

Improving ARC algorithm for Pure Topic Distillation

TH-5093



Developed by:



NUSRAT SHAHEEN
(271- FAS/MSCS/F05)

KHADEEJA-AL-MADNI
(267- FAS/MSCS/F05)

Supervised by:

PROF.DR. M. SIKANDER HAYAT KHIYAL

**Department of Computer Science
Faculty of Basic and Applied Sciences
International Islamic University Islamabad
2008**



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

*In the Name of Allah The Most Beneficent
The Most Merciful*



Department of Computer Science
International Islamic University, Islamabad

Date: 18-09-2008

FINAL APPROVAL

It is certified that we have read the project titled "Improving ARC algorithm for Pure Topic Distillation" submitted by Miss NUSRAT SHAHEEN Reg. No. 271-FAS/MSCS/F05 and Miss KHADEEJA-AL-MADNI Reg. No. 267-FAS/MSCS/F05. It is our judgment that this project is of sufficient standard to warrant its acceptance by International Islamic University, Islamabad for the degree MS in Computer Science.

COMMITTEE

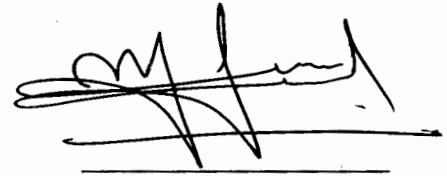
External Examiner:

Dr.A.Sattar
Fmr.D.G.Pakistan Computer Bureau
Islamabad



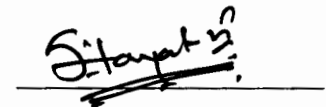
Internal Examiner:

Muhammad Imran Saeed
Assistant Professor
Department of Computer Science
International Islamic University
Islamabad



Supervisors:

Dr. M. Sikandar Hayat Khiyal
Chairperson, Department of Computer Science/
Software Engineering, Fatima Jinnah Women
University, The Mall, Rawalpindi



**A dissertation submitted to the
Department of Computer Science,
International Islamic University, Islamabad
as a partial fulfillment of the requirements
for the award of the degree of
MS in Computer Science**

To Our Loving Parents

“My Lord have Mercy on them (Parents) both as they did care for me when
I was little”

(AL-QURAN 17:24)



Appendix-B

Screen Shots

Appendix B

SCREEN SHOTS

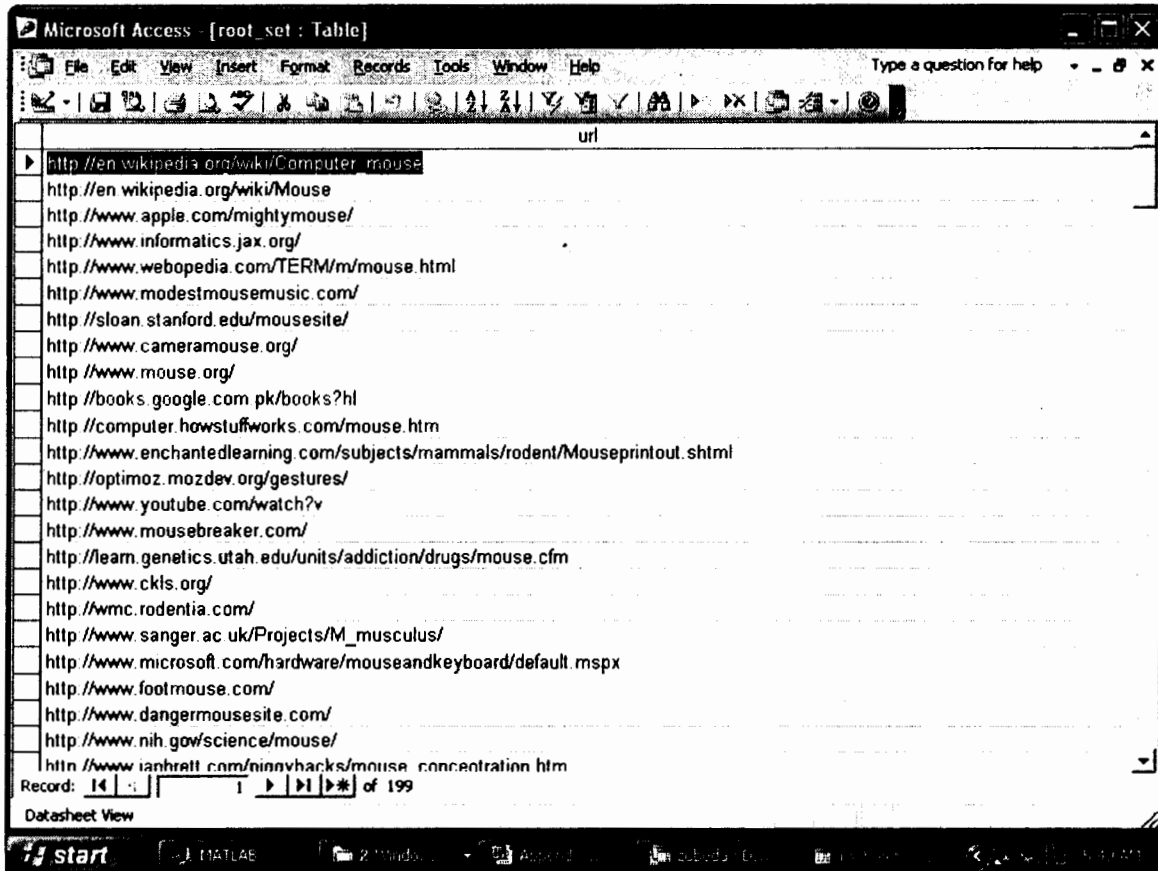


Fig B-1: Root Set

uri	hub
http://www.genome.gov/10001859	10
http://www.informatics.jax.org/mgihome/nomen/	10
http://www.fimre.org/	9
http://www.nature.com/nature/mousegenome/	8
http://www.bcgsc.ca/platform/mapping/mouse	6
http://www.tigr.org/tdb/bac_ends/mouse/bac_end_int	6
http://www.informatics.jax.org/	6
http://www.myspace.com/modestmouse	5
http://itmanagement.earthweb.com/features/article.	4
http://the-mouse-trap.blogspot.com/	4
http://en.wikipedia.org/wiki/Mouse	4
http://www.modestmousemusic.com/	4
http://www.newtonperipherals.com/	1
http://www.newtonperipherals.com/products.html	1
http://www.nervenet.org/main/dictionary.html	1
http://www.mbl.org/	1
http://www.evoluent.com/vm3.html	1
http://www.animax.no/	1
http://tech.ln.lib.mi.us/tutor/welcome.htm	1
http://optimoz.mozdev.org/gestures/installation.ht	1
http://optimoz.mozdev.org/gestures/	1
http://liono.teamrubber.com/rubber_client_work/mou	1
http://www.montrosesecam.com/	1
http://en.wikipedia.org/wiki/Mouse	1

Record: 1 of 25
Datasheet View

Fig B-2: Root_Hub

url	authority
http://www.informatics.jax.org/	6
http://en.wikipedia.org/wiki/Mouse	4
http://phenome.jax.org/pub-cgi/phenome/mpdcgi	2
http://www.eucomm.org/	2
http://www.sanger.ac.uk/Projects/M_musculus/	2
http://www.informatics.jax.org/mgihome/nomen/	2
http://www.miceandrats.com/	1
http://www.modestmousemusic.com/	1
http://computer.howstuffworks.com/mouse.htm	1
http://optimoz.mozdev.org/gestures/	1
http://www.ckis.org/	1
http://www.newtonperipherals.com/	1
http://mousesnp.roche.com/	1
http://www.fimre.org/	1
http://en.wikipedia.org/wiki/Mouse	1
http://www.computer-engineering.org/ps2mouse/	1
http://www.newtonperipherals.com/products.html	1
http://www.tigr.org/tdb/bac_ends/mouse/bac_end_intro.html	1
http://www.sri.com/about/timeline/mouse.html	1
http://solutions.3m.com/wps/portal/3M/en_US/ergonomics/home/products/ergono	1
http://www.emmanet.org/	1
http://optimuz.mozdev.org/gestures/installation.html	1
http://www.highrez.co.uk/downloads/XMouseButtonControl.htm	1
http://www.mousemailer.nrn/	1

Record: 14 of 27
Datasheet View

Fig B-3:Root_Auth

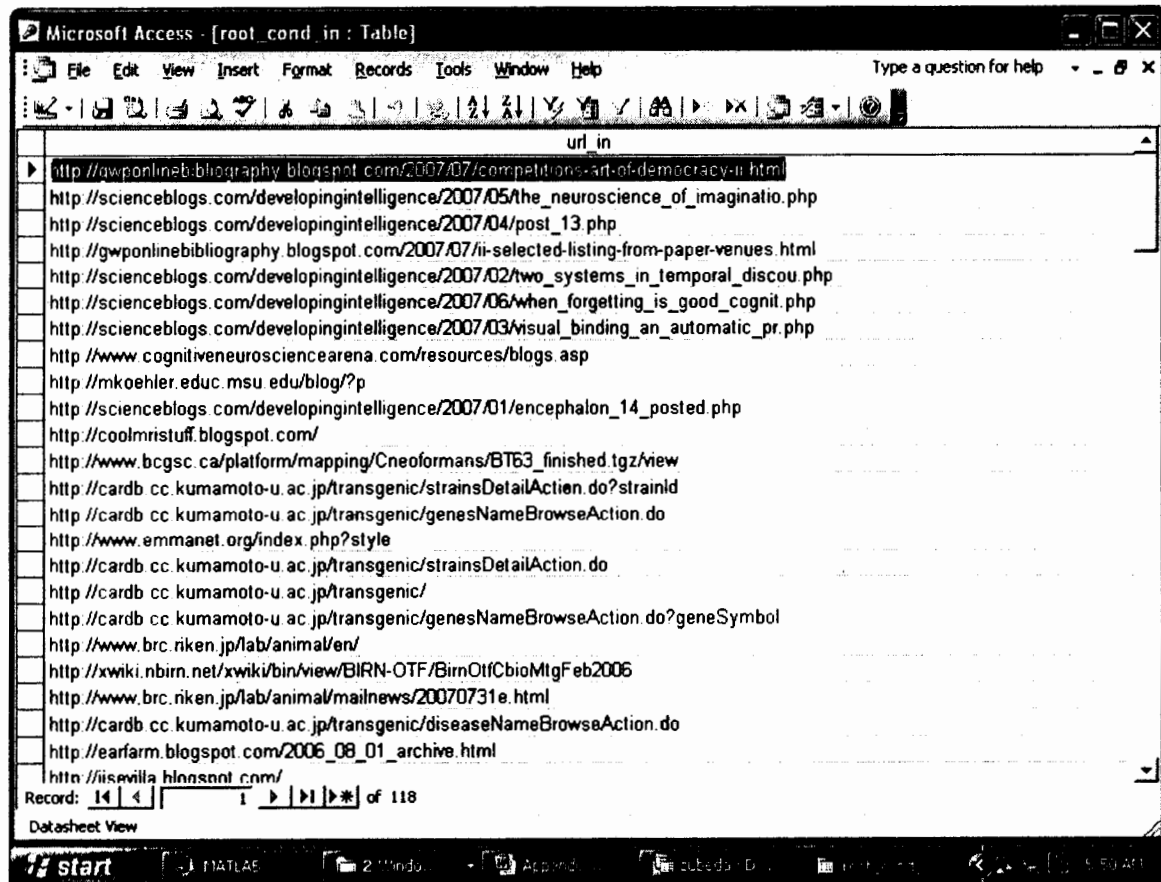


Fig B-4: Candidate Pages_InLinks

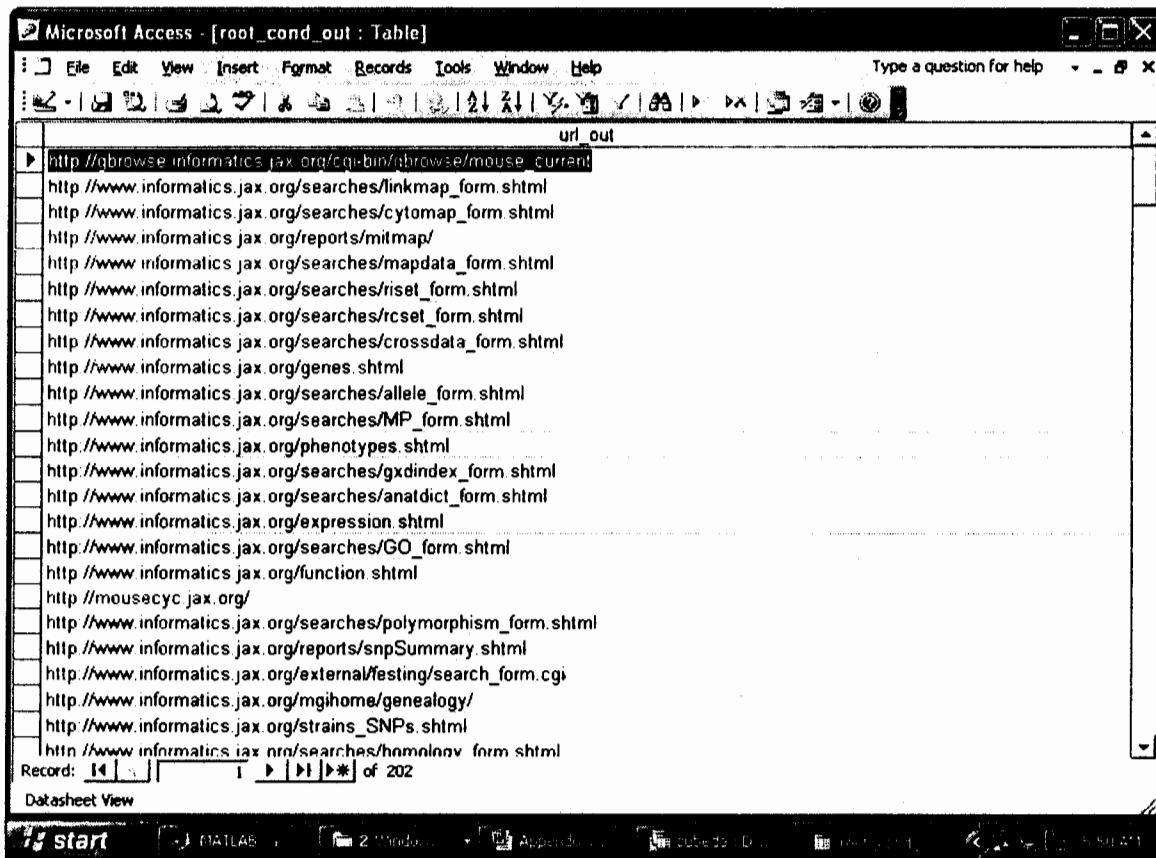


Fig B-5: Candidate Pages_OutLinks

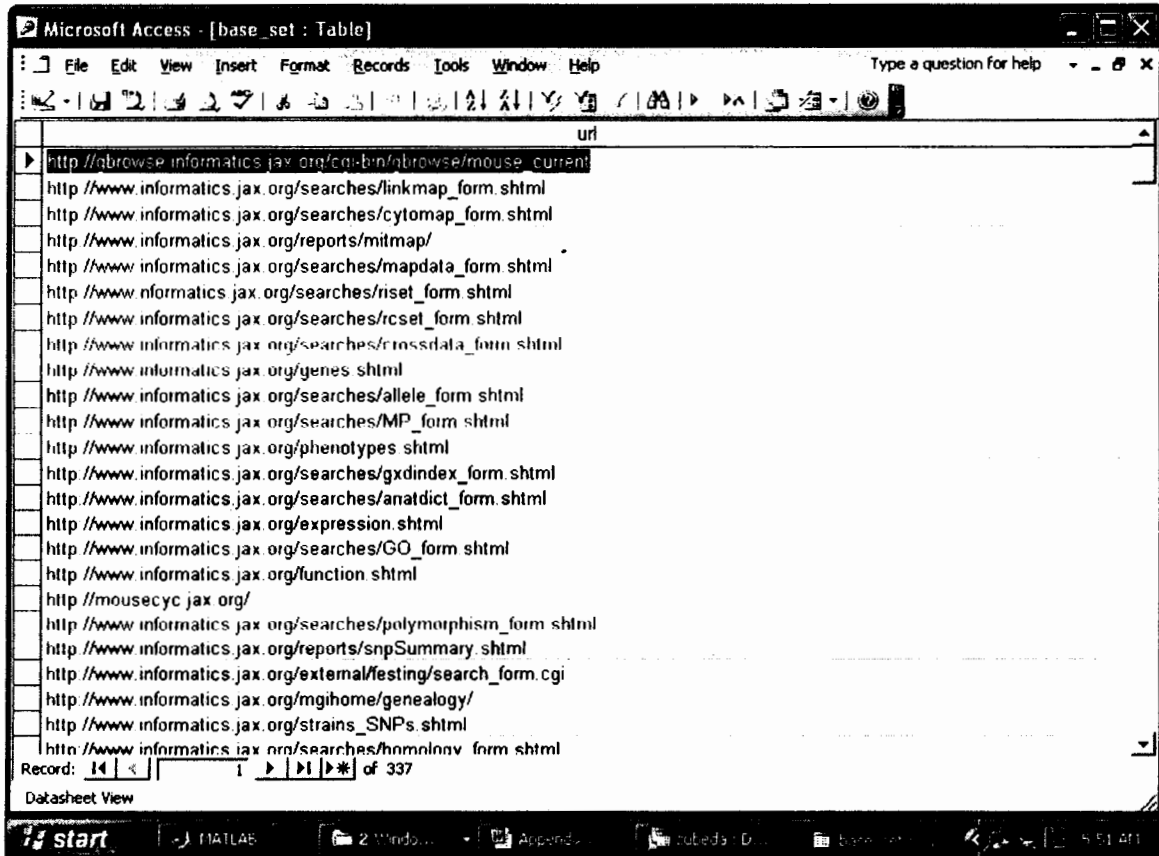
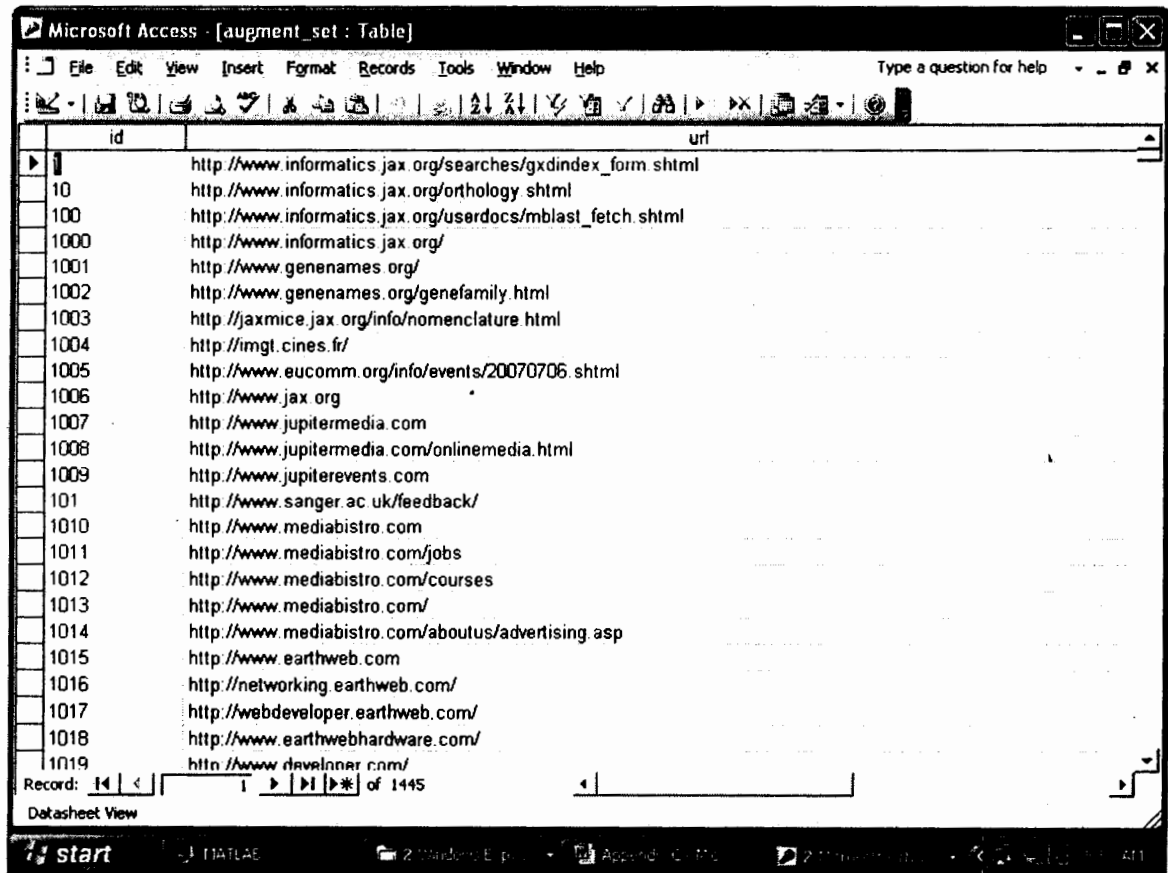


Fig B-6: Base_Set



The screenshot shows a Microsoft Access window titled "Microsoft Access - [augment_set : Table]". The window displays a table with two columns: "id" and "url". The table contains 19 rows of data, with the first row highlighted. The "id" column contains values from 1 to 1019, and the "url" column contains various web addresses. The window also shows a menu bar (File, Edit, View, Insert, Format, Records, Tools, Window, Help) and a toolbar. At the bottom, there is a status bar indicating "Record: 1 of 1445" and "Datasheet View".

id	url
1	http://www.informatics.jax.org/searches/gxindex_form.shtml
10	http://www.informatics.jax.org/orthology.shtml
100	http://www.informatics.jax.org/userdocs/mbblast_fetch.shtml
1000	http://www.informatics.jax.org/
1001	http://www.genenames.org/
1002	http://www.genenames.org/genefamily.html
1003	http://jaxmice.jax.org/info/nomenclature.html
1004	http://imgt.cines.fr/
1005	http://www.eucomm.org/info/events/20070706.shtml
1006	http://www.jax.org
1007	http://www.jupitermedia.com
1008	http://www.jupitermedia.com/onlinemedias.html
1009	http://www.jupiterevents.com
101	http://www.sanger.ac.uk/feedback/
1010	http://www.mediabistro.com
1011	http://www.mediabistro.com/jobs
1012	http://www.mediabistro.com/courses
1013	http://www.mediabistro.com/
1014	http://www.mediabistro.com/aboutus/advertising.asp
1015	http://www.earthweb.com
1016	http://networking.earthweb.com/
1017	http://webdeveloper.earthweb.com/
1018	http://www.earthwebhardware.com/
1019	http://www.devalnner.com/

Fig B-7: Augment_Set

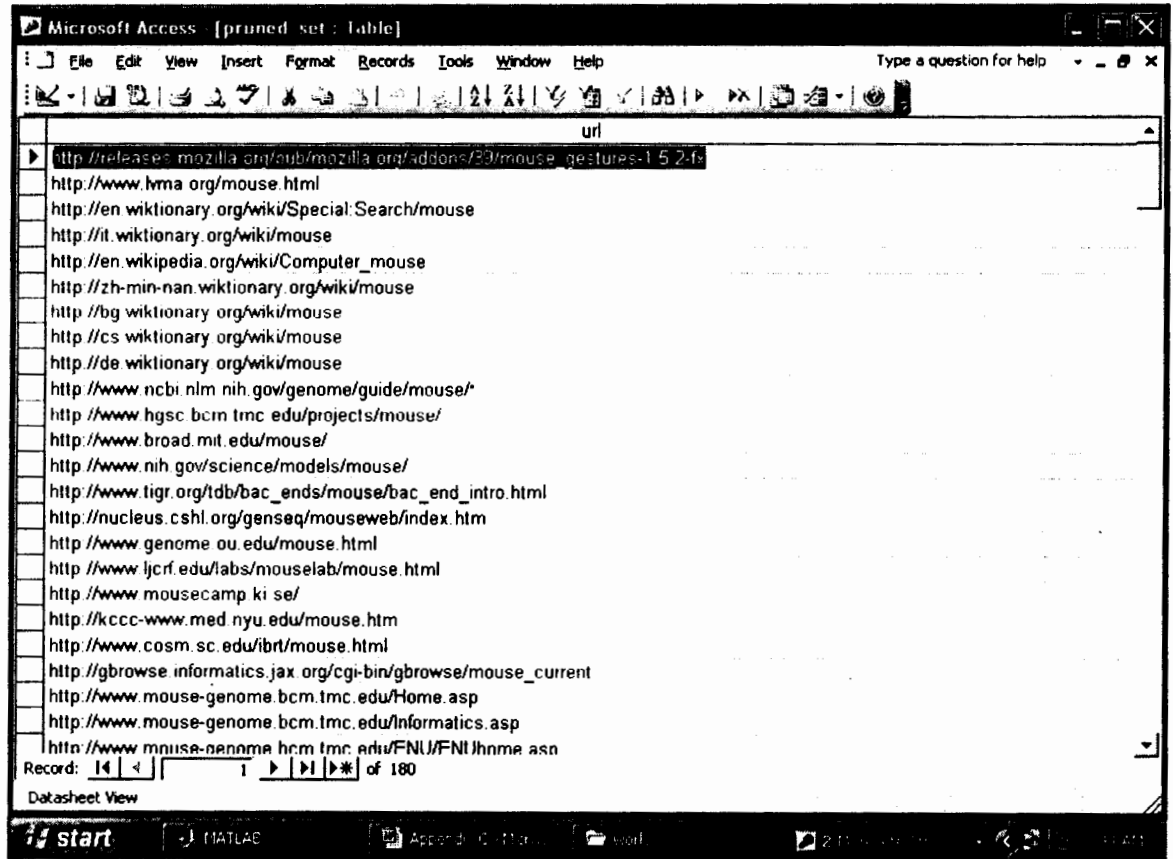


Fig B-8: Pruned_Set

url	hub
http://fr.wiktionary.org/wiki/mouse	33
http://ko.wiktionary.org/wiki/mouse	32
http://hy.wiktionary.org/wiki/mouse	32
http://io.wiktionary.org/wiki/mouse	32
http://id.wiktionary.org/wiki/mouse	33
http://kk.wiktionary.org/wiki/mouse	33
http://lo.wiktionary.org/wiki/mouse	34
http://la.wiktionary.org/wiki/mouse	33
http://lt.wiktionary.org/wiki/mouse	33
http://hu.wiktionary.org/wiki/mouse	35
http://nl.wiktionary.org/wiki/mouse	35
http://ja.wiktionary.org/wiki/mouse	33
http://oc.wiktionary.org/wiki/mouse	34
http://pl.wiktionary.org/wiki/mouse	29
http://pt.wiktionary.org/wiki/mouse	34
http://ru.wiktionary.org/wiki/mouse	31
http://so.wiktionary.org/wiki/mouse	27
http://sr.wiktionary.org/wiki/mouse	34
http://fi.wiktionary.org/wiki/mouse	32
http://sv.wiktionary.org/wiki/mouse	33
http://ta.wiktionary.org/wiki/mouse	34
http://te.wiktionary.org/wiki/mouse	32
http://vi.wiktionary.org/wiki/mouse	33
http://tr.wiktionary.org/wiki/mouse	32

Record: 1 of 78

Datasheet View

Fig B-9: Augment_Hub

url	authority
http://zh-min-nan.wiktionary.org/wiki/mouse	36
http://bg.wiktionary.org/wiki/mouse	36
http://it.wiktionary.org/wiki/mouse	36
http://cs.wiktionary.org/wiki/mouse	34
http://de.wiktionary.org/wiki/mouse	36
http://www.ncbi.nlm.nih.gov/genome/guide/mouse/	5
http://www.nih.gov/science/models/mouse/	25
http://www.tigr.org/tdb/bac_ends/mouse/bac_end_intro.html	8
http://gbrowse.informatics.jax.org/cgi-bin/gbrowse/mouse_current	6
http://www.mouse-genome.bcm.tmc.edu/Home.asp	16
http://www.mouse-genome.bcm.tmc.edu/Informatics.asp	13
http://el.wiktionary.org/wiki/mouse	36
http://es.wiktionary.org/wiki/mouse	36
http://eu.wiktionary.org/wiki/mouse	34
http://fa.wiktionary.org/wiki/mouse	36
http://fr.wiktionary.org/wiki/mouse	36
http://ko.wiktionary.org/wiki/mouse	36
http://hy.wiktionary.org/wiki/mouse	36
http://io.wiktionary.org/wiki/mouse	36
http://id.wiktionary.org/wiki/mouse	23
http://kk.wiktionary.org/wiki/mouse	36
http://lo.wiktionary.org/wiki/mouse	18
http://la.wiktionary.org/wiki/mouse	33
http://li.wiktionary.org/wiki/mouse	34

Record: 14 of 61
Datasheet View

Fig B-10: Augment_Authority

REFERENCES

- [1] Bharat, K., and Henzinger, M. R. Improved algorithms for topic distillation in a hyperlinked environment. In Proceedings of SIGIR-98, 21st ACM International
- [2] Chakrabarti, S. Mining the Web: Discovering Knowledge from Hypertext Data. Conference on Research and Development in Information Retrieval (Melbourne, AU, 1998), pp. 104–111. Morgan-Kaufmann, 2002.
- [3] Chakrabarti, S., Dom, B., Gibson, D., Kleinberg, J., Raghavan, P., and Rajagopalan, S. Automatic resource list compilation by analyzing hyperlink structure and associated text. In Proceedings of the 7th International World Wide Web Conference (1998).
- [4] Chakrabarti, S., Dom, B. E., Kumar, S. R., Raghavan, P., Rajagopalan, S., Tomkins, A., Gibson, D., and Kleinberg, J. Mining the Web's link structure. *IEEE Computer* 32, 8 (1999), 60–67.
- [5] Google api home page, <http://www.google.com/apis/>.
- [6] JON M. KLEINBERG Cornell University, Ithaca, New York 1997 Authoritative Sources in a Hyperlinked Environment
- [7] Selective Hypertext Induced Topic Search Amit C. Awekar NC State University Raleigh, NC 27695, USA acawekar@ncsu.edu Pabitra Mitra Indian Institute of Technology Kharagpur, India – 721302 pabitra@cse.iitkgp.ernet.in Jaewoo Kang NC State University Raleigh, NC 27695, USA kang@csc.ncsu.edu. May 2006
- [8] Yahoo! web search services home page, <http://developer.yahoo.net/>.
-

[9] [http:// altavista.com/links](http://altavista.com/links)

[10] [http:// mathworks.com/help](http://mathworks.com/help)

[11] Brin, S., and Page, L. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* 30, 1–7 (1998), 107–117.

[12] Larson, R. Bibliometrics of the World Wide Web: An exploratory analysis of the intellectual structure of cyberspace. In *Annual Meeting of the American Society for Information Science* (1996).

[13] [http:// wikipedia.com](http://wikipedia.com)



DECLARATION

We, hereby declare that “Improving ARC algorithm for Pure Topic Distillation” software, neither as a whole nor as a part thereof has been copied out from any source. We have developed this software and the accompanied report entirely on the basis of our personal efforts made under the sincere guidance of our supervisor. No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institution of learning.

Nusrat Shaheen
271-FAS/MSCS/F05

Khadeeja-al-Madni
267-FAS/MSCS/F05

ACKNOWLEDGEMENTS

We bow our heads, in deep gratitude, before THE ALMIGHTY ALLAH for Blessing us with the wisdom and the capability and granting us the strength to accomplish this project.

We are very thankful to our kind, dynamic and able supervisor, Prof. Dr.M.Sikandar Hayat Khiyal for taking the time out of his ever busy schedule in providing us the guidance and direction and also always gave us his all out help and assistance right from the tricky and visionary stage of laying down the conceptual framework to the far more exacting stage of actual execution of this project successfully.

We adore and pray for all our teachers and friends, particularly Miss.Zubeda Khannum for guiding and assisting us in acquiring right type of knowledge at right time. We shall ever remain grateful to all of them for their kind help.

We are extremely grateful to our beloved parents for helping us grow mentally right from the day one in school besides catering to all our needs, affording every facility, consoling us when we at times felt dejected and fatigued, inspiring us when we started feeling that the project was getting beyond our control and grasp and praying day in and day out for our success.

We are indeed also very appreciative of the contribution of our younger sisters and brothers for their continuous solace and forbearance, which gave us courage and strength to complete our project in time.

Nusrat Shaheen

Khadeeja-al-Madni

PROJECT IN BRIEF

Project Title:	Improving ARC algorithm for Pure Topic Distillation
Objective:	To overcome the problem of topic distillation in existing ARC algorithm.
Undertaken By:	Nusrat Shaheen Khadeeja-al-Madni
Supervised By:	Prof.Dr. M.Sikander Hayat Khiyal
Starting Date:	February 2007
Completion Date:	August 2008.
Operating Systems:	Windows XP
Tool Used:	Matlab, Microsoft Access
System Used:	Intel Pentium IV, 2.25 MHz processor

ABSTRACT

The rapidly growing World Wide Web now contains more than three billion web pages of text, images and other multimedia information. While this vast amount of information has the potential to benefit all aspects of our society, finding the relevant web pages to satisfy a user's information need still remains an important and challenging task. Many commercial search engines have been developed and used by people all over the world. However, the relevancy of web pages returned by search engine is still lacking, and further research and development are needed to make search engine more effective as a ubiquitous information-seeking tool. To satisfy a user's information need still remains an important and challenging task.

In this thesis we have worked on Automatic Resource Compilation by Analyzing the Hyperlinked Structure and Associated Text algorithm to solve the problem of topic contamination and topic drift. We have improved this algorithm by selectively expanding the root set for distilling pure topic against user query. The main features that differentiate our work are, selective expansion of root set and the way we calculate hub and authority values. As a result we have a much smaller expanded root set and are able to distill a pure topic even if the query is ambiguous. The results have shown that we have overcome the above mentioned problem by getting the most appropriate pages for user queries.

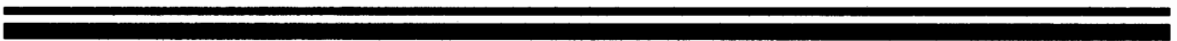
TABLE OF CONTENTS

1.0 INTRODUCTION.....	1
1.1 WEB MINING	3
1.1.1A Taxonomy of web mining.....	3
1.1.2 Web content mining.....	4
1.1.3 Web structure mining.....	5
1.1.4 Web usage mining.....	6
1.2 INFORMATION RETRIEVAL.....	7
1.3 TOPIC DISTILLATION.....	8
1.4 SEARCH ENGINE.....	9
1.5 HYPERLINKS.....	10
1.5.1 Significance of Hyperlink.....	10
1.6 WEB AS A DIRECTED GRAPH.....	11
1.7 EXISTING LINK ANALYSIS ALGORITHM.....	13
1.7.1 Page Rank.....	13
1.7.2 Hypertext Induced Topic Selection (HITS).....	15
1.7.3 SALSA.....	15
1.7.4 Trst Rank.....	16
1.8 EVALUATION OF WEB SEARCH RESULTS.....	16
2.0 LITERATURE SURVEY.....	18
2.1 AUTHORITATIVE SOURCES IN A HYPERLINKED ENVIRONMENT.....	18
2.2 MINING THE LINK STRUCTURE OF WORLD WIDE WEB.....	22
2.3 AUTOMATIC RESOURCE COMPILATION BY ANALYZING THE HYPERLINK STRUCTURE AND ASSOCIATED TEXT.....	23
2.4 SELECTIVE HYPERTEXT INDUCED TOPIC SEARCH.....	24
2.5 IMPROVED ALGORITHM FOR TOPIC DISTILLATION IN A HYPERLINKED ENVIRONMENT.....	25
2.5.1 Mutually Reinforcing Relationships between Hosts.....	25
2.5.2 Automatically Generated Links.....	26
2.5.3 Non-relevant Nodes.....	26
3.0 PROBLEM DEFINITION	27
3.1 BROAD TOPIC QUERIES.....	27
3.1.1 Topic Drift.....	28
3.1.2 Topic Contamination.....	28
4.0 DESIGN	30
4.1 AUTOMATIC RESOURCE COMPILATION BY ANALYZING THE HYPERLINKED STRUCTURE AND ASSOCIATED TEXT.....	30
4.2 SELECTIVE HYPERTEXT INDUCED TOPIC SEARCH.....	30
4.3 SYSTEM ARCHITECTURE.....	31

5.0 IMPLEMENTATION	34
5.1 TECHNOLOGY.....	34
5.1.1 Matlab.....	34
5.1.2 MS-Access.....	35
5.2 IMPLEMENTATION.....	36
5.2.1 Root Set.....	36
5.2.2 Candidate pages.....	36
5.2.3 Base Set.....	37
5.2.4 Augment Set.....	37
5.2.5 Pruned Set.....	37
5.2.6 Hub and Authority.....	38
6.0 TESTING AND RESULTS.....	39
6.1 PURPOSE.....	39
6.2 SYSTEM DESIGN TETSING.....	40
6.3 TESTING DURING DEVELOPMENT.....	40
6.4 TESTING METHODS.....	41
6.4.1 Black box Testing.....	41
6.4.2 White box Testing.....	41
6.5 TESTING OF "IMPROVING ARC ALGORITHM FOR PURE TOPIC DISTILLATION".....	42
6.5.1 Gates.....	43
6.5.2 Mouse.....	42
6.6 ANALYSIS OF RESULTS.....	45
6.6.1 Analysis of query "Mouse" results.....	45
6.6.2 Analysis of query "Gates" results.....	45
7.0 CONCLUSION AND FUTER WORK.....	46
7.1 CONCLUSION.....	46
7.2 FUTURE WORKS.....	47
APPENDIX A - LIST OF ABBREVIATIONS	A-1
APPENDIX B - SCREEN SHOTS	B-1
References	

1

Introduction



INTRODUCTION

With the recent explosive growth of the amount of content on the Internet, it has become increasingly difficult for users to find and utilize information and for content providers to classify and catalog documents. Traditional web search engines often return hundreds or thousands of results for a search, which is time consuming for users to browse. On-line libraries, search engines, and other large document repositories (*e.g.* customer support databases, product specification databases, press release archives, news story archives, *etc.*) are growing so rapidly that it is difficult and costly to categorize every document manually. In order to deal with these problems, researchers look toward automated methods of working with web documents so that they can be more easily browsed, organized, and cataloged with minimal human intervention. In contrast to the highly structured tabular data upon which most machine learning methods are expected to operate, web and text documents are semi-structured. Web documents have well-defined structures such as letters, words, sentences, paragraphs, sections, punctuation marks, HTML tags, and so forth. We know that words make up sentences, sentences make up paragraphs, and so on, but many of the rules governing the order in which the various elements are allowed to appear are vague or ill-defined and can vary dramatically between documents. It is estimated that as much as 85% of all digital business information, most of it web-related, is stored in non-structured formats (*i.e.* non-tabular formats, such as those that are used in databases and spreadsheets). Developing improved methods of performing machine learning techniques on this vast amount of non-tabular, semi-structured web data is therefore highly desirable. Clustering and classification have been useful and active areas of machine learning research that promise to help us cope with the problem of information overload on the Internet. With clustering the goal is to separate a given group of data items (the data set) into groups called clusters such that items in the same cluster are similar to each other and dissimilar to the items in other clusters. In clustering methods no labeled examples are provided in advance for training (this is called unsupervised learning).

Under classification we attempt to assign a data item to a predefined category based on a model that is created from pre-classified training data (supervised learning). In more general terms, both clustering and classification come under the area of knowledge discovery in databases or data mining. Applying data mining techniques to web page content is referred to as web content mining which is a new sub-area of web mining, partially built upon the established field of information retrieval.

Searching information on the World Wide Web (WWW) is now a common practice but not really a simple one. User entered queries to search engines hoping that they will get their required pages, from billions of web servers on the World Wide Web. The size of web is increasing dauntly. Google currently indexes more than 8 billion pages and does not cover the complete Web. In short, the WWW is distributed, heterogeneous and of colossal size. The high rate of change and malicious spamming make the problem of searching on the WWW, even worse. Traditional information retrieval techniques do not perform well on the WWW. Search engines are considered as a solution to this problem. But still many issues are unsolved. Some times pages on the WWW are not honest about their contents. Artificial hyperlinked communities are created purposefully to get higher rank for pages. It is possible that the same information is mirrored at different URLs. We need new models and systems for searching on the WWW. One possible solution is to exploit all the features of the the WWW data like the word content, Document Object Model(DOM) tree, the page structure, the link structure, the URL of page etc. Word content of page can give hints about which topics are addressed in the page. A DOM tree can be used to differentiate between various parts of the page. In links to the page and outthinks from the page can give idea about the context of the page. One can assign different weights to different features. Considering these issues, various algorithms and systems have evolved over the past few years, resulting in an improved user experience. But still we are far from getting completely satisfying answer to our information needs. With the explosive growth of information sources available on the World Wide Web, it has become increasingly necessary for users to utilize automated tools to find the desired information resources, and to track and analyze their usage patterns. These factors give rise to the necessity of creating server side and client side intelligent systems that

can effectively mine for knowledge. Web mining can be broadly defined as the discovery and analysis of useful information from the World Wide Web. This describes the automatic search of information resources available online, i.e. Web content mining, and the discovery of user access patterns from Web servers, i.e., Web usage mining. A distinct feature of the Web is the proliferation of hyperlinks between web pages which allow a user to surf from one webpage to another with a simple click.

1.1 Web Mining

Web Mining is the extraction of interesting and potentially useful patterns and implicit information from artifacts or activity related to the World Wide Web. There are roughly three knowledge discovery domains that pertain to web mining: Web Content Mining, Web Structure Mining, and Web Usage Mining.

1.1.1 A Taxonomy of Web Mining

In this section we present a taxonomy of Web mining along its two primary dimensions, namely Web content mining and Web usage mining.. This taxonomy is depicted in Figure 1.1

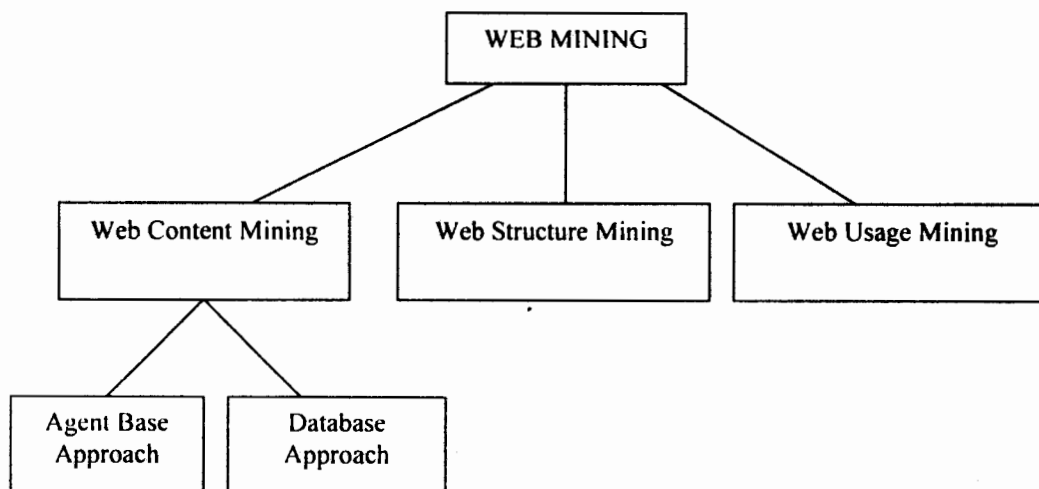


Figure 1.1: Taxonomy of Web Mining

1.1.2 Web Content Mining

In the Web mining domain, web content mining essentially is an analog of data mining techniques for relational databases, since it is possible to find similar types of knowledge from the unstructured data residing in Web documents. The Web document usually contains several types of data, such as text, image, audio, video, metadata and hyperlinks. Some of them are semi-structured such as HTML documents, or a more structured data like the data in the tables or database generated HTML pages, but most of the data is unstructured text data. The unstructured characteristic of Web data forces the Web content mining towards a more complicated approach.

The Web content mining is differentiated from two different points of view: Information Retrieval View and Database View. It shows that most of the researches use bag of words, which is based on the statistics about single words in isolation, to represent unstructured text and take single word found in the training corpus as features. For the semi-structured data, all the works utilize the HTML structures inside the documents and some utilized the hyperlink structure between the documents for document representation. As for the database view, in order to have the better information management and querying on the Web, the mining always tries to infer the structure of the Web site of to transform a Web site to become a database.

S. Chakrabarti [2] provide a in-depth survey of the research on the application of the techniques from machine learning, statistical pattern recognition, and data mining to analyzing hypertext. It's a good resource to be aware of the recent advances in content mining research.

Multimedia data mining is part of the content mining, which is engaged to mine the high-level information and knowledge from the large online multimedia sources. Multimedia data mining on the Web has gained many researchers' attention recently. Working towards a unifying framework for representation, problem solving, and learning from

multimedia is really a challenge, this research area is still in its infancy indeed, many works are waiting to be done.

1.1.3 Web Structure Mining

The goal of Web structure mining is to generate structural summary about the Web site and Web page. Technically, Web content mining mainly focuses on the structure of inner-document, while Web structure mining tries to discover the link structure of the hyperlinks at the inter-document level. Based on the topology of the hyperlinks, Web structure mining will categorize the Web pages and generate the information, such as the similarity and relationship between different Web sites.

Web structure mining can also have another direction -- discovering the structure of Web document itself. This type of structure mining can be used to reveal the structure (schema) of Web pages; this would be good for navigation purpose and make it possible to compare/integrate Web page schemes. This type of structure mining will facilitate introducing database techniques for accessing information in Web pages by providing a reference schema.

The structural information generated from the Web structure mining includes the follows: the information measuring the frequency of the local links in the Web tuples in a Web table; the information measuring the frequency of Web tuples in a Web table containing links that are interior and the links that are within the same document; the information measuring the frequency of Web tuples in a Web table that contains links that are global and the links that span different Web sites; the information measuring the frequency of identical Web tuples that appear in the Web table or among the Web tables.

In general, if a Web page is linked to another Web page directly, or the Web pages are neighbors, we would like to discover the relationships among those Web pages. The relations maybe fall in one of the types, such as they related by synonyms or ontology, they may have similar contents, and both of them may sit in the same Web server therefore created by the same person. Another task of Web structure mining is to discover the nature of the hierarchy or network of hyperlink in the Web sites of a particular

domain. This may help to generalize the flow of information in Web sites that may represent some particular domain; therefore the query processing will be easier and more efficient.

Web structure mining has a nature relation with the Web content mining, since it is very likely that the Web documents contain links, and they both use the real or primary data on the Web. It's quite often to combine these two mining tasks in an application.

1.1.4 Web Usage Mining

The only information left behind by many users visiting a Web site is the path through the pages they have accessed. Most of the Web information retrieval tools only use the textual information, while ignore the link information that could be very valuable.

Web usage mining, the application of data mining techniques to discover user navigation patterns from web data, tries to discover the useful information from the secondary data derived from the interactions of the users while surfing on the Web. It focuses on the techniques that could predict user behavior while the user interacts with Web.

In the process of data preparation of Web usage mining, the Web content and Web site topology will be used as the information sources, which interacts Web usage mining with the Web content mining and Web structure mining. Moreover, the clustering in the process of pattern discovery is a bridge to Web content and structure mining from usage mining.

There are lots of works have been done in the Information Retrieval, Database, Intelligent Agents and Topology, which provide a sound foundation for the Web content mining, Web structure mining. Web usage mining is a relative new research area, and gains more and more attentions in recent years.

1.2 Information Retrieval

Information retrieval (IR) is the science of searching for documents, for information within documents and for metadata about documents, as well as that of searching the World Wide Web. There is overlap in the usage of the terms data retrieval, document retrieval, information retrieval, and text retrieval, but each also has its own body of literature, theory, praxis and technologies. IR is interdisciplinary, based on computer science, mathematics, library science, information science, information architecture, cognitive psychology, linguistics, statistics and physics.

Automated information retrieval systems are used to reduce what has been called "information overload". Many universities and public libraries use IR systems to provide access to books, journals and other documents. Web search engines are the most visible IR applications.

Traditional Information Retrieval (IR) systems work well with controlled, finite collections. Documents in such collections are generally self contained units and they are truthful about their contents. Relevance of a document to the user query can be evaluated easily for such collections. Quality of results can be evaluated in terms of precision and recall. But for the WWW, first of all we cannot measure recall as one cannot have a complete snapshot of the WWW. Also, users are reluctant to go beyond the top few results. Thus the notion of recall holds little meaning in the context of Web IR. Similarly, precision also cannot not be considered as an important measure. Most of the users give short queries and there will be thousands of documents contain in that query. Further, many documents are not truthful about their contents.

The main idea of hyper documents is that documents or parts thereof can be brought into relation to each other and that additional information may be attached to any part of a document. Hypertext links were invented to support the manual browsing through large hypertext or hypermedia collections. However, retrieving specific portions of information in such a collection cannot be achieved by browsing only. There are about million of

pages on the Web today with about more than 1 million being added daily. The retrieval mechanisms are necessary.

In general, users either browse or use the search service when they want to find specific information on the Web. When user submits a query in a search engine to retrieve information, he gets thousands of documents as response to his query. However, few of the results returned by a search engine may be valuable to a user; even they do include the query keywords submitted by the user. It seems that we should find other useful information besides the match of word to improve the information retrieval performance.

1.3 Topic Distillation

Topic distillation can be defined as the process of finding quality documents related to a query topic. It is observed that generally users give very short and ambiguous queries to search engines. Most search engines return pages containing exact matches of the query terms, which may or may not be relevant to the user. Unlike search engines, the aim of topic distillation is not exactly to satisfy the user's precise information need. Rather it takes a broader approach and gives results for the topic of query, so that the user receives a spectrum of information.

Consider a query like Web mining. When the user is searching for web mining then it can be in several contexts such as web mining as a subject, books on web mining, conferences related to web mining, well known people in the field of web mining, important research groups on web mining, companies providing tools for web mining etc. Exact query match can hardly satisfy such broad interests. Rather, the user will need to search many times with query terms adjusted to find some particular aspect of data mining. e.g. "web mining conferences", "web mining tools". But with topic distillation a user can get information about all aspects of web mining with a single search. This is the advantage of topic distillation over simple searching.

Various topic directories like Yahoo!, Google help to get broad topic information on various topics arranged hierarchically. But their limitation is that, topics are hard coded and they involve considerable manual effort. Chakrabarti et al.[3] have conducted

experiments on automatically constructing topic directories. With topic distillation, topic queries can be answered for any broad topic and without any need of expertise or manual effort. The hyperlinks between different pages can provide very important information. The hyperlinks are latent indications of human judgment by the page authors. When page author creates a hyperlink, it is as if he is recommending the destination page or some purpose. So the destination page of the hyperlink gains some prestige. Algorithms like Page Rank rank pages, based on hyperlinks between all the pages in the WWW.

Kleinberg [6] proposed the Hypertext Induced Topic Search (HITS) algorithm for topic distillation on the WWW. It starts with a focused sub graph of the WWW for a query topic, using results from some existing search systems. Then it adds pages from neighborhood of this sub graph to create a larger sub graph. It then does computation to identify good hub pages and authority pages. A good hub page should contain a set of good links related to the query topic, whereas a good authority page should contain comprehensive and trust worthy information about the query topic. Hubs and authorities are found to have mutually enforcing relations. A good hub links to a number of good authorities and a good authority is one, which is pointed to by many good hubs. It may be noted that, the Page Rank of a page is topic independent while the hub and authority values are topic dependent.

1.4 Search Engine

It is a software program that searches a database and gathers and reports information that contains or is related to specified terms. A website whose primary function is providing a search engine for gathering and reporting information available on the Internet or a portion of the Internet.

1.5 Hyperlinks

A hyperlink (often referred to as simply a link), is a reference or navigation element in a document to another section of the same document, another document, or a specified section of another document, that automatically brings the referred information to the user when the navigation element is selected by the user. The two main uses of hyperlink analysis in Web information retrieval are crawling and ranking.

1.5.1 Significance of Hyperlinks

Though links have been heavily used, they have always been fundamental to the Web. There is a popular but flawed assumption about web that all its nodes are equally accessible. Yes, the web has no formalized structure or centralized organization to design web sites, and different web browsers obey and interpret the markup languages and scripting languages in different ways. But, it is a fact that certain web sites are more accessible than others.

Only in the last few years their value has become regulated as search engines and other systems that find and define the structures of the Web increasingly index links and anchor text in addition to keywords and paper content.

Today, Google, a successful and pioneer search engine, has completely taken over the market by using links as the primary method of determining the value and thereby the deserved visibility of a web site. Google interprets links to a web page as objective, peer-endorsed and machine-readable signs of value.

Google indexes links between web sites and interprets a link from A to B as an endorsed of B by A. Links may have different values. If A has a lot of links to it, and C has few links to it, then a link from A to B is worth more than a link from C to B. The value determined in this way is called a page's Page Rank and determines its placement in search results. (The Page Rank is used in addition to conventional text indexing to generate highly accurate search results.) Links can be analyzed more accurately and

usefully than traffic or page views, and have become both measure of success and dispensers of rank.

Links are increasingly being used in preference to content indexing, not only in search engines but for instance to identify communities of web sites, or, on a more local scale, to examine social networks and for the movement among web loggers. There is an assumption in this Page Rank that links provide an objective measure of value and are a sign of peer-endorsement.

1.7 Web as a Directed Graph

We can model the Web as a directed graph, the web pages as the nodes and the hyperlinks as the directed edges. This hyperlink graph contains useful information.

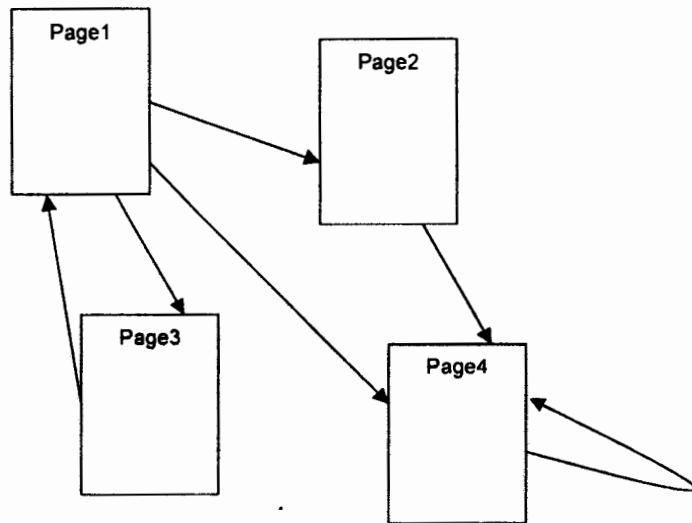


Fig1.2: Hyperlinked pages

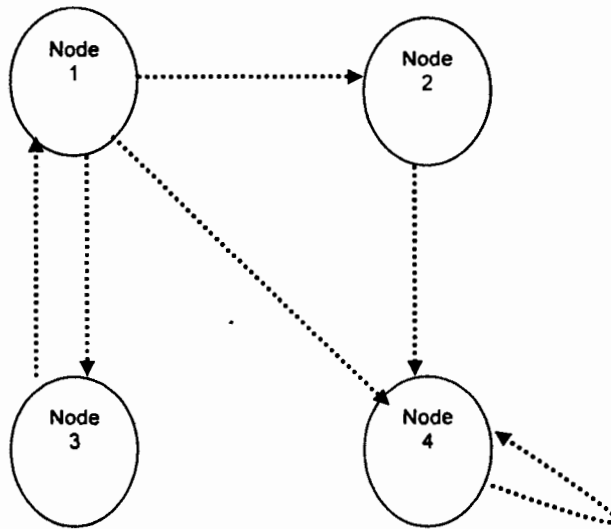


Fig 1.3: Directed graph

A hyperlinked Structure is shown in Figure 1.2 and is converted to Directed Graph in Figure 1.3. A valuable and informative webpage is usually pointed to by a large number of hyperlinks, i.e., it has a large 'in degree'. Such a webpage is called an authority. A webpage that points to many authority WebPages is itself a useful resource and is called a hub. A hub usually has a large 'out degree'. In the context of literature citation, a hub is a review paper which cites many original papers, while an authority is an original seminal paper cited by many papers.

1.8 Existing Link Analysis Algorithms

The three main algorithms for link analysis are

- 1) Page Rank
- 2) HITS
- 3) SALSA.
- 4) Trust Rank

Page Rank and HITS were proposed in 1998 where as SALSA was published in 2000. The feature common to all of them is that they are based on eigenvector computation.

1.8.1 Page Rank

Page Rank is a link analysis algorithm that assigns a numerical weighting to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of "measuring" its relative importance within the set. The algorithm may be applied to any collection of entities with reciprocal quotations and references. The numerical weight that it assigns to any given element E is also called the Page Rank of E and denoted by $PR(E)$. The name Page Rank is a trademark of Google.

Within the past few years, Google has become the far most utilized search engine worldwide. A decisive factor therefore was, besides high performance and ease of use,

the superior quality of search results compared to other search engines. This quality of search results is substantially based on Page Rank, a sophisticated method to rank web documents.

Page Rank is not simply based upon the total number of inbound links. The basic approach of Page Rank is that a document is in fact considered the more important the more other documents link to it, but those inbound links do not count equally. First of all, a document ranks high in terms of Page Rank, if other high ranking documents link to it. So, within the Page Rank concept, the rank of a document is given by the rank of those documents which link to it. Their rank again is given by the rank of documents which link to them.

But it's not just links that matter -- the text included in the links is important too. Google assumes that this clickable text, known as anchor text, describes what the targeted site is about. For example, if links to a bookstore's site include the words "pop-up books" in anchor text, the bookstore's site will rank higher in searches for those terms.

Page Rank is a numeric value that represents how important a page is on the web. Google figures that when one page links to another page, it is effectively casting a vote for the other page. The more votes that are cast for a page, the more important the page must be. Also, the importance of the page that is casting the vote determines how important the vote itself is. Google calculates a page's importance from the votes cast for it. How important each vote is taken into account when a page's PageRank is calculated. PageRank is Google's way of deciding a page's importance. It matters because it is one of the factors that determine a page's rank in the search results. It isn't the only factor that Google uses to rank pages, but it is an important one.

From here on in, we'll occasionally refer to PageRank as "PR". Not all links are counted by Google. For instance, they filter out links from known link farms. Some links can cause a site to be penalized by Google. They rightly figure that webmasters cannot control which sites link to their sites, but they can control which sites they link out to. For this reason, links into a site cannot harm the site, but links from a site can be harmful if

they link to penalized sites. So be careful which sites you link to. If a site has PR0, it is usually a penalty, and it would be unwise to link to it.

1.8.2 Hypertext Induced Topic Selection (HITS)

Hypertext Induced Topic Selection (HITS) is a link analysis algorithm that rates Web pages for their authority and hub values. Authority value estimates the value of the content of the page; hub value estimates the value of its links to other pages. These values can be used to rank Web search results. HITS was developed by Jon Kleinberg. Authority and hub values are defined in terms of one another in a mutual recursion. An authority value is computed as the sum of the scaled hub values that point to that page. A hub value is the sum of the scaled authority values of the pages it points to. Relevance of the linked pages is also considered in some implementations.

HITS, like Page and Brin's PageRank, is an iterative algorithm based on the linkage of the documents on the web. However it does have some major differences:

- It is executed at query time (not at indexing time) with the associated hit on performance that accompanies query-time processing.
- It is not commonly used by search engines. (Though some sources claim a similar algorithm is used by Ask.com.)
- It computes two scores per document (hub and authority) as opposed to a single score.
- It is processed on a small subset of 'relevant' documents, not all documents as was the case with PageRank.

1.8.3 SALSA

Stochastic Algorithm for Link Structure Analysis (SALSA) is a combination of PageRank and HITS. It calculates hub and authority values per query like HITS. However, they are calculated using Markov chains as in PageRank.

1.8.4 Trust Rank

Trust Rank is a link analysis technique. Many Web spam pages are created only with the intention of misleading search engines. These pages, chiefly created for commercial reasons, use various techniques to achieve higher-than-deserved rankings on the search engines' result pages. While human experts can easily identify spam, it is too expensive to manually evaluate a large number of pages.

One popular method for improving rankings is to increase artificially the perceived importance of a document through complex linking schemes. Trust Rank method calls for selecting a small set of seed pages to be evaluated by an expert. Once the reputable seed pages are manually identified, a crawl extending outward from the seed set seeks out similarly reliable and trustworthy pages. Trust Rank's reliability diminishes as documents become further removed from the seed set.

1.9 Evaluation of Web Search Results

Quality of Web search results is definitely a subjective matter. Importance and usefulness of pages will vary from person to person. But, can one objectively infer about the quality of a page? Based on link structure of WWW how can we define a good page? Will it be query dependent? Or can it be query independent? Current Web search systems respond to user queries within a fraction of a second. Users will not mind having a Web search system that responds within a few seconds, provided it returns considerably better results. But as stated by Kleinberg in [6]

"We are lacking objective functions that are both concretely defined and correspond to human notions of quality."

We describe below several parameters to objectively evaluate a page.

- **Popularity**

Popularity of a node can be equated with number in links it has. Here we assume that, if many nodes point to a node then it should be a popular node.

- **Centrality**

Distance from node u to v can be defined as minimum number of links via which we can reach v from u . Radius of a node is its maximum distance from any node in the graph. Center of the graph is the node with the smallest radius. The more central the node, the more easily we can reach other parts of the graph from it.

- **Informativeness**

A node is informative if it points to several nodes that contain useful information. Here we consider not just number of out links, but also quality of nodes pointed.

- **Authority**

Authority of a node is similar to the prestige of the node with the difference that authority is measured with respect to some focused tiny sub graph on a particular topic.

2

Literature Survey

LITERATURE SURVEY

2.1 Authoritative Sources in a Hyperlinked Environment

Kleinberg [6] developed a set of algorithmic tools for extracting information from the link structures of such environments, and report on experiments that demonstrate their effectiveness in a variety of contexts on the World Wide Web. Kleinberg [6] uses link structure for analyzing the collection of pages relevant to a broad search topic, and for discovering the most “authoritative” pages on such topics.

In this paper Kleinberg [6] covers the problem of *searching* on the www, which we could define roughly as the process of discovering pages that are relevant to a given query. It begins from the observation that improving the quality of search methods on the www, is at the present time, a rich and interesting problem that is in many ways orthogonal to concerns of algorithmic efficiency and storage.

In this paper Kleinberg [6] first analyze the hyperlinked structure of World Wide Web and described the way of drawing the focused sub-directed graph from link structure. The notations hub and authority are described. The way of getting root set and expanding it to base set is described. The hub and authority have mutually reinforcing relationship. A good hub points to good Authority and good Authority is pointed by good hub.

An iterative algorithm is defined for hub and authority calculations. These calculations are based on matrices.

The Four basic components for Kleinberg [6] approach is

- For broad topics on the www, the amount of relevant information is growing extremely rapidly, making it continually more difficult for individual users to filter the available resources. To deal with this problem, one needs notions beyond those of relevance and clustering—one needs a way to distill a broad topic, for which there may be millions of relevant pages, down to a representation of very small size. It is for this purpose that Kleinberg [6] define a notion of “authoritative” sources, based on the link structure of the WWW.

- This approach produces a result that is of as high a quality as possible in the context of what is available on the www globally. The underlying domain is not restricted to a focused set of pages, or those residing on a single Web site.
- This approach requires a basic interface to any of a number of standard www search engines, and use techniques for producing “enriched” samples of www pages to determine notions of structure and quality that make sense globally. This helps to deal with problems of scale in handling topics that have an enormous representation on the www.
- In this approach hub pages link densely to a set of thematically related authorities. This equilibrium between hubs and authorities is a phenomenon that recurs in the context of a wide variety of topics on the www.

The **HITS** (“hypertext induced topic selection”) algorithm is an algorithm for rating and ranking Web pages Kleinberg [6]. HITS uses two values for each page, the *authority value* and the *hub value*. Authority and hub values are defined in terms of one another in a mutual recursion. An authority value is computed as the sum of the scaled hub values that point to that page. A hub value is the sum of the scaled authority values. Kleinberg [6] proposed a more refined notion for the importance of web pages. He suggested that web page importance should depend on the search query being performed. Furthermore, each page should have a separate “authority” rating (based on the links going *to* the page) and “hub” rating (based on the links going *from* the page). Kleinberg [6] proposed to use text-based web search engine (such as AltaVista) to get a “Root Set” consisting of a short list of web pages relevant to a given query. Next, the root set is augmented by pages which link to pages in the Root Set, and also pages which are linked to pages in the Root Set, and to obtain a larger base set of web pages.

- **Root Set:**

For a given user query, we obtain a set of relevant documents using some existing search system e.g. Google, Yahoo! This set is called the root set. Procedure for getting a root set is shown in Figure 2.1.

- **Base Set:**

We expand root set by one link neighborhood to obtain the expanded root set or Base Set. Method for generating Base Set from Root Set is shown in Figure 2.1.

- **Hub Page:**

A page that contains links to authoritative pages for the entered query. Example of Hub page is shown in Figure 2.3 (a).

- **Authority page:**

Authority page is a page which contains information about the topic of the query or is directly relevant to the topic of query. Example of an Authority page is shown in Figure 2.3 (b).

- **Co-Reference and Co-Citation:**

The hub and authority matrices have interesting connection to two important concepts, co-citation and co-reference, which are fundamental metrics to characterize the similarity between two.

Co-citation is the number of in-links of a web page and Co-reference is a number of out-links of a web page. The authority matrix is the sum of Co-citation and in degree. The fact that two distinct WebPages co-reference many other WebPages indicates that these have certain commonality. Thus hub matrix is the sum of co-reference and out degree.

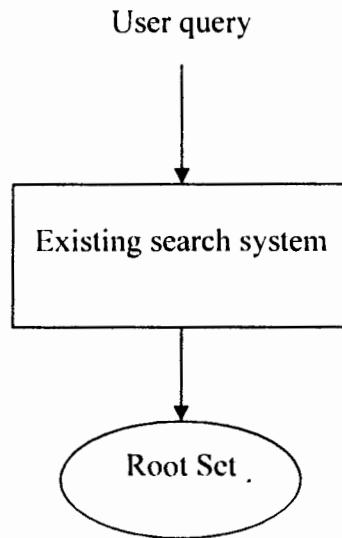


Fig2.1: How to get Root Set

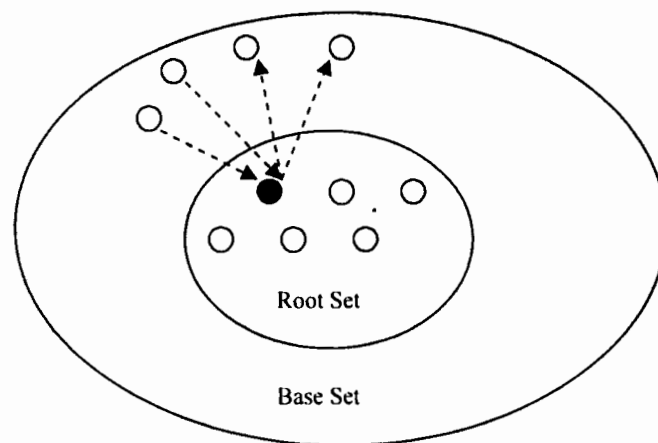


Fig2.2: Generating Base Set of Root Set

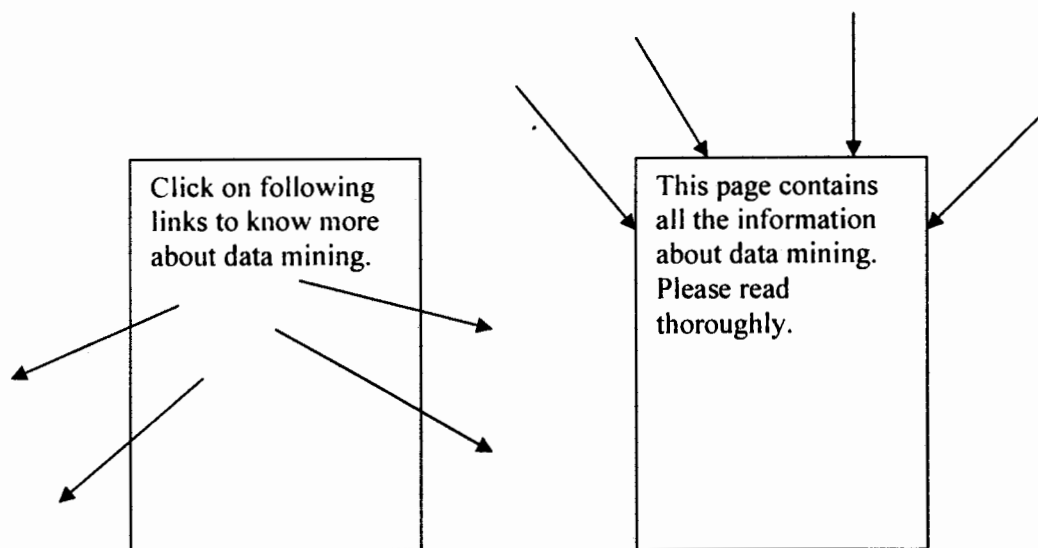


Fig 2.3(a) Hub Page

Fig 2.3(b) Authority Page

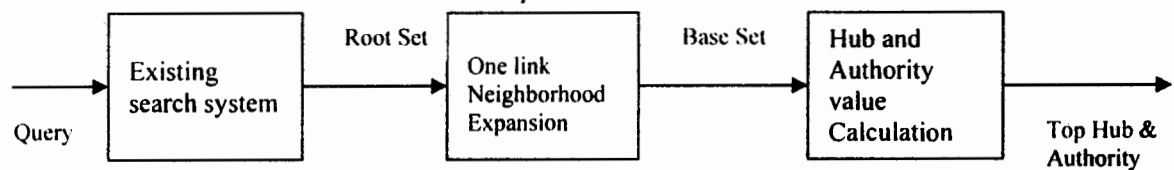


Figure2.4: HITS Algorithm

2.2 Mining the link structure of World Wide Web

Chakrabarti[4] developed an algorithms that exploit the hyperlink structure of the WWW for information discovery and categorization, the construction of high-quality resource lists, and the analysis of on-line hyperlinked communities. This paper addresses the problem of a topic of any breadth which typically contains several thousand or several million relevant Web pages; at the same time, a user will be willing to look at an extremely small number of these pages.

This paper discusses the HITS algorithm in detail along with the way of calculating Hub and Authority. Clever algorithm is an improvement over HITS. In this each Hub and Authority is assigned the non-negative weight. The weight is assigned by considering "href" which is also anchor text. Mini hub and pagelets is made for pages which has large Links.

2.3 Automatic Resource Compilation by analyzing the hyperlink structure and associated text

Chakrabarti [3] discusses the design and evaluation of an *Automatic Resource Compiler*. An automatic resource compiler is a system which, given a topic that is broad and well-represented on the web, will seek out and return a list of web resources that it considers the most authoritative for that topic. This system is built on an algorithm that performs a local analysis of both text and links to arrive at a "global consensus" of the best resources for the topic. This paper describes a user-study, comparing the resource compiler with commercial, human-compiled/assisted services. When web users seek definitive information on a broad topic, they frequently go to a hierarchical, manually-compiled taxonomy such as Yahoo!, or a human-assisted compilation such as Info seeks. The role of such taxonomy is to provide, for any broad topic, such a resource list with high-quality resources on the topic. The goal of ARC is to automatically compile a resource list on any topic that is broad and well-represented on the web. The ARC has three phases. Three phases of an ARC is shown in Figure 2.5.

1) Search and growth phase:

In this phase we get a set of 200 pages and then augment using links to 2-link neighborhood.

2) Weighting Phase:

In this phase we assign to each link (from page p to page q of the augmented set) positive numerical weight that tells how much it is related to the search topic.

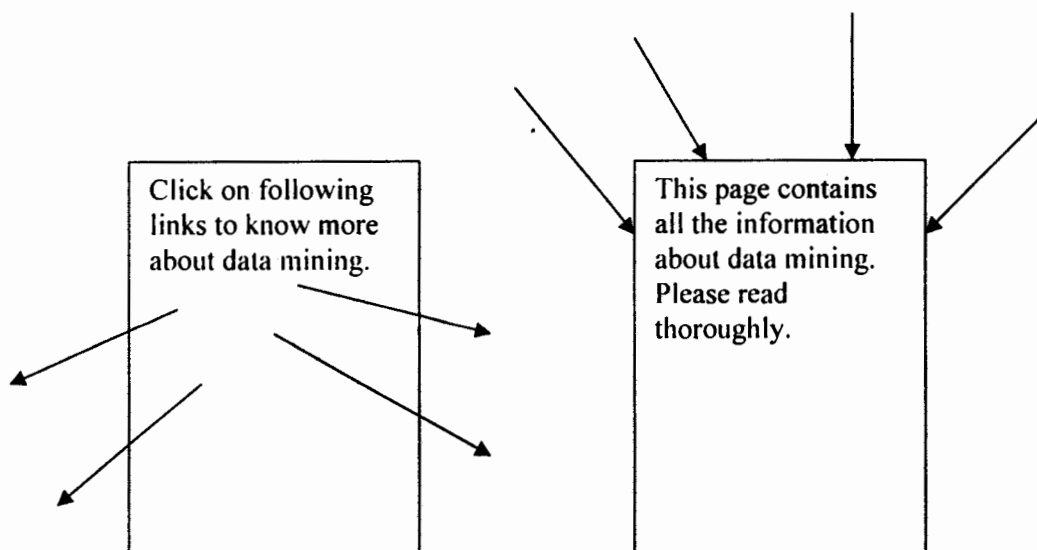


Fig 2.3(a) Hub Page

Fig 2.3(b) Authority Page

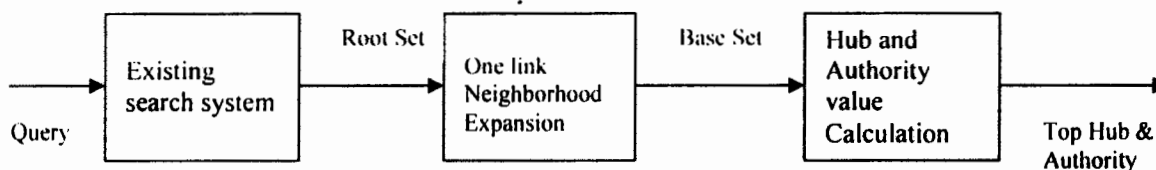


Figure2.4: HITS Algorithm

2.2 Mining the link structure of World Wide Web

Chakrabarti[4] developed an algorithms that exploit the hyperlink structure of the WWW for information discovery and categorization, the construction of high-quality resource lists, and the analysis of on-line hyperlinked communities. This paper addresses the problem of a topic of any breadth which typically contains several thousand or several million relevant Web pages; at the same time, a user will be willing to look at an extremely small number of these pages.

3) Iteration and Reporting Phase:

This phase is to compute vectors \mathbf{h} (for hub) and \mathbf{a} (for authority), with one entry for each page in the augmented set. The entries of the first vector contain scores for the value of each page as a hub, and the second vector describes the value of each page as an authority. Then construct a matrix W that contains an entry corresponding to each ordered pair p, q of pages in the augmented set. This entry is $w(p, q)$ (compute as below) when page p points to q , and 0 otherwise. Let Z be the matrix transpose of W . Then set the vector \mathbf{h} equal to 1 initially and iteratively execute the following two steps k times.

$$\mathbf{a} = \mathbf{W} \mathbf{h}$$

$$\mathbf{h} = \mathbf{Z} \mathbf{a}$$

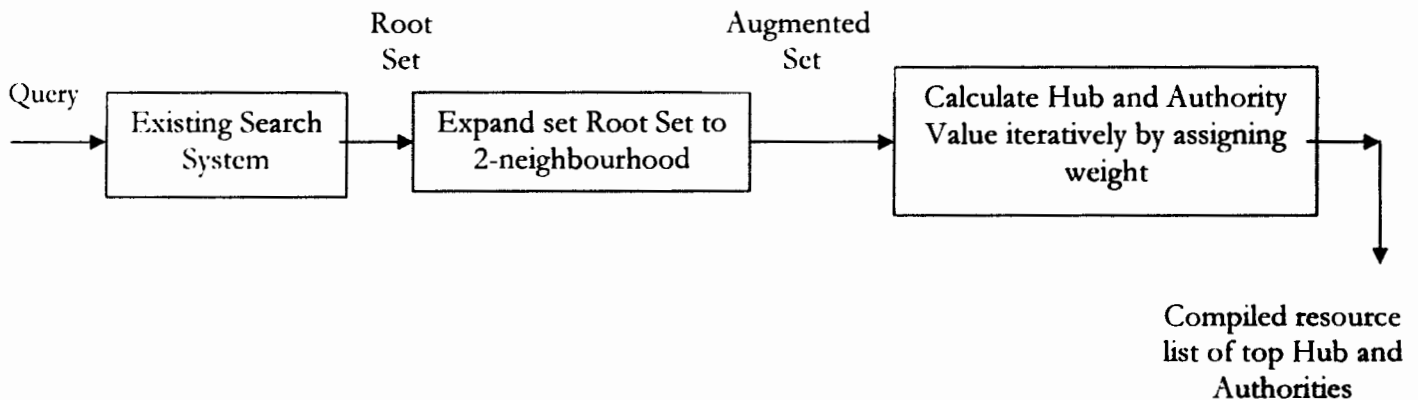


Fig2.5: ARC

2.4 Selective Hypertext Induced Topic Search

Awekar [7] discusses the problem of answering broad-topic queries on the World Wide Web. This paper also presents a link based analysis algorithm SelHITS as shown in Figure 2.6 which is an improvement over HITS algorithm. A novel approach is proposed to calculate hub and authority values on root set. This will return the top hub and authority and then the root set is selectively expanded by considering these scores. This Paper also present a selective expansion method which avoids topic drifts and provides

results consistent with only one interpretation of the query, even if the query is ambiguous. Initial experimental evaluation and user feedback show that SelHITS algorithm indeed distills the most important and relevant pages for broad-topic queries.

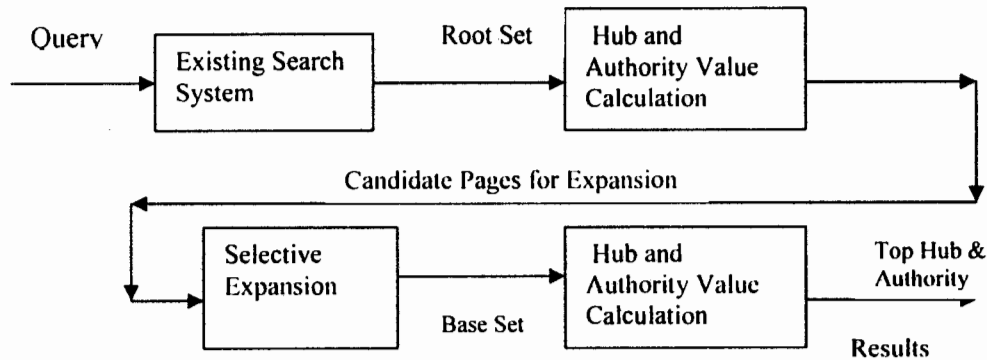


Figure 2.6: SelHITS

2.5 Improved Algorithms for Topic Distillation in a Hyperlinked Environment

Bharat et al [1] described an approach to augment a connectivity analysis based algorithm with content analysis to find quality documents related to the query topic and described the three problems in Hypertext Induced Topic Search (HITS) [6] connectivity analysis algorithm and presented various algorithms to address those problems. Problems in Hypertext Induced Topic Search (HITS) connectivity analysis algorithm are as follows [6].

2.5.1 Mutually Reinforcing Relationships between Hosts.

A set of documents on one host point to a single document on a second host. This drives up the hub scores of the documents on the first host and the authority score of the document on the second host. The reverse case, where there is one document on a first host pointing to multiple documents on a second host, creates the same problem. Since

the set of documents on each host are authored by a single author or organization, these situations give undue weight to the opinion of one person.

2.5.2 Automatically Generated Links:

Web documents generated by tools (e.g., Web authoring tools, database conversion tools) often have links that were inserted by the tool. For example, the Hyper news system, which turns USENET News articles into Web pages, automatically inserts a link to the Hyper news Web site. In such cases human's opinion is represented by the link, does not apply.

2.5.3 Non-relevant Nodes:

The neighborhood graph contains documents not relevant to the query topic. If these nodes are well connected, the topic drift problem arises: the most highly ranked authorities and hubs tend not to be about the original topic. For example, when running the algorithm on the query "jaguar and car" the computation drifted to the general topic "car" and returned the home pages of different car manufacturers as top authorities, and lists of car Non-relevant Nodes manufacturers as the best hubs.

This paper presented one connectivity based algorithms to address the first problem by giving fractional weights to each edge, basically the algorithms are an improvement over Hypertext Induced Topic Search (HITS) algorithm by Kleinberg *et al* (1997)[6]. These algorithms are the combination of content analysis using traditional Information Retrieval techniques.

3

Problem Definition



PROBLEM STATEMENT

Web search can simply be considered as the process of the user entering a query and the search system returning a set of relevant URLs. But not all user queries are same. Kleinberg [6] define user queries as Broad-topic Queries

3.1 Broad Topic Queries

A large number of irrelevant pages are reported when user entered broad topic queries. They do not have exact match. Here user is not just looking for a narrow answer to the query but is also looking for information related to the broad topic. So some URLs will be more relevant while some will be less. Examples of such query can be "Data Mining". Here a user could be looking for broad information about data mining. It can include some introductory articles on data mining by some expert or can be the home page of some well known researcher working in the field of data mining or a page listing important links about data mining. Even a page containing the words "Data Mining" is also relevant to this query. But there will be thousands of such pages. So search system has to choose best pages from thousands of candidate pages. HITS and related algorithms concentrate on latter type, i.e. broad topic queries. As we have large Root set when we apply hits we get many Bi-partite graphs which are related to different aspects of query related information. So our aim is to find the most authoritative and informative pages for the topic of the query based on broad topic queries. Search engines are considered as a solution to this problem. But still many issues are unsolved. Some times pages on the WWW are not honest about their contents. Artificial hyperlinked communities are created purposefully to get higher rank for pages. It is possible that the same information is mirrored at different URLs. We need new models and systems for searching on the WWW.

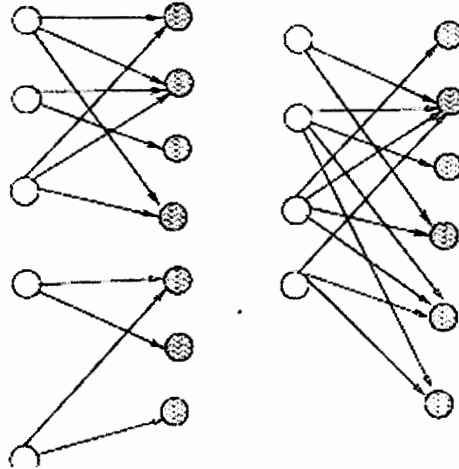


Fig 3.1: Multiple Bipartite Graphs Corresponding to Multiple Topics of a Single Query

3.1.1 Topic Drift

As ARC and other link analysis pages such as HITS add all the pages in one link neighborhood of the root set, many irrelevant nodes are added. It is possible that many such nodes are added and they emerge as a connected community, thus causing their hub and authority values to be hiked. This gives importance to most irrelevant page.

3.1.2 Topic Contamination

Consider queries like "windows", "gates". These queries are ambiguous and have multiple meanings. Windows can be interpreted as windows of a house or the Windows operating system. Gates can be interpreted as Bill Gates or an electronic gate or gate of a house. So depending on interpretation there will be different topics for the query. Simply running ARC and HITS returns results from multiple topics, thus causing topic contamination. The aim of topic distillation process is to deliver results for a single topic only. It is found that generally pages on a single topic form a linked community. It is in the form of a bipartite graph with hubs linking to authorities. So for different interpretations of an ambiguous query we will have several disconnected bipartite cores in the root set. There are hardly any links between pages of different topics. e.g. Pages about Windows operating system will not contain links for windows that we use in our home and vice versa.

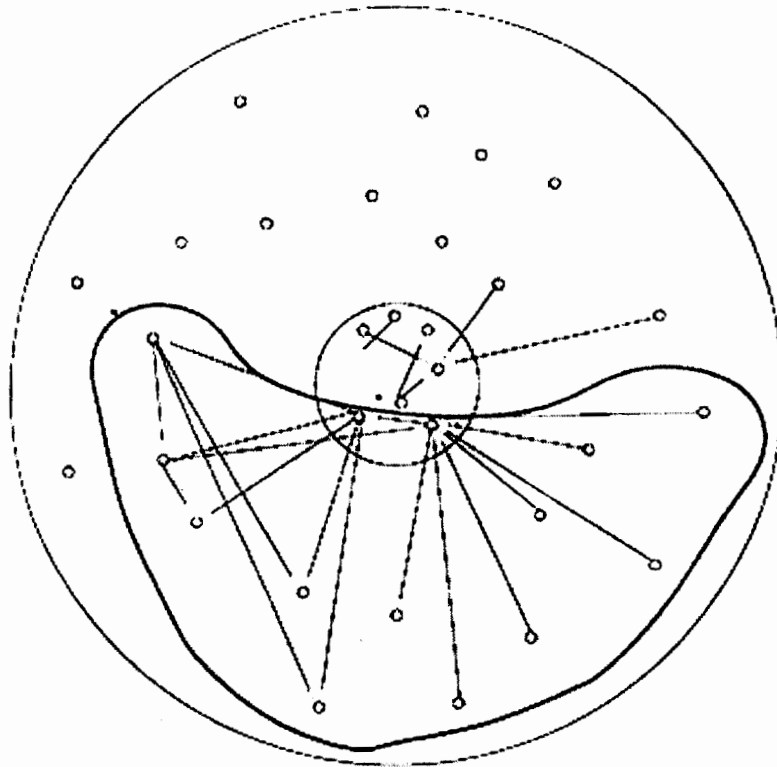


Fig3.2: Pages Causing Topic Drift and Topic Contamination

7H-5093

4

Design



DESIGN

4.1 Automatic Resource Compilation

An automatic resource compiler is a system which, given a topic that is broad and well-represented on the web, will seek out and return a list of web resources that it considers the most authoritative for that topic. The goal of ARC is to automatically compile a resource list on any topic that is broad and well-represented on the web. The ARC has three phases. Three phases of an ARC is shown in Figure 4.1.

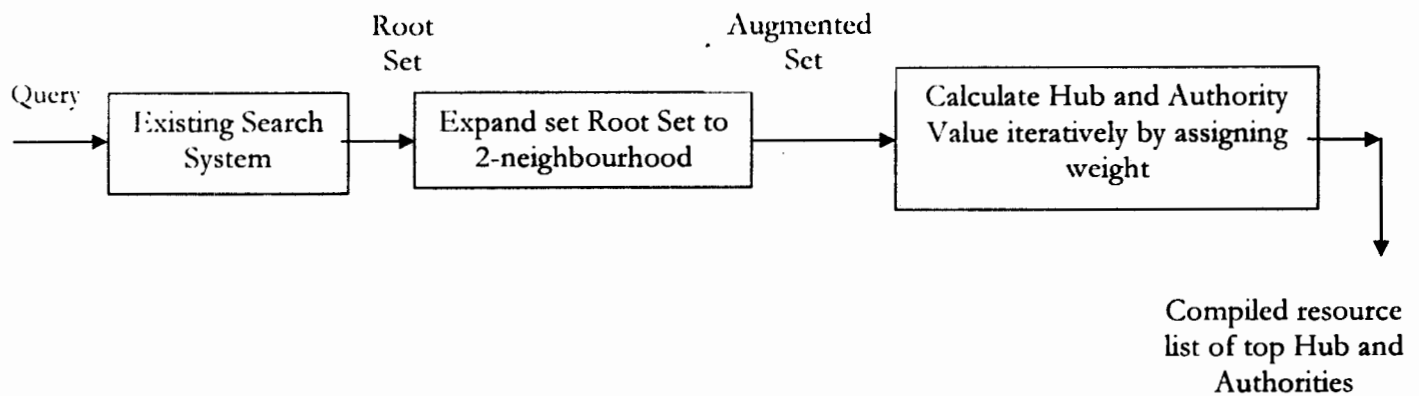


Fig 4.1: ARC

4.2 SelHITS Algorithm

SelHITS algorithm by Mitra *et al*[7] begins with the user query. Then it gets a small root set from some existing search system against user query. The root set is of order of few hundred pages related to query topic. Then it calculates hub and authority values on the root set and select top hubs and top authorities pages as candidate Pages for further expansion. Refer to Figure 4.2. This selective expansion procedure of candidate pages drastically reduces size of the base set, as irrelevant pages are not added to the Candidate Pages to get the base set that avoids time consumption, topic contamination problems. For base set SelHITS repeat the same process that it carried out on the root set. Then it reports top hub and authority pages to the user. Refer to Figure 4.2.

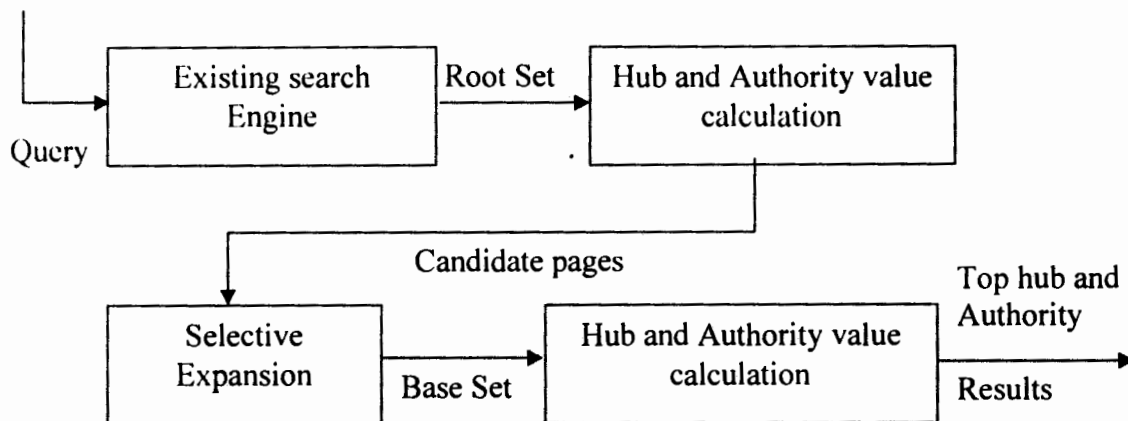


Figure4.2: Architecture of SelHITS Algorithm

4.3 System Architecture

In the implemented system we have applied SelHITS on ARC algorithm to improve the ARC algorithm by eliminating the problem of topic contamination in existing algorithm. Previously the whole root is expanded to base set by considering all the documents and their in-links and out-links. The Blind expansion takes place and because of this large number of irrelevant nodes is added and provides the different interpretation of the same query. This does not fulfill the user requirements and does not provide quality documents. This leads to the problem of Topic Drift and Topic Contamination. By selective expansion of root set only one interpretation of query is obtained. This reduces the problem of Topic Drift and Topic Contamination. The algorithm given by Chakrabarti [3] as discussed in Literature Survey has three phases. The three phases of ARC algorithm, after we have applied SelHITS on root set are discussed as follow:

- **Search and Growth Phase:**

In this phase, given a topic, we have first gathered a collection of pages i.e. similar to Kleinberg's [6] HITS technique. The pages are gathered by entering query into search Engine. We get a root set of 200 pages.

We have first applied the SelHITS technique on the root set by obtaining the top hub and top authorities from root set and the obtained top hub and authorities are the candidate pages from root set which are expanded to one-neighbor hood to make base set. By expanding the obtained candidate pages into base set, we have distilled pure topic related pages as we have expanded only one community in the root set.

The hub and authority scores have been calculated by making mapping table of the Hyperlinks calling each other. Then the obtained base set is expanded to augmented set by considering the out links of pages in base set. The root set is expanded for top hub and authority twice as we considered two neighbor hoods in this algorithm.

- **Weighting Phase:**

In this phase we have assigned a weight to each hub and authority score of a page. We will assign to each link (from page p to page q of the augmented set) a positive numeric weight $w(p,q)$ that increases with the amount of topic related text in the vicinity of the href from p to q .

- **Iteration and Reporting Phase:**

In this phase we have computed two values h (for hub) and a (for authority) with one entry for each page in the augmented set. We will construct a matrix W that contains an entry corresponding to each ordered pair p,q of page in the augmented set. The entry is $W(p,q)$ when page p points to q and 0 otherwise. Let Z be the matrix transpose of W .

We have the vector h equal to 1 initially and execute the following two steps k times.

$$A=Wh$$

$$H=Za$$

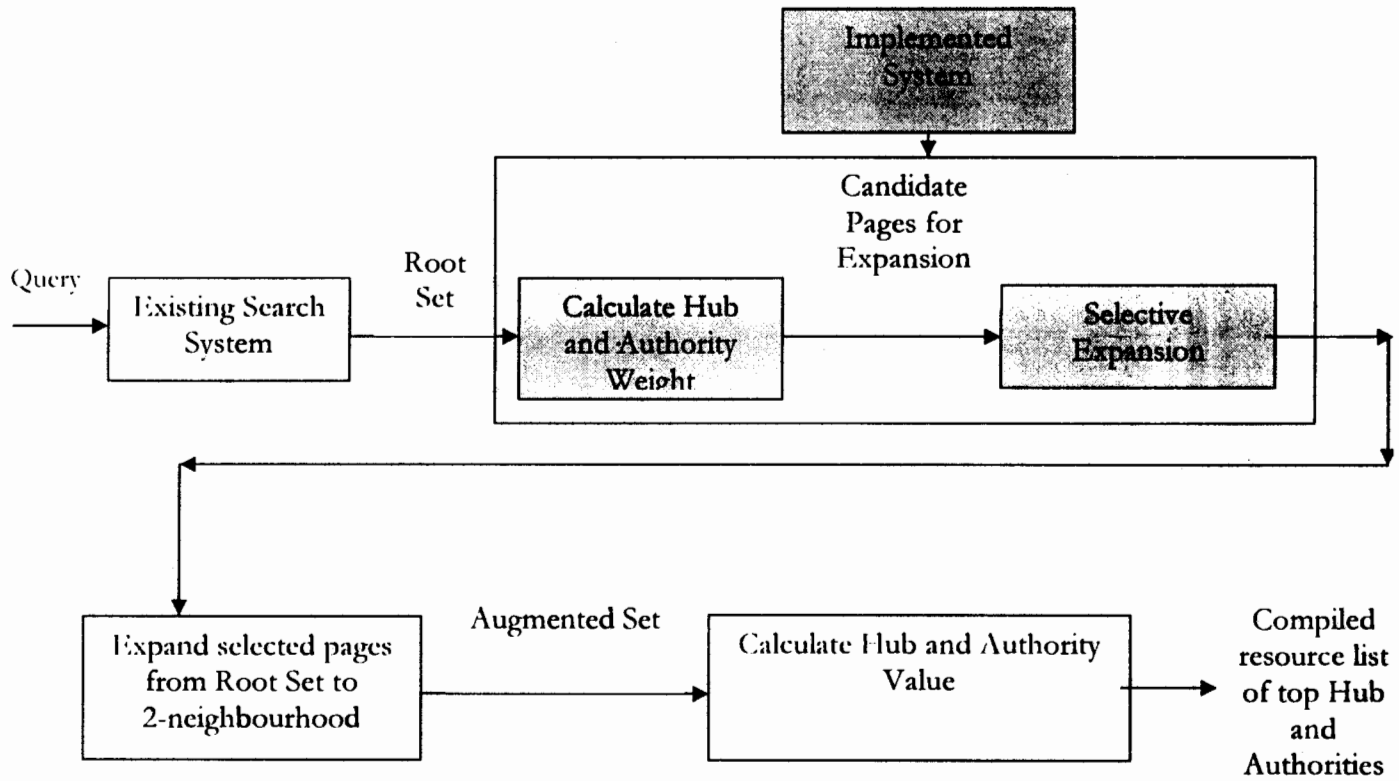


Figure 4.3: SelHITS on ARC

5

Implementation



IMPLEMENTATION

Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy. In computer science, an implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system. Many implementations may exist for a given specification or standard. For example, web browsers contain implementations of World Wide Web Consortium-recommended specifications, and software development tools contain implementations of programming languages.

5.1 Technology

The technology includes computers, operating systems, networks, telecommunication links, storage technologies and the architecture. It provides the base for the data and application architectures. The infrastructure encompasses the hardware and software that are used to support the application and data.

- Our implementation requires Window XP and Internet.
- The tool used are Matlab 7 and MS-Access
- This application can run on Pentium -IV with at least 2.8GHz speed and high internet speed.

5.1.1 MATLAB

MATLAB is computing and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. Using the MATLAB product, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and FORTRAN.

You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. Add-on toolboxes (collections of special-purpose MATLAB functions, available separately) extend the MATLAB environment to

solve particular classes of problems in these application areas. MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications, and distribute your MATLAB algorithms and applications.

Key Features

- High-level language for technical computing
- Development environment for managing code, files, and data
- Interactive tools for iterative exploration, design, and problem solving
- Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration
- 2-D and 3-D graphics functions for visualizing data
- Tools for building custom graphical user interfaces
- Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, Fortran, Java, COM, and Microsoft Excel

5.1.2 MS-Access

Microsoft Access is a computer application used to create and manage computer-based databases on desktop computers and/or on connected computers (a network). Microsoft Access can be used for personal information management (PIM), in a small business to organize and manage data, or in an enterprise to communicate with servers. We have used MS-Access for storing data in data base and data base connectivity is working with Matlab.

5.2 Implementation

Following are the main phases of the system:

- Root Set.
- Candidate Set
- Base Set.
- Augmented Set.
- Pruned Set
- Hub and Authority

Following are the important function designed for system implementation

5.2.1 Root set

The root set is obtained by entering query in any existing search engine. The root set is obtained by taking the set of 200 pages related to entered query.

- **Webbot ()**.

This function is designed to extract URL's from the obtained pages of root set.

5.2.2 Candidate Set

The candidate set contains the top hub and top authority from root set using the following three functions

- **Match()**

This function map the URL's which are calling each others.

- **Authority()**

This function returns the top authority.

- **Hub()**

This function returns the top hubs from match table.

- **Cond()**

This function inserts the obtained top hub and top authority into candidate set.

5.2.3 Base Set

Base set is obtained by taking in links and out links of all the web pages in candidate set.

Following are the functions that are used to complete this task.

- **root_out()**

This function takes all out links of web pages in candidate set.

- **root_in()**

This function takes in links of pages in candidate set

- **Base()**

This function inserts all the obtained in links and out links into base set along with Candidate set pages links.

5.2.4 Augmented Set

Augmented set is found by including all the out links upto 2-neighbour hood .Following is the functions designed to obtain Augment set.

- **Base_out()**

This function takes all the hyperlinks from the pages in base set.

- **Augment()**

This function inserts all the URL's from base set and base_out in the augment set.

5.2.5 Pruned Set

This function takes the URL's from augmented set which contains the string which is entered by user. The text mining is done for getting the links which contain the key word entered by user for searching.

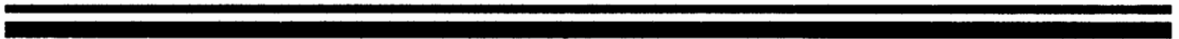
5.2.6 Hub and Authority

Finally the hub and authority values are calculated by using the following functions

- **Augment_match()**
This function maps the hyperlinks in table which are calling each others.
- **Augment_auth()**
This function returns the authority value of each page in augment set and reports the top authority pages to user.
- **Augment_hub()**
This function returns the hub value of each page in augment set and reports the top hub pages to user.

6

Testing and Results



TESTING AND RESULTS

Software testing is the process of checking software, to verify that it satisfies its requirements and to detect errors.

Software testing is an empirical investigation conducted to provide stakeholders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding software bugs.

Testing can never completely establish the correctness of computer software. Instead, it furnishes a criticism or comparison that compares the state and behavior of the product against a specification. Software testing should be distinguished from the separate discipline of Software Quality Assurance, which encompasses all business process areas, not just testing.

6.1 Purpose

A primary purpose for testing is to detect software failures so that defects may be uncovered and corrected. This is a non-trivial pursuit. Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions. The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions as well as examining the quality aspects of code: does it do what it is supposed to do and do what it needs to do. In the current culture of software development, a testing organization may be separate from the development team. There are various roles for testing team members. Information derived from software testing may be used to correct the process by which software is developed.

6.2 System design Testing

It is a key issue to test system design. Design testing is necessary to implement the system. Following issue are reviewed during design testing

- Is the entire requirement for designing the system are fulfilled?
- Design is comprehensive and is fulfilling system requirements.
- Implementation of design is possible or not.
- Method for error handling in a system

The design was reviewed several times and up to extent it is tried to make and implement an error free design.

6.3 Testing during Development

The testing during development includes checking whether the tool used to implement the system is appropriate for this system. The errors and bugs in codes are checked and omitted. All the syntax and semantic errors are checked in this phase. The system is checked whether it is fulfilling the criteria and is giving the required results.

The system is tested by giving different inputs and outputs are checked accordingly.

6.4 Testing methods

Software testing methods are traditionally divided into black box testing and white box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

6.4.1 Black box testing

Black box testing treats the software as a black-box without any understanding of internal behavior. It aims to test the functionality according to the requirements. Thus, the tester inputs data and only sees the output from the test object. This level of testing usually requires thorough test cases to be provided to the tester who then can simply verify that for a given input, the output value (or behavior), is the same as the expected value specified in the test case. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix etc.

6.4.2 White box testing

White box testing, however, is when the tester has access to the internal data structures, code, and algorithms.

Types of white box testing

The following types of white box testing exist:

- **code coverage** - creating tests to satisfy some criteria of code coverage. For example, the test designer can create tests to cause all statements in the program to be executed at least once.
- **mutation testing** methods.

6.5 Testing of “*Improving ARC algorithm for Pure Topic Distillation*”

We have tested our system “Improving ARC algorithm for Pure Topic Distillation” by entering different queries. The system is properly tested by passing it through all the phases of software testing process. The system is tested by the test methods of black box and white box testing.

Following two queries are performed for testing “Improving ARC algorithm for Pure Topic Distillation”

- 1) Gates
- 2) Mouse

6.5.1 Gates

Table 6.1: Results of Query ‘Gates’

HUB	AUTHORITY
url	url
http://www.gates.com/	http://www.gatesfoundation.org/
http://www.gates.com/europe/	http://www.gatesbbq.com/
http://en.wikipedia.org/wiki/Bill_Gates	http://www.gates.scholarships.cam.ac.uk/
http://www.gatesfoundation.org/	http://www.gatestraining.com/
http://www.gatesbbq.com/	http://www.klgates.com/
http://www.gates.scholarships.cam.ac.uk/	http://www.gatesconcreteforms.com/
http://www.gatestraining.com/	http://www.garethgates.com/
http://www.gatesinc.com/	http://www.gatesfoundation.org/ForGrantSeekers/
http://www.klgates.com/	http://www.gateslibrary.org/
http://www.gatesconcreteforms.com/	http://www.linkedin.com/in/billgates
http://www.garethgates.com/	http://www.babygates.com/
http://gatesofvienna.blogspot.com/	http://www.gatesmillsvillage.com/
http://www.gatesfoundation.org/ForGrantSeekers/	http://www.muqshots.org/misc/bill-gates.html
http://news.cnet.com/Gates-big-send-off/2009-1014_3-6242276.htm	http://www.gatescounty.qovoffice2.com/
http://www.nytimes.com/ref/arts/design/GATES-REF.html	http://www.ihsph.edu/gatesinstitute
http://macboy.com/cartoons/switch/gates/	http://en.wikiquote.org/wiki/Bill_Gates
http://www.gateslibrary.org/	http://www.bitstorm.org/gates/
http://www.usnews.com/usnews/tech/billgate/gates.htm	http://www.physics.umd.edu/people/faculty/gates.html
http://www.wired.com/techbiz/people/news/2008/06/gates_bio	http://www.woodengates.com/
http://www.amazinggates.com/	http://www.gates.of.the.arctic.national-park.com/
http://www.gatesfamilyfoundation.org/	http://www.gateschili.org/
http://www.linkedin.com/in/billgates	http://www.7gates.org/
http://prelectur.stanford.edu/lecturers/gates/	http://www.fashiongates.com/
	http://www.nypost.com/seven/06122008/postopinion/oped
	http://www.thesmokingqun.com/muqshots/gatesmuq1.htm
	http://www.natacfd.org/

The table 6.1 show the URL’s of top hub and authority pages for query “Gates”. As seen from the results that all the obtained top hub and authority web pages contains rich information related to Bill Gates –The Founder of Microsoft Corporation and the obtained URL’s are from one community of query.

6.5.2 Mouse

Table 6.2: Results of query 'Mouse'

HUB	AUTHORITY
<p>url</p> <p>http://en.wiktionary.org/wiki/Special:Search/mouse</p> <p>http://hu.wiktionary.org/wiki/mouse</p> <p>http://nl.wiktionary.org/wiki/mouse</p> <p>http://de.wiktionary.org/wiki/mouse</p> <p>http://oc.wiktionary.org/wiki/mouse</p> <p>http://ta.wiktionary.org/wiki/mouse</p> <p>http://sr.wiktionary.org/wiki/mouse</p> <p>http://lo.wiktionary.org/wiki/mouse</p> <p>http://fa.wiktionary.org/wiki/mouse</p> <p>http://es.wiktionary.org/wiki/mouse</p> <p>http://el.wiktionary.org/wiki/mouse</p> <p>http://cs.wiktionary.org/wiki/mouse</p> <p>http://pt.wiktionary.org/wiki/mouse</p> <p>http://fr.wiktionary.org/wiki/mouse</p> <p>http://lt.wiktionary.org/wiki/mouse</p> <p>http://it.wiktionary.org/wiki/mouse</p> <p>http://zh-min-nan.wiktionary.org/wiki/mouse</p> <p>http://sv.wiktionary.org/wiki/mouse</p> <p>http://uk.wiktionary.org/wiki/mouse</p> <p>http://vi.wiktionary.org/wiki/mouse</p> <p>http://kk.wiktionary.org/wiki/mouse</p> <p>http://la.wiktionary.org/wiki/mouse</p> <p>http://id.wiktionary.org/wiki/mouse</p> <p>http://te.wiktionary.org/wiki/mouse</p> <p>http://fi.wiktionary.org/wiki/mouse</p> <p>http://tr.wiktionary.org/wiki/mouse</p> <p>http://eu.wiktionary.org/wiki/mouse</p> <p>http://ja.wiktionary.org/wiki/mouse</p> <p>http://hy.wiktionary.org/wiki/mouse</p> <p>http://zh.wiktionary.org/wiki/mouse</p> <p>http://ko.wiktionary.org/wiki/mouse</p>	<p>url</p> <p>http://kk.wiktionary.org/wiki/mouse</p> <p>http://uk.wiktionary.org/wiki/mouse</p> <p>http://tr.wiktionary.org/wiki/mouse</p> <p>http://te.wiktionary.org/wiki/mouse</p> <p>http://ta.wiktionary.org/wiki/mouse</p> <p>http://so.wiktionary.org/wiki/mouse</p> <p>http://pt.wiktionary.org/wiki/mouse</p> <p>http://zh-min-nan.wiktionary.org/wiki/mouse</p> <p>http://vi.wiktionary.org/wiki/mouse</p> <p>http://ko.wiktionary.org/wiki/mouse</p> <p>http://sv.wiktionary.org/wiki/mouse</p> <p>http://ja.wiktionary.org/wiki/mouse</p> <p>http://it.wiktionary.org/wiki/mouse</p> <p>http://io.wiktionary.org/wiki/mouse</p> <p>http://hy.wiktionary.org/wiki/mouse</p> <p>http://el.wiktionary.org/wiki/mouse</p> <p>http://es.wiktionary.org/wiki/mouse</p> <p>http://fr.wiktionary.org/wiki/mouse</p> <p>http://fa.wiktionary.org/wiki/mouse</p> <p>http://de.wiktionary.org/wiki/mouse</p> <p>http://zh.wiktionary.org/wiki/mouse</p> <p>http://bg.wiktionary.org/wiki/mouse</p> <p>http://fi.wiktionary.org/wiki/mouse</p> <p>http://lt.wiktionary.org/wiki/mouse</p> <p>http://ru.wiktionary.org/wiki/mouse</p> <p>http://oc.wiktionary.org/wiki/mouse</p> <p>http://cs.wiktionary.org/wiki/mouse</p> <p>http://hu.wiktionary.org/wiki/mouse</p> <p>http://eu.wiktionary.org/wiki/mouse</p> <p>http://pl.wiktionary.org/wiki/mouse</p> <p>http://nl.wiktionary.org/wiki/mouse</p>

The table 6.2 shows URL's of the top hub and authority pages for query "Mouse". As seen from the results that all the obtained top hub and authority pages contains rich information related to animal mouse and the obtained URL's are from single interpretation of query "Mouse" i.e. animal mouse.

6.6 Analysis of Results

The results obtained are shown in Table 6.1 and 6.2. From the obtained results, it is clearly shown that the obtained top hub and top authority are from one community of broad topic query entered by user.

6.6.1 Analysis of query “Mouse” results

The first query that we select to test our system is Mouse. The two communities' for this broad topic query are: animal mouse or computer mouse.

The implemented system has shown the remarkable improvement by giving results related to one community that is animal mouse and removes the problem of topic drift and topic contamination to some extent. The user is getting pages which are highly informative related to single aspect of fired query. The achieved results have reduced the time consumption because we have selectively expanded the root set.

By running the same query using ARC algorithm one obtained results related to computer mouse and animal mouse. This shows that user is getting topic contaminated result. Mostly user needs information related to single interpretation of a query. The results obtained mostly contain irrelevant pages which are of no use to user. Also when we expand the root set fully, the size of base set will increase drastically and it is time consuming for a user to search for required web page.

6.6.2 Analysis of query “Gates” results

Similar is the case, when we perform “Gates” query. The three different interpretations for gates are Bill gates, logic gates and home gates. The query gates can be interpreted in various ways. The obtained results are consistent with one interpretation of a query.

As can be seen, all our results are related to Bill Gates, one of the founders of Microsoft. The results also include a link for his philanthropic organization Gates Foundation.

7

Conclusion and Future Works

CONCLUSION AND FUTURE WORKS

7.1 Conclusion

From the obtained results and research done it is concluded that we have improved the ARC Algorithm by selectively expanding the Root Set. The implemented system is successful in giving the desired results to some extent by overcoming the problem of topic contamination and topic drift.

Previously the ARC algorithm blindly expands the root set and this procedure drastically increases size of the base set. Most of the pages added are useless and including them in the base set causes topic drift and it is time consuming to search the relevant page from them. By the selective expansion of root set in ARC algorithm we have distilled pure topic related pages and pages related to single community of entered query to some extent as shown from our results and the obtained top hub and authority pages are related to single interpretation of a query entered by user. Selective expansion has also reduced searching time and large amount of hyperlinks which are of no use to user are now not considered if they are not required.

The topic contamination is removed to some extent in case of broad topic queries. By implementing this technique now user will get results related to single aspect of query to some extent.

The problem of Topic drift has also been overcome by this procedure. By selective expansion, now user will get results according the fired query. There are limited numbers of chances to get pages which are not related to query

7.2 Future Works

- 1) In future we want to make it dynamically for getting the root set. This will give more number of relevant pages related to entered query.
- 2) We also want to improve the ARC algorithm by applying content analysis, to get pages by checking their content. This is helpful in improving the search engine by first checking the contents by applying any text mining algorithm, for getting high percentage of related pages.
- 3) We want to make a search engine after that we will apply text mining on the ARC algorithm.

Appendix-A

List of Abbreviations

Appendix A
DEFINITION OF TERMS

Abbreviations	Full Form
HITS	Hypertext Induced Topic Search
SelHITS	Selective Hypertext Induced Topic Search
H	Hub
A	Authority
SALSA	Stochastic Algorithm for Link Structure Analysis
URL	Universal Resource Locator
HTML	Hypertext Induced Topic Search
WWW	World Wide Web
N/W	Network
ARC	Automatic Resource Compilation
DOM	Document Object Model
IR	Information Retrieval