \tag{2.1}$$

W_i is the influence weight of the i^{th} unit, where S_i is the total number of references from the i^{th} unit to other units. C corresponds to the citation matrix. In the sum, the number of cites to the i^{th} unit from the k^{th} unit is weighted by the weight of k^{th} (referencing) unit.

If a research article receives citations from one or more other research articles, then it is called a *backward citation* and if it issues citations to other research articles, then it is called a *forward citation*. Figure 2.1 below shows a backward citation, where article *A* is cited by articles *B, C* and *D*.

The same principle is applied to the Web for ranking the web pages, where the notion of citations corresponds to the links pointing to a Web page. This simplest ranking of a Web page could be done by summing up the number of links pointing to it. Here, it would favour only the most popular Web sites, such as universally known portals,

news pages, news broadcasters etc. In the Web, the page quality and the content diversity should also be considered.

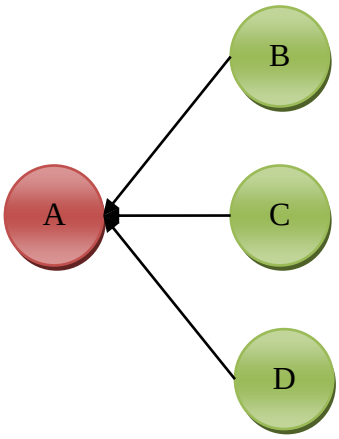


Figure 2.1: Backward Citation

Figure 2.2 shows the forward citation where article *A* is citing articles *B*, *C* and *D*.

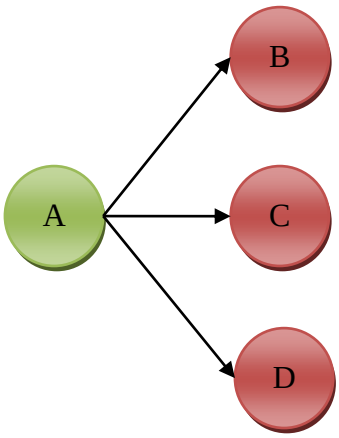


Figure 2.2: Forward Citation

The same principle is applied to the Web for ranking the web pages, where the notion of citations corresponds to the links pointing to a Web page. This simplest ranking of a Web page could be done by summing up the number of links pointing to it. Here, it would favour only the most popular Web sites, such as universally known portals, news pages, news broadcasters etc. In the Web, the page quality and the content diversity should also be considered.

These “hyperlinked communities that appear to span a wide range of interests and disciplines”, are called "Web communities" (Gibson et al. 1998) and the process of identifying them is termed as “trawling”, (Kumar et al. 1999).There are a number of proposed algorithms based on the link analysis. Using Citation analysis, Co-citation algorithm (Dean et al. 1999) and Extended Co-citation algorithm (Hou et al. 2003) were proposed. However, these algorithms are simple and more significant relationships among the pages cannot be discovered.

The following link structure based algorithms which are more complex, address the relationship problems faced by citation algorithms. Six link structures based ranking algorithms, PageRank (PR) (Brin and Page 1998), *Weighted PageRank* (*WPR*) (Xing and Ghorbani 2004), HITS (Kleinberg 1999a), *DistanceRank* (Zareh Bidoki and Yazdani 2008), *DirichletRank* (Wang et al. 2008) and SALSA algorithms (Lempel and Moran 2001) are discussed in detail below.

2.2.2 *PageRank Algorithm*

Brin and Page (1998) developed the PageRank (*PR*) algorithm used by Google based on the citation analysis. They applied the citation analysis in a Web search by treating the incoming links as citations to the Web pages. However, simply applying the citation analysis techniques to the diverse set of Web documents did not result in efficient outcomes. Therefore, the PageRank provides a more advanced way to compute the importance or relevance of a Web page, than just counting the number of pages that are linking to it (called as “backlinks”). If a backlink comes from an “important” page, then that backlink is given a higher weighting than those backlinks from non-important pages. In a simple way, a link from one page to another may be considered a vote. However, not only are the number of votes a page receives considered important, but the “importance” or the “relevance” of the ones that cast these votes is important as well.

The PageRank computation is illustrated below: Assume any arbitrary page, *A*, has pages T_1 to T_n pointing to it (incoming link). PageRank can be calculated using Equation 2.2.

$$PR(A) = (1-d) + d(PR(T_1)/C(T_1) + ... + PR(T_n/C(T_n)) \quad (2.2)$$

The parameter *d* is a damping factor, usually set at 0.85 (Brin and Page 1998) to stop the other pages from having too much influence, this total vote is "damped down" by multiplying it by 0.85. *C(A)* is defined as the number of links going out of page *A*. The PageRanks form a probability distribution over the Web pages, so the sum of all Web pages' PageRank will be one. PageRank can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix of the Web.

PageRank is displayed on the toolbar of the browser if the Google Toolbar is installed. The Toolbar PageRank goes from 0 – 10, like a logarithmic scale with 0 as the low page rank and 10 as the highest page rank. The PageRank of all the pages on the Web changes every month when Google does its re-indexing. Apart from PageRank algorithm, Google uses as many as 200 factors to rank a Web page.

2.2.3 *Weighted PageRank Algorithm*

Xing and Ghorbani (2004) proposed a Weighted PageRank (*WPR*) algorithm, which is an extension of the PageRank algorithm. This algorithm assigns larger rank values to the more important pages, rather than dividing the rank value of a page evenly among its outgoing linked pages. Each outgoing link gets a value proportional to its importance. The importance is assigned in terms of weight values to the incoming and outgoing links and are denoted as $W^{in}(m, n)$ and $W^{out}(m, n)$ respectively. $W^{in}(m, n)$, as shown in Equation 2.3, is the weight of *link*(*m*, *n*) calculated based on the number of incoming links of page *n*, and the number of incoming links of all reference pages of page *m*.

$$W^{in}_{(m,n)} = \frac{I_n}{\sum_{p \in R(m)} I_p} \quad (2.3)$$

$$W^{out}_{(m,n)} = \frac{O_n}{\sum_{p \in R(m)} O_p} \quad (2.4)$$

where I_n and I_p are the number of incoming links of page *n* and page *p* respectively, $R(m)$ denotes the reference page list of page *m*. $W^{out}(m, n)$ is as shown in Equation 2.4. The weight of $link(m, n)$ is calculated based on the number of outgoing links of page *n* and the number of outgoing links of all reference pages of *m*, where O_n and O_p are the number of outgoing links of page *n* and *p* respectively. The formula, which is a modification of the PageRank formula, as proposed by Xing and Ghorbani (2004) for the *WPR* is as shown in Equation 2.5.

$$WPR(n) = (1-d) + d \sum_{m \in B(n)} WPR(m) W^{in}_{(m,n)} W^{out}_{(m,n)} \tag{2.5}$$

To differentiate the *WPR* from the PageRank, Xing and Ghorbani (2004), categorized the resultant pages of a query into four categories based on their relevancy to the given query: They are:

1. *Very Relevant* Pages(*VRP)*: pages that contain very important information related to a given query
2. *Relevant* Pages(*RP*): pages are relevant but do not have important information about a given query
3. *Weak Relevant* Pages(*WRP*): pages may have the query keywords but do not have the relevant information
4. *Irrelevant* Pages(*IRP*): pages do not have any relevant information and query keywords

The PageRank and *WPR* algorithms both provide ranked pages in the sorting order, to users based on the given query. Therefore, in the resultant list, the number of relevant pages and their order are very important for users. Xing and Ghorbani proposed a *Relevance Rule* (2004) to calculate the relevancy value of each page in the list of pages. That makes *WPR* different from PageRank.

Relevancy Rule (RR): The Relevancy Rule is as shown in Equation 2.6. The Relevancy of a page to a given query depends on its category and its position in the

page-list. The larger the relevancy value, the better is the result.

$$k = \sum_{i \in R(p)} (n - i) * W_i \tag{2.6}$$

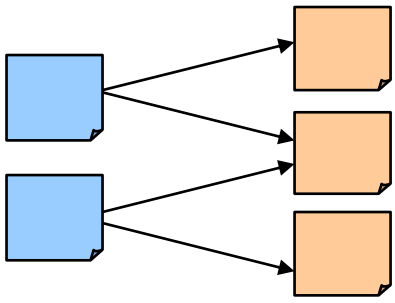
Where i denote the i^{th} page in the result page-list $R(p)$, n represents the first n pages chosen from the list $R(p)$, and W_i is the weight of i^{th} page as given below in Equation 2.7.

$$W_i = (v1, v2, v3, v4) \tag{2.7}$$

Where$v1$, $v2$, $v3$ and $v4$ are the values assigned to a page if the page is *VR*, *R*, *WR* and *IR* respectively. The values are always $v1 > v2 > v3 > v4$. Experimental studies by Wenpu et al. showed that *WPR* produces larger relevancy values than the PageRank.

2.2.4 *The HITS Algorithm - Hubs and Authorities*

Kleinberg (1999a) identifies two different forms of Web pages called *Hubs* and *Authorities*: the former refers to pages with important contents, while the latter are pages that act as resource lists, guiding users to authorities. Thus, a good hub page for a subject points to many authoritative pages on that content, and a good authority page is pointed by many good hub pages on the same subject. Hubs and Authorities are shown in Figure 2.3. Kleinberg says that a page may be a good hub and a good authority at the same time. This circular relationship leads to the definition of an iterative algorithm called HITS.



Hubs Authorities

Figure 2.3: Hubs and Authorities

The HITS algorithm treats WWW as a directed graph $G(V, E)$, where V is a set of Vertices representing pages and E is a set of edges that correspond to links.

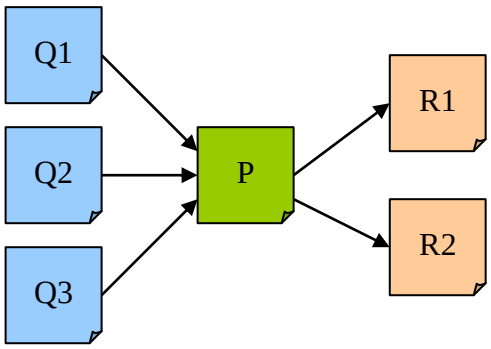
2.2.4.1 *HITS Methodology*

There are two major steps in the HITS algorithm. The first step is the *Sampling Step* and the second step is the *Iterative step*. In the *Sampling step*, a set of relevant pages for the given query are collected i.e. a sub-graph S of G is retrieved which is high in authority pages. This algorithm starts with a root set R, obtains a set of S (keeping in mind that S is relatively small), rich in relevant pages about the query and contains most of the good authorities. The second step, *Iterative step*, finds hubs and authorities using the output of the sampling step using Equations 2.8 and 2.9.

$$H_p = \sum_{q \in I(p)} A_q \tag{2.8}$$

$$A_p = \sum_{q \in B(p)} H_q \tag{2.9}$$

Where H_p is the hub weight, A_p is the Authority weight, $I(p)$ and $B(p)$ denotes the set of reference and referrer pages of page p. The page's authority weight is proportional to the sum of the hub weights of pages that it links to (Kleinberg 1999b). Similarly, a page's hub weight is proportional to the sum of the authority weights of pages that it links to. Figure 2.4 shows an example of the calculation of authority and hub scores.



$A_P = H_{Q1} + H_{Q2} + H_{Q3} H_P = A_{R1} + A_{R2}$

Figure 2.4: Calculation of Hubs and Authorities

2.2.4.2 *Constraints of HITS*

The following are the constraints of the HITS algorithm (Chakrabarti et al. 1999):

- *Hubs and Authorities*: It is not easy to distinguish between *Hubs* and *Authorities* because many sites are both.
- *Topic drift*: Sometime HITS may not produce the most relevant documents for the user queries because of equivalent weights.
- *Automatically generated links:* HITS gives equal importance for automatically generated links which may not produce relevant topics for the user query.
- *Efficiency*: HITS algorithm is not efficient in real time.

The HITS was used in a prototype search engine called Clever (Chakrabarti et al. 1999) for an IBM research project. Because of the above constraints HITS could not be implemented in a real time search engine.

2.2.5 *SALSA Algorithm*

The SALSA algorithm (Stochastic Approach for Link Structure Analysis), proposed by Lempel and Moran (2001)**,** is another link structure based ranking algorithm, that combines the best features from both the PageRank and HITS algorithms. The SALSA algorithm performs a random walk on the hub and authorities of the bipartite graph by alternating between the hub and authority sides. The random walk starts from an authority node, selected uniformly at random and continues by alternating between forward and backward steps. The stochastic matrices for both the hub and authority are shown below:

$$\tilde{h}_{i,j} = \sum_{\{k|(i_h,k_a),(j_h,k_a)\in\tilde{G}\}} \frac{1}{de(i_h)} \cdot \frac{1}{de(k_a)} \qquad (2.10)$$

In Equation 2.10 shown above, $\tilde{h}$ is the hub matrix, $\tilde{G}$ is the authority Markov chain and *de* is the degree of a page. The authority matrix is given below in Equation 2.11.

$$\tilde{a}_{i,j} = \sum_{\{k|(k_h,i_a),(k_{hh},j_a)\in\tilde{G}\}} \frac{1}{de(i_a)} \cdot \frac{1}{de(k_h)} \qquad (2.11)$$

In Equation 2.11, $\tilde{a}$ is the authority matrix. A positive transition probability $\tilde{a}_{i,j} > 0$ implies that a certain page *h* points to both pages *i* and *j*, and hence, page *j* is reachable from page *i* in two steps: retracting along the links $h \rightarrow i$ and then following the link $h \rightarrow j$. The algorithm selects one of the incoming links uniformly at random at the authority node side of the bipartite graph and moves on to a hub node on the hub side. The algorithm selects one of the outgoing links uniformly at random, at the hub node on the hub side of the bipartite graph and moves on to an authority. The authority weights are defined as stationary distribution of this random walk. SALSA is a variation of the HITS algorithm.

2.2.6 *DistanceRank Algorithm*

DistanceRank algorithm proposed by Zareh Bidoki and Yazdani (2008) is a novel recursive method based on reinforcement learning, (Sutton and Barto 1998) which considers distance between pages as punishment, called "DistanceRank" to compute ranks of web pages. The number of 'average clicks' between the two pages is defined as distance. The main objective of this algorithm is to minimize distance or punishment, so that a page with smaller distance can have a higher rank.

Most of the current ranking algorithms have the "rich-get-richer" problem (Cho, Roy and Adams 2005) i.e. the popular high rank web pages become more and more popular and the young high quality pages are not picked by the ranking algorithms. Zareh Bidoki and Yazdani suggested DistanceRank to solve the "rich-get-richer" problem (2008). Cho, Roy and Adams proposed to overcome the "rich-get-richer" problem using Page Quality function (2005). The DistanceRank algorithm is less sensitive to the "rich-get-richer" problem, and finds important pages faster than others. This algorithm is based on the reinforcement learning such that the distance between pages is treated as a punishment factor. Normally related pages are linked to each other so the distance based solution can find pages with high qualities more quickly.

In the PageRank algorithm, the rank of each page is defined as the weighted sum of ranks of all pages having back links or incoming links to the page. A page has a high rank if it has more back links from high page ranks. These two properties are true for DistanceRank also. A page that has many incoming links should have low distance, and if the pages pointing to it have low distance, then subsequently, this page should have a low distance. The above point is clarified using the following definition.

Definition 2.1: If page *a* points to page *b*, then the weight of the link between *a* and *b* is equal to $\text{Log}_{10}O(a)$, where $O(a)$ shows *a*'s out degree or outgoing links.

Definition 2.2: The distance between two pages *a* and *b* is the weight of the shortest path (the path with the minimum value) from *a* to *b*. This is called *logarithmic distance* and is denoted as d_{ab}.

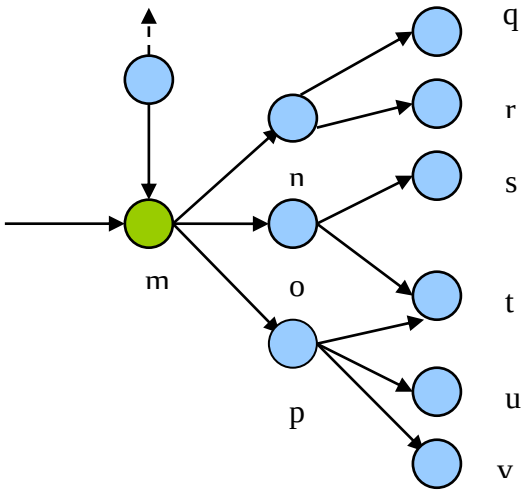


Figure 2.5: A Sample Graph

For example, in Figure 2.5, the weight of out-links or outgoing links in pages *m*, *n*, *o* and *p* is equal to log(3), log(2), log(2) and log(3) respectively, and the distance between *m* and *t* is equal to log(3) + log(2), if the path *m*—*o*—*t* was the shortest path between *m* and *t*. The distance between *m* and *v* is log(3) + log(3) as shown in Figure2.5, even though both *t* and *v* are in the same link level from *m* (two clicks), but *t* is closer to *m*.

Definition 2.3: If d_{ab} shows the distance between two pages *a* and *b* as Definition 2.2, then d_b denotes the average distance of page *b* as shown in Equation 2.12, where *V* shows the number of web pages:

$$d_b = \frac{\sum_{a=1}^{v} d_{ab}}{V} \tag{2.12}$$

In this definition, the researchers used an average click instead of the classical distance definition. The weight of each link is equal to log(*O*(*a*)). If there is no path between *a* and *b*, then d_{ab} will be set a big value. In this method after the distance computation, pages are sorted in the ascending order and pages with smaller average distances will have high ranking.

This method is dependent on the out degree or outgoing links of nodes in the web graph like other algorithms. Apart from that, it also follows the web graph like the random-surfer model (Brin and Page 1998) used in the PageRank, in that each output link of page *a* is selected with probability 1/*O*(*a*). That is, the rank's effect of *a* on page *b* as the inverse product of the out-degrees of pages in the logarithmic shortest path between *a* and *b*. For example, if there is the logarithmic shortest path with single length 3 from *a* to *b* like *a*→*c*→*d*→*b*, then *a*'s effect on *b* is (1/*O*(*a*)) * (1/*O*(*c*)) * (1/*O*(*d*)) * (1/*O*(*b*)). In other words, the probability that a random surfer started from page *a* to reach page *b* is (1/*O*(*a*)) * (1/*O*(*c*)) * (1/*O*(*d*)) * (1/*O*(*b*)).

If the distance between *a* and *b*, d_{ab} is less than the distance between *a* and *c*, d_{ac} then *a*'s rank effect, r_{ab} on *b* is more than on *c*, i.e if $d_{ab}<d_{ac}$ the $r_{ab}>r_{ac}$. In other words, the probability that a random surfer reaches *b* from *a* is more than the probability to reach from *c*.

The purpose of the DistanceRank is to compute the average distance of each page and there is a dependency between the distance of each page and its incoming links or back links. For example, if page *b* has only one back link and it is from page *a*, the average distance for page *b*, d_b is as follows in Equation 2.13.

$$d_b = d_a + \log(O(a)) \tag{2.13}$$

In general, suppose $O(a)$ denotes the number of forwarding or outgoing links from page a and $B(b)$ denotes the set of pages pointing to page b. The *DistanceRank* of page b denoted by d_b is given as follows in Equation 2.14.

$$d_b = \min(d_a + \log O(a)), a \in B(b) \tag{2.14}$$

The distance d_t from Figure 2.5 is calculated as follows.

$d_t = \min\{d_o + \log2, d_p + \log3\} = \min\{d_m + \log3 + \log2, d_m + \log3 + \log3\} = \{d_m + \log3 + \log2\} = d_m + 0.77$.

According to the authors, the DistanceRank is similar to PageRank in ranking pages. Using Equation 2.14, the authors proposed the following formula shown in Equation 2.15 based on the Q-learning, a type of reinforcement learning algorithm (Sutton and Barto 1998) to compute the distance of page b (a links to b).

$$d_{b_{t+1}} = (1-\alpha) * d_{b_t} + \alpha * \min(\log(O(a)) + \gamma^* d_{a_t}), a \in B(b), \\ 0 < \alpha \leq 1, 0 \leq \gamma \leq 1 \tag{2.15}$$

Where α is learning rate and $\log(O(a)$ is the instantaneous punishment it receives in transition state from a to b. d_{b_t} and d_{a_t} show distance of page b and a in time t respectively and $d_{b_{t+1}}$ is distance of page b at time $t + 1$. In other words, the distance of page b at time $t + 1$ depends on its previous distance, its further distance (d_a) and $\log(a)$, the instantaneous punishment from selection page b by the user. The discount factor γ is used to regulate the effects of the distance of pages in the path leading to page b on the distance of page b. For example, if there is a path $m \rightarrow n \rightarrow o \rightarrow p$, then the effect of the distance of m on o is regulated with a γ factor. In this fashion, the sum of received punishments is going to decrease. Since Equation 2.15 is based on the reinforcement learning algorithm, it will converge finally and reach the global optimum state (Sutton and Barto 1998).

Equation 2.16, below, shows the learning rate α, where *t* shows time or iteration number and β is a static value to control regularity of the learning rate. If the learning rate is properly adjusted, the system will converge and reach the stability state very fast with a high throughput. In the beginning the distances of pages are not known, so initially α is set to one and then decreases exponentially to zero.

$$\alpha = e^{-\beta * t} \tag{2.16}$$

According to the authors, the user is an agent surfing the web randomly and in each step it receives some punishment from the environment. The goal is to minimize the sum of punishments. In each state, the agent has some selections, next pages to click, and the page with the minimum received punishment will be selected as the next page for visiting. With that Equation 2.14 can be modified as follows:

d_b = α * (previous punishment of selecting *b*) + (1- α) * (current punishment + instantaneous punishment that user will receive from selection *b*),

So d_b is the total punishment an agent receives from selection page *b*.

This system tries to simulate the real user surfing the web. When a user starts browsing a random page, he/she does not have any background about the web. Then, by browsing and visiting web pages, he/she clicks links based on both the current status of web pages and the previous experiences. As the time goes on, the user gains knowledge in browsing and gets the favourite pages faster. DistanceRank uses the same kind of approach like a real user: it initially sets α = 1 and after visiting more pages and getting more information, α decreases and effectively selects the next pages.

The DistanceRank is computed recursively like PageRank as shown in Equation 2.15. The process iterates to converge. It is possible (Zareh Bidoki and Yazdani 2008) to compute distances with $O(p * |E|)$ time complexity when $p<<V$, which is very close to an ideal state. For instance, *p* is 7 for 7 million pages implying that 7 iterations are enough for an acceptable ranking.

After convergence, the DistanceRank vector is produced. Pages with low DistanceRank will have high ranking and are sorted in the ascending order. The authors used two scenarios for experimental purposes. One is crawling scheduling and the other is rank ordering. The objective of the crawling scheduling is to find more important pages faster. In the rank ordering, DistanceRank is compared with PageRank and Google's rank with and without respect to a user query.

Based on the experimental results done by the authors, the crawling algorithms used by the DistanceRank outperforms (Zareh Bidoki and Yazdani 2008)other algorithms like Breadth-first, Partial PageRank, Back-Link and OPIC (Online Page Importance Computing) in terms of throughput. That is, DistanceRank finds high important pages faster than other algorithms. Also, on the rank ordering, DistanceRank was better than PageRank and Google. The results of DistanceRank are closer to Google than PageRank.

2.2.6.1 *DistanceRank and Ranking Problems*

One of the main problems in the current search engines is the "rich-get-richer" problem that causes the new high quality pages to receive less popularity. To research this problem further, Cho and Roy (2004) proposed two models on how users discover new pages. The Random-Surfer finds new pages by surfing the web randomly without the help of search engines, while the Search-Dominant model searches for new pages by using search engines. The authors found out that it takes 60 times longer for a new page to become popular under the Search-Dominant model than Random-Surfer model. If a ranking algorithm can find new high quality pages and increase their popularity earlier (Cho, Roy and Adams 2005), then that algorithm is less sensitive to the "rich-get-richer" problem. That is, the algorithms should predict the popularity that the pages would get in the future.

The DistanceRank algorithm is less sensitive to the "rich-get-richer" problem and provides good prediction of pages for future ranking. The convergence speed of this algorithm is fast with less iteration. In DistanceRank, it is not necessary to change the web graph for computation. Therefore, some parameters like the damping factor can be removed and one can work on the real graph.

2.2.7 *DirichletRank Algorithm*

The DirichletRank algorithm proposed by Wang et al. (2008) eliminates the zero-one gap problem found in the PageRank algorithm, proposed by Page et al. (1999). The zero-one gap problem occurs due to the current ad hoc way of computing transition probabilities in the random surfing model. The authors suggested the DirichletRank algorithm, which calculates the probabilities using the Bayesian estimation of Dirichlet prior. This zero-one gap problem can be exploited to spam PageRank results and make the state-of-art link-based anti-spamming techniques ineffective. DirichletRank is a form of PageRank and the authors have shown that the DirichletRank algorithm is free from the zero-one gap problem. They have also proved that this algorithm is more robust against several common link spams and is more stable under link perturbations. The authors also claim that this is as efficient as PageRank and it is scalable to large-scale web applications.

Everybody wants their pages to be on the top of the search results. This leads to the Web Spamming, (Gyongyi and Garcia-Molina 2005a) which is a method to maliciously induce bias to the search engines, so that certain target pages will be ranked much higher than they deserve. Consequently, it leads to poor quality of search results and in turn will reduce the search engine reliability.

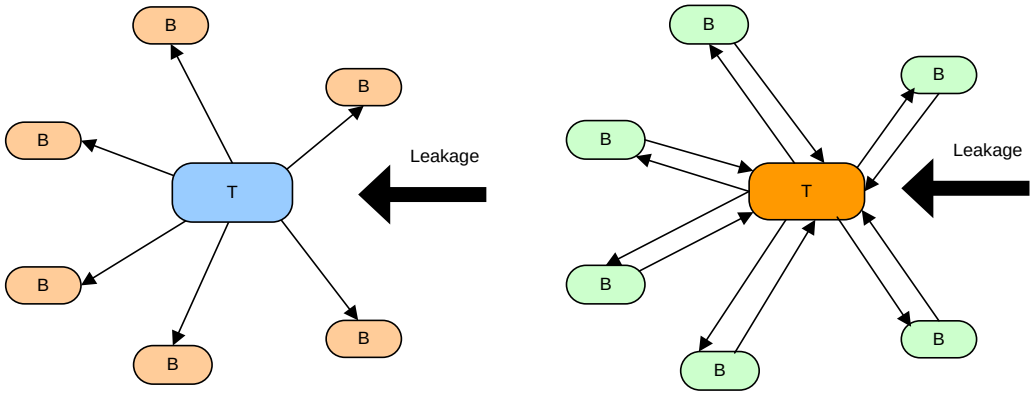
Anti-spamming is now a big challenge for all the search engines. Earlier, Web spamming was done by adding a variety of query keywords on page contents, regardless of their relevance. This type of spamming is easy to detect but now the spammers are trying to use link spamming (Gyongyi and Garcia-Molina 2005b) after the popularity of link-based algorithms like PageRank. In link spamming, the spammers intentionally set up link structures, involving a lot of interconnected pages to boost the PageRank scores of a small number of target pages. This link spamming not only increases rank gains but is also harder to detect by the search engines. Figure 2.6(b) shows a sample link spam structure. Here, the leakage is used to refer to the PageRank scores that reach the link farm from external pages. In this, a web owner creates a large number of bogus web pages called *B's* (their sole purpose is to promote the target page's ranking score), all pointing to and pointed by a single target page *T*. The PageRank assigns a higher ranking score to *T*, more than it deserves (sometime up to 10 times the original score), because it can be deceived by link

spamming.

Wang et al. (2008) proved that PageRank has a zero-one gap flaw which can be potentially exploited by spammers, to easily spam PageRank results. This zero-one gap problem occurs from the ad hoc way of computing the transition probabilities in the random surfing model currently adopted. The probability that the random surfer clicks on one link, is solely given by the number of links on that page. This is why one page's PageRank is not completely passed on to a page it links to, but is divided by the number of links on the page. Therefore, the probability for the random surfer reaching one page is the sum of probabilities for the random surfer following links to this page. Now, this probability is reduced by the damping factor *d*. The justification within the Random Surfer Model, therefore, is that the surfer does not click on an infinite number of links, but gets bored sometimes and jumps to another page at random. The zero-one gap problem refers to the unreasonable dramatic difference between a page with no out-link and one with a single out-link, in their probabilities of randomly jumping to any page. The authors provided a novel DirichletRank algorithm based on the Bayesian estimation, with a Dirichlet prior to solving the zero-one gap problem especially the transition probabilities.

2.2.7.1 *Zero-one gap Problem*

The basic PageRank assumes each row of matrix *M* has at least one non-zero entry, i.e. corresponding node in *G* has at least one out-link. But in reality it does not hold true. Many web pages do not have any out-links and many web applications only consider a sub-graph of the whole web. Even if a page has out-links, it might have been removed when the whole web was projected to a sub-graph. Removing all the pages without out-links is not a solution because it generates new zero-out-link pages. This dangling page problem has been described by Brin and Page (1998), Bianchini, Gori and Scarselli (2005) and Ding et al. (2002). The probability of jumping to a random page is 1 in zero-out-link page, but it drops to λ (in most cases, $\lambda = 0.15$) for a page with a single out-link. There is a big difference between 0 and 1 out-link. This problem is referred to as “zero-one gap” and is a serious flaw in the PageRank, because it allows spammers to manipulate the ranking results of PageRank.



(a) (b)

Figure 2.6: Sample Contrast Structures

Figure 2.6(a) is a structure without link spamming (only out-link) and Figure 2.6(b) shows a typical spamming structure with all bogus pages, *B*'s, having back links to the target page *T*. The authors denote $r_o(.)$ as the PageRank score in Figure 2.6(a) and $r_s(.)$ denotes the PageRank score in Figure 2.6(b). The authors proved that $r_s(T) \geq r_o(T)$ over the range of all λ values. Usually a small λ is preferred in PageRank so the result in $r_s(T)$ is much larger than $r_o(T)$. For example if $\lambda = 0.15$, $r_s(T)$ is about 3 times larger than $r_o(T)$. In Figure 2.6(b), the addition of the bogus pages makes the PageRank score of the target page 3 times larger than before. This is because a surfer is forced to jump back to the target page with a high probability in Figure 2.6(b). With the default value of $\lambda = 0.15$, the single out-link in a bogus page forces a surfer to jump back to the target page with a probability of 0.85. This zero-one-gap problem denotes a serious flaw of PageRank, which makes it sensitive to a local structure change and thus, vulnerable to link spamming.

DirichletRank is an algorithm based on the Bayesian estimation of transition probabilities. According to Wang et al. (2008), this algorithm not only solves the zero-one gap problem, but also the zero-out-link problem. The authors compared the DirichletRank with PageRank, and showed that the former is less sensitive to changes in the local structure and more robust than the latter.

In DirichletRank, a surfer is more likely to follow the out links of the current page, if the page has many out links. Bayesian estimation provides a proper way for setting

the transition probabilities, and Wang et al. (2008) showed that it not only solves the zero-out-link problem, but also the zero-one gap problem. The random jumping probability of DirichletRank is depicted in Equation 2.17.

$$\omega(n)=\frac{\mu}{n+\mu},0\le n\le\infty \tag{2.17}$$

Where n is the number of out-links and μ is the Dirichlet parameter. The researcher set μ = 20, plotted $\omega(n)$ and showed the jumping probability in DirichletRank was smoothed out with no gap between 0 and 1 out-link. They also calculated the DirichletRank scores $d_o(.)$ and $d_s(.)$ for the structures in Figure 2.6(a) and (b), using the following formula shown in Equations 2.18 and 2.19 and $d_S(T)\ge d_O(T)$ for any positive integer k.

$$d_O(T)=\sigma+\frac{\tau}{N} \tag{2.18}$$

$$d_S(T)=\left[1+\frac{k}{\mu^2+(k+1)\mu}\right]\left[\sigma+\frac{k+\mu+1}{\mu+1}\right]\frac{\tau}{N} \tag{2.19}$$

A similar score of PageRank was obtained, i.e. $d_s(T)$ is constantly larger than or equal to $d_o(T)$, but $d_s(T)$ is in fact close to $d_o(T)$. It also shows that there was no significant change in T's DirichletRank scores before and after spamming. Hence, the DirichletRank is more stable and less sensitive to the change of local structure, does not involve extra time cost, and is suitable for Web-scale applications. The study also proved that the DirichletRank is more stable than the PageRank during link perturbation i.e. removing a small number of links or pages. Stability is an important factor for a reliable ranking algorithm, and this study also showed that the DirichletRank is more effective than the PageRank due to its more reasonable allocation of transition probabilities.

Table 2-1 shows the comparison of all the algorithms discussed above. The main

criteria used for comparison are mining techniques used, working method, input parameters, complexity, limitations and the search engine using the algorithm. Among all the algorithms, PageRank and HITS are the most important ones. PageRank is the only algorithm implemented in the Google search engine, while HITS is used in the IBM prototype search engine Clever. Since HITS cannot be implemented directly in a search engine due to its topic drift and efficiency problem, the PageRank algorithm was implemented in the Java program.

Table 2-1: Comparison of Link Structure based Ranking Algorithms

Algorithm / ***Criteria***	***PageRank***	***Weighted PageRank***	***HITS***	***Distance Rank***	***SALSA***	***Dirichlet Rank***
Mining technique used	*WSM*	*WSM*	*WSM & WCM*	*WSM*	*WSM & WCM*	*WSM*
Model	Markov Model of random walk	Markov Model of random walk.	Hubs and Authorities	Recursive method using Reinforcement learning	Hubs, Authorities and Markov chains	Transition probabilities using Bayesian estimation
I/P Parameters	Backlinks	Backlinks, Forward links	Backlinks, Forward Links & content	Backlinks	Backlinks & Forward Links	Backlinks
Complexity(Worst Case)	$O(n)^2$	$< O(n)^2$	$< O(n)^2$	$O(n)^2$	$<O(n)^2$	$O(n)^2$
Limitations	Query independent	Query independent	Topic drift and efficiency problem	Needs to work along with PageRank	Query dependent	Needs to work along with PageRank
Search Engine	Google	Research model	Clever	Research Model	Research Model	Research Model

2.3 HANGING PAGES

In the Web, when a page that does not have any forward or outgoing links then that page can be called as hanging page. Hanging page can be also called dangling page, zero-out-link page, dead end page, sink page etc. For uniformity and consistency purpose, the term 'hanging page' has been used throughout this thesis. There are many reasons for a page to be a hanging page (Eiron, McCurley and Tomlin 2004). They are:

- A page can be naturally hanging i.e. no forward links, like .pdf, .ppt and other attachment files.
- A page producing 403 and 404 HTTP error codes can be considered as a hanging page.
- A page that cannot be crawled by a crawler also can be called as a hanging page.
- A page protected by robots.txt is also called a hanging page.
- A page having no-follow in the meta tag is regarded as a hanging page.
- A page cannot be crawled due to server, router or other problems can also be considered as a hanging page.

This thesis focuses only on handling the hanging pages that occurs naturally in the Web i.e. pages without any forward links. Page et al. (the authors of the Google PageRank algorithm) (Page et al. 1999) have stated the following about hanging pages:

> They affect the model because it is not clear where their weight should be distributed, and there are a large number of them. Often these hanging links are simply pages that we have not downloaded yet……. Because hanging links do not affect the ranking of any other page directly, we simply remove them from the system until all the PageRanks are calculated. After all the PageRanks are calculated they can be added back in without affecting things significantly (page 6).

The first part of the definition holds true, in that hanging pages do not distribute the rank to other pages; instead, they become rank sink (Bianchini, Gori and Scarselli 2005; Langville and Meyer 2004) and many other researchers have reaffirmed this. Equation 2.20 below shows how a rank is calculated for a link structure based Website.

$$W_p = W_p^{in} - W_p^{out} - W_p^{hp} \tag{2.20}$$

In the above equation, W represents a Website with number of pages P. Equation 2.20 shows that, theoretically the ranking of a Website (W_p) can be calculated by adding all the incoming links W_p^{in} from the pages (p) to W, minus the outgoing links W_p^{out} and the hanging links W_p^{hp}. Only the incoming links to a page can count in the rank of a Website in the link structure based ranking algorithms. The outgoing links distribute the rank equally to all the pages that are connected to it. The hanging pages absorb the rank and do not distribute the ranks to other pages. These hanging pages are one of the problems in ranking Web pages, and in turn a problem for Website Optimisation.

The second part of the definition, which states that, hanging pages do not affect the ranking of any other page directly, is not true. While removing hanging pages in the iterative process of the PageRank computation may trigger other pages to become hanging, it also affects the rank of the neighbouring pages. This is shown in Section 3.2.

Hanging pages may have useful information, and particularly the pages with attachment files like .pdf, .ppt and other useful attachment files. They are not included in the PageRank computation and may appear in the index but it may not be their true rank. They may deserve a better rank if the hanging page is a relevant and important one. According to Eiron, McCurley and Tomlin (2004), pages producing 403 and 404 HTTP error code can be called as penalty pages, which occur due to link rot or broken link problems. These penalty pages are not good for a Website and can bring down the rank of that site.

2.3.1 *Existing Methods to Handle Hanging Pages*

This section discusses some former methods for dealing with hanging pages. In the original PageRank algorithm proposed by Brin and Page (1998), the hanging pages were removed from the graph and the PageRank calculated for the non-hanging pages. After calculations, the hanging pages were included without affecting the results. The authors state that a few iterations were enough to remove most of the hanging pages.

Completely removing all the hanging pages would change the results on the non-hanging pages (Haveliwala 1999; Kamvar et al. 2003), since the forward links from the pages were adjusted to consider the lack of links to unreferenced pages. Haveliwala (1999) and Kamvar et al. (2003) suggested jumping to a randomly selected page with probability 1 from every hanging page. For example, the nodes V of the graph ($n = |V|$) can be partitioned into two subsets: (i) S corresponds to a strongly connected sub graph ($|S| = m$) and (ii) The remaining nodes in the subset D have links from S but no forward links. Other research studies (Lempel and Moran 2001; Ng, Zheng and Jordan 2001b) have also proposed methods to handle hanging pages.

A fast two-stage algorithm for computing PageRank and its extensions based on the Markov chain reduction was suggested by Lee, Golub and Zenios (2003). The PageRank vector is considered as the limiting distribution of a homogeneous discrete-time Markov chain that transitions from one web page to another. To compute this vector, they presented a fast algorithm which uses the "lumpability" of the Markov chain and constructed in two stages. In the first stage, they computed the limiting distribution of a chain, where only the hanging pages were combined into one super node. In the second stage, they computed the limiting distribution of a chain where only the non-hanging pages were combined. When this two limiting distributions were concatenated, the limiting distribution of the original chain, the PageRank vector, was produced. According to them, this method can dramatically reduce the computing time and is conceptually elegant. Sargolzaei and Soleymani (2010) also studied the lumping of hanging and non-hanging nodes separately and tried to modify the lumpability. de Jager and Bradley (2009) proposed another method to split the hanging pages into a separate matrix and compute the PageRank.

Another method by Ipsen and Selee (2007) also separated the hanging pages from the non-hanging ones and computed the PageRank. Other methods recommended by Bianchini, Gori and Scarselli (2005), Gleich et al (2010) and Singh, Kumar and Leng (2010; 2012) included hanging pages in the ranking process.

There are two methods proposed in Chapter 3 to include hanging pages in the PageRank computation using *Virtual Node* (*VN*). All the previous methods ignore the hanging pages ranking process which is not fair for the quality hanging pages. The reason to include all the hanging pages in the ranking process is to get fair and relevant ranking for the hanging pages. Also when the hanging pages are connected to the *VN*, they became non-hanging pages and thus satisfying the stochastic requirement of the mathematical model. Chapter 3 describes the methods in detail.

2.4 WEB SPAM

There are two kinds of spamming according to Gyongyi and Garcia-Molina (2005a). They are link spamming and term spamming. Link spamming is a kind of spamming, where the link structure of the Web sites can be altered by using link farms (Baeza-Yates, Castillo and L´opez 2005; Zhang et al. 2004). A link farm is a heavily connected set of pages, created explicitly with the purpose of deceiving a link based search engine's ranking algorithm. Term spamming includes content and meta spamming. Gyongyi et al. (2006) introduced the concept of spam mass and measures the impact of link spamming on a page's ranking. Zhou and Pei (2009) introduce effective detection methods for link spam target pages using page farms.

Bianchini, Gori and Scarselli (2005) worked on the role of hanging pages and their effect on the PageRank. They introduced the notion of energy, which simply represents the sum of PageRanks for all the pages in a given Web site. Equation 2.21 below shows the energy balance which makes it possible to understand the way different Web communities interact with each other, and help to improve the ranking of certain pages.

$$E_I = |I| + E_I^{in} - E_I^{out} - E_I^{hp} \tag{2.21}$$

Let G_1 be a sub graph which represents the energy of a Web site. In the above energy balance equation, $|I|$ denotes the number of pages of G_1, E_I^{in} is the energy that comes to G_1 from other sites. E_I^{out} is the energy that goes out from G_1 which is an energy loss i.e. hyperlinks going out from G_1 decreases the energy. E_I^{hp} is the energy lost in the hanging pages, so, the presence of hanging pages in a Web triggers energy loss. According to Bianchini, Gori and Scarselli (2005), in order to maximize energy, one should not only pay attention to the references received from other sites, but also to the hanging pages and to the external hyperlinks. Hanging pages can be manipulated by spammers to boost the PageRank of Web sites.

Haveliwala and Kamvar (2003) conducted a research study on the second eigenvalue of the Google matrix and the irreducible closed subset, and they mathematically proved the relationship between the second eigenvector and the link spam. According to them, the second eigenvalues are an artefact of certain structures in the Web graph. Wang et al. (2008) addressed a problem called "zero-one gap" in the PageRank algorithm, and developed the DirichletRank algorithm which eliminates the “zero-one gap”; they proved that their algorithm is more resistant to link spamming than the PageRank algorithm. According to Wang et al. (2008), the probability of jumping to a random page is 1, in the case of a hanging page, whereas the probability of a single-out link page drops to 0.15 in most of the cases. There is a big gap between 0 and 1 out link. This gap is referred to as the “zero-one gap”, which allows a spammer to manipulate PageRank to achieve spamming. The DirichletRank proposed by Wang et al. (2008) not only solves the “zero-one gap” problem, but also the hanging page problem. Other researchers like Ipsen and Selee (2007), Langville and Meyer (2004) and Singh, Kumar and Leng (2011) have developed methods to compute PageRank, but they have not explored how hanging pages contribute to link spam.

2.4.1 *TrustRank Algorithm*

The TrustRank (Gyongyi, Garcia-Molina, and Pedersen 2004) is a popular link based Web spam detection algorithm, which works closely with PageRank algorithm. Web Spam (Gyongyi and Garcia-Molina 2005a) refers to the sites/pages that are created with the intention of misleading the search engines. When some sites or pages use various techniques to achieve higher-than deserved ranks, it is called Web spamming

or spamdexing (Gyongyi and Garcia-Molina 2005a). The TrustRank algorithm separates good sites from spam sites using semi-automated methods by, assuming that good sites seldom point to spam or bad sites. TrustRank works by selecting a good seed set. To select this set, it uses Inverse PageRank (IPR) and the link structure of the Web to flow the trust from good pages to other good pages, and separate all the good pages for the seed set. Then it sorts the results in descending order to select top *n* good pages as a seed set. The TrustRank then normalizes the distribution vector by applying the following Equation 2.22:

$$t^* = d.P.t^* + (1-d).dv \tag{2.22}$$

where *d* is the decay or damping factor normally set to 0.85, *P* is the transition matrix, *dv* is the distribution vector after normalization and *t** is the TrustRank score. It is an iterative algorithm like PageRank and gets converged in *M* iterations. A simple example is given using a Web graph in Figure 2.7. The good pages are shown in light blue, i.e. pages 1, 2, 4, 6 and 7 and the bad pages are shown in grey, i.e. pages 3, 5 and 8.

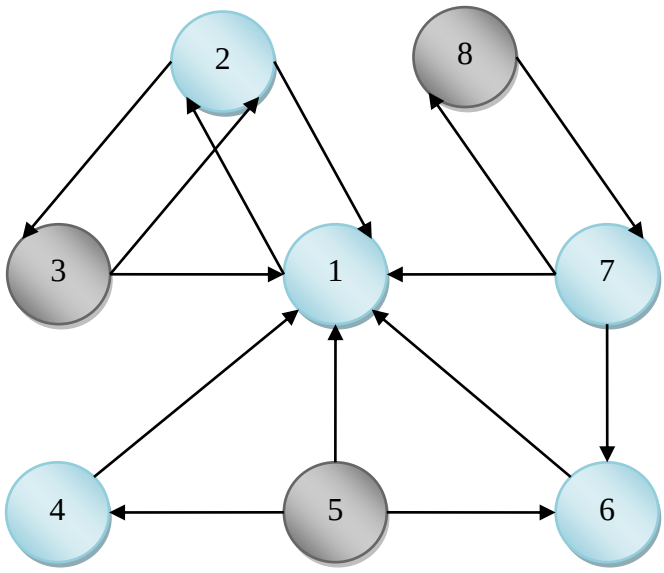


Figure 2.7: A Sample Web Graph for TrustRank

The first step in the TrustRank algorithm is the SelectSeed function (Leng et al. 2012); its goal is to identify desirable pages from the dataset. The SelectSeed

function uses the high inverse PageRank, also known as biased PageRank to find pages that will be most useful in identifying additional good pages. High inverse PageRank pages are likely to point to other high inverse PageRank pages, propagating trust as a result. In the program, 0.85 is used for the damping factor (*d*) with M = 50 iterations. The SelectSeed function returns a vector *s* on the example in Figure 2.7 and shown in Equation 2.23 as follows:

$$s = [0.05,0.08,0.06,0.03,0.06,0.020.19,0.18] \quad (2.23)$$

Next, the rank function will arrange the vectors in descending order and use the oracle function on L most desirable seed pages. Good pages are set to 1 while bad pages and unknown pages are set to 0. This, if the limited budget L is 4, the seed set $S = \{7, 8, 2, 3\}$ and good seed set $S^+ = \{2, 7\}$, while bad seed set $S^- = \{3, 8\}$; the distribution score is shown in Equation 2.24 as follows:

$$dv = [0, 1, 0, 0, 0, 0, 1, 0] \quad (2.24)$$

After that, the algorithm normalizes the static score distribution shown below in Equation 2.25, so that its entries sum up to 1.

$$dv = [0, 0.5, 0, 0, 0, 0, 0.5, 0] \quad (2.25)$$

Finally, the last step computes the TrustRank score with *dv* replacing the score distribution in PageRank algorithm. Again, the damping factor *d* was set to 0.85 and iteration, $M = 50$; the TrustRank algorithm produces the result shown in Equation 2.26:

$$t^* = [0.22, 0.32, 0.14, 0.00, 0.00, 0.03, 0.10, 0.03] \quad (2.26)$$

In this TrustRank, the good seed pages which are pages 7 and 2 have higher scores than most of the pages; some pages like 1 and 3 still have higher scores than page 7. This is because page 1 is a good page, page 3 is pointed by a good page, while page 7 is pointed by a bad page. This TrustRank algorithm is implemented in Chapter 3 and

tested along with the proposed methods to combat Web spam.

2.5 WEBSITE OPTIMISATION

2.5.1 *Introduction to Website Optimisation (WSO)*

Website Optimisation (WSO) started in 1997 when companies started doing business through the Internet. WSO makes a Website friendly, and easy to navigate for users as well as Search engine robots. Consequently, the Website will get more traffic and its rank will improve in an organic way (Kumar, Singh and Mohan 2013).

There are two ways a WSO can be initiated. The first one is before a Website is created i.e. from scratch and the second one is after a Website is created. It is better that optimisation is applied from scratch when a Website is created because altering the link structure of a Web after it is created, would be complicated.

2.5.2 *Website Optimisation Related Terminologies*

There are lots of terminologies related to Website optimisation. A few important and related terminologies are introduced here.

- **SERPs (Search Engine Results Pages)**: Ranking results are displayed by a search engine after a query is typed by a user.
- **Organic Search Results**: Organic search results are the natural way of getting into SERPs due to relevancy of the search terms.
- **Black Hat**: These are improper and illegal methods used by Webmasters to get higher rank in SERPs.
- **White Hat**: These are proper WSO techniques, which follows the best practices and guidelines to get a better rank in SERPs.
- **On-Site**: On-Site WSO factors are those that can be used by the Webmasters, within the Website to improve the ranking in SERPs.
- **Off-Site**: Webmasters have very little control over these factors. Search engines use them to judge the quality of a Website.

2.5.3 *Challenges of Website Optimisation (WSO)*

WSO challenges can be categorized into three categories according to the people involved in it. Killoran (2013), states that there are 3 classes of people involved in shaping the rank of a Website. They are Search engines and their programmers, Webmasters and WSO professionals and Search engine users. The WSO challenges they face are elaborated as follows:

2.5.3.1 *Search Engines and their Programmers*

Given a query, each search engine may produce different ranking orders. Bar-Ilan (2005) and Bar-Ilan, Mat-Hassan and Levene (2006) and many other researchers can attest to this. Even one search engine may produce different answers for a given query at different locations, because its different data centres around the world are not synchronized (Evans 2007). For a given query, one search engine may produce different ranking orders with different browsers, because the search engines like Google monitor the browser's pattern. Mowshowitz and Kawaguchi (2005) state that some search engines favour their own sites and products to appear on top of SERPs rather that of their competitors. For example, Google favours its own products, YouTube and Google+ in the SEPRs. Another challenge is the "rich-get-richer" factor, which occurs because the search engines always give higher ranking for popular and branded sites. Wikipedia, for instance, always comes on top of SERPs in Google and other Search engines. Due to this "rich-get-richer" problem, a newly created quality Website may have to struggle to get into SERPs. Another challenge is the frequent tweaking of ranking algorithms by search engines. Google tweaks its ranking algorithm more than 500 times in a year. Apart from PageRank algorithm, Google uses more than 200 factors to rank a Website. On top of these challenges, search engines do not disclose their ranking algorithms and techniques due to their business competition.

2.5.3.2 *Webmasters and WSO Professionals*

The second challenge in Website optimisation is how much Webmasters and WSO professionals know the policies of WSO and best practices. WSO is very important for commercial and business sites. Some Webmasters wittingly or unwittingly use the Black Hat WSO technique like *keyword stuffing*, *link farming* etc., to achieve a higher ranking in SERPs. *Keyword stuffing* is one of the Black Hat WSO techniques

in which excessive keywords are inserted in many places of a page. *Link farming* (Henzinger, Motwani and Silverstein 2002) is a densely connected page, created explicitly for the purpose of deceiving a link structure based Search engine ranking algorithm. The other challenges are discussed below in the section on On-Site and Off-Site ranking factors.

2.5.3.3 *Search Engine Users*

Search engine users' behaviour and preferences help the search engine to build a better relevancy algorithm based on user's response. A search engine (especially Google) uses the searcher's history and builds two important ranking factors, i.e. C*lick-Through Rate* (CTR) and *Bounce Rate* (BR). CTR is the percentage of times searchers click on SERPs link for a given query. A higher CTR indicates that searchers clicking on a link on the SERPs have a higher relevancy on a given query. BR, which is the opposite of CTR, refers to the percentage of searchers who return to SERPs after clicking a link, due to the irrelevancy of a given query. A higher bounce rate tells the search engine that, the searchers clicking the SERPs link for a given query are disappointed with the Web page. The search engine remembers both CTR and BR in future searches. This way the search engine user helps the Search engines and WSO professionals make some of the best policies for ranking.

2.5.4 *Website Optimisation Stages*

According to Burdon (2005), Website optimisation can be done in four stages:

- Pre-Site Activities - Pre-Site activities occurs before a Website is created and any optimisation process is started. This is all about online strategies for business plan, policies, and strategies, research on market demand, customers and competitors.

- On-Site Activities - On-Site activities are concerned with designing and developing a Website. Keywords optimisation, contents optimisation, structure optimisation as well as internal link optimisation comes under this activity.

- Off-Site Activities - Off-Site activities can help to improve the ranking of a site after it is created. This includes relevant link building, promoting the site through blogs and social networks. Inbound link and social media optimisation comes under this activity.
- Post-Site Activities - WSO is a continuous process, so the Post-Site activities include monitoring and analysing the traffic, customer feedback, link building effects, ranking improvements and competitor's reaction.

The four important stages of optimisation are shown in Figure 2.8 below. If Webmasters or Web developers adopt these stages while designing and developing Websites, their Websites can obtain lot of traffic and will be ranked better in the SERPs.

2.5.4.1 *Pre-Site Stage*

The Pre-Site stage is a very important stage like the analysis stage of Software development. There are two important activities in Pre-Site stage, i.e. planning and research. It is concerned with planning the online business strategy, research on customers interest, competitor's skill etc.

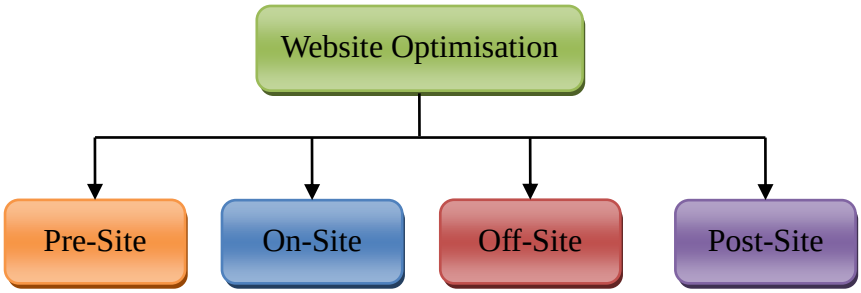


Figure 2.8: Main Stages of WSO

Figure 2.9 shows the important activities in the Pre-Site stage. The first step in Planning is to understand the company's overall business strategy, while the next step is to plan the online business objectives, scope, budget and marketing. Research activities include finding information about market category, competitors in that category, and customers in that category.

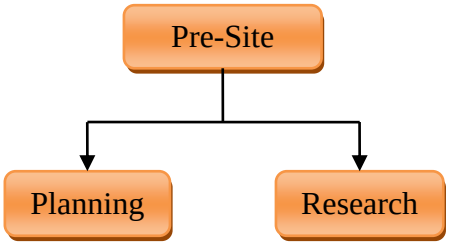


Figure 2.9: Pre-Site Activities in Website Development

The second important research activity is to find the relevant keywords which uniquely identify a business. These keywords are typed by users in the user interface of Search engines, while searching for information. Here, keywords play the same role of keywords in manuscripts which uniquely identify them. Hence, the selection of keywords is very important for the success of an online business. Users should try the keywords in the major search engines before reviewing the results onsite. More about keywords are covered in the next sub section.

2.5.4.2 *On-Site Factors*

Figure 2.10 shows the different ranking factors in the On-Site stage. They are Content, HTML, Internal Links and Architecture.

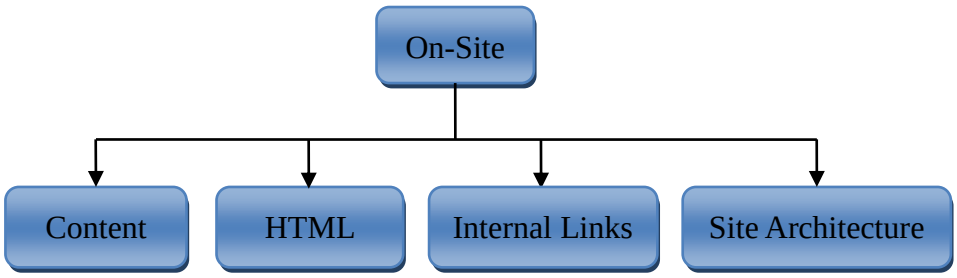


Figure 2.10: On-Site Ranking Factors of WSO

These On-Site ranking factors are a collection of factors which help search engine spiders to determine the characteristics of a page and help the users to understand it. These factors are fully under the control of a Webmaster. When used properly, in a Website, these factors can improve its ranking. Webmasters can also decide the type of content they want to publish.

2.5.4.2.1 *Content*

Content is the king among all the ranking factors in optimisation, and is the only factor which can provide the true nature of a site. The four important elements associated with content:

- Content Quality
- Content Keywords
- Content Engagement
- Content Freshness

2.5.4.2.2 *HTML*

HTML tags can help a user or a search engine determine the relevancy of a Website. The HTML is the basic building block to create Websites. Search engine crawlers can read HTML and HTML related codes. Google, for instance, claims that it can read about 20 over file formats.

2.5.4.2.3 *Internal Links*

Internal links are hyperlinks which are used to connect the pages within the same domain. They help the user and the search engine crawlers to navigate the site and also provide the information hierarchy for a site; this in turn helps to produce the site architecture. The internal links ensure that the ranks are passed to other pages on the site. Normally the links are given between the <*a* and </*a*> using the hyperlink referral, *href*.

Another important element related with link is the *anchor text*, which is used to describe the page to which the link is pointing. This anchor text is largely used by the search engines when identifying the relevancy of a page. Relevant keywords should be used in the anchor text of internal links to improve the ranking of internal pages. The following are a few general rules to be followed in internal links.

- Use anchor text with relevant keywords in internal links.

- Use direct links to more important pages.
- Avoid broken links and hanging links.
- If a hanging page is an important page, then use the methods proposed by Bianchini, Gori and Scarselli (2005) and Singh, Kumar and Leng (2011) to make it a non-hanging page.
- Reduce the number of no-follow links which can bring down the rank of a page.
- Keep a low number of internal outgoing and external outgoing links.
- Keep the number of links on a page to a reasonable number; otherwise search engines may treat your page as link farm.

2.5.4.2.4 *Site Architecture*

Site Architecture helps Search engines and users to easily move around and browse a Website in an efficient way. If the site architecture is easy to navigate, then there is a chance of more pages being indexed by the search engines. According to Vryniotis (2010) of Webseoanalytics.com, there are four types of Site Architecture used by Web developers. They are:

- Complete Link - Here, every page is linked to every other page in the site. This architecture is not really good because of its poor navigation and being ranked lower by search engines. This approach can be used for smaller Websites.
- Deep Link Hierarchy - This approach uses a tree-like structure and only the top level pages are indexed and ranked better in this approach. The navigation is slow and this architecture is not recommended.
- Flat Link Hierarchy - Flat link hierarchy also uses a tree-like structure but it has fewer hierarchy levels than deep link methods. It is one of the best architecture where navigation, indexing and ranking is concerned, and can be used in small, medium and large Websites.

- Overlapping Link Hierarchy - This approach is a form of Flat link hierarchy, where, an N level page is not only linked to N+1 level page, but also to the important pages of N+2 level. This technique can improve the indexing and the rankings of important subcategory pages and can be used to build large Website like e-commerce sites, directories etc.

2.5.4.3 *Off-Site Factors*

Figure 2.11 shows the different Off-Site ranking factors which can be broadly categorized into five types. They are Links (Inbound links), Trust, Social, Personal and User metrics.

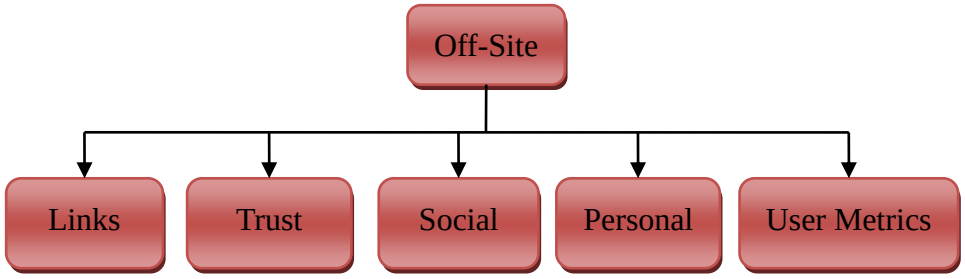


Figure 2.11: Off-Site Ranking Factors of WSO

Off-Site factors provide a very good impression on how other users (including Search engines) and Websites look at a Website. If a Website is relevant and useful, it may get references from other Websites, blogs etc.

- Links - Links refers to the inbound links from external Websites. Inbound links from a reputed and related Website is a good way to improve the Website's authority and relevance. Authority improves trust which in turn, improves the rank of a site. If a Website has good and genuine content, then other relevant sites will link to it. This type of link is called natural or organic. Apart from that, proper marketing technology is required to promote the content. Link factors contribute to nearly 40% of the overall ranking factors in the Google Search engine as shown in Figure 2.12 at the end of this section.

- Trust - Trust is related to link and site authority and the domain history. Quality of the links, social references and site engagement factors contribute to Authority. When the domain and site is older, reputed and operated in the same way, the trust is better.

- Social - Social networks like Facebook, Twitter, Google+ etc. play an important role in promoting Websites. The reputation of the post and sharing is important here, where content is shared with like-minded users. Search engines closely watch the social networks where the contents are shared.

- Personal - Apart from other ranking factors, search engines use personal factors like the country the site is hosted in, who the host is, the site's social connections and how the site is being viewed etc. The above factors help to produce localized search results.

- User Metrics - User metrics are the factors based on user's actions while doing a search on a search engine. Click Through Rate (*CTR*), Bounce Rate (*BR*) and Dwell Time (*DT*) are some user metric factors, which are collected by major search engines and used in their ranking algorithms. Recently, major search engines have started using user metrics also for organic searches. Google collects this information through Google Analytics and this can be checked in the Google Webmaster Tools. The *CTR* is calculated using the following formula in Equation 2.27.

$$CTR = \frac{NPC}{NPD} \tag{2.27}$$

Where *CTR* stands for Click Through Rate, *NPC* for Number of Times a Page is Clicked and *NPD* for Number of Times a Page is Displayed. For example, if a page is clicked 20 times with a display or impression of 100 times, the *CTR* rate is 0.2 or 20%. A higher *CTR* improves the rank of a page. *BR* can be calculated using the following formula in Equation 2.28.

$$B_r = \frac{V_{one}}{V_{total}} \tag{2.28}$$

Where B_r is the Bounce rate of a page, V_{one} is the total number of visitors viewing one page only and V_{total} is the total entries to a page. BR refers to the number of people who visit a site and do not click any page and then return to the SERPs. This shows that the site content is not relevant according to the user search query. A higher *BR* decreases the rank of a page.

DT is simply the time a user spends on a page after clicking the page from SERPs. More *DT* increases the rank of the page. Search engines use the combination of *CTR*, *DT* and few more factors to calculate user metrics.

These Off-Site factors which are not under the control of Webmasters or Web developers, are used by search engines to get more information about the relevancy of a site for a given search query. Webmasters sometimes wittingly or unwittingly manipulate On-Site factors to increase their site ranking. These Off-Site factors cannot be manipulated by Webmasters and generally Search engines combine On-Site and Off-Site factors for ranking.

2.5.4.4 *Post-Site Activities*

Post-Site activities are mainly concerned with monitoring, measuring, updating and improving the performance of the site. The following are the different factors that can be monitored continuously and improved over a period of time.

- Keywords
- Contents
- Links
- Site Architecture
- User metrics

Website Optimisation is a continuous process because Search engines keep tweaking their ranking algorithms and factors frequently. The following questions need to be

asked routinely: Do the keywords help to improve the performance of the site? Do these keywords help to increase the ranking in SERPs and the traffic to the site? Does this help to increase the business? Also, monitor the reaction from the competitor's sites.

Based on this study, the following factors can be considered as Black Hat techniques and should not be used in Website development. The first five factors are On-Site based and the next five are Off-Site based factors.

- Using Spun and Duplicate content
- Creating Content forms (a form of cloaking)
- Using keyword stuffing (in content, titles and other meta descriptions)
- Using tiny text, invisible text, no-frames text, no-script text, alt text (all targeted at Search engine crawlers)
- Using doorway or gateway pages
- Getting hidden inbound links and giving hidden outbound links
- Getting inbound links from link farms
- Purchasing expired domains and redirecting them to a Website
- Links from spam blog comments
- Using social networking spamming methods

There are many research studies and reports about On-Site and Off-Site factors. A report from Sullivan (2013) of Searchengineland.com simulates a periodic table for SEO ranking factors. Weighting was used for all the factors based on a scale from 1 to 3. Negative weighting was also assigned for factors violating the best practices (Black Hat). Another detailed report from Peters (2013) of moz.com analyses both On-Site and Off-Site ranking factors. Apart from that the domain level factors were also added in the report. Most of the WSO companies and consultants develop their

techniques and methodologies mainly for Google Search engine, which controls nearly 89% of the Search engine market. Most of the methodologies discussed in this chapter also work well for the Google Search engine. Figure 2.12 details the latest survey report from moz.com, on the distribution of WSO factors used by the Google algorithm (Peters 2013).

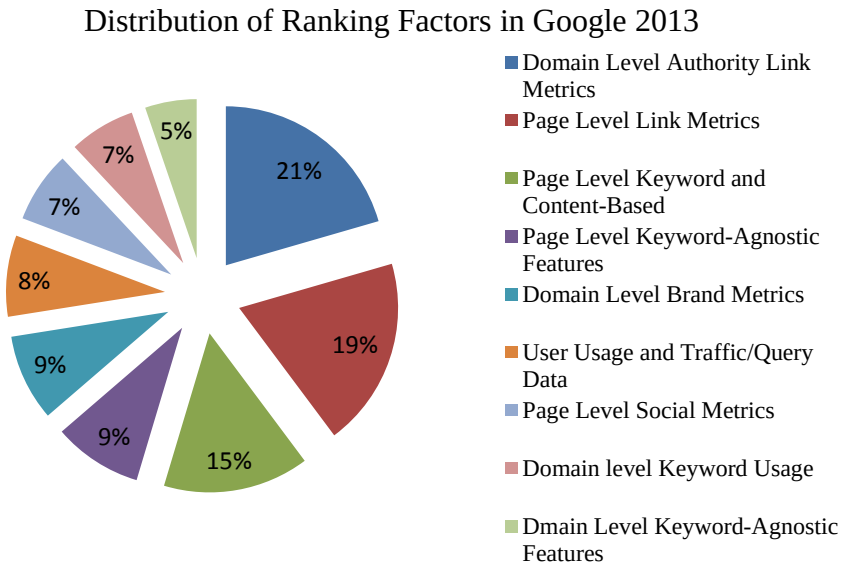


Figure 2.12: Distribution of the latest WSO Factors used by Google

It is evident from the above illustration, that link metrics (Domain Level Authority Link Metrics and Page Level Link Metrics) controls 40% of WSO factors. The next important factors are the Page Level Keyword and Content Based (15%). Google also places importance on user usage, traffic data and page level social metrics. Google, which started as a link structure based search engine, has now evolved into the most trusted search engine by combining more than 200 ranking factors, apart from using PageRank algorithm.

2.6 Preliminaries and Mathematical Definitions

All the preliminaries and the mathematical definitions used in this research study are described here. The *Web graph*, *Markov Chain*, *Adjacency Matrix* and *Transition Probability Matrix* are described, and the datasets are introduced and analysed. Parameter settings for all the algorithms and performance evaluations of the study also discussed here.

2.6.1 *Web Graph*

Graph G consists of two sets, *V* and *E*, where *V* is a finite, nonempty set of *vertices*, and *E* is a set of pairs of vertices; these pairs are called *edges* (Horowitz, Sahni and Rajasekaran 2008). The notations *V*(*G*) and *E*(*G*) represent the sets of vertices and edges, respectively, of graph *G*. A general description of the graph can be denoted as $G=(V,E)$. In an *undirected graph* (*UG*) the pair of vertices representing any edge is unordered. Thus, the pairs (*u*, *v*) and (*v*, *u*) represent the same edge. In a *directed graph* (*DG*), each edge is represented by a directed pair (*u*, *v*); *u* is the *tail* and *v* is the *head* of the edge. Therefore, (*u*, *v*) and (*v*, *u*) represent two different edges.

Figure 2.13 shows a sample directed graph *G*; the set representation for the graph *G* consists of 3 vertices and 5 edges as follows:

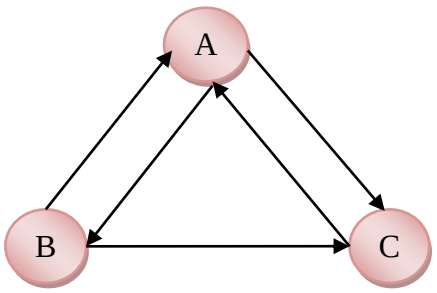


Figure 2.13: Sample Directed Graph *G*

$$V(G)=\{A,B,C\} \qquad E(G)=\{(A,B),(B,A),(A,C),(C,A),(B,C)\}$$

Normally a graph may not have an edge from a vertex *v* back to itself. That is, edges of the form (*v*, *v*) and (*u*, *u*) are not legal. Such edges are known as *self-edges* or *self-loops*. In some cases self-edges or self-loops can be used.

The number of distinct unordered pairs (*u*, *v*) (maximum edges, *ME*) with $u \neq v$ in undirected graph with *n* vertices can be calculated using the following formula shown in Equation 2.29.

$$ME=\frac{n(n-1)}{2} \tag{2.29}$$

An undirected graph is said to be *complete*, if it produces exactly $\frac{n(n-1)}{2}$ edges with *n* vertex. The maximum number of edges (*ME*) in a directed graph can be calculated using the following formula in Equation 2.30.

$$ME = n(n-1) \tag{2.30}$$

If (*u*, *v*) is an edge in *E*(*G*), then the vertices *u* and *v* are *adjacent* and edge (*u*, *v*) is *incident* on vertices *u* and *v*. If (*u*, *v*) is a directed edge, then vertex *u* is adjacent to *v*, and *v* is adjacent from *u*. The edge (*u*, *v*) is incident to *u* and *v*. In the directed graph *G* in Figure 2.13, the edges incident to vertex *B* are (*A*, *B*), (*B*, *A*) and (*B*, *C*). A directed graph is said to be *strongly connected* if for every pair of distinct vertices *u* and *v* in *V*(*G*), there is a directed path from *u* to *v* and also from *v* to *u*.

According to Broder et al. (2000), a Web can be imagined as a large graph containing several hundred million or billions of pages as vertices, and a few billion hyperlinks as edges. It can also be called a *Web Graph* (WG) (Kumar et al. 2000a). Several research studies have been done to analyse the properties of the *graph* (Kumar et al. 2000a; Kleinberg et al. 1999). Stochastic models for the Web graph was analysed by Kumar et al. (2000b). Broder et al. showed the structure of the Web graph looking like a giant bow tie as shown in Figure 2.14 (2000). This Web macroscopic structure has four pieces. The first piece is a central core, all of whose pages can reach one another along directed links -- this "giant strongly connected component" (*SCC*) is at the heart of the Web. The second and third pieces are called *IN* and *OUT*. *IN* consists of pages that can reach the *SCC*, but cannot be reached from it - possibly new sites that people have not yet discovered and linked to. *OUT* consists of pages that are accessible from the *SCC*, but do not link back to it, such as corporate websites that contain only internal links. Finally, the *TENDRILS* contain pages that cannot reach the *SCC*, and cannot be reached from the *SCC*. According to Broader et al., the size of the *SCC* is relatively small compared with *IN*, *OUT* and *Tendrils*. Almost all the sets have roughly the same size. It is evident; therefore, that the Web is growing rapidly and it is a huge structure. In the next section, the definitions and the mathematical model used in this thesis are described.

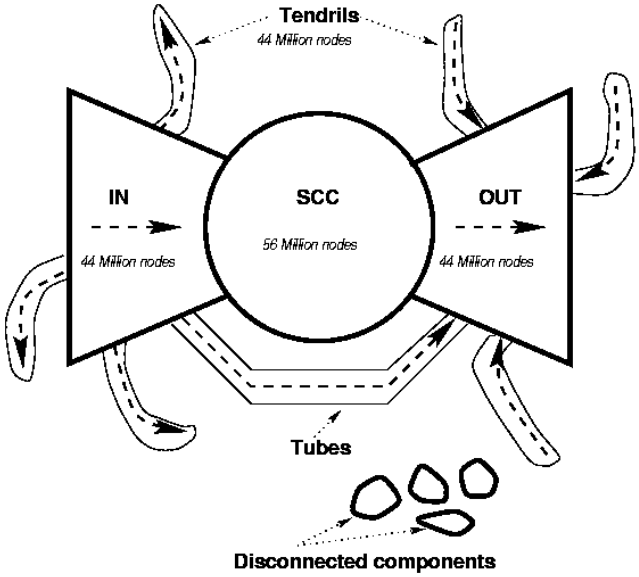


Figure 2.14: Macroscopic Structure of Web (Broder et al. 2000)

A Web can be represented as a directed graph called $G_w(V_w, E_w)$, where V_w denotes a set of Web pages and E_w denotes the hyperlink between pages.

Definition 2.4: The in-degree of a page *i* is the number of edges (incoming links) for which *i* is the head, i.e. $id_{(i)} = \sum_i E_{ij}$.

Definition 2.5: The out-degree of a page *i* is the number of edges (outgoing links) for which *i* is the tail, i.e. $od_{(i)} = \sum_i E_{ij}$.

Definition 2.6: If de_i is the *degree* of vertex *i* in a graph *G* with *n* vertices and *e* edges, then the degree of vertex *de* is the sum of in-degree and out-degree as follows in Equation 2.31.

$$de_i = (id_i + od_i) \tag{2.31}$$

The vertex *B* has in-degree 1, out-degree 2 and degree 3 in the directed graph *G* in Figure 2.13.

On a larger scale, a Web graph can be called as a *host graph* and it can be represented as $G_h = (V_h, E_h)$ where V_h denotes a set of host vertices and E_h denotes a set of

ordered pair of hosts. A host consists of a set of Web pages and under the same domain. Most of the properties of a Web graph apply to a host graph also.

Definition 2.7: An element A_{ij} is equal to 1 if a page *i* have a link to page *j* and is equal to 0 otherwise. It can be represented as an *adjacency* matrix in Equation 2.32. This matrix can also be called as a *link* matrix or *connection* matrix. The Adjacency matrix will be shown in the example later.

$$A_{ij} = \begin{cases} 1 & \text{if } V_i \to V_j \\ 0 & \text{otherwise} \end{cases} \tag{2.32}$$

The generalized $n \times n$ *adjacency* matrix *A* for a directed graph is shown below:

$$A = \begin{bmatrix} a_{1,1} & a_{2,1} & \cdots & a_{n,1} \\ a_{1,2} & a_{2,2} & \cdots & a_{n,2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1,n} & a_{2,n} & \cdots & a_{n,n} \end{bmatrix}$$

Definition 2.8: A *transition probability* matrix is defined as $P_{ij} = A_{ij}/od_{(i)}$ when $de(i) > 0$. For those *i* it is row stochastic which means i^{th} row elements sum to 1. The transition probability matrix can be developed using the following formula as shown in Equation 2.33.

$$P_{ij} = \begin{cases} \dfrac{1}{od_{(i)}} & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{2.33}$$

According to Langville and Meyer, the probability matrix can be also called the transition probability matrix (2005). It is an $n \times n$ matrix, where *n* is the number of Web pages. If a page *i* has $od_{(i)} \geq 1$, then the element in row *i* and column *j* of *P* is $P_{ij} = 1/od_{(i)}$, where $od_{(i)}$ is the number of forward links of Web page *i*. Otherwise, $P_{ij} = 0$ as shown in Equation 2.33. Thus, P_{ij} represents the likelihood that a random surfer will select a link from Web page *i* to Web page *j*. The generalized $n \times n$ *transition probability* matrix *P* for a directed Web graph is shown below:

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & \cdots & p_{n,n} \end{bmatrix}$$

Definition 2.9: Let *P* be a *transition probability* matrix and the sum of a row in a row matrix is 0, i.e. if $\sum_i P_{ij} = 0$ then that corresponding page of the element can be referred to as a hanging page.

2.6.2 *Markov Chain*

The Markov Chain is a random process (Gao et al. 2009)used by a system and states that at any given time $t = 1, 2, 3 \ldots n$ occupies one of a finite number of states. At each time *t*, the system moves from state *i* to *j*, with probability P_{ij} that does not depend on *t*. P_{ij} is a *transition probability* which is an important feature of the Markov chain and it decides the next state of the object by considering only the current state and not any previous ones. A discrete Markov chain can be defined as follows:

Let $\{X_n\}$, n = 0, 1, 2, ..., be an irreducible, aperiodic Markov chain in discrete time, whose state space *S* consists of non-negative integers. The transition probabilities are assumed to be stationary i.e.,

$$P\{X_{n+1} = j | X_n = i\} = P_{ij}, \tag{2.34}$$

In Equation 2.34, where $i, j \in S, n = 0,1,\ldots$, $P_{i,j} \geq 0$, $\sum_j P_{i,j} = 1$. The matrix of transition probabilities is denoted by $P = (p_{ij})$, and its n^{th} power $P^n = \left(P_{ij}^{(n)}\right)$ gives the *n*-step transition probabilities. This type of Markov chain can be referred to as a simple discrete time Markov chain.

The Markov chain (Norris 1996; Gao et al. 2009) was invented by A.A. Markov, a Russian Mathematician in the early 1900's, to predict the behaviour of a system that

moves from one state to another state by considering only the current state. The Markov chain uses only a matrix and a vector to model and predict this state. These chains are used in places where there is a transition of states. It has been utilised in Biology, Economics, Engineering, Physics etc., but the recent application of the Markov chain in the PageRank algorithm Google search engine is interesting and more challenging. The relationship between Markov chain and PageRank algorithm is discussed below in Section 2.6.2.1.

Transition Probability matrix P is an $n \times n$ matrix formed from the *transition probability* of the Markov process, where n represents the number of states. Each entry in the transition matrix P_{ij} is equal to the probability of moving from state j to state i in one time slot, so, $0 \leq P_{ij} \leq 1$ must be true for all $i, j = 1, 2, \ldots, n$. The following example shows a sample transition matrix of a 3 state Markov chain:

$$P_{ij} = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 0 & 1/2 \\ 1/2 & 1/4 & 1/4 \end{bmatrix}$$

The *Transition Probability* matrix must follow the following rules (Atherton, 2005):

- The Transition matrix must be a square matrix. Each entry in the matrix represents a transition state of the Markov chain, so each entry must be between zero and one.
- If the matrix is a row matrix, then the sum of the entries in a row is the sum of the transition probabilities from a state to another state; so, the sum of the entries in any row must equal to one. This is called a *stochastic matrix.* For a column matrix, the sum of the entries in any column must equal to one.

In the above *Transition Probability* matrix, P_{ij}, the probability of moving from one state to another state can be easily seen. For example $P_{3,2}$ = ¼ i.e. the probability of moving from state 2 to 3 is only 25%. Markov chains are used, thus, to predict the probability of an event.

2.6.2.1 *Application of Markov Chain in the PageRank Algorithm*

The PageRank algorithm is the ranking algorithm used to rank the Web pages in the Google Search engine (Brinkmeier 2006). In the original PageRank algorithm by Brin and Page (1998), the Markov chain is not mentioned. But the other researchers like Langville and Mayer (2004; 2006b), Bianchini, Gori and Scarselli (2005)and Brinkmeier (2006), explored the relationship between the PageRank algorithm and the Markov chain. According to Gao et al. (2011), besides the PageRank algorithm, all the other variations of PageRank algorithms can be modelled as a discrete-time Markov process. This section explains the relationship between the PageRank algorithm and the Markov chain. Imagine a random surfer surfing the Web, going from one page to another by randomly choosing an outgoing link from one page to go to the next one. This can sometimes lead to dead ends, i.e. pages with no outgoing links cycle around a group of interconnected pages. Hence, for a certain fraction of time, the surfer chooses a random page from the Web. This theoretical random walk is known as the Markov chain or Markov process. The limiting probability that an infinitely dedicated random surfer visits any particular page is its PageRank.

2.6.3 *Mathematical Definitions Example*

The following Figure 2.15 shows a sample Web graph (G_w) extracted from Curtin University (Sarawak) site (Kumar, Leng and Singh 2013). It contains 7 pages namely, Home, Admin, Staff, Student, Library, Department and Alumni. The following Web graph is used to explain the basic definitions, Adjacency matrix, Transition Probability matrix and Markov chain used in the thesis.

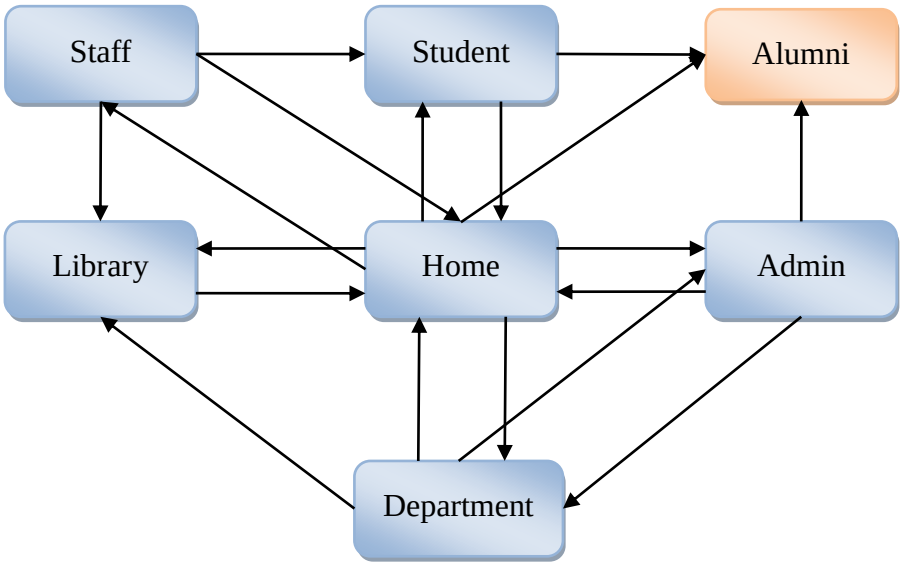


Figure 2.15: A Sample Web Graph G_w

Table 2-2 below shows the In-degree, Out-degree and the degree, for the sample Web graph G_w shown in Figure 2.15, as per the definitions 2.4, 2.5 and 2.6.

Table 2-2: In-Degree, Out-Degree and Degree Calculation for the Web Graph G_w

Page	***In-Degree (id)***	***Out-Degree (od)***	***Degree(de)***
Staff	1	3	4
Student	2	2	4
Alumni	3	0	3
Library	3	1	4
Home	5	6	11
Admin	2	3	5
Department	2	3	5

The adjacency matrix A is created as per definition 2.7 and Equation 2.32 and shown below:

$$A=\begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

The adjacency matrix *A* is a row matrix. The order of row is Staff, Student, Alumni, Library, Home, Admin and Department. For example the first row, Staff page, has a link to the Student, Library and Home pages. Also notice the third row, the Alumni page, which is a hanging page and there are no forward links from that page. That is why the third row has all zeros.

Next, the Transition Probability matrix *P* is created as per definition 2.8 and the Equation 2.33 and shown below:

$$P=\begin{bmatrix} 0 & 1/3 & 0 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 0 & 1/6 & 1/6 \\ 0 & 0 & 1/3 & 0 & 1/3 & 0 & 1/3 \\ 0 & 0 & 0 & 1/3 & 1/3 & 1/3 & 0 \end{bmatrix}$$

2.7 DATA SETS AND FEATURES

Three publicly available datasets were used throughout the whole thesis – WEBSPAM-UK2006 (Castillo et al. 2006), WEBSPAM-UK2007 (Yahoo! Research 2007) and EU2010 (Benczúr et al. 2010). The first two datasets were downloaded from the Laboratory of Web Algorithmics, Università degli Studi di Milano, with the support of the DELIS EU - FET research project. The third one was downloaded from the European Archive Foundation. Apart from the three data sets, live data from the Internet crawled by PyBot program (Leng et al. 2011) and MATLAB program were also used.

WEBSPAM-UK2006 consists of 77,741,046 Web pages, while WEBSPAM-UK2007 consists of 105, 896,555 Web pages. Due to the large collection, the host level was considered instead of page level. The former consists of 11,402 hosts whereas the latter consists of 114,529 hosts. The EU2010 consists of 191,389 hosts. Live data from the Internet sources were used for the page level experiments.

Figure 2.16 shows the graphical representation of hanging and non-hanging hosts for WEBSPAM UK-2006 dataset.

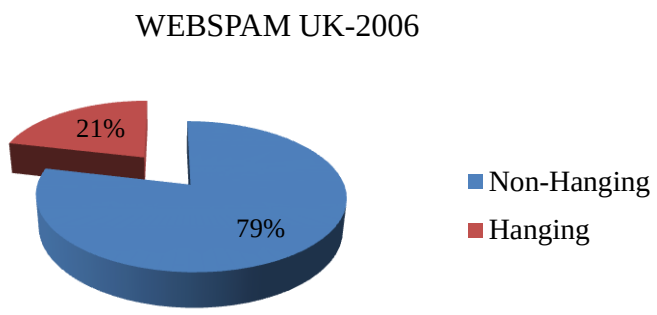


Figure 2.16: Hanging Vs. Non-Hanging Hosts in WEBSPAM UK-2006 Dataset

Figure 2.17 shows the graphical representation of hanging and non-hanging hosts for WEBSPAM UK-2007 dataset.

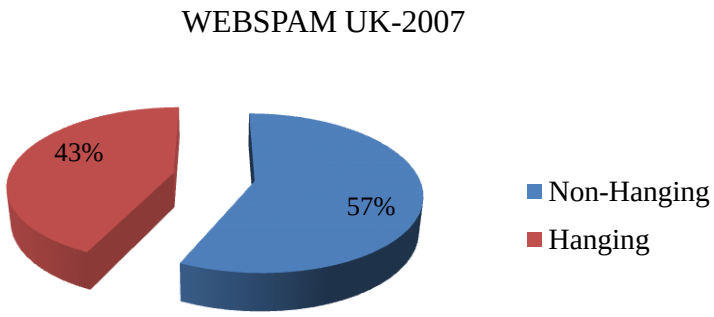


Figure 2.17: Hanging Vs. Non-Hanging Hosts in WEBSPAM UK-2007 Dataset

Figure 2.18 shows the graphical representation of hanging and non-hanging hosts for EU2010 dataset. These figures show that the percentage of hanging hosts/pages have increased over the years.

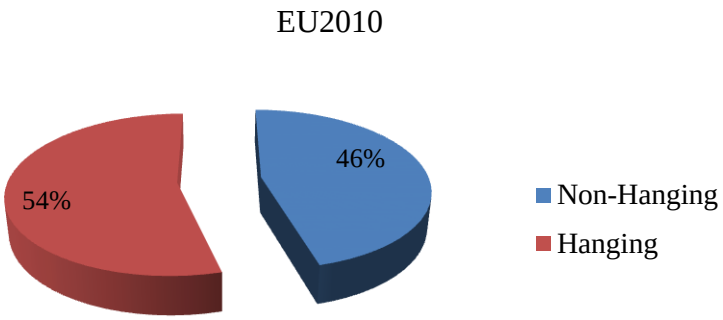


Figure 2.18: Hanging Vs. Non-Hanging Hosts in EU2010 Dataset

Figure 2.19 shows the graphical representation of hanging and non-hanging pages in the Curtin University (Sarawak) Web site.

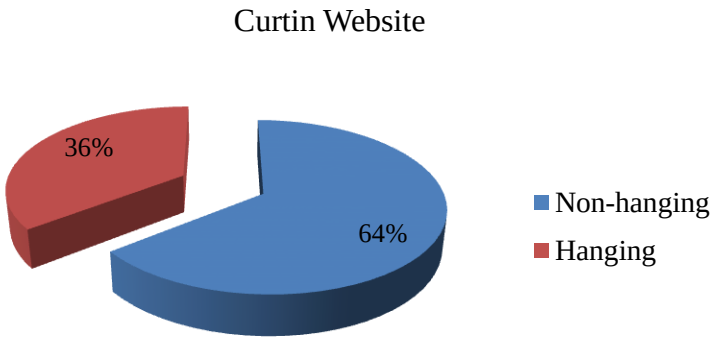


Figure 2.19: Hanging Vs. Non-Hanging Pages in the Curtin Website

2.8 Parameters Settings and Performance Evaluation

In this section, the general parameters settings that were used throughout the research study are discussed. Performance evaluations of the algorithms are shown in the individual chapters.

For the PageRank algorithm and the proposed methodologies, the damping factor *d* was set to 0.85 and the maximum number of iteration was set to 50 for the ranking programs.

2.9 SIMULATION AND EXPERIMENT RESULTS

This section shows the simulation and the experiment results for PageRank algorithm and Weighted PageRank algorithm.

2.9.1 *PageRank Simulation*

An example of the hyperlink structure of four pages *A*, *B*, *C* and *D* is shown in Figure 2.20. The PageRank for pages *A*, *B*, *C* and *D* are computed using the PageRank program created using JAVA and applied on to the graph in Figure 2.20. The input entry screen is shown in Figure 2.21. Users can select the input file which contains the number of nodes, and the number of incoming and outgoing links of the nodes. The output is shown in Table 2-3 and Figure 2.22. Table 2-3 is the output of the PageRank convergence scores and Figure 2.22 is the PageRank convergence chart for the hyperlink structure in Figure 2.20.

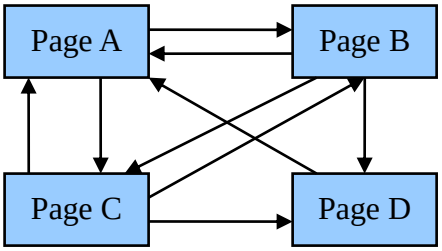


Figure 2.20: Hyperlink Structure for 4 Pages

The initial PageRank is assumed as 1 and calculated accordingly, while the damping factor *d* is set to 0.85. Sample PageRank calculation using Equation 2.2 is shown below:

$PR(A)=(1-d)+ d\ (PR(B)/C(B)+PR(C)/C(C)+PR(D)/C(D))$
$=(1-0.85) + 0.85(1/3+1/3+1/1) = 1.566667$

$PR(B) = (1-d) + d((PR(A)/C(A)+ (PR(C)/C(C)) = 1.099167$

$PR(C) = (1-d) + d((PR(A)/C(A) + (PR(B)/C(B)) = 1.127264$

$PR(D)= (1-d) + d((PR(B)/C(B) + (PR(C)/C(C)) = 0.780822$

The second iteration is shown below by taking the above PageRank value of pages *A*, *B*, *C* and *D* and continuing the iteration.

$PR(A)=0.15+0.85((1.099167/3)+(1.127264/3)+(0.780822/1)=1.444521$

$PR(B) = 0.15 + 0.85((1.444521/2)+(1.127264/3)) = 1.083313$

$PR(C) = 0.15 + 0.85((1.444521/2)+(1.083313/3)) = 1.07086$

$PR(D) = 0.15 + 0.85((1.083313/3)+(1.07086/3)) = 0.760349$

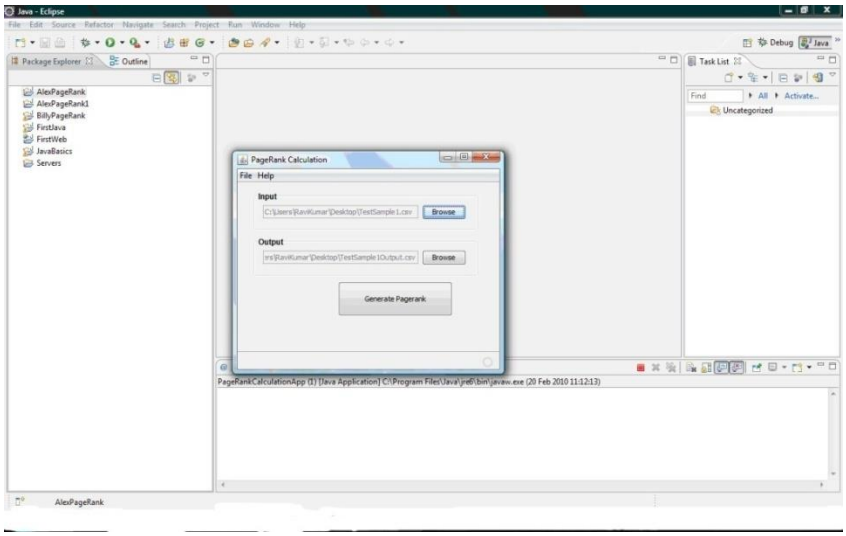


Figure 2.21: PageRank Program Input Entry Window

Table 2-3: PageRank Convergence Scores

Iteration	***A***	***B***	***C***	***D***
1	1	1	1	1
2	1.566667	1.099167	1.127264	0.780822
3	1.444521	1.083313	1.07086	0.760349
4	1.406645	1.051235	1.045674	0.744124
..	..	..	..	..
38	1.313509	0.988244	0.988244	0.710005
39	1.313509	0.988244	0.988244	0.710005
40	1.313509	0.988243	0.988243	0.710005

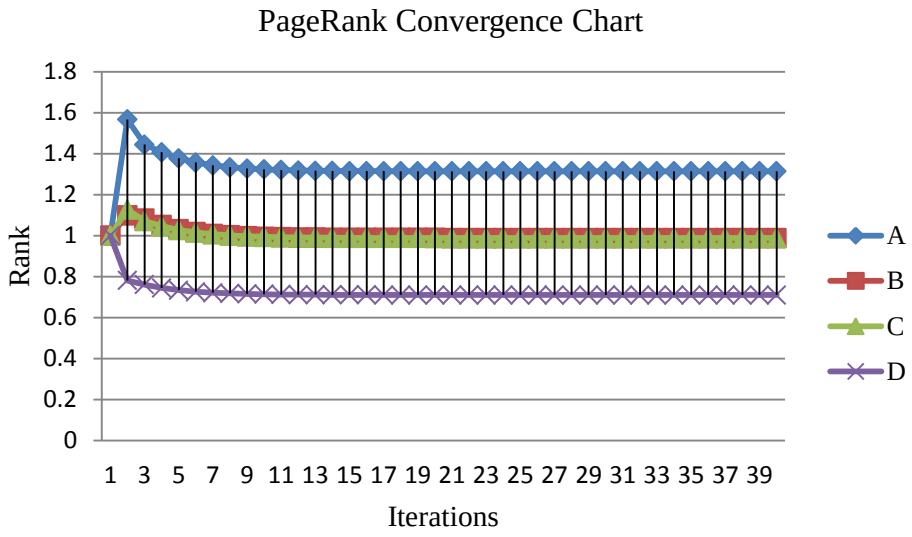


Figure 2.22: PageRank Convergence Chart

2.9.2 *Weighted PageRank Simulation*

The same hyperlink structure as shown in Figure 2.20 was used and the *WPR* calculated. The *WPR* equation for pages *A*, *B*, *C* and *D* are as follows.

$$WPR(A) = (1-d) + d(WPR(B).W^{in}_{(B,A)}.W^{out}_{(B,A)} + WPR(C).W^{in}_{(C,A)}.W^{out}_{(C,A)} + WPR(D).W^{in}_{(D,A)}.W^{out}_{(D,A)}) \quad (2.35)$$

$$WPR(B) = (1-d) + d(WPR(A).W^{in}_{(A,B)}.W^{out}_{(A,B)} + WPR(C).W^{in}_{(C,B)}.W^{out}_{(C,B)}) \quad (2.36)$$

$$WPR(C) = (1-d) + d(WPR(A).W^{in}_{(A,C)}.W^{out}_{(A,C)} + WPR(B).W^{in}_{(B,C)}.W^{out}_{(B,C)}) \quad (2.37)$$

$$WPR(D) = (1-d) + d(WPR(B).W^{in}_{(B,D)}.W^{out}_{(B,D)} + WPR(C).W^{in}_{(C,D)}.W^{out}_{(C,D)}) \quad (2.38)$$

The incoming link and outgoing link weights are calculated as follows:

$$W^{in}_{(B,A)} = I_A/(I_A + I_C) = 3/(3+2) = 3/5$$

$$W^{out}_{(B,A)} = O_A/(O_A + O_C + O_D) = 2/(2+3+1) = 2/6 = 1/3$$

$$W^{in}_{(C,A)} = I_A/(I_A + I_B) = 3/(3+2) = 3/5$$

$$W^{out}_{(C,A)} = O_A/(O_A + O_B + O_D) = 2/(2+3+1) = 2/6 = 1/3$$

$$W^{in}_{(D,A)} = I_A/(I_B + I_C) = 3/(2+2) = 3/4$$

$$W^{out}_{(D,A)} = O_A/O_A = 2/2 = 1$$

By substituting all the above values into Equation 2.35, *WPR* of Page *A* is computed by taking a value of 0.85 for *d* with the initial value of *WPR*(*B*), *WPR*(*C*) and *WPR*(*D*) = 1.

$$WPR(A) = (1-0.85) + 0.85(1*3/5*1/3 + 1*3/5*1/3 + 1*3/4*1) = 1.127$$

$$W^{in}_{(A,B)} = I_B/(I_B + I_C + I_D) = 2/(2+2+2) = 2/6 = 1/3$$

$$W^{out}_{(A,B)} = O_B/(O_B + O_C) = 3/(3+3) = 3/6 = 1/2$$

$$W^{in}_{(C,B)} = I_B/(I_A + I_B) = 2/(3+2) = 2/5$$

$$W^{out}_{(C,B)} = O_B/(O_A + O_B + O_D) = 3/(2+3+1) = 3/6 = 1/2$$

By substituting all the above values into Equation 2.36, *WPR* of Page *B* can be computed by taking *d* as 0.85 and the initial value of *WPR*(*C*) = 1.

$$WPR(B) = (1-0.85 + 0.85((1.127*1/3*1/2 + 1*2/5*1/2) = 0.499$$

$$W^{in}_{(A,c)} = I_C/(I_B + I_C + I_D) = 2/(2+2+2) = 2/6 = 1/3$$

$$W^{out}_{(A,C)} = O_C/(O_B + O_C) = 3/(3+3) = 3/6 = 1/2$$

$$W^{in}_{(B,C)} = I_C/(I_A + I_B) = 2(3+2) = 2/5$$

$$W^{out}_{(B,C)} = O_C/(O_A + O_C + O_D) = 3/(2+3+1) == 3/6 = 1/2$$

By substituting all the above values into Equation 2.37, *WPR* of Page *C* can be computed by taking *d* as 0.85.

$$WPR(C) = (1-0.85) + 0.85((1.127*1/3*1/2) + (0.499*2/5*1/2)) = 0.392$$

$$W^{in}_{(B,D)} = I_D/(I_B + I_C) = 2(2+2) = 2/4 = 1/2$$

$$W^{out}_{(B,D)} = O_D/O_A = 2/2 = 1$$

$$W^{in}_{(C,D)} = I_D/(I_A + I_B) = 2(2+3) = 2/5$$

$$W^{out}_{(C,D)} = O_D/(O_A + O_B + O_D) = 2/2 + 3 + 1 = 2/6 = 1/3$$

By substituting all the above values into Equation 2.38, *WPR* of Page *D* can be computed by taking *d* as 0.85.

$$WPR(D) = (1-0.85) + 0.85((0.499*1/2*1) + (0.392*2/5*1/3)) = 0.406$$

2.9.3 *Simulation Results Discussion*

During the 40th iteration, the PageRank gets converged and the convergence computation ranks are shown in Table 2-3 in the simulation section. The complete convergence rank table is given in Appendix A.

For a smaller set of pages, it is easy to calculate and find out the PageRank values

but for a Web having billions of pages, it is not easy to do the calculation as above. Table 2-3 shows the PageRank of *A* is higher than that of *B*, *C* and *D*. This is because page *A* has 3 incoming links, while pages *B*, *C* and *D* have 2 incoming links as shown in Figure 2.20. Page B has 2 incoming links and 3 outgoing links; page C has 2 incoming links and 3 outgoing links and page *D* has 1 incoming link and 2 outgoing links. It can be seen from Table 2-3, after iteration 40, that the PageRank for the pages gets normalized. Previous experiments (Page et al. 1999; Ridings and Shishigin 2002) showed that the PageRank gets converged to a reasonable tolerance. The convergence of the PageRank calculation is depicted as a graph in Figure 2.22 in the Simulation Result section.

In the WPR, the order of PageRank values is *A, B, D* and *C*. These results show that the page rank order is different from PageRank because WPR do not divide the rank value of a page evenly among its outgoing linked pages rather it assigns a larger rank values to more relevant pages.

2.10 SUMMARY

From a careful review of the published literature, it is clear that although several studies on link analysis Algorithms, PageRank computation, hanging pages and Web spam are reported in literature, but no study has been done on including the relevant hanging pages in the PageRank computation and the contribution of link spam by hanging pages. In addition, most of the existing methods exclude the hanging pages in the rank computation and they get only a minimum rank which is not fair for the relevant hanging pages and they deserve a better rank.

So far, it is found that several researches have neglected the effect of hanging pages in the contribution of link spam because of the assumption that it had small or negligible effects in the link spam. When more and more hanging pages are connected together, they can form an effective link spamming which can affect the rank of Web pages. Hence, it is very important that the hanging pages have to be identified and handled to avoid the link spam. Furthermore, the literature review in this chapter emphasises the importance of the present research, highlighting the research objectives outlined in Chapter 1.

Chapter 3 Methodologies to Handle Hanging Pages

3.1 INTRODUCTION

A random surfer normally surfs the Web, going from one page to another, by randomly choosing a forward link. When a page does not have any forward link or when the surfer gets bored, then he/she chooses a page by other means, like typing a page in URL of a browser. A page that does not have any forward or outgoing links is called a hanging page. Hanging page can be also called dangling page, zero-out-link page, dead end page, sink page etc. These hanging pages are one of the hidden problems of link structure based ranking algorithms, because they do not propagate the rank scores to other pages; this is an important feature in the link structure based ranking methods. Hanging pages keep growing in the Web (Eiron, McCurley and Tomlin2004), and they cannot be left out during the ranking process, because they may contain quality and relevant information.

According to Langville and Meyer (2004; 2006c), the theoretical random walk of the Web can be considered as the *Markov Chain* or *Markov process*. The limiting probability that a dedicated random surfer visits any particular page is its PageRank. A page has a higher PageRank if it has links to and from other pages with high rank as well. Studies on the PageRank algorithm and Hanging Pages have been conducted by the following researchers: Page et al. (1999), Langville and Meyer (2004; 2005; 2006b; 2006c), Ridings and Shishigin (2002),Eiron, McCurley and Tomlin (2004),Bianchini, Gori and Scarselli (2005),Wang et al. (2008),Lee, Golub and Zenios (2003)and Gleich et al. (2010).

In this chapter, the effect of hanging pages in PageRank computing is described first using a sample Web graph. After analysing the effects of hanging pages, two methods are proposed to handle hanging pages. The first method introduces a *VirtualNode* (*VN*) with self-loop, where all the hanging pages are connected to the *VN* in the Web

graph. The second method also uses the *Virtual Node* (*VN*) with self-loop, with all the pages including hanging and non-hanging pages are connected to the *VN* in the Web graph here. The reasons for proposing the above methods are:

- To handle the hanging pages to avoid the rank sink of the hanging pages.
- To make the hanging pages as non-hanging by connecting them to the *Virtual Node*.
- To get fair and decent rank for the hanging pages.

PageRank program was created and tested with the EU2010 dataset. Programs were also created for Methods 1 and 2 and these two methods were compared with each other, and also with the PageRank program. TrustRank algorithm is also implemented and tested with the proposed Method 1 and 2 so that the proposed methods can combat Web spam.

3.2 Effect of Hanging Pages in PageRank Computing

This section describes the effects of hanging pages in PageRank computing. In Section 2.3, the original definition about hanging page is given. The second part of the definition says that the hanging pages do not affect the ranking of any other page directly which is not true. When removing hanging pages, other pages may become hanging and also it affects the rank of neighbouring pages. The following example proves that removing hanging pages affects the rank of neighbouring pages and also it makes other pages to become hanging.

A sample directed graph with 5 nodes is shown in Figure 3.1. The non-hanging pages are shown in blue colour. Page *B* is a hanging page (no out link from page *B*) and is shown in red colour. PageRank is computed for the above graph using the PageRank program. In the first, computation is done by including the hanging pages. The hanging page *B* is having a rank of 0.562. In the second, PageRank computation is done without including the hanging page *B* and page *B* gets only a minimum rank of 0.15. When removing hanging page *B*, page *E* became hanging as shown in Figure 3.2. Also the neighbouring page's rank gets affected as shown in Table 3-1.

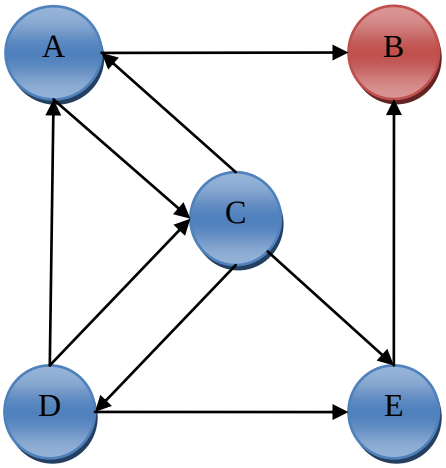


Figure 3.1: Sample Directed Web Graph with 5 Nodes

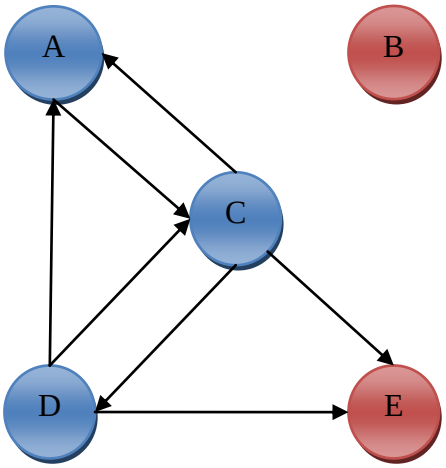


Figure 3.2: Sample Directed Web Graph with 5 Nodes without Hanging Pages

Table 3-1: Effect of Hanging Pages in PageRank Computation

Page	***PageRank with Hanging Pages***	***PageRank w/o Hanging Pages***
A	0.323	1.0
B(H/P)	0.562	0.15
C	0.358	1.298
D	0.252	0.702
E	0.322	0.15

The above example proves that when removing hanging pages, other pages may become hanging and their rank gets affected. In the Web, there are billions of pages and nearly half of them are hanging pages. PageRank algorithm generally leaves the hanging pages and computes the PageRank to reduce the computational complexity. It is important to handle the hanging pages because there are many relevant and quality hanging pages in the Web and they deserve a better rank. There are two methods proposed in this Chapter to handle the hanging pages.

3.3 PROPOSED METHODS

For this research study, two methods are proposed, whereby *VN* is used to handle the hanging pages in the Web graph (Singh, Kumar and Leng 2010; 2011). The Web is organized as a directed graph $G(V, E)$ with a vertex set of V of N pages and a directed edge set E. This directed graph is called Web graph, which can be represented as a matrix. The PageRank creates the graph and matrix before it computes the rank.

In Method 1, a *VN* with self-loop is connected and all the hanging pages are connected to it; this is a similar approach to Bianchini, Gori and Scarselli(2005).In Method 2, a *VN* with self-loop is connected and all the pages including hanging and non-hanging pages are connected to it.

The basic PageRank model treats the whole Web as a directed graph. The following probability matrix, *PV,* with *VN* is an $m \times m$ matrix where $m = (n + 1)$, i.e. the last column and the last row is for the *VN*, which is used for dealing with the hanging pages (Singh, Kumar and Leng 2010).

$$PV = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{vn,1} & p_{vn,2} & \cdots & p_{vn,n} \end{bmatrix}$$

A sample directed graph with 6 nodes is shown in Figure3.3. There are 6 nodes in the directed graph. Nodes *A, B, D* and *E* are non-hanging pages (shown in blue), while nodes *C* and *F* are hanging nodes (shown in red), i.e. they do not have any forward

links. This study shows the effect of hanging nodes in the PageRank computation and, how the neighbouring page ranks get affected by the hanging pages.

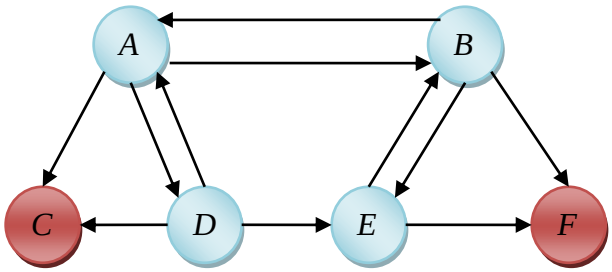


Figure 3.3: A Directed Web Graph *G* with 6 Nodes

The Markov analysis can be defined as any system, which uses the Markov chain to predict the probability of the future state, by taking only the current state. The Markov analysis is commonly used in Biology, Economics, Engineering, Physics and Computer Science as well. For instance, the Google search engine internally follows the Markov chain even though this fact was not mentioned in the first PageRank study (Brin and Page 1998). However, the other researchers (Langville and Meyer 2005; Bianchini, Gori and Scarselli 2005) have proved that Google's PageRank algorithm follows the Markov chain, which uses only a matrix and a vector for modelling and prediction.

The PageRank model uses the random walk theory on the Web graph by randomly moving from one node to another to compute the rank of a page. Here, some nodes are visited more often than others because they have more back links; thus, they are important pages. When a hanging node comes, the random walk cannot proceed further; other than moving from one node to another, it can only progress if the user types the URL on a Web browser. The Stochastic interpretation of PageRank therefore, works only when there are no hangings pages (Bianchini, Gori and Scarselli 2005). But in reality, there are many hanging pages on the Web (shown in the transition probability matrix below), and they cannot be ignored in the PageRank computation due to their importance.

3.3.1 *Transition Probability Matrix Representation*

The transition probability matrix *P* for the graph *G* in Figure 3.3 is shown as follows. It can also be called a hyperlink matrix and it is an $n \times n$ matrix, where n is the

number of Web pages. If Web page i has $d_i \geq 1$ links to other Web pages and Web page i links to Web page j, then the element in row i and column j of P is $P_{ij} = 1/d_i$, where d_i is the number of forward links of Web page i; otherwise, $d_{ij} = 0$. Thus, P_{ij} represents the likelihood that a random surfer will select a link from Web page i to j.

The transition probability matrix P, shown below is produced for Web Graph G in Figure 3.3 by applying Equation 2.5 of Definition 2.5 from Chapter 2:

$$P = \begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 0 & 1/3 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

In the above transition probability matrix P, rows 3 and 6 have only zeros. It means that nodes C and F are hanging nodes and the probability of a random surfer moving from nodes C and F to any other nodes in the directed graph is zero. Matrix P is not stochastic, and it needs to be stochastic as per the PageRank model.

3.3.2 *Method 1*

Method 1 is as follows; a virtual node, *VN*, with self-loop is first connected and then all the hanging nodes are connected to it (shown in orange), as seen in Figure 3.4; the corresponding matrix $PV1$ which is stochastic now, is shown below.

The PageRank formula in Equation 3.1 is applied to compute the PageRank for the graph structure shown in Figure 3.4. The sample calculation for both the Methods 1 and 2, i.e. hanging pages connected to the *VN* and all the pages connected to the *VN*, are shown below.

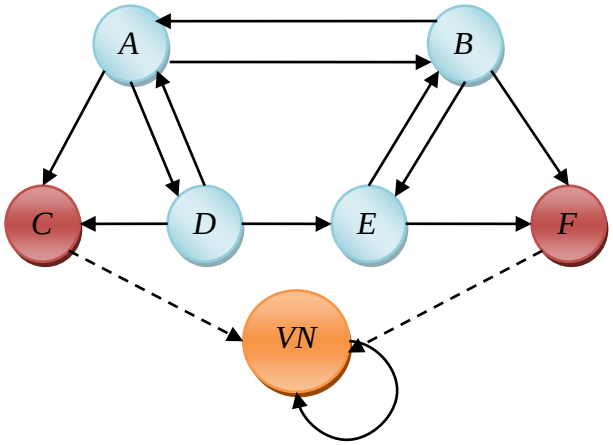


Figure 3.4: Directed Graph with Virtual Node *VN* Using Method 1

$$PV1 = \begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1/3 & 0 & 1/3 & 0 & 1/3 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$PR(p) = d \sum_{q \in pa_p} \frac{PR_q}{O_q} + (1-d) \tag{3.1}$$

In Equation 3.1, p is an arbitrary page that has back links from set of pages pa. O_q is the number of forward links of page q and d is the damping factor such that $0<d<1$, and it is usually set to 0.85. The detailed computation convergence and the chart are shown in Appendix B.

In the PageRank calculation, node A gets back links from only nodes B and D, as shown in Equation 3.2. Only the forward links are different for Methods 1 and 2.

$$PR(A) = d\left(\frac{PR(B)}{O(B)} + \frac{PR(D)}{O(D)}\right) + (1-d) \tag{3.2}$$

It is assumed that the initial page rank of all the nodes as 1 and the damping factor *d* is 0.85. This calculation continues until the PageRank for all the nodes converge. The experimental section shows the PageRank computation using Method 1,with the actual Web data. A sample calculation is shown below:

$$PR(A) = 0.85\left(\frac{1}{3}+\frac{1}{3}\right)+(1-0.85) = 0.717$$

3.3.3 *Method 2*

In Method 2, a Virtual node, *VN*, with self-loop is connected and all the pages including hanging and non-hanging pages are connected to it. The directed Web graph shown in Figure3.3has been modified using Method 2 and is shown below in Figure 3.5:

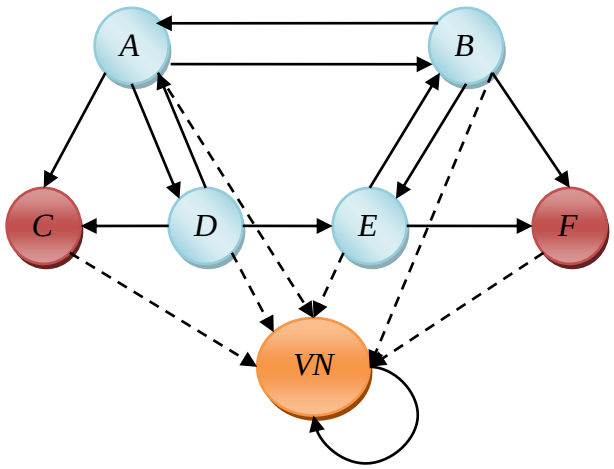


Figure 3.5: Directed Graph with Virtual Node *VN* Using Method 2

The corresponding transition probability matrix *PV*2, for Method 2 is shown below. In the transition probability matrix below, the last column i.e. the virtual node has more transition probability because every node is connected to it. This makes the PageRank of the *VN* increase at a fast rate and does not affect the overall rank of the pages.

$$PV2 = \begin{bmatrix} 0 & 1/4 & 1/4 & 1/4 & 0 & 0 & 1/4 \\ 1/4 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 \\ 0 & 1/3 & 0 & 0 & 0 & 1/3 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The same PageRank formula shown in Equation 3.1 is applied to the directed Web graph in Figure 3.5 using Method 2. Here, every node gets an additional forward link because all the nodes are connected to the virtual node. In this method the PageRank values are reduced for all the nodes but the virtual node gets more back links and its PageRank score increases. In the final ranking order, the virtual node will not be shown. The PageRank of *A* decreases from 0.717 to 0.575, due to the rank distribution; the rank goes down uniformly for all the pages without affecting the order.

$$PR(A) = 0.85\left(\frac{1}{4} + \frac{1}{4}\right) + (1 - 0.85) = 0.575$$

The above calculations are the first iteration of the PageRank computation for Methods 1 and 2, with the PageRank converging after so many iterations. The detailed computation convergence and the chart are shown in Appendix B.

3.4 EXPERIMENTAL RESULTS

The PageRank and the TrustRank program were implemented in the Python program and tested on an Intel Core 2 (2.40 GHz) with 4GB RAM.

3.4.1 *Data Set*

The dataset used in the experiments was provided by the European Archive Foundation, with the support of the Living Web Archives (LiWA) project, known as the EU2010 collection (Benczúr et al. 2010). In this experiment, a host graph was used instead of a Web graph due to the large dataset collection. The original Web graph contains 23m Web pages, while the host graph contains 191388 hosts and 103749 hanging hosts (hosts that are not pointing to other hosts) depicted in Figure

3.6. The table form is shown in Appendix B. In this experiment, the rank of all hanging hosts for Methods 1 and 2 was shown and, the Web Spam detection algorithm, TrustRank (Gyongyi, Garcia-Molina, and Pedersen 2004), applied on the same dataset and compared with the results from the TrustRank with virtual node.

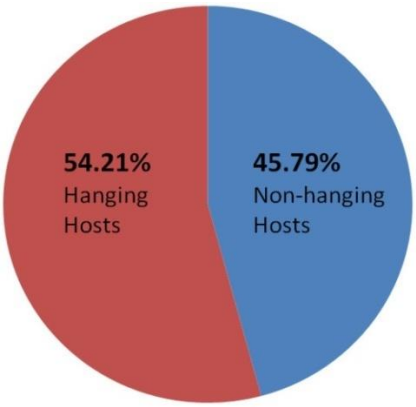


Figure 3.6: Distribution of Hanging and Non-Hanging Hosts

3.4.2 *Pseudo Code*

The following pseudo code given in Figure 3.7 is the same as the PageRank algorithm. The program was implemented using this pseudo code for Methods 1 and 2.

```
Main Procedure
Initialize checkIteration is true
DO
    Call PageRank to calculate the PageRank for every node
    Save the PageRank for every node
    If the PageRanks of last Iteration has the same PageRanks with current Iteration
    checkIteration is false
WHILE quits when checkIteration is false
Procedure PageRank
    Initialize result to 0.15 (1 - the damping factor)
FOR every outgoing nodes of the current node
    Call Calc
Add up result with the results from Calc of all outgoing nodes
Procedure Calc
    Calculate the result by getting the PageRank of the current node divide by the
    numbers of outgoing links of the current node times 0.15 (1 - the damping factor).
```

Figure 3.7: Algorithm to Handle Hanging Hosts using Methods 1 and 2

3.4.3 *Experiments*

For the experiments, a value of 0.85 was used for the damping factor *d*as per the recommendation from many researchers (Langville and Meyer 2004, Bianchini, Gori and Scarselli 2005 and Gleich et al. 2010) and run in 50 iterations, which were sufficient to achieve convergence, because the simulation example in Section 3.3 took less than 50 iterations except for Method 2.The experiments are conducted in two stages. The first one is using the sample Web graph in Figure 3.3 and the second one is using the EU2010 data set.

3.4.3.1 *Experiments with the Web Graph*

First, the PageRank program is applied to the Web graph in Figure 3.3 before applying Method 1 and Method 2. PageRanks are computed and the convergence is shown in Figure 3.8. PageRank convergence table is shown in Appendix B.

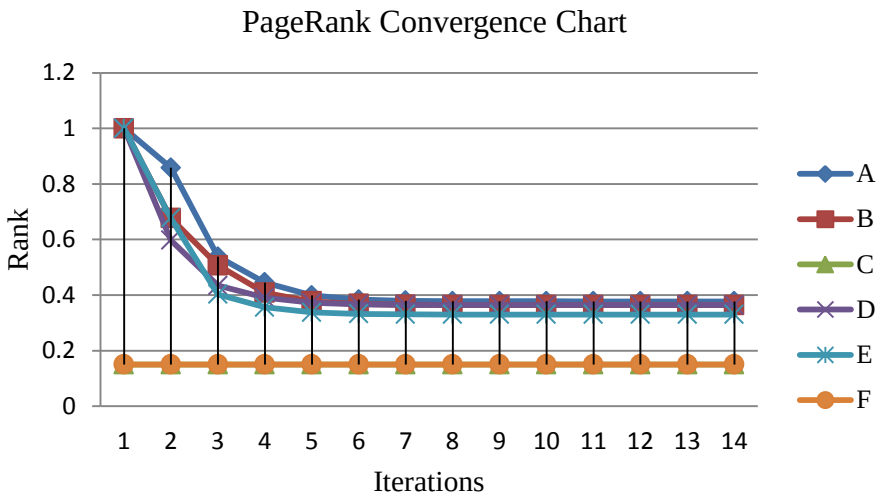


Figure 3.8: Convergence chart for the PageRank

In Figure 3.8, Y axis shows the PageRank and the X axis shows the iterations. PageRank has converged at the 14th iteration. Next, the PageRank algorithm is applied to the modified Web graph in Figure 3.4 using Method 1 (with *VN*). PageRanks are computed and the convergence is shown in Figure 3.9. PageRank convergence table is shown in Appendix B.

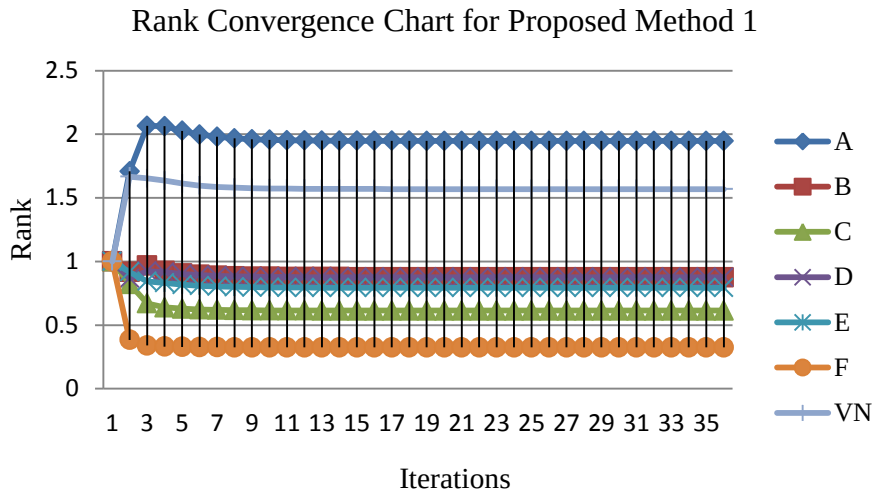


Figure 3.9: Convergence Chart for the Proposed Method 1

In Figure 3.9, Y axis shows the PageRank and the X axis shows the iterations. PageRank has converged at the 36th iteration. Finally, the PageRank algorithm is applied to the modified Web graph in Figure 3.5 using Method 2 (with *VN*). PageRanks are computed and the convergence is shown in Figure 3.10. PageRank convergence table is shown in Appendix B.

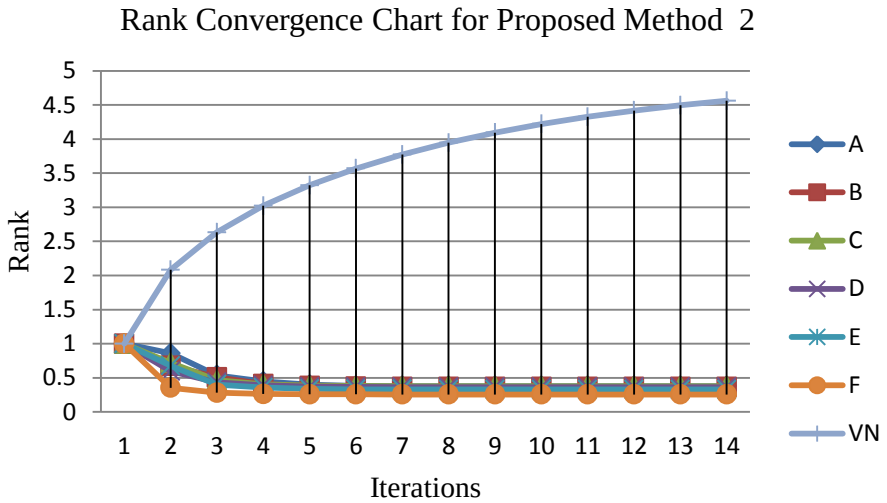


Figure 3.10: Convergence Chart for the Proposed Method 2

In Figure 3.10, Y axis shows the PageRank and the X axis shows the iterations. PageRank has converged only at the 95th iteration. Actually, PageRank has converged at the 14th iteration itself, only the Virtual Node (*VN*) converged at the 95th iteration.

The graph in Figure 3.10 shows only the convergence up to 14th iteration (except the *VN*).

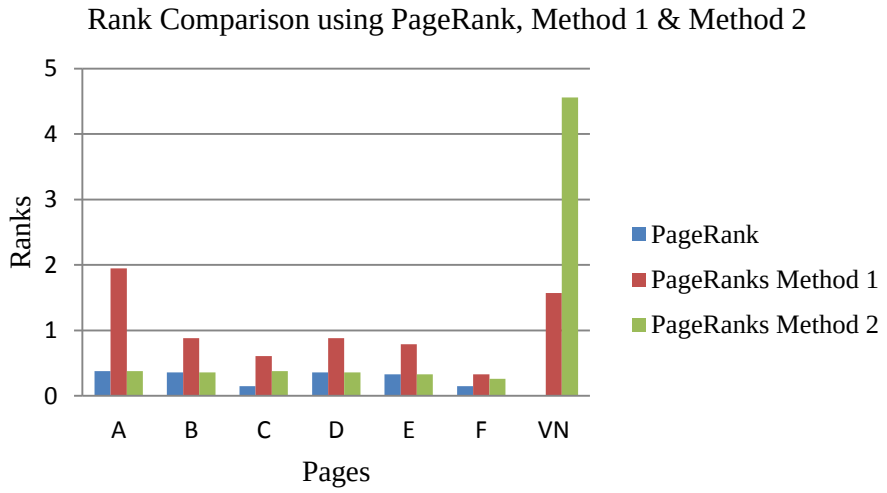


Figure 3.11: Rank Comparison Using PageRank, Method 1 and Method 2

Figure 3.11 shows the rank comparison using PageRank, proposed Method 1 and Method2. Here, X axis shows the pages and Y axis shows the rank.

3.4.3.2 *Experiments with EU2010 Data Set*

Next, experiments are done using the EU2010 data set. This data set is using hosts instead of pages. PageRank is computed for Method 1 and Method 2.

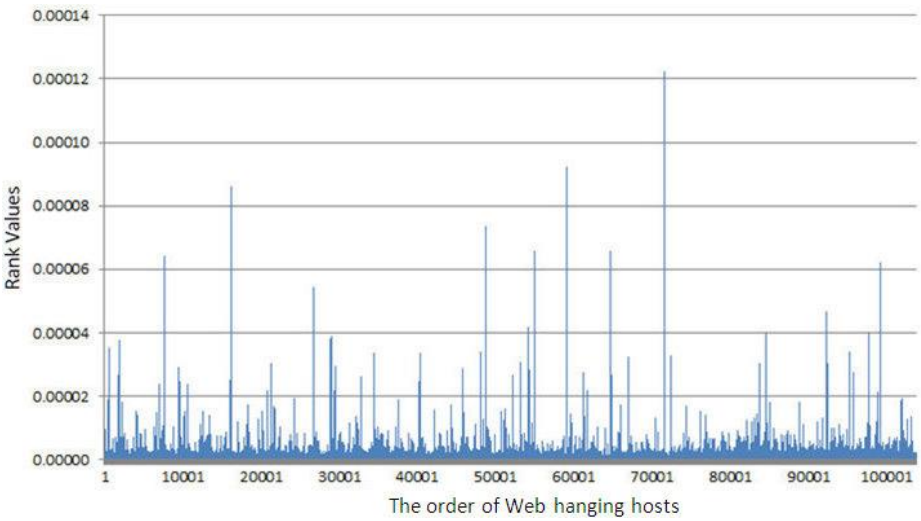


Figure 3.12: Ranking Results of Hanging Hosts for Method 1

In Method 1, a *VN* with self-loop was included and all the hanging hosts were connected to it. Figure 3.12 shows the rank results of the hanging host for Method 1.The Y axis denotes the rank values of the hanging pages, while the X axis denotes the 1sthanging node until the 103749th hanging node (the last node).

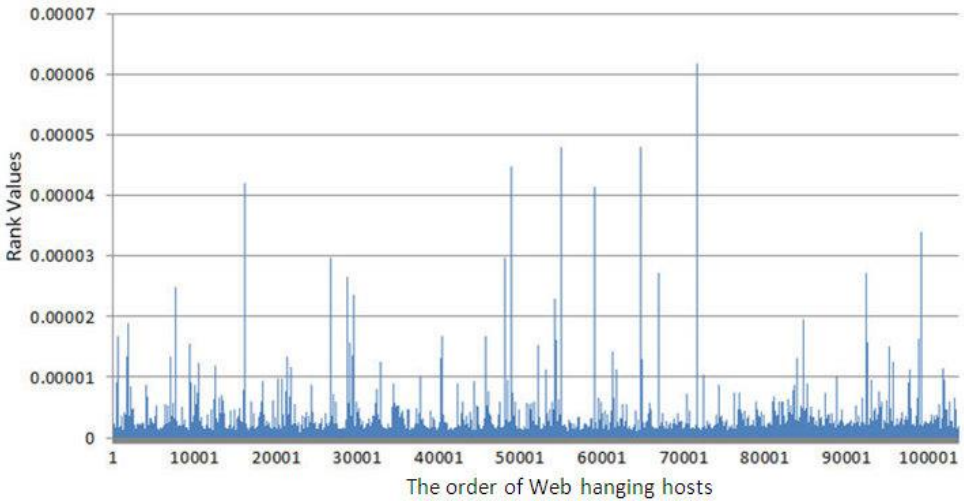


Figure 3.13: Ranking Results of Hanging Hosts for Method 2

In Method 2, all the nodes were connected to the *VN* to make the forward link uniform for ranking purposes. Figure 3.13 shows the rank results of the hanging host for Method 2.

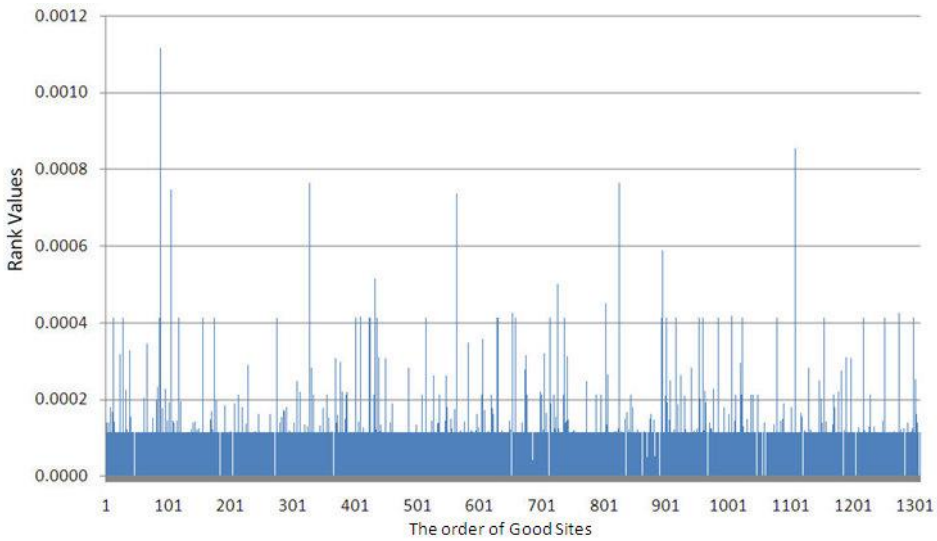


Figure 3.14: Ranking Results from TrustRank

Figure 3.14 shows the ranking results on the good sites on EU2010 using TrustRank,

while Figure 3.15 shows the ranking results on the good sites on EU2010 with *VN*. A sample of 1309 good sites provided by the dataset was tested to see the difference between TrustRank and TrustRank with *VN*.

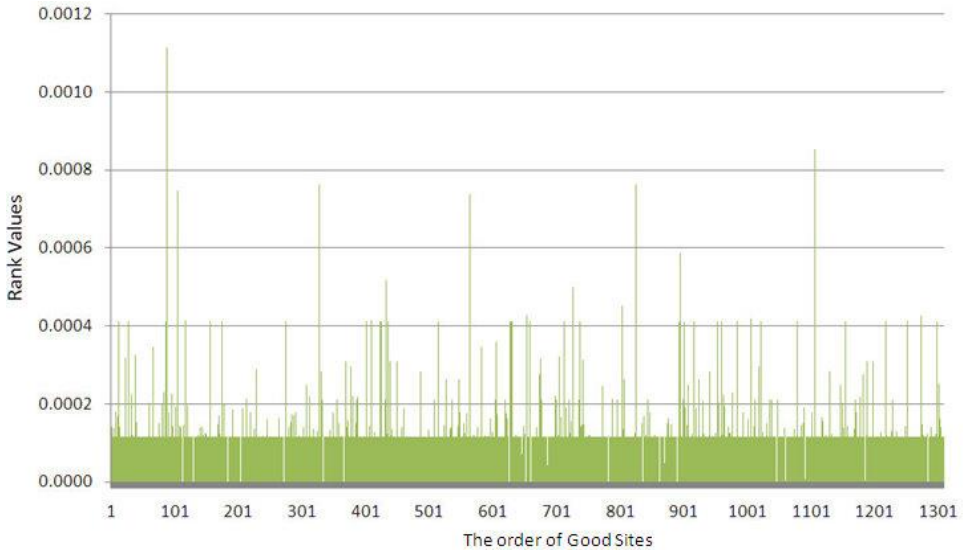


Figure 3.15: Ranking Results from TrustRank with Virtual Node

3.4.4 *Result Analysis*

Original PageRank

When the original PageRank program is applied on to the Web graph, shown in Figure 3.3, hanging pages *C* and *F* gets only a minimum PageRank of 0.15. They may deserve a better ranking. The PageRank get converged in 14^{th} iteration.

Method 1

In the Method 1, *VN* with self-loop was included and all the hanging pages are connected to it. PageRank program is applied on to the modified Web graph, shown in Figure 3.4. Here, the hanging pages *C* and *F* get a decent rank (0.614 and 0.324) and the convergence occurs at the 36^{th} iteration. When compare with original PageRank, Method 1 produces decent rank for hanging pages with a moderate convergence rate. The same analysis goes for the EU2010 data set. PageRank results are shown in Figure 3.12. Here, the page ranks are fair by including all the hanging hosts in the ranking with a moderate convergence. The output (Figure 3.12) in a chart format shows the rank values on the Y axis and the order of hanging hosts on the X axis.

Method 2

In Method 2, all the hanging hosts as well as the non-hanging hosts were connected to the *VN* to make the out link uniform for ranking purposes. PageRank program is applied on to the modified Web graph, shown in Figure 3.5. Here, the hanging pages *C* and *F* get a decent rank (0.377 and 0.257) and the convergence occurs only at the 95^{th} iteration due to the Virtual Node (*VN*). Method 2 also produces decent rank for hanging pages but with a high convergence rate. The same analysis goes for the EU2010 data set also. The output is shown in Figure 3.13 in a chart format. The page rank value reduced a little bit here, compared with Method 1, because the forward links of all the hosts were connected to the *VN*. The original PageRank method was not suitable as far as the hanging pages were concerned because they were omitted in the computation. In Method 1, the hanging pages gets a decent rank and the number of iterations was less when compared with Method 2. The results proved that Method 1 is better when compared with PageRank and Method 2 because it not only reduces the number of iterations in the computation but also produces a fair and accurate ranking of results as for as the hanging pages are concerned. It is very clear that there are more hanging than non-hanging pages on the Web (54% are hanging pages in the sample data set) and this rate keeps increasing. The hanging pages therefore, cannot be neglected in the ranking process due to their importance.

In Figures3.11 and 3.12, there is no significant difference with/without *VN* in the calculation of TrustRank. Methods 1 and 2 are, therefore capable of combating Web spam with the inclusion of TrustRank.

3.4.5 *Computation of Complexity*

The basic PageRank model treats the whole web as a directed graph $G = (V, E)$, where, a set V of vertices consists of n pages, and the set E of directed edges (i, j), which exist if and only if page i has a hyperlink to page j. The directed graph can be represented as an $n \times n$ matrix.

According to Augeri (2008) and Safronov and Parashar (2003),where the PageRank algorithm for every Web page is concerned, one needs to find all the pages to which the new page links. This requires a full array of scan so that every element is

checked. If n is the number of Web pages, an assumption can be made, that n-1 is the maximum number of links on a page; therefore, the worst case performance is $O(n^2)$. The PageRank algorithm which is essentially a power method algorithm has a lower and upper bound of $\Omega(n^2 \log n)$ and $O(n^2 \cdot t)$, where $n = |V|$, $t = \log_d \tau$, d is a damping factor, usually 0.85 and $\tau = 1/n$. If sparse matrices are used, the lower and upper bound of the PageRank algorithm are $\Omega(e^2 \log n)$ and $O(e \log_d \tau)$respectively, where e denotes the number of edges contained in the graph.

The proposed method for handling hanging pages involves adding a *VN* into the calculation. A *VN* can be denoted by ε_v, so the lower and upper bound of $\Omega(n^2 \log n)$ and $O(n^2 t)$, where $n = |V + \varepsilon_v|$, $t = \log_d \tau$, d is a scaling factor, usually 0.85 and $\tau = 1/n$, d. This would not affect the PageRank algorithm, and by adding the *VN*, it actually takes all the hanging pages into account. Intuitively, the proposed algorithm (Methods 1 and 2) has the same computation power as the PageRank algorithm, and produced more relevant results by including hanging pages into consideration.

Complexity Calculation

Computing the PageRank is actually populating the matrix and then calculating its principal eigenvector. It is calculated using matrix-vector multiplication and addition. The cost of multiplying an $n \times m$ matrix by an $m \times p$ matrix is $O(nmp)$. In this case, it is $n \times n$ by $n \times 1$ matrix and the complexity is $O(n^2)$.

3.5 SUMMARY

This chapter has proposed two methods, Methods 1 and 2 using a Virtual Node (*VN*), to calculate PageRank in dealing with the problem of hanging pages. Method 1 took less iteration and also produced a fair and accurate ranking of pages compared with Method 2. Both Methods 1 and 2 produced relevant results when compared with the original PageRank algorithm. But both methods took more iteration to converge when compare with PageRank algorithm. Most Web ranking algorithms are kept as trade secrets due to competition, so it is difficult to know how the ranking algorithms are implemented in reality. But with the limited resources, the PageRank algorithm was implemented and it handled the hanging pages efficiently. The TrustRank algorithm was also implemented to combat spamming in the proposed Methods 1 and 2, compared with each other and then compared with the standard PageRank

algorithm.

The next chapter discusses the experiments that deal with only the relevant hanging pages, instead of including all the hanging pages in the rank computation.

Chapter 4 Relevancy of Hanging Pages

4.1 INTRODUCTION

As more and more meaningful hanging pages increases in the Web, their relevancy has to be determined according to keywords or query terms, to make the Search Engine Result Pages (SERPs) fair and relevant. In this chapter, an algorithm called Hanging Relevancy Algorithm (HRA) is introduced and implemented to determine the relevancy of hanging pages in the link structure based ranking algorithms(Kumar et al. 2014). This method includes the relevant hanging pages in the ranking algorithm along with the non-hanging pages to reduce the complexity over Methods 1 and 2. The relevancy function is used to determine the relevancy of a hanging page with respect to keywords or query terms. Stability analysis is also done to show that the perturbation of link structure does not affect the order of the perturbed pages. The hanging relevancy algorithm, therefore, is the first kind of approach in determining the relevancy of hanging pages, and includes only the relevant hanging pages in the ranking process. This algorithm is a trade-off between complexity and relevancy by increasing computational complexity and at the same time producing more relevant results. The architecture of the proposed hanging relevancy method is shown in Figure 4.1.

All the existing methods to handle hanging pages either exclude the hanging pages or include the hanging pages in the rank computation. If all the hanging pages are included in the computation, the computation complexity increases. If they are excluded in the ranking, the ranking results are not fair and relevant. To make a trade-off between complexity and fair results, the proposed method includes only the relevant hanging pages in the ranking process. The important ranking methods are tabulated in Table 4-1, according to computational complexity. It is assumed that there are N number of Web pages, which consists of N_1number of non-hanging pages, and N_2number of hanging pages.

Table 4-1: Inclusion of Hanging Pages in Computing

Ranking Method	***Inclusion of Hanging Pages***	***Computational Complexity***
Page et al. (1999)	No	$O(N_1^2)$
Kamvar et al. (2003)	No	$O(N_1^2)$
Lee Golub and Zenios* (2003)	Yes	$O(N_1^2)+O(N_2^2)$
de. Jager and Bradley* (2009)	Yes	$O(N_1^2)+O(N_2^2)$
Ipsen and Selee* (2007)	Yes	$O(N_1^2)+O(N_2^2)$
Bianchini, Gori and Scarselli** (2005)	Yes	$O(N^2)$
Singh, Kumar and Leng ** (2011)	Yes	$O(N^2)$

---
* separates hanging and non-hanging pages and computes the rank using matrix lumpability.
** includes both hanging and non-hanging pages in the computing

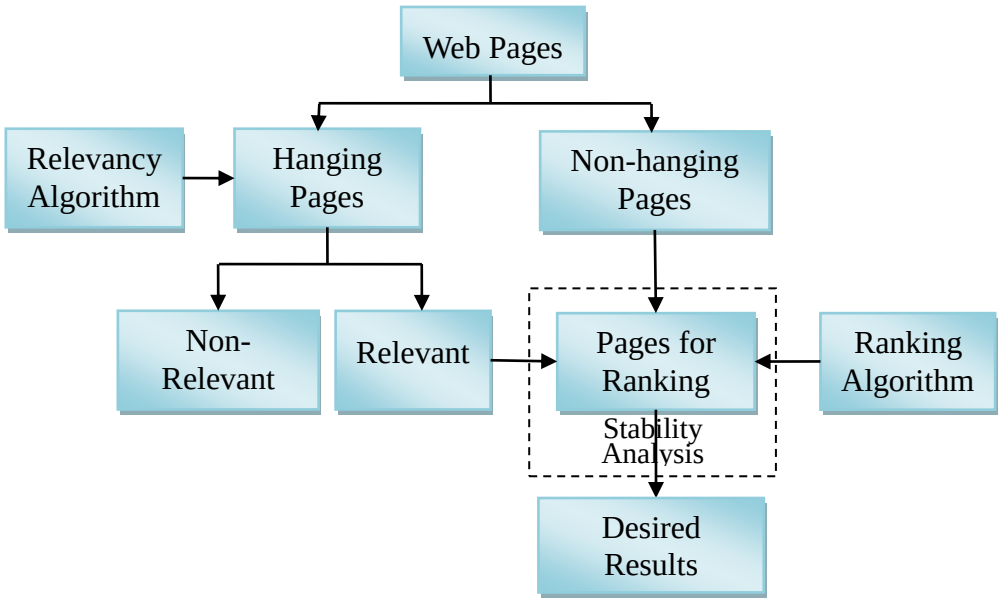


Figure 4.1: Architecture of the Proposed Hanging Relevancy Method

Web Pages contain both hanging pages and non-hanging pages. The proposed

Relevancy Algorithm is applied to the hanging pages and determines their relevancy according to keywords or query terms, by looking into their Anchor Text (AT). Stability analysis is applied to the link structure to make sure that the alteration of links does not change the order of the non-hanging pages. In this way the proposed method produces fairer and more relevant results compared to other link structure based ranking algorithms. For this study, experiments were done on the WEBSPAM UK 2006, WEBSPAM UK 2007 and EU2010 data sets, to determine the percentage of hanging and non-hanging hosts. A crawler program was created and it crawled the Curtin University (Sarawak) Web site. In the downloaded pages, hanging pages and non- hanging pages were separated and the PageRank program applied; the results are shown.

4.2 ANCHOR TEXT

The Anchor text which is the visible hyperlink text on a Web page is usually used to indicate the subject matter of the page to which it links. According to Zhicheng et al. (2009), the initial purpose of the anchor text is for users to navigate from one page to another, and to describe briefly the document content. Eiron and McCurley (2003) calls anchor text as 'highlighted clickable text'. Proper use of anchor text can increase the visibility of a page, and in turn increase the page rank. Anchor text is one of the important ranking factors in link structure based search engines. Inclusion of keywords in the anchor text can increase the value of a target page. In this study, the algorithm uses the anchor text to find the relevancy of hanging pages according to keywords or user query terms. There are two important reasons for using anchor text to find the relevancy of hanging pages:

- The first one is, generally anchor text describe the target document short and precise which exactly the way a user type a query in the interface of the Search engines.
- The second reason for selecting anchor text is, it is one of the important link based ranking factors and this research is based on the link structure ranking algorithm.

4.3 HANGING RELEVANCY USING RELEVANCY ALGORITHM

Let $G_w(V_w, E_w)$ be a Web graph consisting of non-hanging and hanging pages that can be determined by creating a transition probability matrix, using Equation 2.5 of Definition 2.5from Chapter 2.

Relevant hanging pages are pages without any outgoing links with relevancy to a particular keyword or query. On the other hand, non-relevant hanging pages have no relevancy to the keyword or query term. The objective of this research study is to determine whether a hanging page is relevant to a particular query term or not. All the hanging nodes in the graph will be determined, either as a relevant hanging node or non-relevant hanging node by applying the Relevancy Algorithm.

4.3.1 *Methodology*

Let $G_w = (V_w, E_w)$ be a *Web graph* with vertices V_w as the set of Web pages and E_w as the hyperlink between pages. Reference can be made to Chapter 2 for the generalized $n \times n$ probability transition matrix P and the definitions.

Definition 2.6 from Chapter 2, $\sum p_{(i,j)} = 0$, will be utilised i.e. to find out the hanging pages from a Web graph using the probability matrix. This definition can be used to determine whether a page in the Web graph G_w is a hanging or non-hanging page.

After a page is determined as a hanging page, the following Relevancy function in Equation 4.1is applied to graph G_w to determine the relevancy of that hanging page for a specific query term.

$$R(p) = \begin{cases} \text{if } AT(G_w(V_w, E_w)) = QT \text{ then} \\ \quad \text{connect the hanging page to the home page} \\ \text{else} \\ \quad \text{discard the hanging page} \end{cases} \qquad (4.1)$$

In the above Relevancy function, *AT* is the Anchor Text of a hanging page and *QT* is the Query Term or keyword, which is used by the relevancy algorithm to check the relevancy of a hanging page. The relevancy function compares the query term (*QT*) with the anchor text (*AT*) of a hanging page. If the anchor text exactly matches the query term, then the hanging page is connected back to the home page to make it

stochastic. Otherwise, the hanging page is discarded from the probability matrix *P*. This modified probability matrix is defined as the proposed probability matrix *PP*. The generalized form of the proposed probability matrix *PP* consists of only the non-hanging pages and the relevant hanging pages (for a query term).

$$PP = \begin{bmatrix} pp_{1,1} & pp_{1,2} & \cdots & pp_{1,n} \\ pp_{2,1} & pp_{2,2} & \cdots & pp_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ pp_{n,1} & pp_{n,2} & \cdots & pp_{n,n} \end{bmatrix}$$

This proposed probability matrix, *PP*, is a fairer and more relevant matrix than probability matrix *P*, because it has only fewer pages to compute by including only the relevant pages in the computing. The hanging relevancy methodology is shown below in Figure 4.2. The hanging pages are selected from the graph and the relevancy algorithm is applied to determine the relevancy. Link adjustments are done for the hanging pages which passed the relevancy test to become non-hanging pages. Other non-relevant hanging pages are discarded.

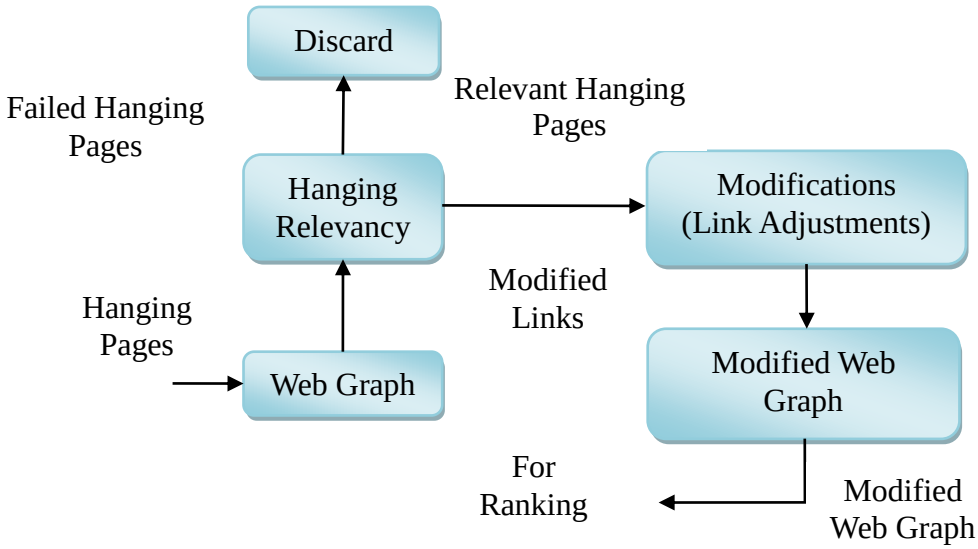


Figure 4.2: Hanging Relevancy Methodology

The hanging relevancy algorithm uses the relevancy function to determine whether a hanging page is relevant or non-relevant. This algorithm compares the anchor text of the hanging page with the keyword or query term, and if it matches, considers that

hanging page as relevant. Then the algorithm converts the relevant hanging page into a non-hanging page, and discards the non-relevant hanging page.

4.3.2 *Algorithm*

The complete algorithm of the proposed method is shown in Figure 4.3.

Let P be the Probability Matrix for the Web Graph, $G_w(V_w,E_w)$ which consists of all non-hanging and hanging pages.

1. Starting with the first row, read all the rows from matrix P and repeat step2.
   2. if $\sum p_{(i,j)} = 0$ (in the matrix P)Then

$$R(P) = \begin{cases} \text{if } AT(G_w(V_w, E_w)) = QT \text{ then} \\ \quad \text{connect the hanging page to the home page} \\ \text{else} \\ \quad \text{discard the hanging page} \end{cases}$$

   i.e (if AT(Anchor Text)of $G_w(V_w,E_w)$= QT (Query Term) w.r.t to $P_{(i,j)}$then)

   $P_{(i,j)}$=1

   connect the relevant hanging page to the home page

   else

   discard the row from $P_{(i,j)}$

3. Call the new matrix as PP which consists of only non-hanging pages and the relevant hanging pages and apply the ranking algorithm to produce a fair and efficient ranking order.

Figure 4.3: Hanging Relevancy Algorithm

4.3.3 *Example*

A sample Web graph with 8 nodes is shown in Figure 4.4. The non-hanging pages are shown in green (nodes *A*, *B*, *C*, *D* and *G*) and the hanging pages are shown in red (nodes *E*, *F* and *H*).

The Adjacency Matrix A for the graph G_w is computed and shown below, as per Equation 2.4 of Definition 2.4 from Chapter 2.

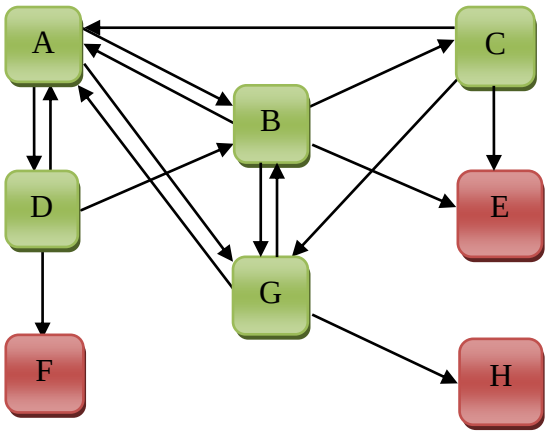


Figure 4.4: A Sample Web Graph G_w with 8 Nodes

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The Probability matrix P for the sample Web Graph G_w is shown below as per Equation 2.5 of Definition 2.5 from Chapter 2.

$$P = \begin{bmatrix} 0 & 1/3 & 0 & 1/3 & 0 & 0 & 1/3 & 0 \\ 1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 0 & 1/3 & 0 \\ 1/3 & 1/3 & 0 & 0 & 0 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

From the above probability matrix P, as per Definition 2.6 of Chapter 2, there are 3 hanging pages in the graph namely, page E, F and H. Only the hanging pages, E and F passed the relevancy test. The algorithm is applied to the matrix P, and the proposed probability matrix PP is produced. There are two important steps: the first one is to find out the hanging pages from matrix P, and the second step is to apply

the relevancy function. Matrix *PP* includes the relevant hanging pages (pages *E* and *F*) as well as the non-hanging pages shown below. Pages *E* and *F* (rows 5 and 6) are connected back to the home page *A* to make it stochastic as per the algorithm. This also makes the pages *E* and *F* becomes non-hanging. According to Langville and Meyer (2004) and Singh, Kumar and Leng (2010), matrix *PP* is *stochastic* and *primitive* like the original matrix proposed by Brin and Page (1998). A matrix becomes a stochastic matrix if the sum of rows is equal to 1. A positive, irreducible matrix is *primitive* if it has only one eigenvalue on its spectral circle. A matrix is *irreducible* if its graph shows that every node is accessible from every other node.

$$PP = \begin{bmatrix} 0 & 1/3 & 0 & 1/3 & 0 & 0 & 1/3 \\ 1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 \\ 1/3 & 0 & 0 & 0 & 1/3 & 0 & 1/3 \\ 1/3 & 1/3 & 0 & 0 & 0 & 1/3 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The modified Web graph G'_w for the above proposed probability matrix is as shown below in Figure 4.5. Here, the hanging node (non-relevant one) is removed and the links are adjusted. This produces better and fairer rank results, which are discussed in the Experimental Section.

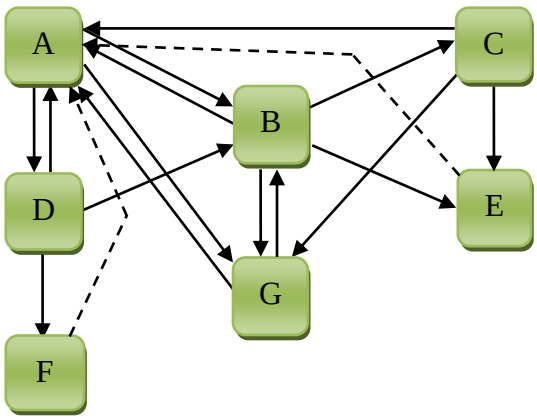


Figure 4.5: Modified Web Graph G'_w

Table 4-2 shows the results of the relevancy function for the above example. Only

the hanging pages *E* and *F* (relevant) have passed the relevancy test. Page *G* has failed the relevancy test and it was not included in the ranking process.

4.3.4 *Stability Analysis*

According to Ng, Zheng and Jordan (2001b), stability is one of the features to be measured for link structure based ranking algorithms, since there are always perturbations on the Web. Deleting some links on the Web should not affect the ranking of a popular Web site. If deleting links on a Web site changes its rank dramatically, then the consistency of the ranking algorithm has to be checked.

Table 4-2: Relevancy Function Results for the Graph G_w

Page	***Page Type***	***Relevancy Test***
A	Non-Hanging	N/A
B	Non-Hanging	N/A
C	Non-Hanging	N/A
D	Non-Hanging	N/A
E	Hanging	Pass
F	Hanging	Pass
G	Non-Hanging	N/A
H	Hanging	Fail

Most of the popular link structure based ranking algorithms, like HITS and PageRank create a matrix and compute the principal eigenvector for stability analysis. In the proposed probability matrix *PP*, when a link is deleted (link from page *G* to page *H*), and the stability analysis has to be done for that matrix. It can be done by computing the principal eigenvector and the eigenvalues.

Eigenvectors and eigenvalues are produced to prove that the proposed probability matrix *PP* is stable. The following theorem proves the stability of the proposed algorithm, which follows the basic PageRank algorithm.

THEOREM 4.1 Let *P* be the probability matrix, and *pe* the principal right eigenvector of $(dU+(1-d)P)^T$, where *d* is the damping parameter usually set to

0.1- 0.2, and U is the transition matrix of uniform transition probabilities. Let nodes n_1, n_2... n_k be altered in any way and PP be the corresponding new transition matrix. Then the new PageRank scores pe satisfies as per Equation 4.2:

$$\left\| \overline{pe} - pe \right\|_1 \leq \frac{2\sum_{j=1}^{k} pe_{n_j}}{d} \tag{4.2}$$

Assuming d is not close to 0, would mean that if the perturbed nodes or pages do not have a high overall PageRank scores, as compared to the unperturbed PageRank scores, pe, then the perturbed PageRank scores $\overline{pe}$ will be close from the original.

Proof. The proof for Theorem 4.1 can be seen in Appendix D.

4.4 EXPERIMENTAL RESULTS

4.4.1 *Rank Computation*

First, the PageRank program was used to calculate the ranking order for the example in Figure 4.4, i.e. before applying the hanging relevancy algorithm. PageRank formula can be referred from Equation 2.2 or Equation 3.1. Next, the PageRank program was used to calculate the ranking order for the example in Figure 4.5, i.e. after applying the hanging relevancy algorithm.

Damping factor d was set to 0.85 and the number of iterations at50. The results are summarized in the following Table 4-3.

Table 4-3: PageRank Results for the Graph G_w and G'_w

Page	***PageRank (Before Hanging Relevancy)***	***PageRank (After Hanging Relevancy)***
A	**1.468**	**2.137**
B	1.30	1.480
C	0.517	0.465
D	0.566	0.756
E	**0.15**	**0.596**
F	**0.15**	**0.364**

G	1.153	1.202
H	0.15	0.15

The summary of the results are shown in Figure 4.6 in the graphical form.

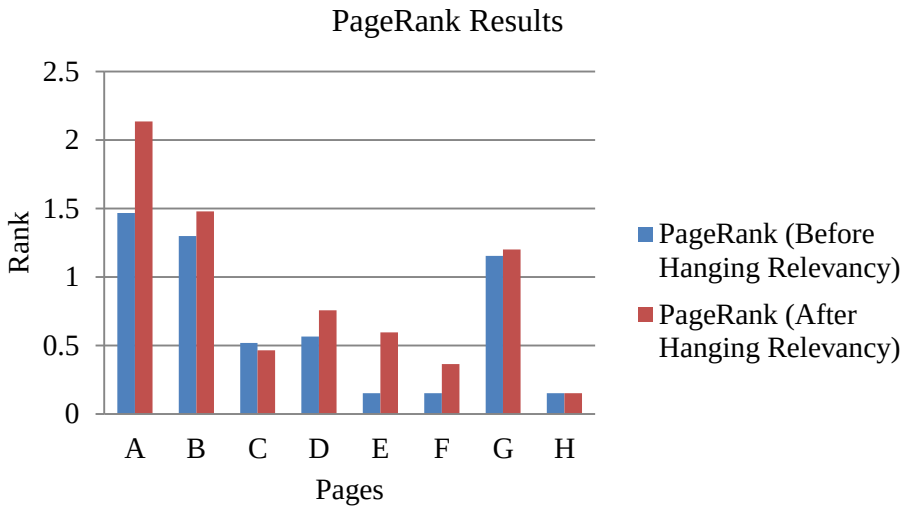


Figure 4.6**:** PageRank Results Comparison Graph

4.4.2 *Experiment on WWW*

For the live experiment on the Web, the PyBot program (Web Crawler) of Leng et al. (2011) was used to crawl the Curtin University (Sarawak Malaysia) Website and to download Web pages. The PyBot Crawler, developed using Python 2.7 was implemented using Tree search along with a Queue (First-In-First-Out) structure. The Crawler downloaded both hanging and non-hanging pages. Due to the volume and the complexity of the World Wide Web, the crawler, crawled only the internal links of the domain site.

The Curtin Website consists of 1728 non-hanging pages and 954 hanging pages (refer Figure 2.19). Figure 2.16 shows the number of non-hanging pages and hanging pages for other publicly available datasets like WEBSPAMUK-2006 of Castillo et al. (2006), WEBSPAMUK-2007 of Yahoo! Research (2007) and EU2010 of Benczúr et al. (2010). The experiment result shows that more than 20% of Web pages are hanging pages, which may provide valuable information to the user.

The PageRank was applied to the Curtin Web pages and the non-hanging results are shown in Figure 4.7. The PageRank simply removes the hanging pages due to computation complexity. PageRank results for hanging nodes are shown in Figure 4.8.

The Web pages were retrieved and indexed based on their anchor texts. Table 4-4 shows the top most indexed keywords, with keyword "Curtin" being the most indexed (as much as 66 Web pages).

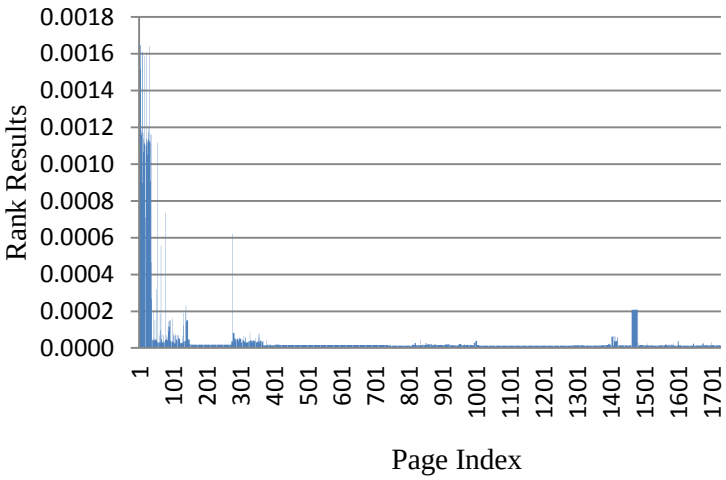


Figure 4.7: Rank Results on the Non-Hanging Nodes

The proposed relevancy algorithm was applied to the downloaded pages to rank them. The results were ranked using six query terms – "*Curtin*", "*Learning*", "*Teaching*", "*University*", "*Research*" and "*Students*". As a sample, only the rank results of the query term "*Research*" and the page names are shown in Table 4-5.

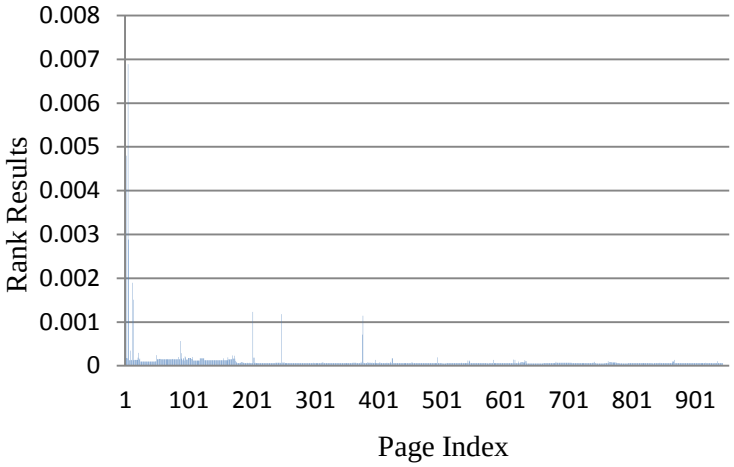


Figure 4.8: Rank Results on the Hanging Nodes

Table 4-4: Top Most Indexed Keywords

Index Keyword	***Number of Links***
Curtin	66
Learning	54
Teaching	37
University	36
Research	10
Students	0

Table 4-5: Hanging Pages for the Query *'Research'*

Rank	***Page Name***
0.0001284	research_profile/proj_EngSc.htm
0.0001199	brc2010/press_clippings/Borneo_Post_09-0628.pdf
0.0001199	brc2010/press_clippings/Borneo_Post_10-0409.pdf
0.0001113	doc/Research_Project_Approval_Application.doc
0.0001113	doc/Research_Procedures.pdf
0.0000937	doc/ERPC_Form.pdf
0.0000919	doc/HREC_FormA_Mar2008.doc
0.0000919	doc/HREC_FormC.doc
0.0000919	doc/HREC_FormC_GuidelinesMar2008.doc
0.0000881	download/CSRCF_Application.pdf

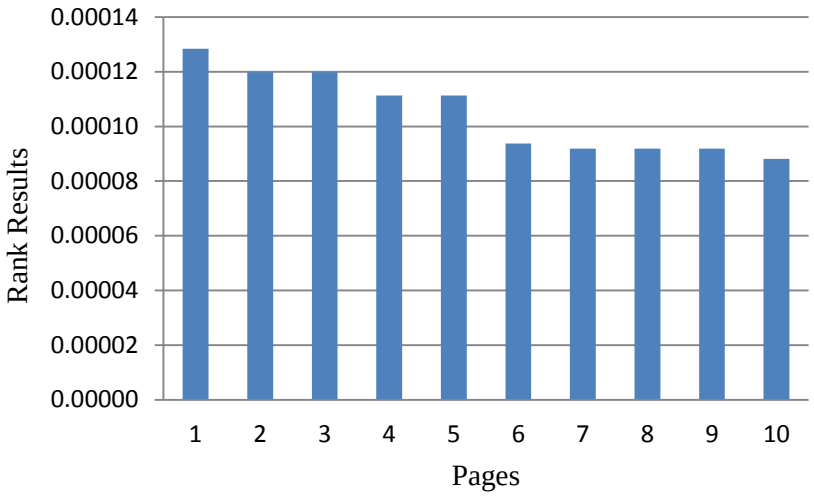


Figure 4.9: Ranking Order of the Hanging Pages for Query '*Research'*

Table 4-5 and Figure 4.9 illustrate the results and also the ranks of the hanging pages, for the query search term '*Research'*. Actually, the entire hanging page results returned relevant to the query search term, '*Research*', with the first hanging page returned as the highest rank, signifying the most relevant hanging page.

4.4.3 *Experiment on Stability Analysis*

The MATLAB (V 8.0.0.783, R2012b) program was used to produce the eigenvector and eigenvalues for both the probability matrix *P* and the proposed probability matrix *PP*, for the example shown in Figure 4.4. Table 4-6 gives the eigenvalues of the matrix *P* (for the graph in Figure 4.4), which are the diagonal elements *d* of the eigenvector *v*.

Table 4-6: Eigenvalues of the Matrix *P*

A	***B***	***C***	***D***	***E***	***F***	***G***	***H***
0.7764	0	0	0	0	0	0	0
0	-0.4632	0	0	0	0	0	0
0	0	-0.1566 + 0.2245i	0	0	0	0	0
0	0	0	-0.1566 - 0.2245i	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Table 4-7 gives the eigenvalues of the matrix *PP* (for the graph in Figure 4.5), which are the diagonal elements *d* of the eigenvector *v*.

Table 4-7: Eigenvalues of the Matrix *PP*

A	***B***	***C***	***D***	***E***	***F***	***G***
0.9952	0	0	0	0	0	0
0	-0.1883 + 0.4307i	0	0	0	0	0
0	0	-0.1883 - 0.4307i	0	0	0	0
0	0	0	-0.3755 + 0.1160i	0	0	0
0	0	0	0	-0.3755 - 0.1160i	0	0
0	0	0	0	0	0.1323	0
0	0	0	0	0	0	-0

In Table 4-6, the eigenvalues of the last 4 rows are 0 because of the hanging pages *E* and *F* of the Web graph in Figure 4.4. Generally, when eigenvalues are negative, the

system is stable. Both eigenvalues in Table 4-6 and Table 4-7 are mostly negative, and the ranking order does not change much on the probability matrix *PP*.

4.4.4 *Result Analysis*

In Table 4-3, the top 3 pages before applying the relevancy function (original PageRank algorithm) are pages *A*, *B* and *G*. After applying the relevancy function, the top 3 pages remained in the same order (*A*, *B* and *G*), but their rank values increased a bit. The ranking order of relevant hanging page *E* has moved to 5 from 6. The summary of the results can be seen in Figure 4.6. This show that the order of relevant hanging pages can improve using this hanging relevancy algorithm than the PageRank algorithm and at the same time reduce the computational complexity over Methods 1 and 2 in Chapter 3. It also improves the ranking of the home page (in this study, page *A* is the home page) considerably, because the relevant hanging pages are connected back to the home page. Before applying the hanging relevancy algorithm, PageRank gets converged at the 45^{th} iteration. After applying the hanging relevancy algorithm, the PageRank gets converged at the 47^{th} iteration (see Appendix D for the detailed results). The experiment also showed that the ranks of certain relevant hanging pages (page *E*) had increased by 4 times. This proved that relevant hanging pages deserves a better ranking.

The proposed hanging relevancy algorithm compromises between complexity and relevancy. It may slow down the ranking process due to the query dependent approach, but it produces fair ranking results by including only the relevant hanging pages. The Query dependent approach can be used only, when the query independent approach does not produce fair results. The main focus in this method was to include the relevant hanging pages in the ranking process and produce fair ranking results.

4.5 SUMMARY

In this chapter, the hanging relevancy algorithm was proposed to include only the relevant hanging pages in the ranking process, based on the link structure. Most of the link structure based ranking algorithms just ignore the hanging pages during ranking. These relevant hanging pages are deprived of their ranks by not showing their true ranking order. When all the hanging pages are included in the ranking, the computational complexity increases. This relevancy function is used to trade-off

between complexity and relevancy, by increasing computational complexity and at the same time producing more relevant results. Considering the amount of hanging pages on the Web, it is really necessary to determine the relevancy of hanging pages according to keywords or query term. It may slow down the ranking process due to query dependency, but it produces fair ranking results. Hence, this relevancy approach can be used only when the traditional search methods do not produce the relevant results. As shown from the examples and the experiments, the hanging relevancy algorithm produces more relevant results with average computational complexity. The experiment also showed that the ranks of certain relevant hanging pages had increased by 4 times. This shows that relevant hanging pages deserves a better ranking.

The next chapter analyses another problem, namely link spam, associated with the hanging pages.

Chapter 5 Link Spam Detection

5.1 INTRODUCTION

Web Spam is a major challenge in the area of Web information retrieval (WIR), which is a method of deliberately manipulating the Search Engine Result Pages (SERPs) in an unethical manner. It is also called spamdexing (Gyongyi and Garcia-Molina 2005b), i.e. using spamming techniques to improve the Web site index in the search engine rankings. As per the prediction of Henzinger, Motwani and Silverstein (2002), Web spam has become the most important challenge of the Web search engine, and it became more active after the introduction of e-commerce in the late 1990s and the advent of fierce competition among search engines in WIR. Generally, Web users look at only the first few pages of the search engine results. This is one of the reasons why commercial and business companies push their Web sites to appear at the top of search engine results. There is also financial gain for the companies when more visitors visit their Web site. Moreover, Web users believe that the search engine results are authentic information, even though majority of the search engine results are unrelated and unauthentic. The order of search engine results in the SERPs is the main reason for spamming in WIR. The intention of Web spammers is to mislead search engine ranking algorithms by promoting certain pages to an undeserved rank. Consequently, they mislead the Web users with irrelevant information. This can affect the creditability of search engines in the WIR.

There are many ways to achieve spamming. *Content* spamming and *Link* spamming are the two popular techniques used in WIR. Link spamming is a type of spamming used to improve the ranking of certain web pages, by having illegitimate links, and it is the most effective way of achieving Web spam. As the internet grows in an exponential way, Web spamming also grows accordingly. Web spammers are looking for every opportunity to induce spamming. One such opportunity are the Web hanging pages, which do not have any outgoing links, but may have one or more incoming links. They receive a share of rank from other pages, but do not propagate their rank to other pages. Hanging pages are described in detail in Chapter 2. These

pages are one of the potential targets for spammers, because in link structure based ranking algorithms, the ranking of Web pages is decided only by the number of incoming links and not by the contents of the Web pages.

Link structure based ranking algorithms like PageRank (Page et al. 1999), HITS (Kleinberg 1999a) and SALSA (Lempel and Moran 2001) can be affected with this kind of link spam. Among these three ranking algorithms, PageRank is the most affected algorithm, because it is the only algorithm that is used commercially in the search engines (Google) for ranking Web pages. This kind of spamming can be a threat to the integrity of the PageRank algorithm.

This chapter proposes Link Spam Detection (LSD) algorithm to detect link spam, in the form of irreducible closed subsets contributed by hanging pages in the Web (Kumar, Singh and Mohan 2014a). A simulation is done to show the contribution of link spam by hanging pages using Web graph, Adjacency matrix and Probability matrix. The methodology first induces link spam using hanging pages and then detects the link spam. Experiments are done using one of the top 10 Web sites (Amazon.com). A crawler program is created in MATLAB which is used to download pages from Amazon.com. A PageRank program is also created using MATLAB and the program is used to rank the Web pages before and after spam. The results proved that link spam can be induced in hanging pages, and can be detected using eigenvectors and eigenvalues.

5.2 PROPOSED METHODOLOGY

Consider a Web graph G_w (V_w, E_w) which has both hanging and non-hanging pages. There are two steps in this methodology to induce link spam. The first step is to identify a target page say, T, and remove all the forward links of target page T to make it a hanging page. Equation 5.1 can be used to remove all the forward links of the target page T.

$$\sum_i T_{ij} = 0 \qquad (5.1)$$

$$p_{i,j} = \begin{cases} a_{i,j} / c_j & if\ c_j \neq 0 \\ 1 & if\ c_j = 0 \end{cases} \quad (5.2)$$

The second step is by using Equation5.2, i.e. all the hanging pages that get an incoming link from the target node *T*, must be connected back to the target node.

After applying Equations5.1 and 5.2, the graph can induce an irreducible closed subset that can create link spam and promote ranks for the target page. Also, this method works only when the graph G_w contains two irreducible closed subsets, which can absorb lots of energy and are not propagated outside. That is why the pages in the irreducible closed subsets have higher ranks than other pages. The link spam can be detected by studying the eigenvector and the eigenvalues, particularly, the second eigenvector. The proposed method can detect link spam contributed by hanging pages using eigenvector and eigenvalues.

Definition 5.1: A set of states *T* is an *irreducible closed subset* of the Markov chain, corresponding to the *transition probability* matrix *P*, if and only if *T* is a closed subset, and no other subset of *T* is a closed subset (Haveliwala and Kamvar 2003).

Definition 5.2: A *Jump Probability (JP)* matrix can be created by adding a damping factor *d* in the *transition probability* matrix. It is used to simulate the random Web surfer model and is also called the Google matrix. It can be defined as follows.

$$JP = dP + \frac{1-d}{n} E \quad (5.3)$$

In the above Equation 5.3, *P* is the *transition probability* matrix, *d* is the damping factor, and usually set at 0.85, *n* is the number of nodes in the graph and *E* is the $n \times n$ matrix of all ones.

5.2.1 *Eigen Vector*

To understand the importance of the second eigenvalue, one needs to review Definition 2.6 from Chapter 2 about the hanging pages, $\sum_i P_{ij} = 0$ and Definition

5.1(irreducible closed subset).

Definition 2.6 from Chapter 2 refers to a hanging or zero-out link node, i.e. in a closed Markov chain; a node can get an incoming link and no outgoing link from that node. In the real Web, there may be many irreducible closed subsets and hanging pages. Analysing the second eigenvalue can determine the link spam associated with the hanging pages.

The first eigenvector is actually the PageRank values (Langville and Meyer 2005) of the jump probability matrix, which can be calculated by Equation 5.4.

$$JP\, g^{(1)} = \lambda_1 g^{(1)} \tag{5.4}$$

In Equation 5.4, $g^{(1)}$ is the distribution of the visiting frequency of each page in the random web surfer model. $g^{(1)}$ is the unique dominant eigenvector corresponding to the dominant eigenvalue λ_1=1. To show that λ_1=1 exists and is unique; the Perron-Frobenious theorem (Meyer 2000) can be used for the Markov matrix *JP*.

A matrix is irreducible if its graph shows that every node is reachable from every other node (Haveliwala and Kamvar 2003) and (Langville and Meyer 2004). An irreducible Markov chain with a primitive transition matrix is called an aperiodic chain (Langville and Meyer 2004). As mentioned before, $\lambda_1 = 1$ is the dominant eigenvalue and the corresponding eigenvector is the PageRank vector g^1. The second largest eigenvalue is λ_2, which is always less than λ_1 i.e. $\lambda_1 = 1 > \lambda_2$.

Theorem 5.1: The second eigenvector g^2 of *JP* is orthogonal to *e*: $e^T g^2 = 0$.

The proof for Theorem 5.1 is given in Appendix E. Here, *e* is the vector of all ones. From theorem 5.1, $e^T g^2 = 0$, therefore, the second eigenvector of *JP* only depends on *P* in Equation 5.3.

Theorem 5.2: The second eigenvalue of *JP*, $\lambda_2 = d$ if *P* has at least two irreducible closed subsets.

The proof for Theorem 5.2 is given in Appendix E and the results are shown in the experimental section. Theorem 5.2 has the following inferences for the PageRank algorithm.

PageRank convergence: Power method used by PageRank has the conversion rate equal to $\lambda_2/\lambda_1 = d$. *Stability of PageRank algorithm*: According to Haveliwala and Kamvar (2003), when the eigengap i.e. $|\lambda_1| - |\lambda_2|$ is greater, a more stable stationary distribution of the Markov chain occurs. *Spam Detection*: The eigenvectors corresponding to $\lambda_2 = d$ is an artifact of certain structures in the Web (Haveliwala and Kamvar 2003). This can help to detect link spamming.

According to Bianchini, Gori and Scarselli (2005), Langville and Meyer (2004) and Boldi, Vigna and Santini (2005), when the value of *d* is higher, an accurate PageRank will be produced. When the value of *d* is lower, a faster convergence and a more stable distribution will occur. The initial value of *d* used by Google is 0.85 and the best value of *d* is also 0.85, as suggested by other researchers. (Haveliwala and Kamvar 2003; Langville and Meyer 2004; Boldi, Vigna and Santini 2005). Hence 0.85 was also used as the value for *d* in the experiment for this thesis. By studying the eigenvalues and the eigenvectors, link spam can be detected in the ranking process. In the Power method, first eigenvector is actually the PageRank vector.

5.2.2*Power Method*

The Power method is the simplest and most popular method to find the eigenvalue and eigenvector of a matrix. When power method to matrix *JP* in Equation 5.3, the convergence of the method for diagonalizable matrices is proved, provided $|\lambda_1| > |\lambda_2|$.

If matrix *JP* is diagonalizable, then there exist *n* independent vectors of *JP*. Let the eigenvectors be $g^1, \ldots g^n$, then $g^1, \ldots g^n$ forms a basis of T^n. The initial vector $v^{(0)}$ can be written as:

$$v^{(0)} = a_1 g^1 + a_2 g^2 + \ldots + a_n g^n \qquad (5.5)$$

In Equation 5.5, $a_1, \ldots, a_n$ are scalars and multiplying both sides of Equation 5.5 by JP^k produces:

$$JP^k v^{(0)} = JP^k (a_1 g^1 + a_2 g^2 + \ldots + a_n g^n)$$

$$= a_1 JP^k g^1 + a_2 JP^k g^2 + \ldots + a_n JP^k g^n$$

$$= a_1 \lambda_1^k g^1 + a_2 \lambda_2^k g^2 + \ldots + a_n \lambda_n^k g^n$$

$$= a_1 \lambda_1^k \left(g^1 + \sum_{j=2}^{n} \frac{a_j}{a_1} \left(\frac{\lambda_j}{\lambda_1} \right)^k g^j \right)$$

If $|\lambda_1| > |\lambda_2| \geq \ldots |\lambda_n|$ then λ_1 can be called a dominant eigenvalue. For example $\left(\frac{\lambda_j}{\lambda_1} \right)^k \rightarrow 0$ and if $a_1 \neq 0$, $JP^k v^{(0)} \rightarrow a_1 JP^k g^1$. The power method normalizes the product $JPv^{(k-1)}$ and it converges to g^1. Here, each iteration is a single matrix-vector multiplication and it can be performed very efficiently rather than a matrix-matrix multiplication. The convergence factor is determined by the second most dominant term, $a_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k g_2$ and the rate of convergence is equal to $|\lambda_2|/|\lambda_1|$. The algorithm used for creating the program is given below.

5.2.3 *Algorithm*

The proposed algorithm to detect link spam caused by hanging pages is shown below in Figure 5.1.

1) Let *G(V,E)* be a directed graph with a set of vertices and edges.
2) Let *SG* be a sub graph with hanging pages and non-hanging pages.
3) Call PageRank
4) Select a random target page say *T* and follow the steps.

a) $\sum_i T_{ij} = 0$ (Remove all the outgoing links from the target node *T*)

b) Look for all the hanging pages having an incoming link from the target node *T* and connect back all those hanging pages back to the target node *T.*

5) Check the sub graph *SG* having two irreducible closed subsets.
6) Create an adjacency matrix for the sub graph *SG* by using the following formula:

$$A_{ij} = \begin{cases} 1 \ if \ (i, j) \in E \\ 0 \ \ \text{otherwise} \end{cases}$$

7) Create a transition probability matrix *P* for the sub graph by using the following formula:

$$p_{i,j} = \begin{cases} a_{i,j} / c_j & if \ c_j \neq 0 \\ 0 & if \ c_j = 0 \end{cases}$$

8) Create the jump probability matrix (*JP)* using the following formula:

$$JP = dP + \frac{1-d}{n} E$$

(In the above formula, the value of *d* is 0.85 and *P* is the probability matrix created in step 7, *E* is the *n x n* vectors of all ones and *n* is the number of nodes in the sub graph).

9) Call the PageRank to calculate the new page rank values.
10) Call the eig function to calculate the eigenvalues and eigenvector from the jump probability matrix (*JP*).
11) Check the second eigenvector whether the target node *T* is in the irreducible closed subset by using the following formula:

$$\begin{cases} v_j \neq 0 \text{ if node j} \in \text{irreducible closed subset} \\ v_j = 0 \text{ if node j} \notin \text{irreducible closed subset} \end{cases}$$

Procedure PageRank

a. Create sparse transition probability matrix
b. Calculate the PageRank using the power method by assigning the damping factor, no. of iterations.
c. Create bar chart for pageranks.

Figure 5.1: Algorithm to Detect Link Spam

5.2.4 *Example for Link spam*

Consider the following sample Web graph G_w with 8 nodes and 12 edges shown in

Figure 5.2, which also shows the PageRanks of all the 8 nodes. Colour codes are used in this graph to differentiate nodes, blue (nodes 3, 4, 5 and 6) indicates non hanging pages, orange (node 8) denotes a hanging page and green (nodes 1 and 2) denotes nodes in irreducible closed subset. The sum of column 8 is zero (using Equation 5.1) and this indicates that page 8 is a hanging page. Let node 7 shown in red be the target node for link spam. Nodes 1 and 2 have high PageRanks (0.25 and 0.27) among the 8 nodes because of the irreducible property; in addition they don't propagate their score to other nodes.

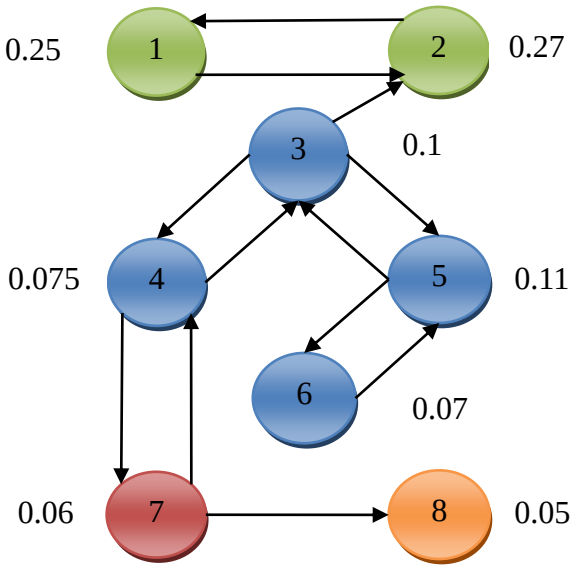


Figure 5.2: Sample Web Graph G_w before Link Spam

The *adjacency* matrix (column matrix) *A* is generated for the graph G_w in Figure 5.2, as per Equation 2.4 from Chapter 2 and shown below. The last column represents the out-degree (*od*) of node 8, while the last row represents the in-degree (*id*) of node 8. The sum of the columns in the matrix *A* gives the *od* and the sum of the rows gives the *id*.

In the *adjacency* matrix *A*, the eighth column represents the hanging page for node 8 by having all zero entries.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The *transition probability* matrix P is computed for the graph G_w in Figure 5.2 as per the Equation 2.5 from Chapter 2 and is shown below.

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/3 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$

This *transition probability* matrix P is not stochastic because column 8 does not sum up to 1. This is due to the hanging page.

The Web graph G_w in Figure 5.2 is modified according to the proposed method and shown in Figure 5.3 as follows. In the first step all the outgoing links from the target node (node 7) is removed. In the second step, hanging node 8 is connected back to the target node 7 as shown in Figure 5.3. Now this modified graph G'_w has two irreducible closed subsets, nodes 1 and 2 and nodes 7 and 8. The PageRanks results are computed and shown in Figure 5.3 after the link spam is induced. The target node rank has increased from 0.06 to 0.185 (more than 3 times) due to proposed link spam methodology. Also notice that ranks of node 7 and 8 has increased to 0.185 and 0.179 respectively due to the irreducible property as shown in Figure 5.3.

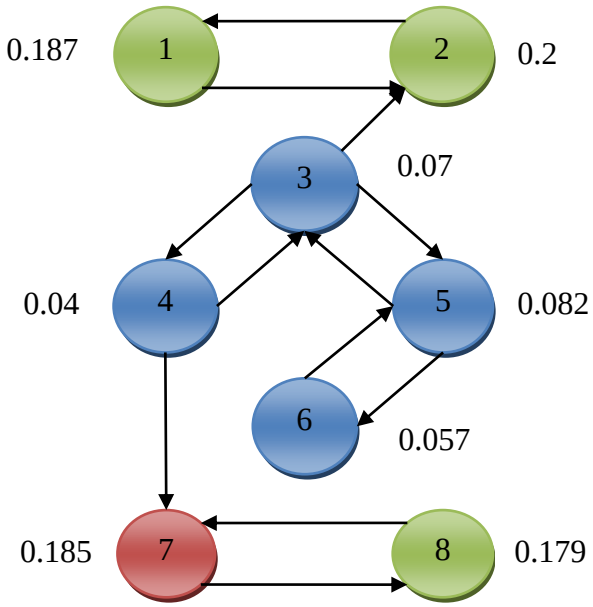


Figure 5.3: Modified Web Graph G'_w after Link Spam

The *adjacency* matrix (*A*) for the modified graph G'_w is shown below.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

In the above *adjacency* matrix (*A*), node 8 is no more a hanging page because it is connected back to node 7 (target node) as per the proposed rule in Equation 5.2.

The *transition probability* matrix (*P*) (column matrix) for the graph G'_w in Figure 5.3, can be developed by using the formula in Equation5.2.

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The above *probability* matrix *P* has the following problems.

- It does not model the random jump to another page (1-*d*). The first eigenvectors are not necessarily unique because matrix *P* is reducible (because of node 1 and 2 and 7 and 8).
- The computation of the first eigenvector becomes difficult because of the reducibility of the matrix.
- This matrix is not stochastic (in the above example *P* is stochastic because there are no other hanging pages in the graph, but in the real web it is not the case).

All the above problems, i.e. reducibility, random surfer model and stochastic are addressed in the *jump probability* matrix (*JP*), which can be obtained by using the following formula in Equation5.6.

$$JP_{i,j} = \begin{cases} da_{i,j} / c_j + (1-d)/n & if\ c_j \neq 0 \\ 1/n & if\ c_j = 0 \end{cases} \quad (5.6)$$

This is the same as Equation5.3which is shown below.

$$JP = dP + \frac{1-d}{n} E$$

When Equation5.4is applied to the *probability* matrix (*P*), the following *jump*

probability (*JP*) matrix is produced as follows:

$$JP = \begin{bmatrix} 0.019 & 0.869 & 0.019 & 0.019 & 0.019 & 0.019 & 0.019 & 0.019 \\ 0.869 & 0.019 & 0.302 & 0.019 & 0.019 & 0.019 & 0.019 & 0.019 \\ 0.019 & 0.019 & 0.019 & 0.444 & 0.444 & 0.019 & 0.019 & 0.019 \\ 0.019 & 0.019 & 0.302 & 0.019 & 0.019 & 0.019 & 0.019 & 0.019 \\ 0.019 & 0.019 & 0.302 & 0.019 & 0.019 & 0.869 & 0.019 & 0.019 \\ 0.019 & 0.019 & 0.019 & 0.019 & 0.444. & 0.019 & 0.019 & 0.019 \\ 0.019 & 0.019 & 0.019 & 0.444. & 0.019 & 0.019 & 0.019 & 0.869 \\ 0.019 & 0019 & 0.019 & 0.019 & 0.019 & 0.019 & 0.869 & 0.019 \end{bmatrix}$$

MATLAB (Version R2012b) was used to calculate the eigenvectors and eigenvalues for the *jump probability* matrix, *JP*. The following are the eigenvectors and eigenvalues:

v =

0.4725	**0.5000**	0.4566	0.2280	0.2280	0.4564	-0.7071	-0.0167
0.5005	**0.5000**	0.3867	0.0777	-0.0777	-0.3864	0.7071	0.0167
0.1831	0.0000	-0.3880	-0.6052	-0.6053	-0.3882	0.0000	-0.0000
0.0996	0.0000	-0.1526	-0.5914	0.5914	0.1527	-0.0000	-0.0000
0.2191	0.0000	-0.5044	0.1789	-0.1789	0.5047	-0.0000	0.0000
0.1409	0.0000	-0.2979	0.2625	0.2625	-0.2981	0.0000	-0.0000
0.4667	**-0.5000**	0.2285	0.1140	0.1140	0.2282	-0.0024	-0.7069
0.4439	**-0.5000**	0.2698	0.3346	-0.3346	-0.2696	0.0024	0.7069

The above *v* is the eigenvector produced by the MATLAB for the graph in Figure 5.3. The first column is the first eigenvector which is the PageRank values of nodes 1 to 8. The second column refers to the second eigenvector. The right eigenvector $v^{(i)}$ of *JP* i.e. $v^{(i)} = (v_1, ..., v_n)$ has the following properties:

$$\begin{cases} v_j \neq 0 \text{ if node } j \in \text{irreducible closed subset} \\ v_j = 0 \text{ if node } j \notin \text{irreducible closed subset} \end{cases} \quad (5.7)$$

The above Equation5.7 shows that the second eigenvector will have a non-zero value, if a node is in an irreducible closed sub set; otherwise they will have zero

values as seen in the second column of *v*. This second eigenvector indicates that the pages in the irreducible closed subset contribute to link spam.

It can be observed that the two irreducible closed subsets (nodes 1, 2 and nodes 7, 8) have non-zero values (0.5000, 0.5000 and -0.5000, -0.5000) and the other nodes have zero values. This indicates that irreducible closed subsets contribute to link spam. Hanging pages play an important role in forming the irreducible closed subset, and in turn contribute to link spam. In the experiment, the PageRank order of the target node (node 7) was increased from 7 to 3. The eigenvalues for the *jump probability* matrix *JP* is shown below.

e =

1.0019	0	0	0	0	0	0	0
0	0.8500	0	0	0	0	0	0
0	0	0.7197	0	0	0	0	0
0	0	0	0.2897	0	0	0	0
0	0	0	0	-0.2897	0	0	0
0	0	0	0	0	-0.7197	0	0
0	0	0	0	0	0	-0.8500	0
0	0	0	0	0	0	0	-0.8500

The second eigenvalue, as depicted the above eigenvalues *e* is 0.85, which is the same as the damping factor used for the *jump probability* matrix (*JP*). According to Haveliwala and Kamvar(2003), if the *transition probability* matrix has at least two irreducible closed subsets, then the second eigenvector of the Google matrix or jump probability matrix is $\lambda_2 = d$ (Theorem 5.2). The sample experiment also produced $\lambda_2 = d$ (0.85). The detailed results are shown in the experimental section.

5.3 EXPERIMENTAL RESULTS

The first task in the experiment was to prove how the link spam could increase the PageRank values. The PageRank program was created using MATLAB (R2012b) to calculate the rank before and after spam. The basic PageRank algorithm by Moler (2011) was modified to include the proposed method, and the program was tested on an Intel i7 Processor (1.70 Ghz) with 6GB RAM. To begin with, the PageRank

program was used for the sample graph in Figure 5.2 i.e. the graph before link spam, and it produced the following results.

The target node for the link spam is node 7 and it is currently ranked no 7. The order of rank for the 8 nodes from high to low is node 2, 1, 5, 3, 4, 6, 7 and 8. The second program, which included the proposed methodologies were applied to the graph in Figure 5.2 and the output is shown in Figure 5.4.

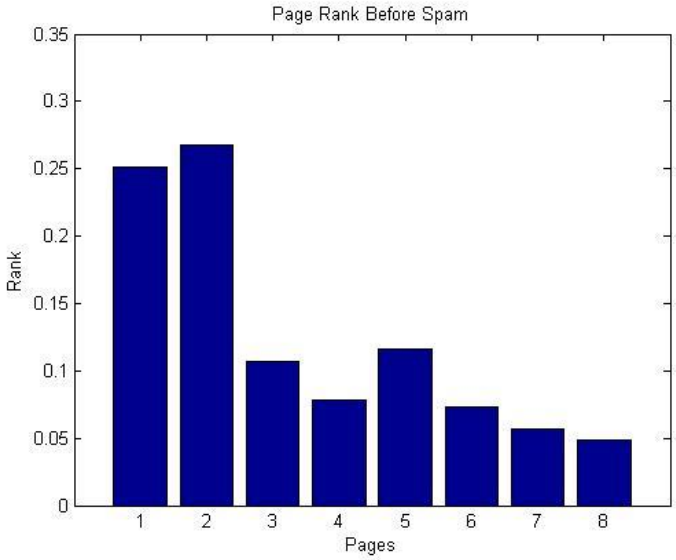


Figure 5.4: PageRank Results before Link Spam

In Figure 5.5 below, the order of rank for the 8 nodes is now node 2, 1, 7, 8, 5, 3, 6 and 4. The target node 7 order has increased from 7 to 3. Just by connecting a hanging node back to the target node can increase the rank significantly. Similarly, in a Web, when many hanging pages are connected to a target page for link spam purposes, the PageRank score can increase significantly.

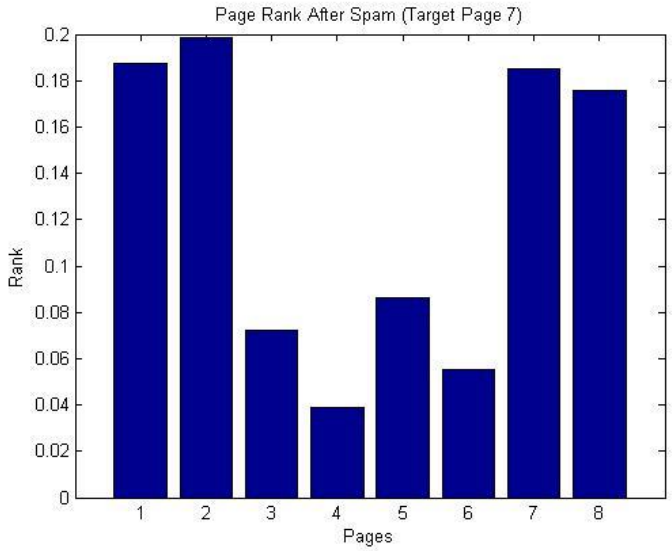


Figure 5.5: PageRank Results after Link Spam

5.3.1 *Experiments with Amazon.com*

To prove further the proposed methodology, experiments were done with live Internet data. Table 5-1 below shows the top 10 Web sites in the world and their incoming links.

Table 5-1: Top 10 Web Sites in the World (Source Alexa.com)

Rank	***Website Name***	***URL***	***In-Links***
1	Facebook	www.facebook.com	8,296,430
2	Google	www.google.com	4,656,505
3	YouTube	www.youtube.com	3,802,453
4	Yahoo!	www.yahoo.com	1,804,470
5	Baidu.com	www.baidu.com	304,348
6	Amazon.com	www.amazon.com	1,148,899
7	Wikipedia	www.wikipedia.org	2,171,478
8	QQ.com	www.QQ.com	445,248
9	Windows Live	www.live.com	134,048
10	Taobao.com	www.taobao.com	163,653

Due to the huge Web size and the computational complexity, experiments were conducted with only one site (amazon.com) from the top 10 of the world's best web sites. First, using the surfer program from MATLAB (Moler 2011), Webpages were

downloaded from amazon.com. Due to the size complexity, only the first 50 pages from amazon.com are shown in the *adjacency* matrix as shown in Figure 5.6. Table 5-2 shows the list of first 50 pages in the amazon.com. Due to the computational complexity, only the first 20 pages were taken from amazon.com (some images and pictures were omitted) and the methodology applied. Let page 15 be the target page for the link spam and the PageRank program developed in MATLAB was used to calculate the PageRank.

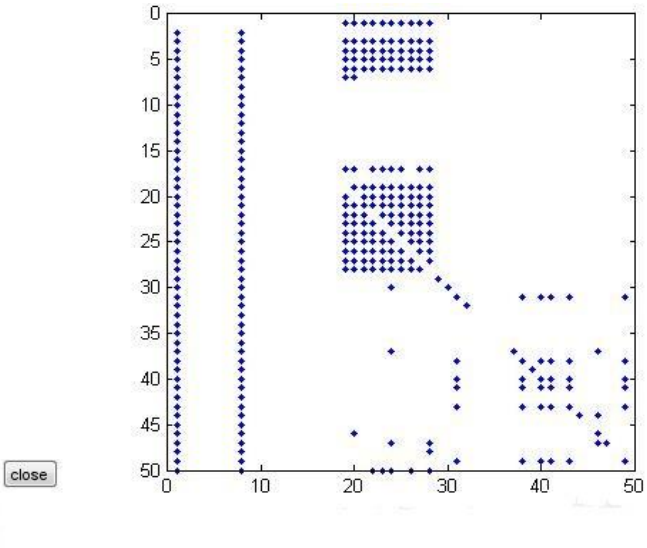


Figure 5.6: Adjacency Matrix for Amazon.com for the First 50 Pages

Table 5-2: List of First 50 Pages from Amazon.com

Page No	***Pages***
1	'http://www.amazon.com'
2	'http://www.amazon.com.br'
3	'http://www.amazon.ca'
4	'http://www.amazon.cn'
5	'http://www.amazon.fr'
6	'http://www.amazon.de'
7	'http://www.amazon.in'
....	.....
49	'http://www.look.com'
50	'http://www.myhabit.com'

Figure 5.7 shows the PageRank results for the first 20 pages of amazon.com before the link spam was introduced. Table 5-3 shows the summary of the results before and after link spam.

The *transition probability* matrix *P* (column matrix) is computed and shown below after the link spam is introduced. It is a sparse matrix as can be seen below. Generally, the *transition probability* matrix for the real Web is a sparse matrix. The *jump probability* matrix (*JP*) is not shown here due to the huge size.

$$P = \begin{bmatrix}
0 & 1/3 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/2 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1/3 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1/2 & 0 & 1/2 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1/2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1/2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

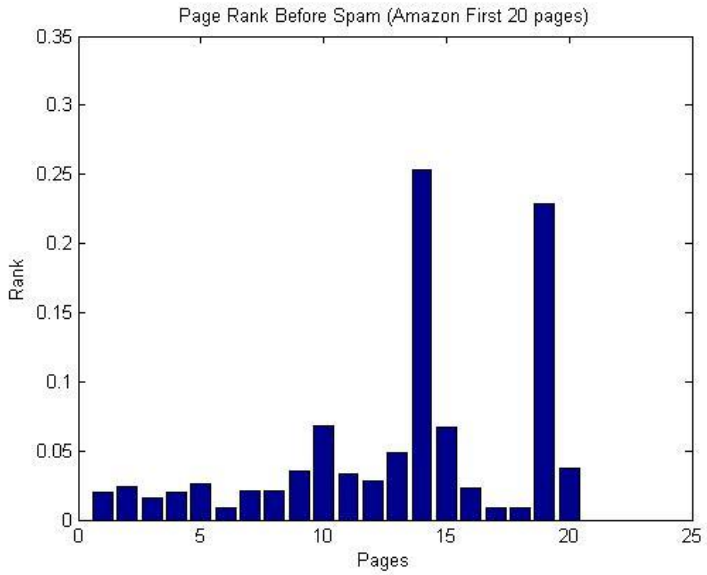


Figure 5.7: PageRank Results before Link Spam for Amazon.com

Figure 5.8 shows the PageRank results after the link spam was introduced. The target page 15 (http://amazonlocal.com)* in Table 5-3 is shown in bold face.

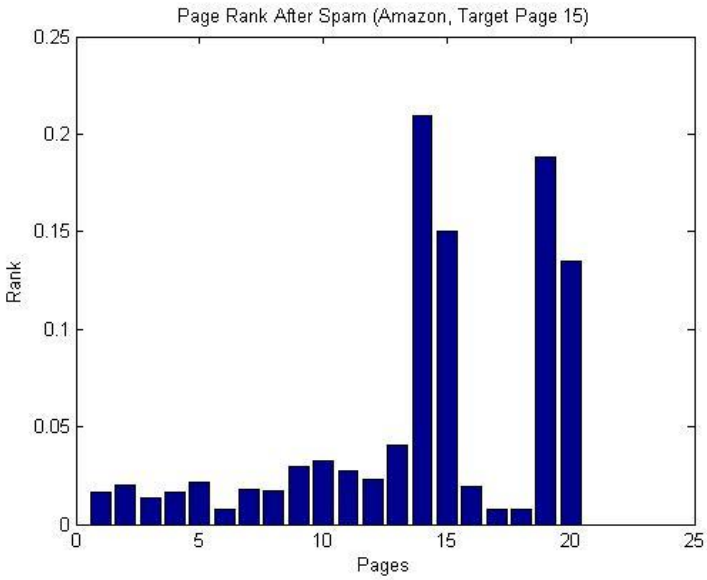


Figure 5.8: PageRank Results after Link Spam for Amazon.com

The comparison graph before link spam and after link spam is shown below in Figure 5.9.

---

* This is not the actual PageRank of amazon.com. It is one of the pages in amazon.com and the PageRanks are based on the proposed method.

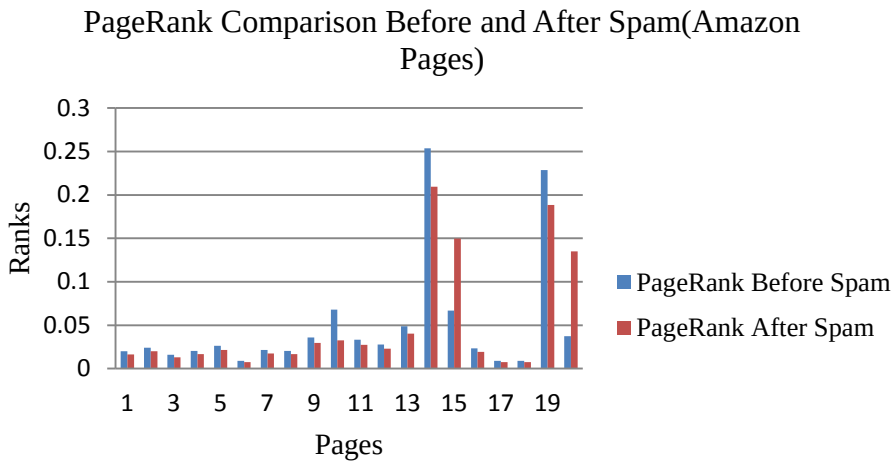


Figure 5.9: PageRank Comparisons before and after Link Spam

Table 5-3: Experimental Results Showing the PageRank and Second Eigenvectors and Eigenvalues

Page No	***Amazon Pages***	***PageRank before Spam***	***PageRank after Spam***	***Second Eigenvector***	***Second Eigenvalue***
1	'http://www.amazon.com.br'	0.0199	0.0164	0.0000	0
2	'http://www.amazon.ca'	0.0243	0.0201	-0.0000	**0.85**
3	'http://www.amazon.cn'	0.016	0.0132	0.0000	0
4	'http://www.amazon.fr'	0.0203	0.0167	-0.0000	0
5	'http://www.amazon.de'	0.0263	0.0217	-0.0000	0
6	'http://www.amazon.in'	0.0091	0.0075	0.0000	0
7	'http://www.amazon.it'	0.0214	0.0176	0.0000	0
8	'http://www.amazon.co.jp'	0.0205	0.0169	0.0000	0
9	'http://www.amazon.es'	0.0358	0.0295	0.0000	0
10	'http://www.amazon.co.uk'	0.0679	0.0326	-0.0000	0
11	'http://www.6pm.com'	0.0335	0.0276	0.0000	0
12	'http://www.abebooks.com'	0.0279	0.023	0.0000	0
13	'http://www.afterschool.com'	0.0489	0.0403	0.0000	0
14	'http://fresh.amazon.com'	0.2537	0.2093	**0.5000**	0
15	**'http://amazonlocal.com'**	**0.0668**	**0.1499**	**-0.5000**	0
16	'http://www.amazonsupply.com'	0.0233	0.0192	0.0000	0
17	'http://aws.amazon.com'	0.0091	0.0075	0.0000	0
18	'http://askville.amazon.com'	0.0091	0.0075	0.0000	0
19	'http://www.audible.com'	0.2286	0.1886	**0.5000**	0
20	'http://www.beautybar.com'	0.0375	0.1349	**-0.5000**	0

Next, eigenvalues and eigenvectors for the *Jump Probability* matrix (*JP*) were produced by the program. The first eigenvector is actually the PageRank values (After Spam). Table 5-3 above shows the second eigenvector and the second eigenvalue for the first 20 pages of amazon.com along with the PageRank values.

5.3.2 *Result Analysis*

Figure 5.9 shows the comparison graph before link spam and after link spam. Notice that the PageRank for the target page 15 has increased from 0.0668 to 0.149. Before spam the order of the target page is 4. After the link spam is introduced, the PageRank of page 15 has more than doubled and the order of the target page is promoted to 3, as shown in Table 5-3. The important observation in the Table 5-3 is the second eigenvector which shows that pages 14 and 15 and pages 19 and 20 are two irreducible closed subsets. As per Equation5.7, they have non-zero values and all the other pages have zero values. This clearly proves that node 15 (target node) is in the irreducible closed subset, which contributes to link spam and this can be detected using the second eigenvector. This method induces link spam using hanging pages in the form of irreducible closed subset and the second eigenvector detects this link spam. Results in Figure 5.9 and Table 5-3 clearly proved that hanging pages can contribute link spam and this link spam can be detected using the second eigenvector.

5.4 SUMMARY

This chapter explores the contribution of hanging pages in the link spam, and proposed a method to form and detect link spam using hanging pages. For this experiment, the PageRank algorithm of Google was used as the base algorithm and included in the methodology.

In doing this, the mathematical models behind the Google search engine like adjacency matrix, transition probability matrix, Google or jump probability matrix, Markov chain, eigenvectors and eigenvalues were explored.

An important finding in this study is the significant role played by the hanging pages in forming the irreducible closed subsets. These subsets absorb lot of energy and get a high PageRank because they do not propagate their ranks to other pages. If more and more hanging pages are connected to an irreducible closed subset, an efficient

link spamming can be achieved. Another finding in this method is the detection of the irreducible closed subset by the second eigenvector of the jump probability matrix.

Live pages from amazon.com were taken and experiments were conducted. The methodology was simulated using the amazon.com Web pages and ranking was done. The experiment also gave the same results as the example shown in the proposed methodology. If Web site developers or Search Engine Optimization (SEO) professionals create Web sites without hanging pages or fix the hanging pages, this kind of link spamming can be controlled.

The challenges of Website Optimisation with regards to hanging pages are examined in the next chapter.

Chapter 6 Website Optimisation

6.1 INTRODUCTION

In the 1990s, many companies realized the value of the Internet and quickly moved their business operations online. When more and more companies started doing business through the Internet, the competition became very stiff; as a result, these companies started working on their Websites so that it would appear on top of the Search Engine Result Pages (SERPs). Search Engine Optimisation (SEO) companies and professionals help e-commerce sites to improve their rank in an organic way, which in turn helps their business to grow.

In this thesis, Search Engine Optimisation (SEO) is called as Website Optimisation (WSO) because the Websites are the one optimised to suit the search engine needs and the not the search engines. Throughout this chapter the term WSO is used instead of SEO. WSO is a set of guidelines, methodologies and techniques for a Website to increase the volume of traffic in a natural or organic way and to obtain a high rank in the SERPs.

Search engines and their relevancy algorithms are constantly being challenged by Black Hat techniques and spammers, due to business competition. Apart from that, there are other hidden challenges in the Web in the form of hanging pages and broken links. Hanging pages are Web pages that do not have any forwarding links or the pages for which the forwarding links are not identified (Eiron, McCurley, and Tomlin 2004). A link that was working once and does not work anymore is called a broken link.

In this chapter, the effects of hanging pages in Website optimisation are studied, and methods are provided to overcome the effects (Kumar, Singh and Mohan 2014b). Problems of hanging pages in Website optimisation are described with an example. Experiments were done using live data from Curtin.edu.my site and the results are shown.

6.2 Role of Hanging Pages in WSO

6.2.1 *Effect of Hanging Pages in Search Engine Ranking Algorithms*

Removing hanging pages from the Web graph can affect the ranking of neighbouring pages. The following example shows two problems associated with hanging pages. The first one is how a hanging page accumulates rank and does not distribute it to other pages. The second one is how the rank of neighbouring pages can be affected when removing a hanging page.

Example

The following example discusses how a PageRank is affected with and without the hanging pages. Figure 6.1 shows a sample Web Graph G_w with 6 pages where, pages *A*, *B*, *C*, *D* and *E* are non-hanging pages, while page *F* is a hanging page because there is no out link from it. The PageRank program was used to compute the PageRank of all the 6 pages. Table 6-1 shows the PageRank results with and without hanging pages.

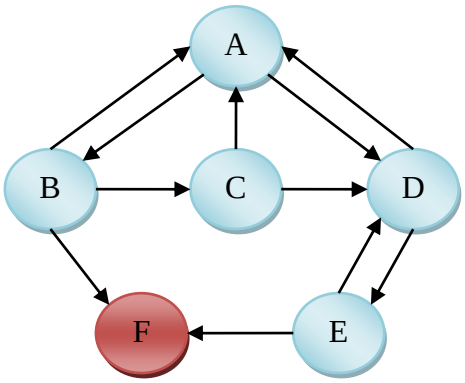


Figure 6.1: A Sample Web Graph G_w with 6 Pages

Table 6-1: PageRank Results with and without Hanging Pages

Page	***PageRank with Hanging Pages***	***PageRank w/o Hanging Pages***
A	0.735	1.345
B	0.463	0.722
C	0.281	0.457
D	0.788	1.633
E	0.485	0.844
F (*HP*)	0.487	0.15

The second column in the above table shows PageRank results with hanging pages for Graph G_w as shown in Figure 6.1. Pages *D* and *A* depict high PageRanks because both of them have 3 incoming links. Hanging page *F* has higher PageRanks than pages *B*, *C* and *E* because it does not distribute its rank to other pages.

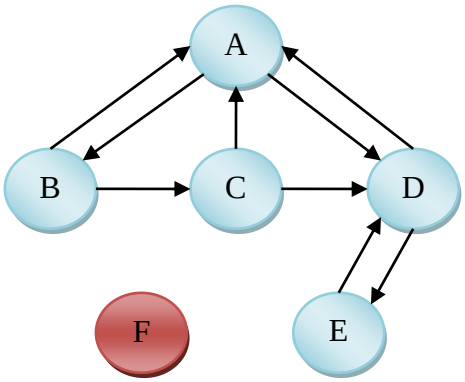


Figure 6.2: Modified Web Graph G^1_wwithout Hanging Pages

In the above Figure 6.2, Web graph G_w is modified so that the graph does not have any hanging pages. There is only one hanging page (page *F*) in graph G_w. An algorithm was used to convert the Web graph G_w into G^1_w so that G^1_w does not have any hanging pages. The PageRank program was then applied to the modified Web graph G^1_w; the results are shown in the third column of Table 6-1. Here, hanging page *F* has only a minimum PageRank of 0.15 and removing this page affects the rank of almost all the other pages. However, the PageRank has improved for the rest of the pages. Thus, the Google PageRank algorithm works by leaving out the

hanging pages to reduce the computational complexity. The real Web is so complex with billions of pages and the computation of PageRank is not easy even with powerful computers and large storage devices. The PageRank results in Table 6-1 are shown in graph form in Figure 6.3 as follows:

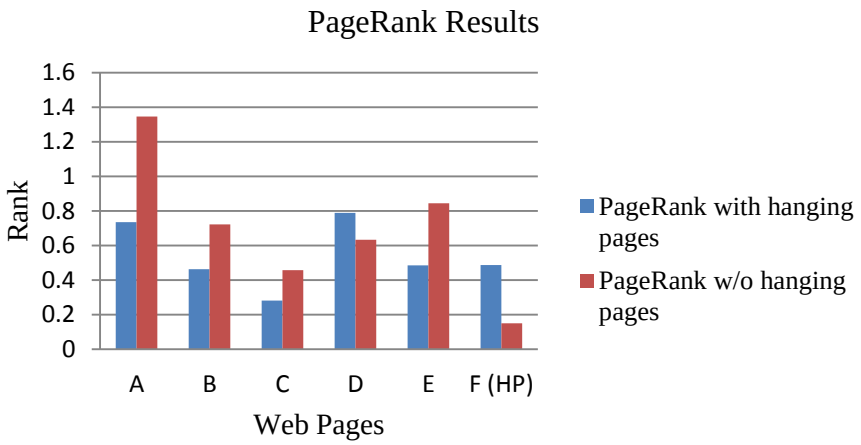


Figure 6.3: PageRank Results with and without Hanging Pages

The above example reflects two important issues:

- The first one is that hanging pages accumulate PageRank and do not distribute it to other pages.
- The second one is that removing hanging pages can affect the PageRank of neighbouring pages and in turn will affect the rank of a Website.

Excluding all the hanging pages when computing a PageRank would not be advisable because hanging pages may have relevant and important information. If a hanging page is important, then that hanging page should be converted into a non-hanging page by using one of the methods proposed in Chapters 3 and 4.

6.2.2 *Methods to Overcome Broken Links and Hanging Pages in WSO*

Broken links and hanging pages are the major obstacles in optimizing a Website. Broken links are the links that lead to pages that do not exit. When clicking on a broken link page, a 404 HTTP error is produced, indicating that the requested URL is

not found. This would be very disappointing for the user who was expecting some pertinent content. Broken link errors can, therefore, affect the rank of a Website in SERPs.

6.2.2.1 *How Links get Broken*

Broken links occur due to one of the following reasons:

- A Web page on the Website is moved or deleted.
- A Web page on another site is moved or deleted.
- A Website is pulled out from the Web server or has ceased to exist.
- A typo or incorrect URL address has been entered.

6.2.2.2 *Methods to Overcome Broken Links*

There are many tools and utilities to find and fix the broken links. Web administrators and WSO professionals need to check and fix the broken links on a regular basis if many additions and deletions occur in a Website. The following are a few strategies to fix broken links.

- If the link is necessary, then the broken link should be found and updated with the proper link.
- If the link is not necessary, then it should be deleted.
- If it is a typo, the URL address should be corrected.

Google Webmaster tools, link checker tools etc. can be used to find and fix broken links.

6.2.3 *Methods to Overcome Hanging Pages in WSO*

It is the duty of the Web developers, administrators and WSO professionals to develop a Web site without hanging pages. If a hanging page is important like .pdf, .ppt or any attachment file then that page has to be converted into a non-hanging

page. There are few methodologies proposed in Chapter 3 and 4 to convert a hanging page into a non-hanging page.

Bianchini, Gori and Scarselli (2005) proposed a method to connect all the hanging pages to a hypothetical node. Singh, Kumar and Leng (2011) suggested two more methods to handle hanging pages. The first one is to connect all the hanging pages to a virtual node and then connect the virtual node back to it. The second method is to connect all the hanging and non-hanging pages to the virtual node and connect the virtual back to it. All the hanging pages can be connected to the home page. More details can be found in Chapters 3 and 4.

6.3 EXPERIMENTAL RESULTS

Experiments for this research study were conducted on the WSO ranking factors in the Curtin University Website (http://www.curtin.edu.my) using the PageRank program and the SEO free tool from Webseoanalytics.com. The number of incoming links, URL's link information, domain's link information and also the PageRank score was noted. Table 6-2 gives the global rank of Curtin, PR score and the number of incoming links.

Table 6-2: Curtin University Domain's Score and Authority

Google's PR	***No. of Incoming Links***
7	141,490

6.3.1 *Back Link Analysis*

Figure 6.4 shows the percentage of internal and external back links for the Curtin site and table form is shown in Appendix F. Curtin's Website Google PageRank is 7 (on a scale of 10);this is because of many external back links and also most of them are from .edu and .gov domains.

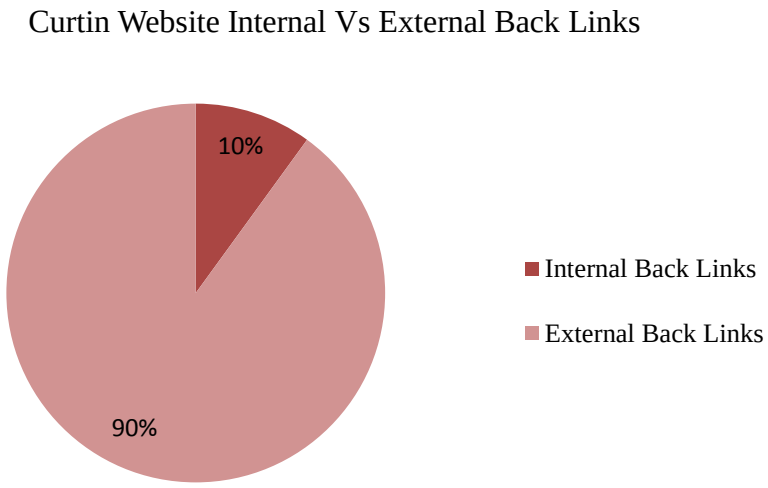


Figure 6.4: Curtin University Website's Internal Vs. External Back Links

Figure 6.5 shows followed VS no-followed back links in the Curtin Website and this no-followed links are a kind of hanging pages. The equivalent table is show in Appendix F. The followed back links passes the PageRank to the linked page and the no-followed back links do not pass the PageRank to these pages.

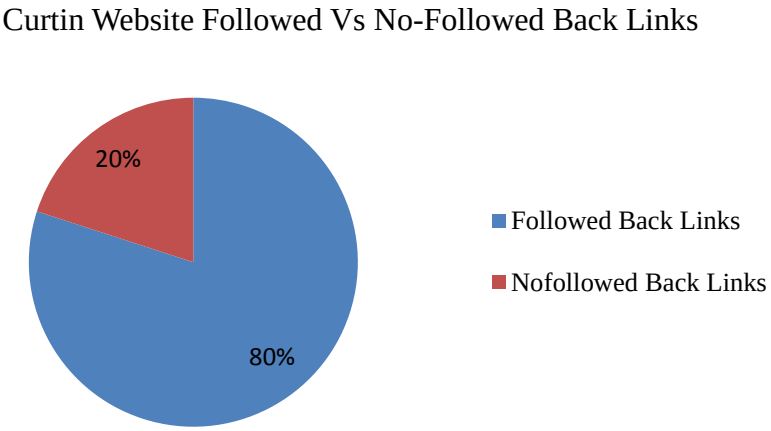


Figure 6.5: Curtin University Website's Followed Vs. No-Followed Back Links

Table 6-3 shows the URL's external back links and domain's information. It also lists the number of .edu and .gov domains.

Table 6-3: Curtin University URL's External Links and Domain Information

URL's External Backlinks		***URL's External Domains***		***Domain's External Backlinks***		***Domain's External Domains***	
235238		1923		335361		3469	
.edu	*.gov*	*.edu*	*.gov*	*.edu*	*.gov*	*.edu*	*.gov*
83479	640	61	10	89562	806	94	16

6.3.2 *Broken Link Analysis*

The Broken Link Analysis was conducted on a sample of 3000 pages from the Curtin Website, using the WebSeoAnalytics tool. The percentage of good and broken links is shown in the graph form below in Figure 6.6 and the equivalent table is shown in Appendix F:

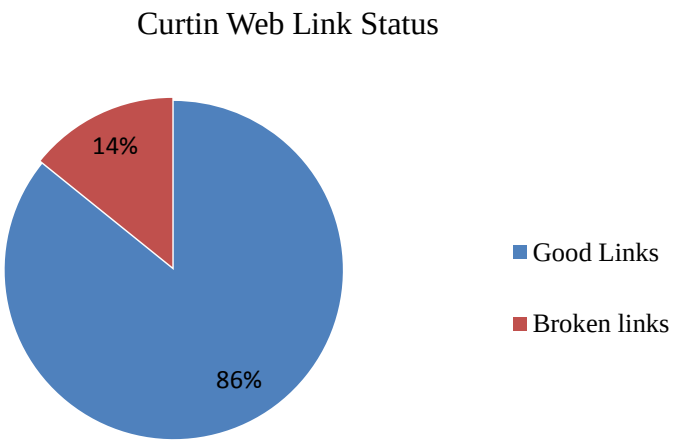


Figure 6.6: Curtin University Link Statuses

Figure 6.7 shows the type of broken links. The majority of them (90%) are 404 Not Found error, which is a client side error saying that the requested resource (page) could not be found. 500 and 504 errors are server side errors. The type of broken link statistics is also shown in the table format in Appendix F.

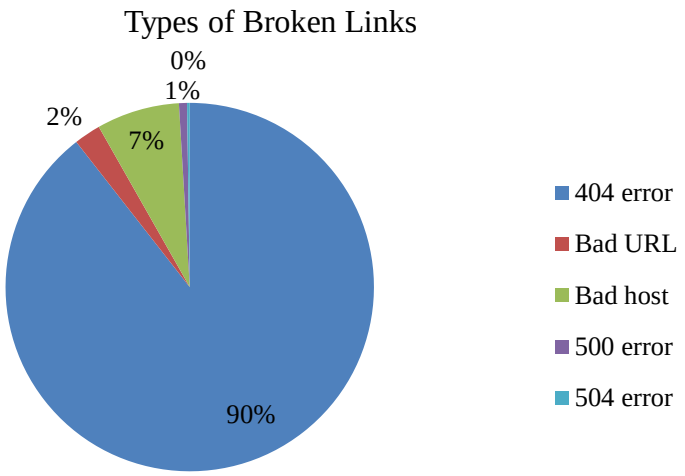


Figure 6.7: Types of Broken Links

Finally, experiments were conducted on On-Site statistics like title relevancy, description relevancy and keyword relevancy for the Curtin Website; the results are depicted in Table 6-4.

Table 6-4: Curtin University "On-Site" Statistics

Title Length	***Title Relevancy***	***Description Length***	***Description Relevancy***	***Keywords Length***	***Keyword Relevancy***	***Av. HTML Headings Relevancy***
25	100%	267	100%	416	100%	75%

The above experiment provides that the Curtin title relevancy is 100%, title description relevancy is 100%, keywords relevancy is 100% and only the HTML headings relevancy is 75%.

6.3.3 *Result Analysis and Discussion*

The results in Figure 6-3 using the sample Web graph shows that hanging pages accumulates PageRank and affects the rank of neighbouring Web pages and in turn affect the Website optimisation process. Based on the experiment and the analysis, the following On-Site suggestions are given to improve the ranking of the Curtin University (Sarawak) site.

1. The page meta description length is 267; it should be between 50 and 150 characters long.
2. The number of H2 Tags are 8; it should be lower than 4.
3. The current number of H3 Tags are 10; it should be lower than 5.
4. Out of 25 images, 3 of them do not have alt Text.

In the broken link analysis, 90% of the broken links are due to 404 errors. This 404 error is a client side error that the requested pages is not available or moved. Even though it is a client side error, this error can be disturbing for users and need to be fixed by the Web administrators. Check the URL for spelling or the correct slashes.

The general rule in WSO is, optimise a Website for users and not for search engines. Based on the research and experiments, it is suggested that the following On-Site methodologies be considered, while optimizing a Website.

- Keep good quality and fresh content.
- Use optimised Website titles and descriptions.
- Use proper URL structure.
- Use Keywords at the right place and keep a maximum of 3% (more than that, and it may become keyword stuffing, which is a Black Hat WSO) keywords density in the site.
- Create user friendly navigation by using breadcrumbs, sitemaps etc.
- Use optimised internal links.
- Use Alt Tag for describing image and area.
- Use Text formatting like h1, h2, bold, italic etc.
- Use external links only to good and relevant sites and make sure there are no broken links.

- There should not be any hanging pages, as they can absorb the ranks and do not distribute the ranks to other pages (Bianchini, Gori and Scarselli 2005; Singh, Kumar and Leng 2011).

Based on this research study, the following Off-Site methodologies can be considered to optimise a Website.

- Create links to Websites or blogs having similar interest, and if a user's Website is useful and genuine, then they will link back to the same user's Website. This natural or organic link helps to improve rankings in SERPs.
- It is good to have few relevant incoming links from reputed sites rather than have many incoming links from irrelevant sites.
- Promote a user's Website through social networks like Facebook, Twitter, and Google+ by sharing things with like-minded users to show the user's active participation. A recent survey by Searchmetrics ("Searchmetrics" 2013) says that Google+ has high weighting in Off-Site WSO ranking factors.
- Webmasters can write useful and unique content about their sites, not only in their own blogs but also in other service related blogs (Vaidhya 2008). They can also post their comments in service related forums. Such blogs and forums allow links which can be crawled by search engines, and in the process promote the Websites and increase Off-Site WSO ranking factors.
- Share documents like brochures, slides and other related ones in common sharing sites like Google Docs, slideshare etc. This will help the site to acquire the qualities of branded Websites.

6.4 SUMMARY

This chapter has explored the problem of hanging pages in WSO and proposed methods to overcome the effects. Experiments were first carried out to show the effect of hanging pages on WSO and subsequently, conducted on the Curtin University Website, to show both the On-Site and Off-Site ranking factors. Finally, a

few On-Site ranking factors were recommended to improve the ranking of Curtin University Website.

Chapter 7 Conclusion

As the Web is nearing one trillion pages, retrieving the relevant and authentic information from it has become a challenging task. While the Web pages are increasing at an enormous rate, the hanging pages in the Web are concurrently multiplying. These pages, especially the relevant ones, deserve to be ranked fair in the SERPS, but are ignored by the link structure based ranking algorithms during ranking. Subsequently, these relevant hanging pages are deprived of their rank evaluation and result in obtaining unfair ranks in the SERPS. This thesis has, therefore, examined the various problems associated with hanging pages in Web Information Retrieval, and proposed solutions to those problems.

A comparative study of link structure based ranking algorithms was initially conducted. The PageRank algorithm was taken as the base algorithm for this research study, because it is the most affected algorithm by the hanging pages. It was implemented and modified according to the various problems detailed in this study. The PageRank algorithm was simulated for a sample hyperlink structure and the PageRanks were computed. The PageRank was converged in the 40th iteration. This experiment has proved that when a page gets more incoming links, its PageRank can increase. The important parameters of link structure based ranking algorithms like model, mining technique used, complexity, limitation etc., were also analysed and compared.

In order to comprehend the current situation of hanging pages on the Web, three datasets were analysed. The experiments showed the percentage of hanging pages in the following datasets: WEBSPAM-UK2006 - 21.35%, WEBSPAM-UK2007 - 43.11%, EU2010 - 54.21% and the Curtin University (Sarawak) Website - 35.57%. It shows that the percentage of hanging pages has increased on the Web. The study has also successfully implemented various algorithms to handle hanging pages in the link structure based ranking algorithms and found that relevant hanging pages deserved a better ranking.

To deal with the problem of hanging pages, this research study proposed two methods to calculate the Page Rank using the Virtual Node (*VN*). In Method 1, all the hanging nodes were identified and connected to a self-loop *VN* and the PageRank was computed. In Method 2, all the hanging and non-hanging nodes were connected to the self-loop *VN* to make the out link uniform for the ranking purpose. The PageRank program was modified according to Methods 1 and 2 and applied to the EU2010 data set. In this experiment, Host graph was used instead of Web graph due to the large dataset collection. The percentage of hanging and non-hanging hosts were analysed, and it was found that nearly 54% of the hosts were hanging hosts, indicating that there are more hanging than non-hanging pages in the Web, and this needs to be addressed. Method 1 produced fair ranking results by including all the hanging hosts in the ranking computation, and also took less number of iterations (36) compared with Method 2. In Method 2, the PageRank values were reduced a little for all the hosts, when compared with Method 1 because the forward links of all the hosts were connected to the Virtual Node. Method 2 also produced fair ranking results but it took more iteration (95) when compared to Method 1. The TrustRank was also implemented and included in both Methods so that they were capable of combating Web spam. Overall, Method 1 performed better because it produced fair and relevant results apart from taking less iteration to converge, when compared with Method 2.

A PageRank simulation program with hanging relevancy function was developed and experiments were carried out using a hyperlink structure with 8 pages. The PageRank results before and after applying the relevancy function were compared. Before applying the hanging relevancy algorithm, the PageRank converged at the 45^{th} iteration, but after applying the algorithm, the convergence occurred at the 47^{th} iteration (only 2 higher than the original PageRank algorithm). The PageRank values of relevant hanging pages were increased after applying the relevancy function, thus, implying that the relevancy function could assist in improving the rank of relevant hanging pages.

To further consolidate the results, a crawler program was created to download the Curtin University Web pages and the hanging relevancy algorithm was applied on to

the downloaded pages. The experiments showed that nearly 36% of the downloaded pages from the Curtin Website were hanging pages. The hanging relevancy algorithm had produced more relevant results with less computation time compared to Methods 1 and 2. The experiments further consolidated the simulation program results, in that the ranks of all the relevant hanging pages had improved; for example the rank of some pages had increased by as many as four, indicating that these pages actually deserved a better ranking.

Stability analysis was applied on the Web graph to show that the perturbation of the link structure did not affect the overall rank of Websites. A program was created in MATLAB to produce eigenvectors and eigenvalues to study the stability analysis. The eigenvalues produced by the experiments were mostly negative, which indicated that the system was stable and the overall rank of the Website was not affected. The hanging relevancy algorithm, which uses the relevancy function to determine whether a hanging page is relevant or non-relevant, is a first kind of approach in determining the relevancy of hanging pages, and includes only the relevant hanging pages in the ranking process. This relevancy function is a trade-off between complexity and relevancy, i.e., it increases computational complexity but produces more relevant results. The use of the relevancy function can be a hybrid approach in determining the relevancy of hanging pages. Whenever the traditional ranking methods do not produce the relevant search results, the hanging relevancy algorithm can be used as an alternative.

Link structure based algorithms can be affected by link spam. This research study has also proposed a Link Spam Detection (LSD) algorithm to detect link spam, in the form of irreducible closed subsets contributed by hanging pages in the Web. In the simulation example, a target page was selected randomly and link spam was introduced according to the proposed methodology. A program, which included PageRank, was created in MATLAB and applied to the Web Graph before and after the introduction of link spam and the results were compared. The PageRank order of target page 7 was promoted from 7 to 3 in the simulated example, and the rank increased by nearly 3 times, showing that hanging pages had contributed to link spam. Live Web pages were also downloaded from Amazon.com and the PageRanks were calculated before link spam was introduced. A target page was selected, link

spam introduced and the PageRank applied to it. The results showed that the rank of the target page had doubled and the rank order was also promoted, thus, consolidating the simulation results that the hanging pages can contribute to link spam.

The second eigenvector and the eigenvalues were computed using the MATLAB program to detect the link spam contributed by the hanging pages. The results indicated that non-zero values of the irreducible closed subsets of the second eigenvector had helped to detect the link spam. The findings were consistent with the simulated examples, and also validated the fact that the hanging pages contributed to link spam.

This study also examined different types of hanging pages and their problems in optimising a Website, and suggested methodologies to handle hanging pages in Website optimization. A crawler was used to download pages from the Curtin University (Sarawak) Website, and the On-Site factors and Off-Site factors were examined. It was found that Curtin University (Sarawak) has a Google's PageRank of 7, which is considered as a good rank on the Google's Tool bar.

Additionally, a back link analysis carried out on the Curtin site, showed that 90% of Curtin's back links are external, while only 10% are internal. External back links from relevant and authentic sites (.gov and .edu) improved the PageRank of the Curtin site. Only 20% of the Curtin's links are no-followed links (one of the reasons for forming hanging pages). The analysis also showed that only 14% of the Curtin links are broken links. The broken links were further analysed and the results showed that 90% of the broken links were due to HTTP 404 Not Found error. This meant that the requested page is not available on the client side. Broken links are another reason for forming hanging pages in the Web.

The effect of hanging pages on Website optimisation was examined, and both On-Site and Off-Site ranking factors for constructing optimised Websites, were suggested for the Web administrators or Web masters. Finally, the experiments also provided On-Site statistics for the Curtin site like title length, title relevancy, keywords length and keywords relevancy.

This thesis has contributed significantly to the overall body of knowledge in the computing field, by identifying different types of hanging pages in the Web and recommending methodologies for relevant hanging pages to obtain fair ranks in the SERPs. In addition, this thesis has suggested methods to combat the link spam which accompany hanging pages, and also proposed On-Site and Off-Site ranking factors to build optimised Web sites, in order to obtain better ranking in the link structure based ranking algorithms.

The research limitations of the study can be described as follows: the first one was the computational complexity of computing the large matrix to find out the PageRank. To overcome the above problems, only a portion of the dataset was taken for computation due to the limitation of computing resources. The second one was finding out the second eigenvector, due to poor convergence of the non-unique values of the second eigenvector in detecting the link spam contributed by hanging pages.

To explore further, on the effect of hanging pages and link spam in the link structure based ranking algorithms, future research strategies should include the following:

a) Apply machine learning algorithms to predict the relevancy of hanging pages while indexing or ranking. Machine learning is the process of construction and study of systems that can learn from data using artificial intelligence. Machine learning algorithms can be used to train on hanging pages to distinguish between relevant and non-relevant hanging pages.

b) Apply machine learning algorithms to analyse and predict the type of incoming links such as link farms, reciprocal links, sponsored links, paid links, pure links etc. and help to combat link spam.

c) Apply machine learning algorithms in Search Engine Optimisation (SEO) to predict the user's browsing pattern and user metrics like *Click Through Rate* (*CTR*), *Bounce Rate* (*BR*), Dwell time etc.

References

Aldous, D., 1983. "Random walks on finite groups and rapidly mixing Markov chains.". Lecture Notes in Mathematics in Mathematics, *Springer-Verlag*. 986: 243-297. doi: 10.1007/BFb0068322.

Atherton, R., 2005. "A Look at Markov Chains and Their Use in Google". Master's thesis, Iowa State University, Ames, Accessed March 2014, http://www.math.bas.bg/~jeni/Rebecca_Atherton.pdf, last.

Augeri, C. J. 2008. "On Graph isomorphism and the PageRank Algorithm". PhD dissertation, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio.

Baeza-Yates, R. and Ribeiro-Neto, B., 1999.*Modern Information Retrieval*: Addison-Wesley Longman Publishing Co., Inc.

Baeza-Yates, R., 2003. "Information Retrieval in the Web: beyond current search engines." *International Journal of Approximate Reasoning, Elsevier Group*. 34 (2-3): 97-104.doi: 10.1016/j.ijar.2003.07.002.

Baeza-Yates, R., Castillo, C. and L´opez, V., 2005. "Pagerank Increase under Different Collusion Topologies", 1st *International Workshop on Adversarial Information Retrieval on the Web*, 10-14 May 2005; Chiba, Japan.

Bar-Ilan, J., 2005. "Comparing rankings of search results on the web.", *Information Processing and Management, Elsevier Group* 41(6):1511-1519. doi: 10.1016/j.ipm.2005.03.008.

Bar-Ilan, J., Mat-Hassan, M. and Levene, M., 2006. "Methods for comparing rankings of search engine results.", *Computer Networks: The International Journal of Computer and Telecommunications Networking - Web dynamics.* 50 (10) : 1448-1463.doi: 10.1016/j.comnet.2005.10.020.

Becchetti, L., Carlos, C., Debora, D., Baeza-Yates, R. and Leonardi, S., 2008. "Link Analysis for Web Spam Detection." *ACM Transactions on the Web.* 2 (1): 1-42. doi: 10.1145/1326561.1326563.

Benczúr, A. A., Castillo, C., Erdélyi, M., Gyöngyi, Z., Masanes, J. and Matthews, M. 2010. ECML/PKDD 2010 Discovery Challenge Data Set. Crawled by the *European Archive Foundation*.

Bianchini, M., Gori, M. and Scarselli, F., 2005. "Inside PageRank" *ACM*

Transactions on Internet Technology (*TOIT*). 5(1): 92-128. doi: 10.1145/1052934.1052938.

Bing, L. 2007. *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data*: Springer-Verlag Berlin Heidelberg.

Boldi, P., Vigna, S. and Santini, M., 2005. "PageRank as the Function of the Damping Factor.", *Proceedings of the 14th International conference on World Wide Web*, Chiba, Japan, 557-566.

Brin, S. and Page, L., 1998. "The Anatomy of a Large Scale Hypertextual Web search engine.", *Computer Network and ISDN Systems*. 30 (1-7): 107-117. doi: 10.1016/S0169-7552(98)00110-X.

Brinkmeier, M., 2006. "Pagerank Revisited.", *ACM Transactions on Internet Technology.* 6 (3): 282-301. doi: 10.1145/1151087.1151090.

Borodin, A., Roberts, G. O., Rosenthal, J. S. and Tsaparas, P., 2005. "Link Analysis Ranking Algorithms, Theory, and Experiments.", *ACM Transactions on Internet Technology (TOIT).* 5 (1): 231-297.doi: 10.1145/1052934.1052942.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A. and Wiener, J., 2000. "Graph Structure in the Web", *Computer Networks: The International Journal of Computer and telecommunications Networking*. 33 (1-6): 309-320.doi: 10.1016/S1389-1286(00)00083-9.

Broder, A., 2002. "Web searching technology overview.", In *Advanced school and Workshop on Models and Algorithms for the World Wide Web*, Udine, Italy.

Burdon, D., 2005. "The Basics of Search Engine Optimisation", *SEO Book*. Accessed on 07th May 2014, http://www.studymode.com/essays/The-Basics-Of-Search-Engine-Optimisation-327336.html.

Castillo, C., Donato, D., Becchetti, L., Boldi, P., Santini, M. and Vigna, S., 2006. "A Reference Collection for Web Spam." *SIGIR Forum* 40 (2).

Chakrabarti, S., Dom, B., Gibson, D., Kleinberg, J., Kumar, R., Raghavan, P., Rajagopalan, S. and Tomkins, A., 1999." Mining the Link Structure of the World Wide Web", *IEEE Computer Society Press*, 32(8): 60-67.doi: 10.1109/2.781636.

Chang, G**.,** Healey, M**.,** McHugh, J.A.M. and Wang**,** T.L., 2001.*Mining the World Wide Web - An Information Search Approach*: The Kluwer International Series on Informational Retrieval Publications, 10.

Cho, J. and Roy, S., 2004. "Impact of Search Engines on Page Popularity". *Proceedings of the 13th International Conference on WWW,* 20-29.

Cho, J., Roy, S. and Adams, R. E., 2005."Page Quality: In search of an unbiased web ranking". *Proceedings of ACM International Conference on Management of*

Data, 551-562.doi: 10.1145/1066157.1066220.

Cooley, R., Mobasher, B. and Srivastava, J., 1997."Web Mining: Information and Pattern Discovery on the World Wide Web." *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence, (ICTAI'97)*, Newport Beach, CA, USA.

Cormen, T. H., Stein, C., Rivest, R.L. and Leiserson, C.E., 2001. *Introduction to Algorithms*: McGraw-Hill Higher Education.

da Gomes, M. G. Jr. and Gong Zhiguo. 2005. "Web Structure Mining: An Introduction", *Proceedings of the IEEE International Conference on Information Acquisition*. doi: 10.1109/ICIA.2005.1635156.

Davison, B. D., 2000. "Recognizing nepotistic links on the Web.", In *Artificial Intelligence for Web Search*. AAAI Press, TX, 23–28.

Dean, J. and Henzinger, M. 1999."Finding Related Pages in the World Wide Web", *Proceedings of the 8thInternational World Wide Web Conference*, 1467-1479.doi: 10.1016/S1389-1286(99)00022-5.

de Jager, D. V., and Bradley, J. T., 2009. "PageRank: Splitting Homogeneous Singular Linear Systems of Index One.", *Proceedings of the 2nd International Conference on Theory of Information Retrieval: Advances in Information Retrieval Theory*. 17-28. doi: 10.1007/978-3-642-04417-5_3.

Ding, C. H. Q., He, X., Husbands, P., Zha, H. and Simon, H. D., 2002."PageRank: HITS and a Unified Framework for Link Analysis". *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 353-354. doi: 10.1145/564376.564440.

Drost, I. and Scheffer, T. 2005. "Thwarting the nigritude ultramarine: learning to identify link spam.", *Proceedings of the 16th European Conference on Machine Learning (ECML)*, 96-107.doi: 10.1007/11564096_14.

Duhan, N., Sharma, A. K. and Bhatia, K. K., 2009." Page Ranking Algorithms: A Survey", *Proceedings of the IEEE International Conference on Advance Computing*. 1530-1537. doi: 10.1109/IADCC.2009.4809246.

Eiron, N. and McCurley, K. S., 2003. "Analysis of Anchor Text for Web Search", In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 459-460. NY, USA. doi:10.1145/860435.860550.

Eiron, N., McCurley, K.S. and Tomlin, J.A., 2004. "Ranking the Web Frontier." In *Proceedings of the 13thInternational conference on World Wide Web, New York, NY, USA*, 309-318. ACM, New York, NY, USA. doi: 10.1145/988672.988714.

Evans, M.P., 2007. "Analysing Google rankings through search engine optimisation data.", *Internet Research,* 17 (1): 21-37.doi: 10.1108/10662240710730470.

"Factshunt", 2013.Total number of Websites and Size of the Internet as of 2013, Accessed 6th April 2013.http://www.factshunt.com/2014/01/total-number-of-websites-size-of.html.

Gao, B., Liu, T. Y., Ma, Z., Wang, T. and Li, H., 2009. "A General Markov Framework for Page Importance Computation", *Proceedings of the 18th ACM conference on Information and knowledge management*. 1835-1838. doi: 10.1145/1645953.1646243.

Gao, B., Liu, T. Y., Liu, Y., Wang, T., Ma, Z. M. and Li, H., 2011. "Page importance computation based on Markov processes", *Journal of Information Retrieval, Springer*, 14 (5): 488-514.doi: 10.1007/s10791-011-9164-x.

Garfield, E., 1972. "Citation Analysis as a Tool in Journal Evaluation", *Science.* 178 (4060): 471-479.doi: 10.1126/science.178.4060.471.

Gibson, D., Kleinberg, J. and Raghavan, P., 1998."Inferring Web Communities from Link Topology", *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia*. 225-234. doi: 10.1145/276627.276652.

Gillies, J. and Cailliau, R., 2000. *How the Web Was Born: The Storey of the World Wide Web*: Oxford University Press.

Gleich, D. F, Gray, A. P., Greif, C. and Lau, T., 2010. "An Inner-Outer Iteration for Computing PageRank.", *SIAM Journal on Scientific Computing (SISC).* 32 (1): 348-371. doi: 10.1137/080727397.

Grimmett, G. and Stirzaker, D., 1989. *Probability and Random Processes*. Oxford University Press.

Gyongyi, Z., Garcia-Molina, H. and Pedersen. J., 2004. "Combating Web Spam with Trustrank", *Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada,*30, 576-587, 1316740: VLDB Endowment.

Gyongyi, Z. and Garcia-Molina, H., 2005a. "Web Spam Taxonomy.", *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web,* May 10-14, 2005, Chiba, Japan: http://ilpubs.stanford.edu:8090/646/

———. 2005b."Link Spam Alliances." *Proceedings of the 31st International Conference on Very Large DataBases* (VLDB), Trondheim, Norway, 517-528, 1083654: VLDB Endowment.

Gyongyi, Z., Berkhin, P., Garcia-Molina, H. and Pedersen, J., 2006. "Link Spam Detection Based on Mass Estimation" *Proceedings of the 32nd International Conference on Very Large Data Bases, Seoul, Korea,* 439-450, 1164166: VLDB Endowment.

Haveliwala, T., 1999. "Efficient Computation of PageRank.", *Technical Report*, Stanford University, http://ilpubs.stanford.edu:8090/386/

Haveliwala, T. and Kamvar, S. D., 2003. "The Second Eigenvalue of the Google Matrix.", *Technical Report* 2003-20, Stanford University, http://ilpubs.stanford.edu:8090/582/.

Henzinger, M. R., Motwani, R. and Silverstein, C., 2002. "Challenges in Web Search Engines", *Journal of ACM SIGIR*, 36 (2): 11-22.doi: 10.1145/792550.792553.

Horowitz, E., Sahni, S. and Rajasekaran, S., 2008. *Fundamentals of Computer Algorithms*: Galgotia Publications Pvt. Ltd.

Hou, J. and Zhang, Y., 2003. "Effectively Finding Relevant Web Pages from Linkage Information.", *IEEE Transactions on Knowledge and Data Engineering*, 15(4): 940-951.doi:10.1109/TKDE.2003.1209010.

"InternetLiveStats", 2014. April 2014 Internet Users, Accessed 21st April 2014. http://www.internetlivestats.com/internet-users/

Iosifescu, M., 1980.*Finite Markov Processes and Their Applications*. John Wiley and Sons, Inc.,

Ipsen, I. C. F. and Selee, T.M., 2007. "PageRank Computation, With Special Attention to Dangling Node." *Society for Industrial and Applied Mathematics (SIAM)*, 29 (4):1281-1296.doi: 10.1137/060664331.

Jansen, B. J., Spink, A., Bateman, J. and Saracevic, T., 1998. "Real life Information Retrieval: A study of user queries on the Web.", *ACM SIGIR Forum*, 32 (1): 5–17.doi: 10.1145/281250.281253.

Kamvar, S. D., Haveliwala, T., Manning, C. D. and Golub, G. H., 2003. "Exploiting the Block Structure of the Web for Computing PageRank". *Technical Report, 2003-17*, Stanford University, http://ilpubs.stanford.edu:8090/579/.

Killoran, J. B., 2013. "How to Use Search Engine Optimisation Techniques to Increase Website Visibility", *IEEE Transactions on Professional Communication*, 56 (1): 50-66.doi: 10.1109/TPC.2012.2237255.

Kleinberg, J., 1999a. "Authoritative Sources in a Hyper-Linked Environment." *Journal of the ACM*, 46(5): 604-632.doi: 10.1145/324133.324140.

———. 1999b. "Hubs, Authorities and Communities", *ACM Computing Surveys*, 31(4), doi: 10.1145/345966.345982.

Kleinberg, J., Kumar, R., Raghavan, P., Rajagopalan, S. and Tompkins, A., 1999. "Web as a Graph: Measurements, models and methods", *Proceedings of the 5th Annual International Conference on Computing and Combinatorics*, 1-17.

Kosala, R. and Blockeel, H., 2000. "Web Mining Research: A Survey." *SIGKDD Explorations, Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining*.2 (1): 1-15.doi:10.1145/360402.360406.

Kumar, R., Raghavan, P., Rajagopalan, S. and Tomkins, A., 1999. "Trawling the Web for Emerging Cyber-Communities", *Computer Networks: The International Journal of Computer and Telecommunications Networking*. 31 (11-16): 1481-1493.doi: 10.1016/S1389-1286(99)00040-7.

Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tompkins, A. and Upfal, E., 2000a. "Web as a Graph", *Proceedings of the Nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Database systems*, 1-10.doi: 10.1145/335168.335170.

———. 2000b. "Stochastic Models for the Web Graph.", *Proceedings of the 41st Annual Symposium on Foundations of Computer Science FOCS*, 57.

Kumar, P. R. and Singh, A. K., 2010. "Web Structure Mining: Exploring Hyperlinks and Algorithms for Information Retrieval", *American Journal of Applied Sciences* 7(6): 840-845.

Kumar, P. R., Leng, G. K. and Singh, A. K., 2013. "Application of Markov Chain in the PageRank Algorithm.", *Pertanika Journal of Science & technology, JST*, 21 (2): 541-554.

Kumar, P.R., Singh, A. K. and Mohan, A., 2013. "Efficient Methodologies to Optimise Website for Link Structure based Search Engines.", *Proceedings of the IEEE 2013 International Conference on Green Computing, Communication and Conservation of Energy* (ICGCE), Chennai, India.

———. 2014a. "A New Algorithm for Detection of Link Spam Contributed By Zero-out Link Pages.", *Turkish Journal of Electrical Engineering and Computer Sciences* (Accepted and waiting for publication).

———. 2014b. "Efficient Methodologies to Overcome the effects of Hanging Pages in Website Optimisation.", *International Journal of Web Engineering and Technology* (Under Review).

Kumar, P. R., Leng, G. K., Singh, A. K. and Mohan, A., 2014. "Efficient Methodologies to determine the Relevancy of Hanging Pages with Stability Analysis", *Cybernetics and Systems: An International Journal* (Under Review).

Langville, A. N. and Meyer, C. D., 2004. "Deeper Inside PageRank.", *Internet Mathematics*. 1 (3): 335-380. doi**:**10.1080/15427951.2004.10129091.

———. 2005. "A Survey of Eigenvector Methods of Web Information Retrieval." *Journal SIAM Review*. 47(1): 135-161.doi:10.1137/S0036144503424786.

———. 2006a. *Google's PageRank and Beyond: The Science of Search Engine*

Rankings: Princeton University Press.

———. 2006b. "Updating Markov Chains with an eye on Google's PageRank." *SIAM Journal on Matrix Analysis and Applications,* 27 (4): 968-987. doi:10.1137/040619028.

———. 2006c. "A Reordering for the PageRank Problem.", *SIAM Journal on Scientific Computing (SISC).*27 (6): 2112-2120. doi: 10.1137/040607551.

Lee, P. C., Golub, G. H. and Zenios, S.A., 2003. "A Fast Two-stage Algorithm for Computing PageRank and its Extensions.", *Technical Report SCCM*-2003-15, *Scientific Computation and Computational Mathematics,* Stanford University.

Leiner, B. M., Cerf, V. G., Clerk, D. D., Khan, R. E., Kleinrock, L., Lynch, D. C., Postel, J., Roberts, L. G. and Wolff, S., 2009. "A Brief History of the Internet." *ACM SIGCOMM Computer Communications Review,* 39(5):(22-31). doi: 10.1145/1629607.1629613.

Lempel, R. and Moran, S., 2001. "Salsa: The Stochastic Approach for Link-Structure Analysis." *ACM Transactions Information System.* 19 (2): 131-160. doi: 10.1145/382979.383041.

Leng, G. K., Kumar, P. R., Singh, A.K. and Dash, R.K., 2011. "PyBot: An Algorithm for Web Crawling.", *Proceedings of the IEEE International Conference on Nano Science, Technology & Societal Implications*.1-6, India. doi: 10.1109/NSTSI.2011.6111993.

Leng, G.K., Kumar, P. R., Singh, A. K and Mohan, A., 2012. "Link Based Spam Algorithms in Adversarial Information Retrieval.", *Cybernetics and Systems: An International Journal.* 43(6):459-475. doi:10.1080/01969722.2012.707491.

Meyer, C. D., 2000. "*Matrix Analysis and Applied Linear Algebra", Chapter 7 and 8*, Society for Industrial and Applied Mathematics.

Mowshowitz, A. and Kawaguchi, A., 2005. "Measuring search engine bias.", *Information Processing and Management,* 41 (5):1193-1205.

Moler, C., 2011.*Experiments with MATLAB, Chapter 7: Google PageRank,* MathWorks, Inc.,

"Netcraft", 2014. April 2014 Web Server Survey, Accessed 6th April 2014. http://news.netcraft.com/archives/2014/04/02/april-2014-web-server-survey.html.

"NetMarketShare", 2014. March 2014 Search Engine Market Share, Accessed 6th April 2014. http://www.netmarketshare.com/search-engine-market-share.aspx?qprid=4&qpcustomd=0&qpct=3.

Ng, A. Y., Zheng, A. X. and Jordan, M. I., 2001a. "Link analysis, eigenvectors and

stability.", *Proceedings of 17th International Joint Conference on Artificial Intelligence.* 2: 903-910.

———. 2001b. "Stable Algorithms for Link Analysis.", *Proceedings of the 24th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2001)*. ACM, New York. 258-266. doi: 10.1145/383952.384003.

Norris, R., 1996. *Markov Chains*: Cambridge University Press.

Page, L., Brin, S., Motwani, R. and Winograd, T., 1999. "The Pagerank Citation Ranking: Bringing order to the Web"., *Technical Report, Stanford Digital Libraries* SIDL-WP-1999-0120. http://ilpubs.stanford.edu:8090/422/.

Perkins, A., 2001. "The Classification of Search Engine Spam", Accessed on 15th April 2014, http://www.silverdisc.co.uk/articles/spam-classification.

Peters, M., 2013. "Search Engine Ranking Factors", Accessed on 16th February 2014, http://moz.com/blog/ranking-factors-2013.

Pinski, G. and Narin, F., 1976. "Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics", *Information Processing and Management*, 12 (5): 297-312.

Ridings, C. and Shishigin, M., 2002. "PageRank Uncovered". *Technical Report*. http://www.voelspriet2.nl/PageRank.pdf.

Safronov, V. and Parashar, M., 2003. "Optimizing Web servers using Page Rank Prefetching for Clustered Accesses", *Information Sciences*, 150 (3-4): 165:176.

Sargolzaei, P. and Soleymani, F., 2010 "PageRank Problem, Survey and Future Directions", *International Mathematical Forum*, 5 (19): 937-956.

"Searchmetrics", 2013. Ranking Factors 2013, Accessed on 16th February 2014, http://www.searchmetrics.com/en/knowledge-base/ranking-factors-uk-2013/.

Seymour, T., Frontsvog, D. and Kumar, S., 2011. "History of Search Engines.", *International Journal of Management and Information Systems*, 15 (4).

Silverstein, C., Marais, H., Henzinger, M. and Moricz, M., 1999. "Analysis of a very large web search engine query log.", *ACM SIGIR Forum*, 33 (1): 6-12. doi: 10.1145/331403.331405.

Singh, A. K. and Kumar, P. R., 2009. "A Comparative Study of Page Ranking Algorithms for Information Retrieval", *World Academy of Science, Engineering and Technology*, 3 (4).

Singh, A.K., Kumar, P. R. and Leng, G.K., 2010."Efficient Algorithm for Handling Dangling Pages Using Hypothetical Node.", *Proceedings of the IEEE 13th*

International Conference on Digital Content, Multimedia Technology and its Applications (IDC), Seoul, S. Korea. 44-49.

———. 2011."Efficient Methodologies to Handle Hanging Pages Using Virtual Node", *Cybernetics and Systems: An International Journal*, 42 (8), 621-635.doi: 10.1080/01969722.2011.634679.

———.2012. "Solving Hanging Relevancy Using Genetic Algorithm.", *Proceedings of the IEEE International Conference on Uncertainty Reasoning and Knowledge Engineering*. Jakarta. 9-12. doi: 10.1109/URKE.2012.6319593.

Spirin, N. and Han, J., 2011. "Survey on Web Spam Detection: Principles and Algorithms.",. *ACM SIGKDD Explorations Newsletter,* 13 (2): 50-64. doi: 10.1145/2207243.2207252.

Sullivan, D., 2013. "Periodic Table of SEO success factors", Accessed on 16th February 2014, http://searchengineland.com/seotable.

Sutton, R. S. and Barto, A. G., 1998. "*Reinforcement Learning: An Introduction*.", Cambridge, MA: MIT Press.

Vaidhya, 2008. "21 SEO strategies to build your online reputation", Accessed on 16th February 2014, http://moz.com/ugc/21offpage-seo-strategies-to-build-your-online-reputation.

Vryniotis, V., 2010. "Link Structure: Analysing the Most Important Methods", Accessed 16th February 2014, http://www.webseoanalytics.com/blog/link-structure-analyzing-the-most-important-methods/.

Wang, X., Tao, T., Sun, J. T., Shakery, A. and Zhai, C., 2008. "DirichletRank: Solving the Zero-One Gap Problem of PageRank.", *ACM Transaction on Information Systems* (TOIS), 26 (2).doi: 10.1145/1344411.1344416.

Wu, B. and Davison, B. D., 2005. "Cloaking and Redirection: A Preliminary Study", *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, Chiba, Japan.

Xing, W. and Ghorbani, A., 2004. "Weighted PageRank Algorithm.", *Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR'04)*, *IEEE,* Canada. 305-314. doi: 10.1109/DNSR.2004.1344743.

Yahoo! Research, 2007. "Web Spam Collections.", http://barcelona.research.yahoo.net/webspam/datasets/.

Zareh Bidoki, A. M. and Yazdani, N., 2008. "DistanceRank: An intelligent ranking algorithm for web pages.", *Information Processing and Management*, 44 (2): 877-892. doi: 10.1016/j.ipm.2007.06.004.

Zhang, G. Q., Qiang, Z. G., Yang, Q. F., Cheng, S. Q. and Zhou, T., 2008. "Evolution

of the Internet and its cores.", *New Journal of Physics*, 10 (12). doi: 10.1088/1367-2630/10/12/123027.

Zhang, H., Goel, A., Govindan, R., Mason, K., and Van Roy, B., 2004. "Making eigenvector- based reputation systems robust to collusion.", *3rd Workshop on Web Graphs (WAW)*. *Lecture Notes in Computer Science*, *Springer*, Rome, Italy, 3243, 92–104.doi: 10.1007/978-3-540-30216-2_8.

Zhicheng, D., Ruihua, S., Jian, Y. N. and Ji, R. W., 2009. "Using Anchor Texts with their Hyperlink Structure for Web Search.", *Proceedings of the 32nd International ACM SIGIR conference on Research and Development in Information Retrieval*, 227-234.doi: 10.1145/1571941.1571982.

Zhou, B. and Pei, J., 2009. "Link Spam Target Detection Using Page Farms.", *ACM Transactions on Knowledge Discovery from Data* (TKDD), 3(3).doi: 10.1145/1552303.1552306.

Appendices

APPENDIX A - CHAPTER 2 RESULTS

The PageRank convergence computation for the hyperlink structure in Figure 2.20, and the convergence chart are shown below:

A. PageRank Convergence Computation

Iteration No	***Page A***	***Page B***	***Page C***	***Page D***
1	1	1	1	1
2	1.566667	1.099167	1.127264	0.780822
3	1.444521	1.083313	1.07086	0.760349
4	1.406645	1.051235	1.045674	0.744124
5	1.37663	1.031342	1.027281	0.733277
6	1.356562	1.017602	1.014859	0.725864
7	1.342848	1.008254	1.006382	0.720814
8	1.333505	1.001881	1.000606	0.717371
9	1.327137	0.997538	0.996669	0.715025
10	1.322797	0.994578	0.993986	0.713427
11	1.319839	0.992561	0.992157	0.712337
12	1.317823	0.991186	0.990911	0.711594
13	1.316449	0.990249	0.990061	0.711088
14	1.315513	0.98961	0.989482	0.710743
15	1.314874	0.989175	0.989088	0.710508
16	1.314439	0.988878	0.988819	0.710348
17	1.314143	0.988676	0.988636	0.710238
18	1.313941	0.988538	0.988511	0.710164
19	1.313803	0.988444	0.988426	0.710113
20	1.313709	0.98838	0.988368	0.710079
21	1.313645	0.988337	0.988328	0.710055
22	1.313602	0.988307	0.988301	0.710039
23	1.313572	0.988287	0.988283	0.710028
24	1.313552	0.988273	0.98827	0.710021

25	1.313538	0.988264	0.988262	0.710015
26	1.313529	0.988257	0.988256	0.710012
27	1.313522	0.988253	0.988252	0.71001
28	1.313518	0.98825	0.988249	0.710008
29	1.313515	0.988248	0.988247	0.710007
30	1.313513	0.988246	0.988246	0.710006
31	1.313511	0.988245	0.988245	0.710006
32	1.313511	0.988245	0.988245	0.710005
33	1.31351	0.988244	0.988244	0.710005
34	1.313509	0.988244	0.988244	0.710005
35	1.313509	0.988244	0.988244	0.710005
36	1.313509	0.988244	0.988244	0.710005
37	1.313509	0.988244	0.988244	0.710005
38	1.313509	0.988244	0.988244	0.710005
39	1.313509	0.988244	0.988244	0.710005
40	1.313509	0.988243	0.988243	0.710005

APPENDIX B - CHAPTER 3 RESULTS

The number of hanging and non-hanging nodes in the EU2010 sample data set is shown in Table A.

A. Sample Dataset Types for EU2010

Type	*Number of Hosts*
Hanging	103749
Non-hanging	87639
Total	191388

The PageRank convergence computation for the directed Web graph with 6 nodes in Figure 3.3, and the convergence chart are shown below:

B. Convergence Computation

Iteration No	*A*	*B*	*C*	*D*	*E*	*F*
1	1	1	0.15	1	1	0.15
2	0.858333	0.676528	0.15	0.597437	0.676528	0.15
3	0.537893	0.506646	0.15	0.43435	0.402772	0.15
4	0.445209	0.408089	0.15	0.390417	0.356027	0.15
5	0.398316	0.377131	0.15	0.371984	0.337734	0.15
6	0.3837	0.367843	0.15	0.366312	0.332135	0.15
7	0.379319	0.36507	0.15	0.364614	0.330459	0.15
8	0.378011	0.364243	0.15	0.364107	0.329959	0.15
9	0.377621	0.363996	0.15	0.363955	0.32981	0.15
10	0.377505	0.363922	0.15	0.36391	0.329766	0.15
11	0.37747	0.363901	0.15	0.363897	0.329752	0.15
12	0.377459	0.363894	0.15	0.363893	0.329749	0.15
13	0.377456	0.363892	0.15	0.363892	0.329747	0.15
14	0.377455	0.363891	0.15	0.363891	0.329747	0.15

The rank convergence computation for Proposed Method 1 for the Web graph in Figure 3.4 and the convergence chart are shown below:

C. Convergence Computation for Proposed Method 1

Iteration No	***A***	***B***	***C***	***D***	***E***	***F***	***VN***
1	1	1	1	1	1	1	1
2	1.70833	0.91736	0.82321	0.86727	0.91736	0.38324	1.666937
3	2.06536	0.96843	0.67017	0.92507	0.84377	0.33988	1.653512
4	2.06337	0.9245	0.63921	0.91573	0.83092	0.33111	1.634654
5	2.02618	0.9052	0.62852	0.90217	0.8179	0.32808	1.612057
6	1.99791	0.89416	0.62297	0.89258	0.80903	0.32651	1.596373
7	1.97945	0.88735	0.61964	0.88641	0.80335	0.32556	1.586296
8	1.96772	0.88308	0.61755	0.88249	0.79976	0.32497	1.579919
9	1.96032	0.8804	0.61624	0.88003	0.7975	0.3246	1.575896
10	1.95565	0.8787	0.61542	0.87847	0.79607	0.32437	1.573361
11	1.95271	0.87764	0.6149	0.87749	0.79517	0.32422	1.571763
12	1.95086	0.87696	0.61457	0.87687	0.79461	0.32413	1.570756
13	1.94969	0.87654	0.61437	0.87648	0.79425	0.32407	1.570121
14	1.94895	0.87627	0.61424	0.87624	0.79402	0.32403	1.569721
15	1.94849	0.87611	0.61415	0.87608	0.79388	0.32401	1.569469
16	1.9482	0.876	0.6141	0.87599	0.79379	0.324	1.56931
17	1.94801	0.87593	0.61407	0.87592	0.79374	0.32399	1.56921
18	1.9479	0.87589	0.61405	0.87589	0.7937	0.32398	1.569147
19	1.94782	0.87586	0.61404	0.87586	0.79368	0.32398	1.569107
20	1.94778	0.87585	0.61403	0.87585	0.79366	0.32398	1.569082
21	1.94775	0.87584	0.61402	0.87584	0.79366	0.32397	1.569066
22	1.94773	0.87583	0.61402	0.87583	0.79365	0.32397	1.569056
23	1.94772	0.87583	0.61402	0.87583	0.79365	0.32397	1.56905
24	1.94771	0.87582	0.61402	0.87582	0.79364	0.32397	1.569046
25	1.94771	0.87582	0.61402	0.87582	0.79364	0.32397	1.569044
26	1.9477	0.87582	0.61402	0.87582	0.79364	0.32397	1.569042

27	1.9477	0.87582	0.61402	0.87582	0.79364	0.32397	1.569041
28	1.9477	0.87582	0.61402	0.87582	0.79364	0.32397	1.569041
29	1.9477	0.87582	0.61402	0.87582	0.79364	0.32397	1.56904
30	1.9477	0.87582	0.61402	0.87582	0.79364	0.32397	1.56904
31	1.9477	0.87582	0.61402	0.87582	0.79364	0.32397	1.56904
32	1.9477	0.87582	0.61402	0.87582	0.79364	0.32397	1.56904
33	1.9477	0.87582	0.61402	0.87582	0.79364	0.32397	1.56904
34	1.9477	0.87582	0.61402	0.87582	0.79364	0.32397	1.56904
35	1.9477	0.87582	0.61402	0.87582	0.79364	0.32397	1.56904
36	1.9477	0.87582	0.61402	0.87582	0.79364	0.32397	1.569039

The rank convergence computation for Proposed Method 2 for the Web graph in Figure 3.5 and the convergence chart are shown below:

D. Convergence Computation for Proposed Method 2

(Only the node, *VN* converged in the 95th Iteration)

Itr. No	***A***	***B***	***C***	***D***	***E***	***F***	***VN***
1	1	1	1	1	1	1	1
2	0.858333	0.676528	0.720858	0.597437	0.676528	0.354243	2.08287
3	0.537893	0.506646	0.465693	0.43435	0.402772	0.281946	2.631993
4	0.445209	0.408089	0.403323	0.390417	0.356027	0.264275	3.021672
5	0.398316	0.377131	0.385158	0.371984	0.337734	0.259128	3.321682
6	0.3837	0.367843	0.379753	0.366312	0.332135	0.257597	3.567109
7	0.379319	0.36507	0.378141	0.364614	0.330459	0.25714	3.772855
8	0.378011	0.364243	0.37766	0.364107	0.329959	0.257004	3.946883
9	0.377621	0.363996	0.377516	0.363955	0.32981	0.256963	4.094552
10	0.377505	0.363922	0.377473	0.36391	0.329766	0.256951	4.219993
11	0.37747	0.363901	0.37746	0.363897	0.329752	0.256947	4.326596
12	0.377459	0.363894	0.377457	0.363893	0.329749	0.256946	4.417202
13	0.377456	0.363892	0.377455	0.363892	0.329747	0.256946	4.494215
14	0.377455	0.363891	0.377455	0.363891	0.329747	0.256946	4.559675

APPENDIX C - ASYMPTOTIC NOTATION

O-notation

O-notation or Big-Oh notation provides an upper bound on a function to within a constant factor. For a given function $g(n)$, $O(g(n))$ denotes the set of functions if there exist positive constant c and n_0 such that $0 \le f(n) \le cg(n)$ for all $n \ge n_0$.

$g(n)$ is an asymptotic upper bound for $f(n)$.

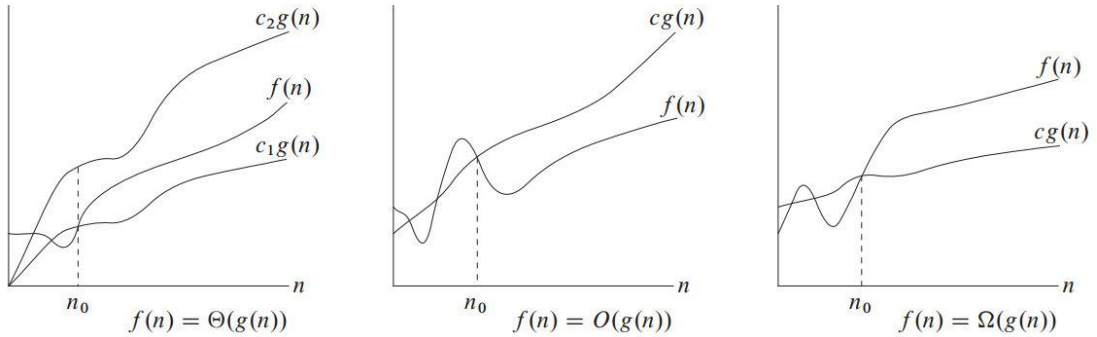
Ω-notation

Ω-notation or Big-Omega notation provides a lower bound on a function to within a constant factor. For a given function $g(n)$, $\Omega(g(n))$ denotes the set of functions if there exist positive constant c and n_0 such that $0 \le cg(n) \le f(n)$ for all $n \ge n_0$.

$g(n)$ is an asymptotic lower bound for $f(n)$.

Θ-notation

For a given function $g(n)$, $\Theta(g(n))$ denotes the set of functions if there exist positive constant c_1, c_2, and n_0 such that $0 \le c_1 g(n) \le f(n) \le c_2 g(n)$ for all $n \ge n_0$. $g(n)$ is an asymptotic tight bound for $f(n)$.



Graphic examples of the Θ, *O*, and Ω notations.

Source: (Cormen et al. 2001, Introduction to Algorithms).

APPENDIX D - CHAPTER 4 THEOREMS PROOF AND RESULTS

Theorem 4.1 from Chapter 4 is given below:

THEOREM 4.1 Let *P* be the probability matrix, and let *pe* be the principal right eigenvector of $(dU+(1-d)P)^T$, where *d* is the damping parameter usually set to 0.1- 0.2, and *U* is the transition matrix of uniform transition probabilities. Let nodes n_1, n_2... n_k be altered in any way and *PP* be the corresponding new transition matrix. Then the new PageRank scores *pe* satisfies as per Equation 4.2:

$$\left\|\overline{pe} - pe\right\|_1 \leq \frac{2\sum_{j=1}^{k} pe_{n_j}}{d}$$

Assuming *d* is not close to 0, this means that if the perturbed nodes or pages do not have high overall PageRank scores as compared to the unperturbed PageRank scores *pe*, then the perturbed PageRank scores $\overline{pe}$ will be close from the original.

Proof:

The coupled matrices are *P* and *PP* which use the transition probabilities $(dU+(1-d)P)^T$ and $(dU+(1-d)PP)^T$ respectively. A coupled Markov chain $\{(X_t, Y_t): t \geq 0\}$ over pairs of Web pages/documents is as follows: $X_0 = Y_0$ is drawn according to the probability *pe*, that is, from the stationary distribution of the PageRank random surfer model. The state transitions works as follows: One step *t*, the probability *d* to reset both chains, in which case X_t and Y_t can set to the same page chosen uniformly at random collection. If no reset occurs, and if $X_{t-1} = Y_{t-1}$ is one of the unperturbed pages, then $X_t = Y_t$ is chosen to be the random page linked to by page X_{t-1}, and independently of it, Y_t is chosen to be a random page linked to by page Y_{t-1}.

Now there are two "coupled" Markov chains, X_t and Y_t, the former, using the transition probabilities $(dU+(1-d)P)^T$, and latter $(dU+(1-d)PP)^T$, so that their transitions are correlated. For instance, the resets to both chains always occur in lock-step. But since each chain is following its own state transition distribution, the

asymptotic distributions of X_t and Y_t must respectively be pe and $\overline{pe}$. Now, let $d_t = P(X_t \neq Y_t)$. Note $d_0 = 0$, since X0 = Y0 always. Let Q denote the set of perturbed pages, and:

$$d_{t+1} = P(X_{t+1} \neq Y_{t+1})$$
$$= P(X_{t+1} \neq Y_{t+1} \mid \text{reset at } t+1)P(\text{reset}) + P(X_{t+1} \neq Y_{t+1} \mid \text{no reset at } t+1)P(\text{no reset})$$

$$= 0.d + (1-d)P(X_{t+1} \neq Y_{t+1} \mid \text{no reset at } t+1)$$

$$= (1-d)[P(X_{t+1} \neq Y_{t+1}, X_t \neq Y_t \mid \text{no reset at } t+1)$$
$$+ P(X_{t+1} \neq Y_{t+1}, X_t = Y_t \mid \text{no reset at } t+1)]$$

$$\leq (1-d)[P(X_t \neq Y_t \mid \text{no reset at } t+1)$$
$$+ P(X_{t+1} \neq Y_{t+1}, X_t = Y_t, X_t \in Q \mid \text{no reset at } t+1)]$$

$$\leq (1-d)P(X_t \neq Y_t + P(X_t \in Q \mid \text{no reset at } t+1))$$

$$\leq (1-d)(d_t + \sum_{i \in Q} pe)$$

Where to derive the first inequality, it uses the fact by construction and, the event $X_{t+1} \neq Y_{t+1}, X_t \neq Y_t$ is possible only if X_t is one of the perturbed pages. Using the fact that $d_0 = 0$ and by iterating this bound on d_{t+1} in terms of d_t, an asymptotic upper-bound, $d_\infty \leq (\sum_{i \in Q} pe)/d$ can be obtained. Thus, if (X_∞, Y_∞) is drawn from the stationary distribution of the correlated chains, so the marginal distributions of X_∞ and Y_∞ are respectively given by pe and $\overline{pe}$, then $P(X_\infty \neq Y_\infty) = d_\infty \leq (\sum_{i \in Q} pe)/d$. But if two random variables have only a small d_∞ chance of taking different values, then their distributions must be similar. More precisely, by the coupling Lemma (Aldous 1983) the *variational distance* $(1/2)\sum_i | pe - \overline{pe} |$ between the distributions must also be bound by the same quantity d_∞. This shows $\left\| \overline{pe} - pe \right\|_1 \leq 2\, d_\infty$, which concludes the proof.

Source: (Ng, Zheng and Jordan 2001a, Link analysis, eigenvectors and stability)

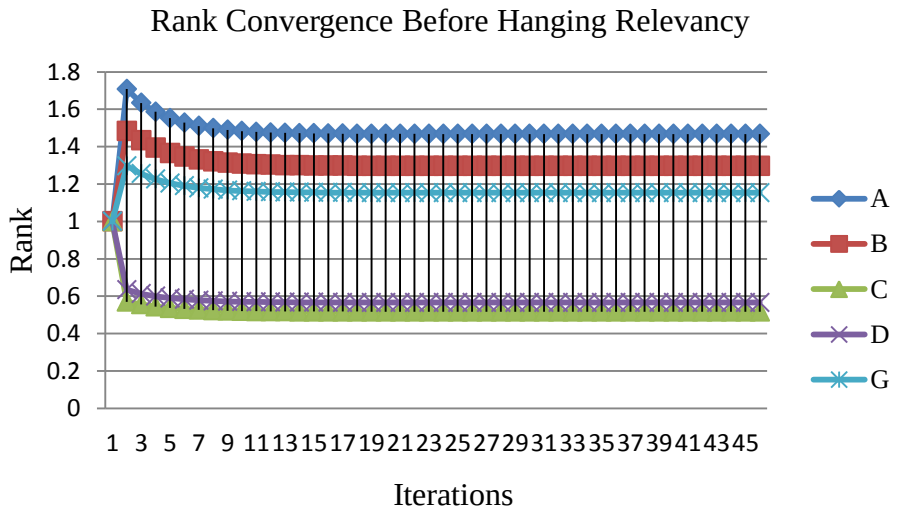
The rank convergence computation before hanging relevancy for the Web graph in Figure 4.4 and the convergence chart are shown below:

A. Convergence Computation before Hanging Relevancy

Iteration No	***A***	***B***	***C***	***D***	***G***
1	1	1	1	1	1
2	1.708333	1.484028	0.570475	0.634028	1.296954
3	1.633593	1.433519	0.556164	0.612851	1.255385
4	1.586534	1.393518	0.54483	0.599518	1.225901
5	1.552186	1.365589	0.536917	0.589786	1.204893
6	1.527845	1.345628	0.531261	0.582889	1.189937
7	1.510498	1.331426	0.527237	0.577975	1.179288
8	1.498149	1.321312	0.524372	0.574476	1.171705
9	1.489357	1.314111	0.522332	0.571984	1.166307
10	1.483096	1.308984	0.520879	0.570211	1.162463
11	1.478639	1.305334	0.519845	0.568948	1.159726
12	1.475465	1.302735	0.519108	0.568048	1.157778
13	1.473205	1.300884	0.518584	0.567408	1.15639
14	1.471596	1.299567	0.518211	0.566952	1.155402
15	1.470451	1.298628	0.517945	0.566628	1.154699
16	1.469635	1.29796	0.517755	0.566397	1.154198
17	1.469054	1.297485	0.517621	0.566232	1.153842
18	1.468641	1.297146	0.517525	0.566115	1.153588
19	1.468346	1.296905	0.517456	0.566031	1.153407
20	1.468137	1.296733	0.517408	0.565972	1.153278
21	1.467987	1.296611	0.517373	0.56593	1.153187
22	1.467881	1.296524	0.517348	0.5659	1.153121
23	1.467805	1.296462	0.517331	0.565878	1.153075
24	1.467752	1.296418	0.517318	0.565863	1.153042
25	1.467713	1.296387	0.51731	0.565852	1.153018
26	1.467686	1.296364	0.517303	0.565844	1.153001
27	1.467666	1.296348	0.517299	0.565839	1.152989

28	1.467653	1.296337	0.517295	0.565835	1.152981
29	1.467643	1.296329	0.517293	0.565832	1.152975
30	1.467636	1.296323	0.517292	0.56583	1.152971
31	1.467631	1.296319	0.51729	0.565829	1.152968
32	1.467627	1.296316	0.51729	0.565828	1.152965
33	1.467625	1.296314	0.517289	0.565827	1.152964
34	1.467623	1.296313	0.517289	0.565826	1.152963
35	1.467622	1.296312	0.517288	0.565826	1.152962
36	1.467621	1.296311	0.517288	0.565826	1.152961
37	1.46762	1.29631	0.517288	0.565826	1.152961
38	1.46762	1.29631	0.517288	0.565826	1.152961
39	1.467619	1.29631	0.517288	0.565825	1.15296
40	1.467619	1.296309	0.517288	0.565825	1.15296
41	1.467619	1.296309	0.517288	0.565825	1.15296
42	1.467619	1.296309	0.517288	0.565825	1.15296
43	1.467619	1.296309	0.517288	0.565825	1.15296
44	1.467619	1.296309	0.517288	0.565825	1.15296
45	1.467619	1.296309	0.517288	0.565825	1.15296
46	1.467618	1.296309	0.517288	0.565825	1.15296

B. Rank Convergence chart Before Hanging Relevancy



The rank convergence computation after hanging relevancy for the Web graph in

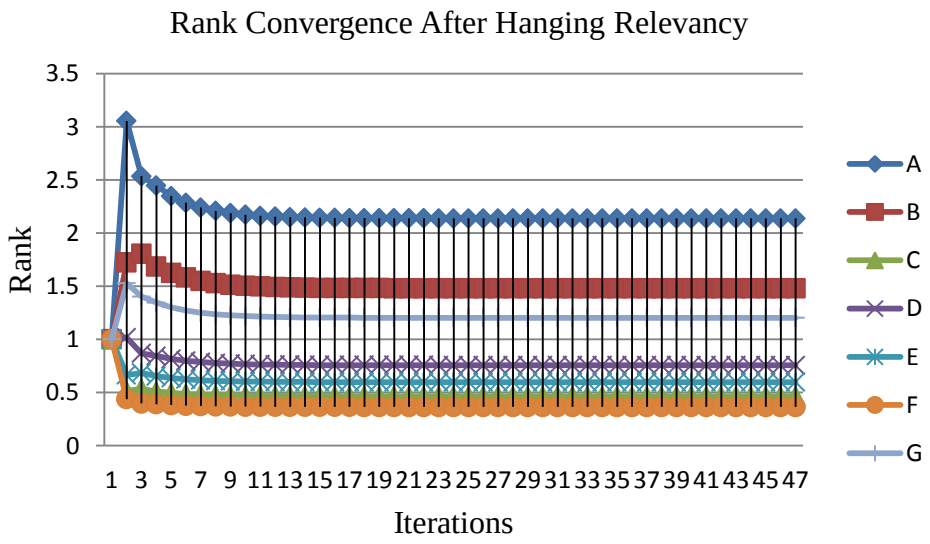
Figure 4.5 and the convergence chart are shown below:

C. Convergence Computation after Hanging Relevancy

Itr. No	***A***	***B***	***C***	***D***	***E***	***F***	***G***
1	1	1	1	1	1	1	1
2	3.054167	1.723681	0.516282	1.015347	0.662562	0.437682	1.527909
3	2.534812	1.80524	0.533614	0.868197	0.684804	0.395989	1.403001
4	2.445743	1.685225	0.50811	0.84296	0.652075	0.388839	1.345035
5	2.34733	1.625556	0.495431	0.815077	0.635803	0.380938	1.300879
6	2.283845	1.580902	0.485942	0.797089	0.623625	0.375842	1.270714
7	2.239068	1.550298	0.479438	0.784402	0.615279	0.372247	1.249682
8	2.208039	1.528973	0.474907	0.775611	0.609464	0.369756	1.235075
9	2.186464	1.514161	0.471759	0.769498	0.605424	0.368024	1.224923
10	2.171473	1.503867	0.469572	0.765251	0.602617	0.366821	1.217868
11	2.161054	1.496713	0.468052	0.762299	0.600666	0.365985	1.212965
12	2.153814	1.491742	0.466995	0.760247	0.59931	0.365403	1.209558
13	2.148783	1.488287	0.466261	0.758822	0.598368	0.365	1.20719
14	2.145286	1.485886	0.465751	0.757831	0.597714	0.364719	1.205545
15	2.142857	1.484218	0.465396	0.757143	0.597259	0.364524	1.204401
16	2.141168	1.483059	0.46515	0.756664	0.596942	0.364388	1.203607
17	2.139995	1.482253	0.464979	0.756332	0.596723	0.364294	1.203054
18	2.139179	1.481693	0.46486	0.756101	0.59657	0.364229	1.202671
19	2.138612	1.481304	0.464777	0.75594	0.596464	0.364183	1.202404
20	2.138219	1.481033	0.46472	0.755829	0.59639	0.364151	1.202219
21	2.137945	1.480845	0.46468	0.755751	0.596339	0.364129	1.20209
22	2.137755	1.480715	0.464652	0.755697	0.596303	0.364114	1.202
23	2.137623	1.480624	0.464633	0.75566	0.596279	0.364104	1.201938
24	2.137531	1.480561	0.464619	0.755634	0.596261	0.364096	1.201895
25	2.137467	1.480517	0.46461	0.755616	0.596249	0.364091	1.201865
26	2.137422	1.480487	0.464603	0.755603	0.596241	0.364088	1.201844
27	2.137392	1.480466	0.464599	0.755594	0.596235	0.364085	1.20183
28	2.13737	1.480451	0.464596	0.755588	0.596231	0.364083	1.20182

29	2.137355	1.480441	0.464594	0.755584	0.596229	0.364082	1.201813
30	2.137345	1.480434	0.464592	0.755581	0.596227	0.364081	1.201808
31	2.137338	1.480429	0.464591	0.755579	0.596225	0.364081	1.201804
32	2.137333	1.480425	0.46459	0.755578	0.596224	0.36408	1.201802
33	2.137329	1.480423	0.46459	0.755577	0.596224	0.36408	1.2018
34	2.137327	1.480421	0.464589	0.755576	0.596223	0.36408	1.201799
35	2.137325	1.48042	0.464589	0.755575	0.596223	0.36408	1.201798
36	2.137324	1.480419	0.464589	0.755575	0.596223	0.36408	1.201798
37	2.137323	1.480419	0.464589	0.755575	0.596222	0.36408	1.201797
38	2.137323	1.480418	0.464589	0.755575	0.596222	0.36408	1.201797
39	2.137322	1.480418	0.464589	0.755575	0.596222	0.364079	1.201797
40	2.137322	1.480418	0.464589	0.755575	0.596222	0.364079	1.201797
41	2.137322	1.480418	0.464589	0.755575	0.596222	0.364079	1.201797
42	2.137322	1.480418	0.464589	0.755574	0.596222	0.364079	1.201797
43	2.137322	1.480417	0.464589	0.755574	0.596222	0.364079	1.201797
44	2.137322	1.480417	0.464589	0.755574	0.596222	0.364079	1.201797
45	2.137322	1.480417	0.464589	0.755574	0.596222	0.364079	1.201797
46	2.137322	1.480417	0.464589	0.755574	0.596222	0.364079	1.201797
47	2.137321	1.480417	0.464589	0.755574	0.596222	0.364079	1.201797

D. Rank Convergence chart After Hanging Relevancy



APPENDIX E - CHAPTER 5 THEOREMS PROOF AND RESULTS

Theorem 5.1 from Chapter 5 is given below:

Theorem 5.1: The second eigenvector g^2 of *JP* is orthogonal to *e*: $e^T g^2 = 0$.

Proof: Since $|\lambda_2| < |\lambda_1|$ (by Lemma 5.1), the second eigenvector of g^2 of *JP* is orthogonal to first eigenvector of *P* by using the following theorem:

If *P* is the transition probability matrix for a finite Markov chain, then the multiplicity of the eigenvalue 1 is equal to the number of irreducible closed subsets of the chain.

From Section 5.3, the first eigenvector of *JP* is *e*. Therefore g^2 is orthogonal to *e*.

Lemma 5.1: The second eigenvalue of *JP* has modulus $|\lambda_2| < 1$.

Proof: Consider the Markov chain corresponding to *P* has only one irreducible closed sub chain *S*, and if *S* is aperiodic, then the chain corresponding to *P* must have a unique eigenvector with eigenvalue 1, by the Ergodic theorem (Grimmett, Stirzaker 1989). So it simply shows that the Markov chain corresponding to *P* has a single irreducible closed sub chain *S*, and this sub chain is aperiodic.

Lemma 5.1.1 shows that *P* has a single irreducible sub chain *S*, and Lemma 1.2 shows this sub chain is aperiodic.

Lemma 5.1.1: There exists a unique irreducible closed subset *S*, of the Markov chain corresponding to *P*.

Proof: This proof was split into a proof of existence and a proof of uniqueness.
Existence: Let the set *U* be the states with nonzero components in*v*. Let *S* consists of the set of all states reachable from *U* along nonzero transitions in the chain. *S* trivially forms a closed subset. Further, since every state has a transition to *U*, no sub

set of *S* can be closed. Therefore, *S* must be the unique irreducible closed subset of the chain.

Lemma 5.1.2: The unique irreducible closed subset S is an aperiodic sub chain.

Proof: According to Iosifescu (1980), two distinct states belonging to the same class (irreducible closed subset) have the same period. In other words, the property of having period *d* is a class property. Therefore, if at least one state in *S* has a self-transition, then the subset *S* is aperiodic. Let *u* be any state in *U*. By construction, there exists a self-transition from *u* to itself. Therefore *S*, must be aperiodic.

From Lemmas 5.1.1 and 5.1.2, and the Ergodic Theorem, $|\lambda_2| < 1$and Lemma 5.1 is proved.

The Theorem 5.2 from Chapter 5 is given below:

Theorem 5.2: The second eigenvalue of *JP*, $\lambda_2 = d$ if *P* has at least two irreducible closed subsets.

Proof:

Case 1: $d = 0$

If $d = 0$, then from Equation 5.3, $JP = E$, since *E* is a rank one matrix and $|\lambda_2| = 0$. Thus, Theorem 5.2 is proved for$d = 0$.

Case 2: $d = 1$

If $d = 1$, then from Equation 5.3, $JP = P$. Since *P* is a column-stochastic matrix, $|\lambda_2| \leq 1$. Thus, Theorem 5.2 is proved for $d = 1$.

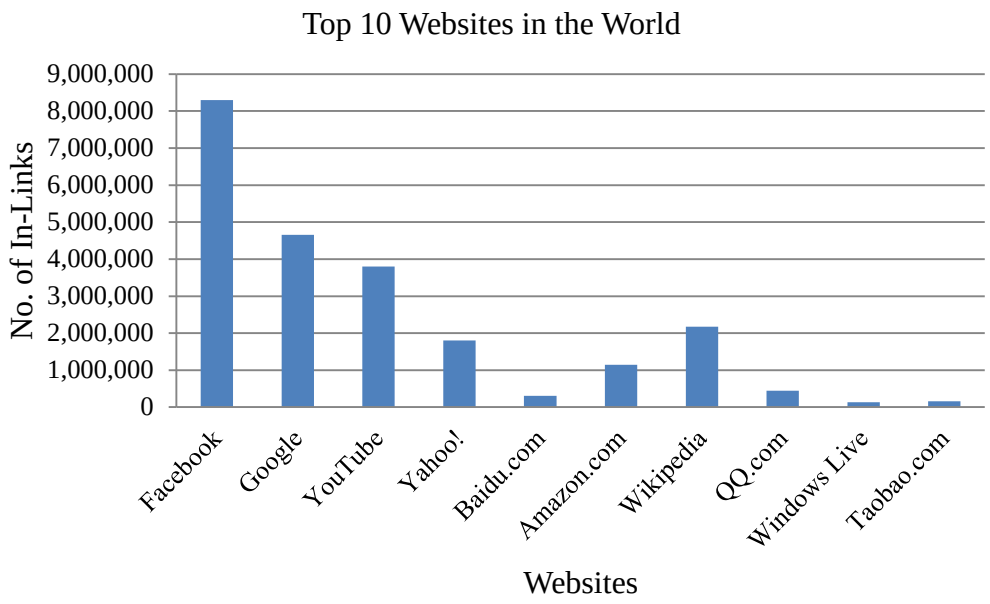
Case 3: $0 < d < 1$

This can be proved as follows: It is assumed that *P* has two irreducible closed subsets. A vector g^i that is an eigenvector of *JP* and whose corresponding eigenvalue is $\lambda_i = d$. Therefore, $|\lambda_2| \geq d$, and there exists $\lambda_i = d$. Therefore, if *P* has at least two irreducible closed subsets, $\lambda_2 = d$.

Source :(Haveliwala and Kamvar.2003.The Second Eigenvalue of the Google Matrix)

The top 10 Websites in the world are shown in graph form below:

A. Top 10 Websites in the World



APPENDIX F - CHAPTER 6 RESULTS

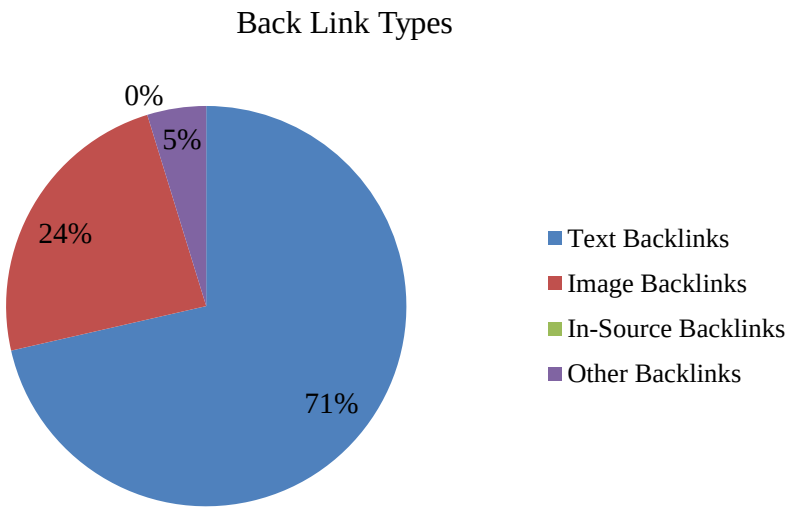
Curtin University's internal and external back links are compared in Table A and shown below.

A. Curtin University Internal Vs. External Back Links

Internal Back Links	*External Back Links*
10%	90%

The different type of back links of Curtin's site is shown below in the graph:

B. Curtin's Back Link Types



The details of broken links of Curtin's site are shown below in the table:

C. Curtin's Broken Links Details

No	*Broken link*	*Page where found*	*Server response*
1	http://www.curtin.edu.my/map.htm	url src	404
2	http://hris.staff.curtin.edu.my/ehr/cgi-bin/tsehr.dll/init >>	url src	bad host
3	http://www.curtin.edu.my/future/fees/index.htmS	url src	404
4	http://www.curtin.edu.my/Student_Complaints/index.htm	url src	404

5	http://www.curtin.edu.my/image_video/sch_dept/A-Z_inde	>>	url src	404
6	http://www.curtin.edu.my/image_video/contact_us_all.htm		url src	404
7	http://www.curtin.edu.my/UniversityLife/contact_us_all.htn	>>	url src	404
8	http://www.curtin.edu.my/sch_dept/international/index.htm		url src	404
9	http://www.curtin.edu.my/sch_dept/R&D/index.htm		url src	404
10	http://www.curtin.edu.my/sch_dept/CurtinSarawak/about.h		url src	404
11	http://www.curtin.edu.my/sch_dept/contactWeb.asp		url src	404
12	http://www.curtin.edu.my/A-Z_index.htm		url src	404
13	http://www.curtin.edu.my/sch_dept/current/index.htm		url src	404
14	http://www.curtin.edu.my/biovalley/enrolment/index.htm		url src	404
15	file:///m|/learning%20centre/index.htm/m|/Learning%20Centr	>>	url src	bad url
16	http://www.curtin.edu.my/Corp%20Comm/index.htm		url src	404
17	http://www.curtin.edu.my/future/housing.htm		url src	404
18	http://www.curtin.edu.my/University%20Life/jcw_history.h	>>	url src	404
19	http://www.curtin.edu.my/University%20Life/john_curtin_a	>>	url src	404
20	http://www.curtin.edu.my/University%20Life/volunteers_c	>>	url src	404
21	http://www.curtin.edu.my/University%20Life/staff.asp		url src	404
22	http://www.youtube.com/watch?v=l60qqicm1qY		url src	404
23	http://www.curtin.edu.my/University%20Life/http/cv.curtin.	>>	url src	404
24	http://www.curtin.edu.my/UniversityLife/maito:%20universi	>>	url src	404
25	http://www.curtin.edu.my/staff/staffIndex.asp		url src	404
26	http://www.curtin.edu.my/UniversityLife/UL_services/conta	>>	url src	404
27	http://www.curtin.edu.my/oweek/sch_dept/A-Z_index.htm		url src	404
28	http://www.curtin.edu.my/oweek/Student_Complaints/index	>>	url src	404
29	http://www.curtin.edu.my/oweek/contact_us_all.htm		url src	404
30	http://www.curtin.edu.my/oweek/CurtinSarawak/about.htm		url src	404
31	http://www.curtin.edu.my/sch_dept/pre_university/index.h	>>	url src	404
32	http://www.curtin.edu.my/sch_dept/engineering_science/ir	>>	url src	404
33	http://www.curtin.edu.my/sch_dept/business/Postgraduate	>>	url src	404
34	http://www.curtin.edu.my/University%20Life/jcw.htm		url src	404
35	http://www.curtin.edu.my/University%20Life/picture_galler	>>	url src	404
36	http://www.curtin.edu.my/sch_dept/Administrative/General	>>	url src	404
37	http://www.curtin.edu.my/sch_dept/Administrative/General	>>	url src	404
38	http://www.curtin.edu.my/sch_dept/Administrative/General	>>	url src	404

39	http://www.curtin.edu.my/sch_dept/Administrative/General	>>	url src	404
40	http://www.curtin.edu.my/sch_dept/Administrative/General	>>	url src	404
41	http://www.curtin.edu.my/sch_dept/Administrative/General	>>	url src	404
42	http://www.curtin.edu.my/sch_dept/Administrative/General	>>	url src	404
43	http://www.curtin.edu.my/sch_dept/Administrative/General	>>	url src	404
44	http://www.curtin.edu.my/sch_dept/Administrative/General	>>	url src	404
45	http://www.curtin.edu.my/sch_dept/Administrative/General	>>	url src	404
46	http://www.curtin.edu.my/sch_dept/Administrative/General	>>	url src	404
47	http://www.curtin.edu.my/sch_dept/Administrative/General	>>	url src	404
48	http://www.curtin.edu.my/University%20Life/future/index.h	>>	url src	404
49	http://www.curtin.edu.my/University%20Life/international/i	>>	url src	404
50	http://www.curtin.edu.my/University%20Life/CurtinSarawa	>>	url src	404
51	http://www.curtin.edu.my/University%20Life/R&D/index.htr		url src	404
52	http://www.curtin.edu.my/University%20Life/current/index.	>>	url src	404
53	http://www.curtin.edu.my/University%20Life/staff/staffInde	>>	url src	404
54	http://library.curtin.edu.au/copyright/		url src	404
55	http://www.curtin.edu.my/Bookshop/contact_us_all.htm		url src	404
56	http://www.bookshop.curtin.edu.au/oasis.html		url src	404
57	http://www.curtin.edu.my/sch_dept/MassComm/index.htm		url src	404
58	http://webr.curtin.edu.my/login/login.php		url src	bad host
59	http://amuse.staff.curtin.edu.my/ttx/ttx.cgi		url src	bad host
60	http://www.curtin.edu.my/current/Curtin%20Pool%20Club		url src	404
61	http://www.curtin.edu.my/current/SPE%20student%20cha	>>	url src	404
62	http://www.curtin2.edu.my/contact_us_all.htm		url src	bad host
63	http://hris-net.staff.curtin.edu.my		url src	bad host
64	http://www.curtin.edu.my/faq.htm		url src	404
65	http://www.miri.net.my		url src	bad host
66	http://www.curtin.edu.my/staff/Departments/Student_Serv	>>	url src	404
67	http://www.curtin.edu.my/staff/contactWeb.asp		url src	404
68	http://internal.curtin.edu.my/event/login/login.php		url src	bad host
69	http://www.curtin.edu.my/sch_dept/ICT/CurtinSarawak/abc	>>	url src	404
70	.http://www.curtin.edu.my/campusnews/index.htm		url src	bad url
71	http://www.curtin.edu.my/events/CurtinSarawak/about.htm		url src	404
72	.http://handbook.curtin.edu.au/courses/31/312097.html		url src	bad url

73	http://soes.curtin.edu.my/applied-geology/courses/		url src	404
74	http://www.curtin.edu.my/sch_dept/ICT/sch_dept/A-Z_inde	>>	url src	404
75	http://www.curtin.edu.my/sch_dept/ICT/Student_Complaint	>>	url src	404
76	http://www.curtin.edu.my/sch_dept/ICT/contact_us_all.htm		url src	404
77	https://staf.curtin.edu.my:4097/mail/		url src	bad host
78	http://www.curtin.edu.my/sch_dept/business/international/	>>	url src	404
79	http://www.curtin.edu.my/sch_dept/business/CurtinSaraw	>>	url src	404
80	http://www.curtin.edu.my/sch_dept/business/R&D/index.ht		url src	404
81	http://www.curtin.edu.my/sch_dept/business/Commerce/in	>>	url src	404
82	http://www.curtin.edu.my/sch_dept/SOBusiness/contactW	>>	url src	404
83	http://www.curtin.edu.my/sch_dept/business/BusinessAdr	>>	url src	404
84	http://www.curtin.edu.my/sch_dept/media_culture_commu	>>	url src	404
85	http://www.curtin.edu.my/sch_dept/media_culture_commu	>>	url src	404
86	http://www.curtin.edu.my/R&D/sch_dept/A-Z_index.htm		url src	404
87	http://www.curtin.edu.my/R&D/contact_us_all.htm		url src	404
88	http://www.curtin.edu.my/prospective/contact_us_all.htm		url src	404
89	http://www.curtin.edu.my/prospective/Student_Complaints/	>>	url src	404
90	http://www.curtin.edu.my/learning_centre/IEP/IEP_Adms.ht		url src	404
91	http://www.curtin.edu.my/sch_dept/pre_university/Founda	>>	url src	404
92	http://www.curtin.edu.my/sch_dept/pre_university/Diploma	>>	url src	404
93	http://www.curtin.edu.my/csri/index.htm		url src	404
94	http://www.imi.gov.my/index.php/en/visa		url src	404
95	http://www.curtin.edu.my/future/future/shuttle.htm		url src	404
96	http://www.curtin.edu.my/current/publicholiday.htm		url src	404
97	http://www.curtin.edu.my/University%20Life/organization.h	>>	url src	404
98	http://www.curtin.edu.my/University%20Life/Photo%20Gal	>>	url src	404
99	http://www.curtin.edu.my/University%20Life/Photo%20Gal	>>	url src	404
100	http://www.curtin.edu.my/University%20Life/Photo%20Gal	>>	url src	404
101	http://www.curtin.edu.my/University%20Life/Photo%20Gal	>>	url src	404
102	http://www.curtin.edu.my/University%20Life/Photo%20Gal	>>	url src	404
103	http://www.curtin.edu.my/University%20Life/Photo%20Gal	>>	url src	404
104	http://www.curtin.edu.my/University%20Life/Photo%20Gal	>>	url src	404
105	http://lsn.curtin.edu.au/open/		url src	bad host
106	http://otl.curtin.edu.au/teaching_learning/		url src	404

107	http://www.cutin.edu.my		url src	bad host
108	http://www.curtin.edu.my/sch_dept/engineering_science/S	>>	url src	404
109	http://www.curtin.edu.my/sch_dept/engineering_science/S	>>	url src	404
110	http://www.curtin.edu.my/sch_dept/engineering_science/S	>>	url src	404
111	http://www.curtin.edu.my/sch_dept/engineering_science/S	>>	url src	404
112	http://www.curtin.edu.my/sch_dept/engineering_science/S	>>	url src	404
113	http://www.curtin.edu.my/sch_dept/engineering_science/S	>>	url src	404
114	http://www.curtin.edu.my/sch_dept/engineering_science/S	>>	url src	404
115	http://www.curtin.edu.my/sch_dept/engineering_science/S	>>	url src	404
116	http://www.curtin.edu.my/R&D/research_profile/Dr%20Chu	>>	url src	404
117	http://www.curtin.edu.my/current/calendar2014.htm		url src	404
118	http://www.curtin.edu.my/current/accommodation/index.htr		url src	404
119	http://www.curtin.edu.my/current/sch_dept/A-Z_index.htm		url src	404
120	http://www.curtin.edu.my/current/contact_us_all.htm		url src	404
121	http://www.curtin.edu.my/current/T&L/index.htm		url src	404
122	http://www.curtin.edu.my/current/international/index.htm		url src	404
123	http://www.curtin.edu.my/current/CurtinSarawak/about.htr		url src	404
124	http://www.curtin.edu.my/current/R&D/index.htm		url src	404
125	http://www.curtin.edu.my/current/examination/index.htm		url src	404
126	http://www.curtin.edu.my/Bookshop/CurtinSarawak/about.		url src	404
127	http://www.curtin.edu.my/campusnews/maito:yeeboon@c	>>	url src	404
128	http://www.curtin.edu.my/University%20Life/career_curre	>>	url src	404
129	http://www.curtin.edu.my//Learning%20Centre/Outreach/ir	>>	url src	404
130	https://oasis.curtin.edu.au/Auth/LogOn		url src	500
131	http://www.yayasan.org.my/download2.html		url src	404
132	http://app.mphe.gov.my/forum/		url src	bad host
133	http://labs.curtin.edu.my		url src	bad host
134	http://it-assistant01.staff.curtin.edu.my/hesk		url src	bad host
135	http://internal.curtin.edu.my/timer/index.html		url src	bad host
136	http://staffprofile.curtin.edu.my		url src	bad host
137	http://amuse.staff.curtin.edu.my/cs/ttx.cgi		url src	bad host
138	http://www.curtin.edu.my/R&D/researchers/ToR_RDC.htm		url src	404
139	http://www.curtin.edu.my/future/booking_application.htm		url src	404
140	http://www.curtin.edu.my/future/CurtinSarawak/about.htm		url src	404
141	http://www.curtin.edu.my/future/future/index.htm		url src	404

142	http://www.curtin.edu.my/future/international/index.htm		url src	404
143	http://www.curtin.edu.my/future/R&D/index.htm		url src	404
144	http://www.curtin.edu.my/future/current/index.htm		url src	404
145	http://www.curtin.edu.my/future/staff/staffIndex.asp		url src	404
146	http://www.curtin.edu.my/prospective/index.htm		url src	404
147	http://www.curtin.edu.my/prospective/apply_online.htm		url src	404
148	http://www.curtin.edu.my/prospective/fees/index.htm		url src	404
149	http://www.curtin.edu.my/future/academic_req.htm		url src	404
150	http://www.curtin.edu.my/Learning%20Centre/IEP/IEP.htm		url src	404
151	http://www.curtin.edu.my/sch_dept/engineering_science/a	>>	url src	404
152	http://www.curtin.edu.my/sch_dept/MassComm/requiremer	>>	url src	404
153	http://www.curtin.edu.my/sch_dept/SOEngineering/Postgra	>>	url src	404
154	http://www.curtin.edu.my/staff/Departments/Student_Serv	>>	url src	404
155	http://graduations.curtin.edu.au/graduate/ceremony/registe	>>	url src	404
156	http://www.curtin.edu.my/sch_dept/sch_dept/A-Z_index.h		url src	404
157	http://www.curtin.edu.my/sch_dept/Student_Complaints/in	>>	url src	404
158	http://www.curtin.edu.my/sch_dept/contact_us_all.htm		url src	404
159	https://staff.curtin.edu.my:4097/mail/		url src	bad host
160	http://www.curtin.edu.my/sch_dept/ICT/maito:it.helpdesk@	>>	url src	404
161	http://www.curtin.edu.my/international/international_office/	>>	url src	404
162	http://www.hornbill.bizland.com/index.html		url src	404
163	http://www.curtin.edu.my/learning_centre%20or%20call%:	>>	url src	404
164	http://www.curtin.edu.my/january_intake/		url src	404
165	http://www.curtin.edu.my/sch_dept/pre_university/Diploma	>>	url src	404
166	file:////moon/website$/contactweb.asp///moon/website$/cc	>>	url src	bad url
167	http://www.curtin.edu.my/Learning%20Centre/cute.htm		url src	404
168	http://research.curtin.edu.au/graduate/register.html		url src	404
169	http://www.ipedr.com/proceeding.htm		url src	404
170	http://handbook.curtin.edu.au/units/30/301065.html		url src	404
171	http://handbook.curtin.edu.au/units/12/12881.html		url src	404
172	http://handbook.curtin.edu.au/units/38/3822.html		url src	404
173	http://handbook.curtin.edu.au/units/10/10808.html		url src	404
174	http://handbook.curtin.edu.au/units/12/12598.html		url src	404
175	http://handbook.curtin.edu.au/units/12/12599.html		url src	404
176	http://www.curtin.edu.my/future/online_accommodation.as		url src	404

177	http://www.curtin.edu.my/sch_dept/Administrative/General	>>	url src	404
178	http://www.curtin.edu.my/future/future/accomodation.htm		url src	404
179	http://www.curtin.edu.my/future/representatives/future/ind	>>	url src	404
180	http://www.curtin.edu.my/future/representatives/internatio	>>	url src	404
181	http://www.curtin.edu.my/future/representatives/CurtinSar	>>	url src	404
182	http://www.curtin.edu.my/future/representatives/R&D/inde	>>	url src	404
183	http://www.curtin.edu.my/future/representatives/current/in	>>	url src	404
184	http://www.curtin.edu.my/future/representatives/staff/staf	>>	url src	404
185	http://www.curtin.edu.my/future/contactWeb.asp		url src	404
186	http://www.curtin.edu.my/UniversityLife/jcw_event.htm		url src	404
187	http://www.curtin.edu.my/R&D/ORD/sch_dept/A-Z_index.h		url src	404
188	http://www.curtin.edu.my/R&D/ORD/contact_us_all.htm		url src	404
189	http://www.curtin.edu.my//Learning%20Centre/LC_SProfile	>>	url src	404
190	http://www.curtin.edu.my/sch_dept/PreUniversity/FP_SPro	>>	url src	404
191	http://www.curtin.edu.my/sch_dept/media_culture_commu	>>	url src	404
192	http://www.curtin.edu.my/sch_dept/MassComm/SProfile.as	>>	url src	404
193	http://www.curtin.edu.my/sch_dept/PreUniversity/FP_SPro	>>	url src	404
194	http://www.curtin.edu.my/sch_dept/pre_university/PreU_S	>>	url src	404
195	http://www.curtin.edu.my//sch_dept/MassComm/SProfile.a	>>	url src	404
196	http://www.curtin.edu.my/sch_dept/PreUniversity/FP_SPro	>>	url src	404
197	http://www.curtin.edu.my/sch_dept/PreUniversity/FP_SPro	>>	url src	404
198	http://www.curtin.edu.my/Learning%20Centre/LC_SProfile.	>>	url src	404
199	http://www.curtin.edu.my/University%20Life/e-fact.htm		url src	404
200	http://www.curtin.edu.my/University%20Life/cv.htm		url src	404
201	http://www.curtin.edu.my/sch_dept/SOEngineering/Departr	>>	url src	404
202	http://shauntmw.youbloog.com/greenflow		url src	bad host
203	http://www.doiop.com/projectcare		url src	404
204	http://www.curtn.edu.my		url src	bad host
205	http://mastraveller.com/detikMH/		url src	404
206	http://research.humanities.curtin.edu.au/conferences/DDR/	>>	url src	404
207	http://research.humanities.curtin.edu.au/conferences/DDR/		url src	404
208	http://research.humanities.curtin.edu.au/conferences/DDR/	>>	url src	404
209	http://www.curtin.edu.my/sch_dept/pre_university/PreU_S	>>	url src	404
210	http://examinations.curtin.edu.au/students/intro_exams.cfm	>>	url src	404

211	hhttp://www.curtin.edu.my/milestones/milestones_2013.htr		url src	bad url
212	http://www.find.curtin.edu.au/indexschools.cfm		url src	404
213	http://www.curtin.edu.my//oweek/sch_dept/A-Z_index.htm		url src	404
214	http://www.curtin.edu.my//oweek/Student_Complaints/inde	>>	url src	404
215	http://www.curtin.edu.my//oweek/CurtinSarawak/about.htr		url src	404
216	http://www.curtin.edu.my//oweek/contact_us_all.htm		url src	404
217	http://www.curtin.edu.my/future/astro_tm_install.asp		url src	404
218	http://www.curtin.edu.my/UniversityLife/Career_Alumni/ca	>>	url src	404
219	http://www.curtin.edu.my/R&D/Student_Complaints/index.h		url src	404
220	http://www.curtin.edu.my/R&D/future/index.htm		url src	404
221	http://www.curtin.edu.my/R&D/international/index.htm		url src	404
222	http://www.curtin.edu.my/R&D/current/index.htm		url src	404
223	http://www.curtin.edu.my/R&D/ethics/index.htm		url src	404
224	http://www.curtin.edu.my/R&D/staff/index.htm		url src	404
225	http://www.curtin.edu.my/R&D/alumni/index.htm		url src	404
226	http://www.curtin.edu.my/R&D/campusnews/index.htm		url src	404
227	http://www.curtin.edu.my/R&D/jobs@curtin/index.htm		url src	404
228	http://www.curtin.edu.my/R&D/upcoming_conferences/ind	>>	url src	404
229	http://www.curtin.edu.my/R&D/campusnews/calendarever		url src	404
230	http://www.curtin.edu.my/R&D/learning_centre/index.htm		url src	404
231	http://www.curtin.edu.my/R&D/sch_dept/pre_university/inc	>>	url src	404
232	http://www.curtin.edu.my/R&D/sch_dept/business/index.ht		url src	404
233	http://www.curtin.edu.my/R&D/sch_dept/media_culture_co	>>	url src	404
234	http://www.curtin.edu.my/R&D/sch_dept/business/Postgra	>>	url src	404
235	http://www.curtin.edu.my/R&D/map.htm		url src	404
236	http://www.curtin.edu.my/R&D/Bookshop/index.htm		url src	404
237	http://www.curtin.edu.my/R&D/current/calendar.htm		url src	404
238	http://www.curtin.edu.my/R&D/future/fees/index.htm		url src	404
239	http://www.curtin.edu.my/R&D/scholarship/scholarship.htn		url src	404
240	http://www.curtin.edu.my/R&D/UniversityLife/index.htm		url src	404
241	http://www.curtin.edu.my/R&D/T&L/index.htm		url src	404
242	http://www.curtin.edu.my/R&D/CurtinSarawak/about.htm		url src	404
243	http://www.curtin.edu.my/R&D/staff/Departments/index.htr		url src	404
244	http://www.curtin.edu.my/R&D/current/forms.htm		url src	404
245	http://www.curtin.edu.my/R&D/courses/index.htm		url src	404
246	http://www.curtin.edu.my/R&D/events/index.htm		url src	404

247	http://www.curtin.edu.my/R&D/learning_centre/Outreach/ir	>>	url src	404
248	http://www.curtin.edu.my/R&D/UniversityLife/jcw.htm		url src	404
249	http://www.curtin.edu.my/R&D/UniversityLife/picture_galler	>>	url src	404
250	http://www.curtin.edu.my/R&D/UniversityLife/cv.htm		url src	404
251	http://www.curtin.edu.my/R&D/Miri/index.htm		url src	404
252	http://www.curtin.edu.my/R&D/staff/staffIndex.asp		url src	404
253	http://www.curtin.edu.my/jobs@curtin/about.htm		url src	404
254	http://www.curtin.edu.my/R&D/R&D/index.htm		url src	404
255	http://www.curtin.edu.my/jobs@curtin/CurtinSarawak/abou	>>	url src	404
256	http://www.miricouncil.gov.my/web/introduction_miri.html		url src	404
257	(http://students.curtin.edu.au/local/docs/certification_guidel	>>	url src	bad url
258	http://www.curtin.edu.my/campusnews/archives/future/inc	>>	url src	404
259	http://www.curtin.edu.my/campusnews/archives/internatic	>>	url src	404
260	http://www.curtin.edu.my/campusnews/archives/CurtinSa	>>	url src	404
261	http://www.curtin.edu.my/campusnews/archives/R&D/inde	>>	url src	404
262	http://www.curtin.edu.my/campusnews/archives/current/i	>>	url src	404
263	http://www.curtin.edu.my/campusnews/archives/staff/sta	>>	url src	404
264	http://www.curtin.edu.my/campusnews/archives/contactV	>>	url src	404
265	http://www.curtin.edu.my/campusnews/PR_11-23.htm		url src	404
266	http://www.curtin.edu.my/campusnews/PR_11-22.htm		url src	404
267	http://www.curtin.edu.my/campusnews/PR_11-21.htm		url src	404
268	http://www.curtin.edu.my/campusnews/PR_11-20.htm		url src	404
269	http://www.curtin.edu.my/campusnews/PR_11-19.htm		url src	404
270	http://www.curtin.edu.my/campusnews/PR_11-17.htm		url src	404
271	http://www.curtin.edu.my/campusnews/PR_11-18b.htm		url src	404
272	http://www.curtin.edu.my/campusnews/PR_11-18.htm		url src	404
273	http://www.curtin.edu.my/campusnews/PR_11-16.htm		url src	404
274	http://www.curtin.edu.my/campusnews/PR_11-15.htm		url src	404
275	http://www.curtin.edu.my/campusnews/archives/2007/arti	>>	url src	404
276	http://www.curtin.edu.my/campusnews/archives/2007/arti	>>	url src	404
277	http://www.curtin.edu.my/campusnews/mediarelease/Loca	>>	url src	404
278	http://www.curtin.edu.my/shortvideocompetition		url src	404
279	file:////moon/website$/learning%20centre/index.htm///Moon	>>	url src	bad url
280	file:////moon/website$/events/index.htm///Moon/website$/e	>>	url src	bad url

281	http://www.curtin.edu.my/sch_dept/ICT/students/maito:it.he	>>	url src	404
282	https://payloan.mohe.gov.my/MyBrain15/index2.php		url src	bad host
283	https://payloan.mohe.gov.my/MyBrain15/index_mymaster.pl		url src	bad host
284	https://payloan.mohe.gov.my/MyBrain15/index_myphd.php		url src	bad host
285	http://www.curtin.edu.my/Health_Counseling/counselling.ht	>>	url src	404
286	http://www.curtin.edu.my/Health_Counseling/counselling_s	>>	url src	404
287	http://www.curtin.edu.my/Records_Archive/campus_life_t	>>	url src	404
288	http://www.curtin.edu.my/University%20Life/referral.htm		url src	404
289	http://www.curtin.edu.my/R&D/phd_scholarships		url src	404
290	http://www.facebook.com/%20CurtinUniversitySarawakMa	>>	url src	404
291	http://green2012.co.cc		url src	504
292	http://www.curtin.edu.my.%20more		url src	bad host
293	http://www.curtin.edu.my/sch_dept/pre_university/Diploma	>>	url src	404
294	http://www.curtin.edu.my/sch_dept/engineering_science/a	>>	url src	404
295	http://www.karyawanmuda.com.my		url src	bad host
296	http://www.curtin.edu.my/csri/		url src	404
297	http://www.facebook.com/profile.php?id=685093063		url src	404
298	http://www.curtin.edu.my/osh/%20or%20call%20085-443		url src	404
299	http://www.curtin.edu.my/campusnews/mediarelease/200	>>	url src	404
300	http://www.curtin.edu.edu.my		url src	bad host
301	http://www.curtin.edu.my/campusnews/mediarelease/200	>>	url src	404
302	http://www.curtin.edu.my/cutse2007/		url src	404
303	http://www.dph.gov.my/vektor/eng/index.htm		url src	bad host
304	http://www.curtin.edu.my/cutse2006		url src	404
305	http://www.curtin.edu.my/sch_dept/MassComm/BA_Mass(	>>	url src	404
306	http://www.curtin.edu.my/R&D/forStudents/Master.htm		url src	404
307	http://www.curtin.edu.my/R&D/forStudents/PhD.htm		url src	404
308	http://www.cisco.com/web/learning/netacad/index.html)		url src	404
309	http://library.curtin.edu.au/dblist/index.html		url src	404
310	http://library.curtin.edu.au/ebooks/index.html		url src	404
311	http://library.curtin.edu.au/theses/index.html		url src	404
312	http://lis.curtin.edu.my		url src	bad host
313	http://www.curtin.edu.my/campusnews/mediarelease/200	>>	url src	404
314	http://www.curtin.edu.my/campusnews/mediarelease/200	>>	url src	404
315	http://www.curtin.edu.my/current/contact_us.htm		url src	404

316	http://www.curtin.edu.my//map.htm		url src	404
317	http://www.curtin.edu.my//current/accommodation/index.ht		url src	404
318	http://www.mapmyevent.com/map/index.php?eid=25974		url src	500
319	http://www.curtin.edu.my/sch_dept/media_culture_commu	>>	url src	404
320	http://www.curtin.edu.my/future/complaint.asp		url src	404
321	http://www.curtin.edu.my/future/changing_room.asp		url src	404
322	http://www.curtin.edu.my/future/complaint_test.asp		url src	404
323	http://www.curtin.edu.my/future/astro_tm_install_test.asp		url src	404
324	(http://apps.curtin.edu.my/ERO/Admin/Event.aspx?id=209		url src	bad url
325	http://www.nestle.com.my/Pages/Nestle.aspx		url src	404
326	http://www.curtin.edu.my/UniversityLife/Career_Alumni/ev	>>	url src	404
327	http://www.curtin.edu.my/staffIndex.asp		url src	404
328	http://www.curtin.edu.my/campusnews/archives/2005/lev	>>	url src	404
329	http://www.curtin.edu.my/campusnews/archives/2005/chr	>>	url src	404
330	http://www.curtin.edu.my/enrolment/index.htm		url src	404
331	https://staf.curtin.edu.my:4097/mail/schedule.html		url src	bad host
332	http://www.curtin.edu.my/University%20Life/request.htm		url src	404
333	http://www.curtin.edu.my/University%20Life/housing/index	>>	url src	404
334	http://www.curtin.edu.my/University%20Life/request.asp		url src	404
335	http://www.curtin.edu.my//sch_dept/Administrative/Genera	>>	url src	404
336	http://www.curtin.edu.my//sch_dept/Administrative/Genera	>>	url src	404
337	http://www.curtin.edu.my//sch_dept/pre_university/index.h	>>	url src	404
338	http://www.curtin.edu.my//sch_dept/engineering_science/i	>>	url src	404
339	http://www.curtin.edu.my//sch_dept/Administrative/Genera	>>	url src	404
340	http://www.curtin.edu.my//sch_dept/Administrative/Genera	>>	url src	404
341	http://www.curtin.edu.my//sch_dept/Administrative/Genera	>>	url src	404
342	http://www.curtin.edu.my//sch_dept/Administrative/Genera	>>	url src	404
343	http://www.curtin.edu.my//sch_dept/Administrative/Genera	>>	url src	404
344	http://www.curtin.edu.my//sch_dept/Administrative/Genera	>>	url src	404
345	http://www.curtin.edu.my//sch_dept/Administrative/Genera	>>	url src	404
346	http://www.curtin.edu.my//sch_dept/Administrative/Genera	>>	url src	404
347	http://www.curtin.edu.my//sch_dept/Administrative/Genera	>>	url src	404
348	http://www.curtin.edu.my//sch_dept/Administrative/Genera	>>	url src	404
349	http://www.curtin.edu.my//University%20Life/future/index.l	>>	url src	404

350	http://www.curtin.edu.my//University%20Life/international/i	>>	url src	404
351	http://www.curtin.edu.my//University%20Life/CurtinSaraw	>>	url src	404
352	http://www.curtin.edu.my//University%20Life/R&D/index.ht		url src	404
353	http://www.curtin.edu.my//University%20Life/current/index	>>	url src	404
354	http://www.curtin.edu.my//University%20Life/staff/staffInd	>>	url src	404
355	http://www.curtin.edu.my//sch_dept/international/index.htm	>>	url src	404
356	http://www.curtin.edu.my//sch_dept/R&D/index.htm		url src	404
357	http://www.curtin.edu.my//sch_dept/CurtinSarawak/about.	>>	url src	404
358	http://www.curtin.edu.my//sch_dept/contactWeb.asp		url src	404
359	http://www.curtin.edu.my//Student_Complaints/index.htm		url src	404
360	http://www.curtin.edu.my//current/Curtin%20Pool%20Club		url src	404
361	http://www.curtin.edu.my//current/SPE%20student%20cha	>>	url src	404
362	http://www.curtin.edu.my//sch_dept/ICT/CurtinSarawak/ab	>>	url src	404
363	http://www.curtin.edu.my//staff/staffIndex.asp		url src	404
364	http://www.curtin.edu.my//events/CurtinSarawak/about.htr		url src	404
365	http://www.curtin.edu.my//sch_dept/MassComm/index.htm		url src	404
366	http://www.curtin.edu.my//sch_dept/business/Postgraduat	>>	url src	404
367	http://www.curtin.edu.my//University%20Life/picture_galle	>>	url src	404
368	http://www.curtin.edu.my//Bookshop/contact_us_all.htm		url src	404
369	https://oasis.curtin.edu.au/LoginContactUs		url src	500
370	http://www.rswit.rsb.my		url src	bad host
371	http://www.jtkswk.mohr.com.my		url src	bad host
372	http://www.shippingcorp.com.my		url src	bad host
373	http://www.curtin.edu.my/cutse2013/maito:CUTSE2013Sec	>>	url src	404
374	http://www.curtin.edu.my/cutse2012/maito:CUTSE2012Sec	>>	url src	404
375	httw://www.megahotel.com.my		url src	bad url
376	http://www.curtin.edu.my/10thanniversary/gallery/open_da	>>	url src	404
377	http://www.curtin.edu.my//sch_dept/current/index.htm		url src	404
378	http://www.curtin.edu.my//sch_dept/ICT/sch_dept/A-Z_ind	>>	url src	404
379	http://www.curtin.edu.my//sch_dept/ICT/Student_Complaint	>>	url src	404
380	http://www.curtin.edu.my//sch_dept/ICT/contact_us_all.htn		url src	404
381	http://www.curtin.edu.my//current/international/index.htm		url src	404
382	http://www.curtin.edu.my//current/CurtinSarawak/about.ht		url src	404
383	http://www.curtin.edu.my//current/R&D/index.htm		url src	404
384	http://www.curtin.edu.my//current/examination/index.htm		url src	404

385	http://www.curtin.edu.my//sch_dept/business/international	>>	url src	404
386	http://www.curtin.edu.my//sch_dept/business/R&D/index.h		url src	404
387	http://www.curtin.edu.my//sch_dept/business/CurtinSaraw	>>	url src	404
388	http://www.curtin.edu.my//sch_dept/business/Commerce/ir	>>	url src	404
389	http://www.curtin.edu.my//sch_dept/SOBusiness/contactW	>>	url src	404
390	http://www.curtin.edu.my//sch_dept/business/BusinessAd	>>	url src	404
391	http://www.curtin.edu.my//sch_dept/ICT/maito:it.helpdesk@	>>	url src	404
392	http://www.curtin.edu.my//prospective/Student_Complaints	>>	url src	404
393	http://www.curtin.edu.my//prospective/contact_us_all.htm		url src	404
394	http://www.curtin.edu.my//R&D/sch_dept/A-Z_index.htm		url src	404
395	http://www.curtin.edu.my//R&D/contact_us_all.htm		url src	404
396	http://www.curtin.edu.my//R&D/research_profile/Dr%20Ch	>>	url src	404
397	http://www.curtin.edu.my//future/future/shuttle.htm		url src	404
398	http://www.curtin.edu.my//faq.htm		url src	404
399	http://www.curtin.edu.my//campusnews/maito:yeeboon@c	>>	url src	404
400	http://www.curtin.edu.my//Corp%20Comm/index.htm		url src	404
401	http://www.curtin.edu.my//future/housing.htm		url src	404
402	http://www.curtin.edu.my//University%20Life/john_curtin_a	>>	url src	404
403	http://www.curtin.edu.my//University%20Life/jcw_history.h	>>	url src	404
404	http://www.curtin.edu.my//University%20Life/volunteers_c	>>	url src	404
405	http://www.curtin.edu.my//University%20Life/staff.asp		url src	404
406	http://www.curtin.edu.my//UniversityLife/maito:%20univers	>>	url src	404
407	http://www.curtin.edu.my//University%20Life/http/cv.curtin	>>	url src	404
408	http://www.curtin.edu.my//learning_centre/IEP/IEP_Adms.ht	>>	url src	404
409	http://www.curtin.edu.my//sch_dept/pre_university/Founda	>>	url src	404
410	http://www.curtin.edu.my//sch_dept/pre_university/Diploma	>>	url src	404
411	http://www.curtin.edu.my/RecordsArchive/campus_life_tes	>>	url src	404
412	http://www.curtin.edu.my/csri/softlaunch.htm		url src	404
413	http://www.pertanika2.upm.edu.my/jpertanika/index.htm		url src	404
414	http://www.xfab.com/index.php?id=1		url src	404
415	http://www.waset.org/journals/		url src	404
416	http://www.sains.com.my/sains/html/index.shtml		url src	404
417	http://www.curtin.edu.my//sch_dept/media_culture_commu	>>	url src	404
418	http://www.curtin.edu.my//sch_dept/media_culture_commu	>>	url src	404

419	http://www.curtin.edu.my//sch_dept/sch_dept/A-Z_index.h		url src	404
420	http://www.curtin.edu.my//sch_dept/Student_Complaints/in	>>	url src	404
421	http://www.curtin.edu.my//sch_dept/contact_us_all.htm		url src	404
422	http://www.curtin.edu.my//future/online_accommodation.as		url src	404
423	http://www.curtin.edu.my//staff/Departments/Student_Serv	>>	url src	404
424	http://www.curtin.edu.my//staff/contactWeb.asp		url src	404
425	http://www.curtin.edu.my//jobs@curtin/about.htm		url src	404
426	http://www.curtin.edu.my//jobs@curtin/CurtinSarawak/abo	>>	url src	404

Curtin University's followed and no-followed back links are compared in Table D and shown below:

D. Curtin University Followed Vs. No-Followed Back Links

Followed Back Links	***No-Followed Back Links***
80%	20%

Curtin University's total pages, good links and the broken links are shown below in Table E.

E. Curtin University Broken Links

Total Pages	***Good Links***	***Broken Links***
3000	2574	426

Curtin University's different types of broken links are shown in Table F below:

F. Curtin University type of Broken Links

Total Pages	***Broken Links***	***404 Error***	***Bad URL***	***Bad Host***	***500 Error***	***504 Error***
3000	426	381	10	31	3	1