

# Improved Algorithm for Topic Distillation Using SelHITS

TH-5256



DATA ENTERED

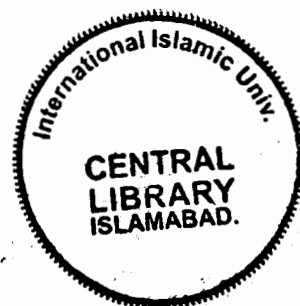
*Developed by:*

**ZUBEDA KHANUM**  
(266- 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*

MS

006.3

ZUG



Accession No TH 5256

Mail  
ToS256E2008CSMS

8/12/2

data mining  
algorithm for distillation

**Department of Computer Science**  
**International Islamic University, Islamabad**

**Date: 21-08-08**

**FINAL APPROVAL**

It is certified that we have read the project titled "Improved Algorithm for Topic Distillation Using SelHITS" submitted by **Miss ZUBEDA KHANNUM Reg. No. 266-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**

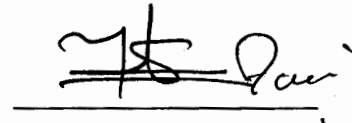
**External Examiner:**

Dr. A. Sattar  
Fmr. Director General  
Pakistan Computer Bureau,  
Islamabad.



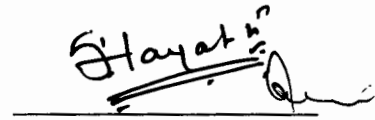
**Internal Examiner:**

Dr. Muhammad Sher  
Head of Department  
Department of Computer Science  
International Islamic University,  
Islamabad.



**Supervisors:**

Dr. M. Sikandar Hayat Khiyal  
Chair Person 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)

## **DECLARATION**

I, hereby declare that “Improved Algorithm for Topic Distillation Using SelHITS” software, neither as a whole nor as a part thereof has been copied out from any source. I have developed this software and the accompanied report entirely on the basis of my 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.

Zubeda Khanum

266-FAS/MSCS/F05

## **ACKNOWLEDGEMENTS**

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

I am very thankful to my kind, dynamic and able supervisor, Dr. M. Sikandar Hayat Khiyal for taking the time out of his ever busy schedule in providing me the guidance and direction and also always gave me 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.

I adore and pray for all my teachers for guiding and assisting me in acquiring right type of knowledge at right time. I shall ever remain grateful to all of them for their kind help.

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

I am indeed also very appreciative of the contribution of my younger sisters and brothers for their continuous solace and forbearance, which give us courage and strength to complete my project in time.

Zubeda Khanum



## **PROJECT IN BRIEF**

**Project Title:** Improved Algorithm for Topic Distillation Using SelHITS

**Objective:** To improve content and connectivity based algorithm for topic distillation.

**Undertaken By:** Zubeda Khanum

**Supervised By:** Dr. M. Sikander Hayat Khiyal

**Starting Date:** February 2007

**Completion Date:** August 2008

**Operating Systems:** Windows XP

**Tool Used:** Matlab, Microsoft Access, Visual C++

**System Used:** Intel Pentium IV, 2.25 MHz processor

## **ABSTRACT**

*Searching information on the World Wide Web (WWW) has now become a common practice but this is not simple one. Search engines have been developed and used by people all over the world. People fire queries to search engines and hoping that they will find exactly what they want, from the vast amount of information scattered on millions of web servers but finding the relevant web pages to satisfy a user's information need still remains an important and challenging task. The relevancy of web pages returned by search engine is still lacking, for example some times pages on the WWW are not honest about their contents. Therefore further research and development are needed to make search engine more effective.*

*In thesis we have worked on webminig algorithm to solve the problem of topic contamination and topic drift. We have improved the connectivity and content algorithm for search engines by applying the SelHITS technique on them. The results have shown that we can overcome the above mentioned problem by getting the most appropriate pages for user queries.*

---

**TABLE OF CONTENTS**

<b>1.0 INTRODUCTION .....</b>	<b>1</b>
1.1 TOPIC DISTILLATION.....	2
1.2 HYPERLINKS.....	3
1.3 INFORMATION RETRIEVAL.....	4
1.4 WEB MINING.....	5
1.5 MODELING THE WEB AS A DIRECTED GRAPH.....	6
1.6 EXISTING LINK ANALYSIS ALGORITHM.....	8
1.6.1 Page Rank.....	8
1.6.2 Hypertext Induced Topic Selection (HITS).....	11
1.6.3 SALSA.....	12
1.7 CONTENT ANALYSIS.....	12
1.7.1 Types Of Content Analysis.....	12
<b>2.0 LITERATURE SURVEY .....</b>	<b>14</b>
2.1 HYPERTEXT INDUCED TOPIC SEARCH.....	14
2.2 PAGE RANK ,HITS AND A UNIFIED FRAME WORK FOR LINK ANALYSIS.....	16
2.3 THE CONNECTIVITY SERVER: FAST ACCESS TO LINKAGE INFORMATION ON THE WEB.....	17
2.3.1 Internal Structure.....	18
2.3.2 Performance.....	21
2.4 SELECTIVE HYPERTEXT INDUCED TOPIC SEARCH.....	22
2.5 IMPROVED ALGORITHM FOR TOPIC DISTILLATION IN A HYPERLINKED ENVIRONMENT.....	23
2.5.1 Mutually Reinforcing Relationships Between Hosts.....	24
2.5.2 Automatically Generated Links.....	24
2.5.3 Non-relevant Nodes.....	24
2.6 AUTOMATIC RESOURCE COMPILATION BY ANALYZING THE HYPERLINK STRUCTURE AND ASSOCIATED TEXT.....	25
<b>3.0 PROBLEM DEFINITION .....</b>	<b>27</b>
3.1 BLIND EXPANSION OF ROOT SET.....	27
3.2 DISTILLING PURE TOPIC.....	27
<b>4.0 DESIGN .....</b>	<b>29</b>
4.1 CONTENT AND CONNECTIVITY BASED ALGORITHMS.....	29
4.2 SELHITS ALGORITHM.....	31
4.3 DESIGN OF PROPOSED SYSTEM.....	33

<b>5.0 IMPLEMENTATION</b> .....	<b>36</b>
5.1 TECHNOLOGY.....	36
5.1.1 Matlab.....	36
5.1.2 C#.....	36
5.2 IMPLEMENTATION.....	37
5.2.1 Root Set.....	38
5.2.2 Candidate pages.....	38
5.2.3 Base Set.....	39
5.2.4 Pruned Set.....	39
5.2.5 Top Hub and Authority.....	40
5.2.5.1 Top hub and Authority through <i>med</i> algorithm.....	40
5.2.5.2 Top hub and Authority through <i>Start Set Median</i> algorithm.....	40
5.2.5.3 Top hub and Authority through <i>Fraction of Maximum Weight</i> Algorithm.....	41
<b>6.0 TESTING AND RESULTS</b> .....	<b>42</b>
6.1 PURPOSE.....	42
6.2 TESTING PRINCIPLES.....	43
6.3 TESTING SPECIFICATION PLAN.....	43
6.4 TESTING DURING DESIGN.....	44
6.5 TESTING DURING CODING.....	44
6.6 TESTING OF “IMPROVED ALGORITHM FOR TOPIC DISTILLATION USING SELHITS”.....	45
6.6.1 Mouse.....	46
6.6.2 Windows.....	49
6.7 ANALYSIS OF RESULTS.....	52
<b>7.0 CONCLUSION AND FUTURE WORK</b> .....	<b>53</b>
7.1 PERFORMANCE.....	53
7.2 FUTURE WORKS.....	53
<b>APPENDIX A - LIST OF ABBREVIATIONS</b>	<b>A-1</b>
<b>APPENDIX B - SCREEN SHOTS</b>	<b>B-1</b>
<b>References</b>	

***1***

***Introduction***



---

## **INTRODUCTION**

Searching information on the World Wide Web (WWW) is now a common practice but not really a simple one. People fire queries to search engines hoping that they will find exactly what they want, from the haystack of data scattered on millions of web servers. The size of web is daunting. 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 outlinks 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.

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. 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. 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 data mining. When the user is searching for data mining then it can be in several contexts such as data mining as a subject, books on data mining, conferences related to data mining, well known people in the field of data mining, important research groups on data mining, companies providing tools for data 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. "data mining conferences", "data mining tools". But with topic distillation a user can get information about all aspects of data 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 iterative, eigenvector based 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.

Kleinberg's work was further augmented by Bharat et al.[1] and Chakrabarti et al. [3] by combining link analysis with content analysis of pages. Algorithms like Stochastic Algorithm for Link Structure Analysis (SALSA) combine HITS with PageRank .

## 1.2. Hyperlinks

Hyperlinking by means of citation is a concept of great vintage. In 1945 Vannevar Bush developed first modern hyperlinked system called Memex which could create and follow hyperlinks between documents In 1990 Tim Berners-Lee named his hypertext browsing



system as "World Wide Web". Later, Hyper Text Transfer Protocol (HTTP) was developed for exchange of hypertext. At the same time CERN developed Hyper Text Mark up Language (HTML). Since then the WWW has experienced an exponential growth and it is still growing today.

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.

### **1.3. Information Retrieval**

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. How can one decide the most relevant top 10-20 pages for a query yielding thousands of candidate pages? Traditional models like Term Frequency and Inverse Document Frequency (TF-IDF) vector representation and *cosine similarity* with the query are also not good measures. As opposed to plain text documents, links between pages is an important factor for hypertext documents on the WWW.

The main idea of hyper documents is that documents or parts there of 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 300 million pages on the Web today with about 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.

## **1.4 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. Web content mining is the process of extracting knowledge from the content of documents or their descriptions. Web document text mining, resource discovery based on concepts indexing or agent based technology may also fall in this category. Web structure mining is the process of inferring knowledge from the World Wide Web organization and links between references and referents in the Web. Finally, web usage mining, also known as Web Log Mining, is the process of extracting interesting patterns in web access logs.

- **Web Content Mining**

Web content mining is an automatic process that goes beyond keyword extraction. Since the content of a text document presents no machine readable semantic, some approaches have suggested restructuring the document content in a representation that could be exploited by machines. The usual approach to exploit known structure in documents is to use wrappers to map documents to some data model. Techniques using lexicons for content interpretation are yet to come. There are two groups of web

content mining strategies: Those that directly mine the content of documents and those that improve on the content search of other tools like search engines.

- **Web Structure Mining**

World Wide Web can reveal more information than just the information contained in documents. For example, links pointing to a document indicate the popularity of the document, while links coming out of a document indicate the richness or perhaps the variety of topics covered in the document. This can be compared to bibliographical citations. When a paper is cited often, it ought to be important. The PageRank and CLEVER methods take advantage of this information conveyed by the links to find pertinent web pages. By means of counters, higher levels cumulate the number of artifacts subsumed by the concepts they hold. Counters of hyperlinks, in and out documents, retrace the structure of the web artifacts summarized.

- **Web Usage Mining**

Web servers record and accumulate data about user interactions whenever requests for resources are received. Analyzing the web access logs of different web sites can help understand the user behavior and the web structure, thereby improving the design of this colossal collection of resources. There are two main tendencies in Web Usage Mining driven by the applications of the discoveries: General Access Pattern Tracking and Customized Usage Tracking. The general access pattern tracking analyzes the web logs to understand access patterns and trends. These analyses can shed light on better structure and grouping of resource providers.

## **1.5 Modeling the Web as 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:

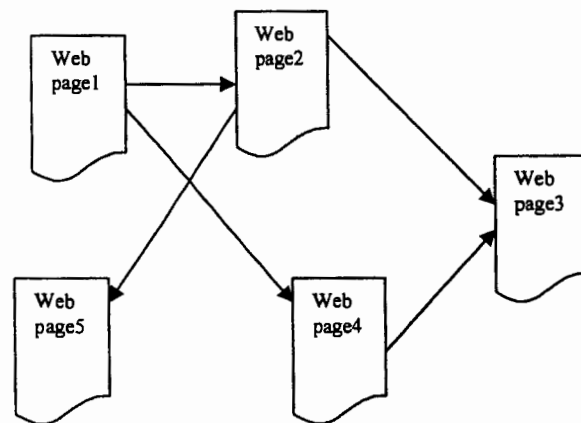


Figure1.1: Hyperlinked Structure

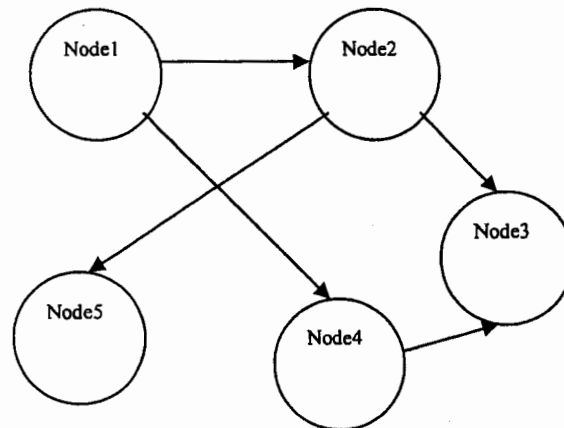


Figure1.2:Directed Graph

A hyperlinked Structure is shown in Figure 1.1 and is converted to Directed Graph in Figure 1.2. 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.

## 1.6 Existing Link Analysis Algorithms

The three main algorithms considered are PageRank, HITS and SALSA. PageRank 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.

### 1.6.1 Page Rank

PageRank 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 ranking 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.

- **How is PageRank calculated?**

To calculate the PageRank for a page, all of its inbound links are taken into account. These are links from within the site and links from outside the site.

$$PR(A) = (1-d) + d(PR(t1)/C(t1) + \dots + PR(tn)/C(tn))$$

That's the equation that calculates a page's PageRank. It's the original one that was published when PageRank was being developed, and it is probable that Google uses a

variation of it but they aren't telling us what it is. It doesn't matter though, as this equation is good enough.

In the equation ' $t_1 - t_n$ ' are pages linking to page A, 'C' is the number of outbound links that a page has and 'd' is a damping factor, usually set to 0.85.

We can think of it in a simpler way:-

a page's PageRank =  $0.15 + 0.85 * (\text{a "share" of the PageRank of every page that links to it})$

"share" = the linking page's PageRank divided by the number of outbound links on the page.

A page "votes" an amount of PageRank onto each page that it links to. The amount of PageRank that it has to vote with is a little less than its own PageRank value (its own value \* 0.85). This value is shared equally between all the pages that it links to.

From this, we could conclude that a link from a page with PR4 and 5 outbound links is worth more than a link from a page with PR8 and 100 outbound links. The PageRank of a page that links to yours is important but the number of links on that page is also important. The more links there are on a page, the less PageRank value your page will receive from it.

If the PageRank value differences between PR1, PR2,.....PR10 were equal then that conclusion would hold up, but many people believe that the values between PR1 and PR10 (the maximum) are set on a logarithmic scale, and there is very good reason for believing it. Nobody outside Google knows for sure one way or the other, but the chances are high that the scale is logarithmic, or similar. If so, it means that it takes a lot more additional PageRank for a page to move up to the next PageRank level that it did to move up from the previous PageRank level. The result is that it reverses the previous conclusion, so that a link from a PR8 page that has lots of outbound links is worth more than a link from a PR4 page that has only a few outbound links.

Whichever scale Google uses, we can be sure of one thing. A link from another site increases our site's PageRank. Just remember to avoid links from link farms.

Note that when a page votes its PageRank value to other pages, its own PageRank is not reduced by the value that it is voting. The page doing the voting doesn't give away its PageRank and end up with nothing. It isn't a transfer of PageRank. It is simply a vote according to the page's PageRank value. It's like a shareholders meeting where each shareholder votes according to the number of shares held, but the shares themselves aren't given away. Even so, pages do lose some PageRank indirectly, as we'll see later.

For a page's calculation, its existing PageRank (if it has any) is abandoned completely and a fresh calculation is done where the page relies solely on the PageRank "voted" for it by its current inbound links, which may have changed since the last time the page's PageRank was calculated.

The equation shows clearly how a page's PageRank is arrived at. But what isn't immediately obvious is that it can't work if the calculation is done just once. Suppose we have 2 pages, A and B, which link to each other, and neither have any other links of any kind. This is what happens:-

**Step 1:** Calculate page A's PageRank from the value of its inbound links

Page A now has a new PageRank value. The calculation used the value of the inbound link from page B. But page B has an inbound link (from page A) and its new PageRank value hasn't been worked out yet, so page A's new PageRank value is based on inaccurate data and can't be accurate.

**Step 2:** Calculate page B's PageRank from the value of its inbound links

Page B now has a new PageRank value, but it can't be accurate because the calculation used the new PageRank value of the inbound link from page A, which is inaccurate.

It's a Catch 22 situation. We can't work out A's PageRank until we know B's PageRank, and we can't work out B's PageRank until we know A's PageRank.

Now that both pages have newly calculated PageRank values, can't we just run the calculations again to arrive at accurate values? No. We can run the calculations again using the new values and the results will be more accurate, but we will always be using inaccurate values for the calculations, so the results will always be inaccurate.

The problem is overcome by repeating the calculations many times. Each time produces slightly more accurate values. In fact, total accuracy can never be achieved because the calculations are always based on inaccurate values. This is precisely what Google does at each update, and it's the reason why the updates take so long.

One thing to bear in mind is that the results we get from the calculations are proportions. The figures must then be set against a scale (known only to Google) to arrive at each page's actual PageRank. Even so, we can use the calculations to channel the PageRank within a site around its pages so that certain pages receive a higher proportion of it than others.

### **1.6.2 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.

### 1.6.3 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.

## 1.7 Content Analysis.

Content analysis is a research tool used to determine the presence of certain words or concepts within texts or sets of text. It is the process of analyzing the content of document to find the relevant information or the process of analyzing text to extract information that is useful for particular purposes. We can find the more relevant documents on World Wide Web by the combination of Content analysis techniques with connectivity analysis methods.

### 1.7.1 Types of Content Analysis

There are two general categories of content analysis: conceptual analysis and relational analysis.

- **Conceptual analysis**

Conceptual analysis can be thought of as establishing the existence and frequency of concepts – most often represented by words or phrases – in a text. For instance, say you have a hunch that your favorite poet often writes about hunger. With conceptual analysis you can determine how many times words such as “hunger,” “hungry,” “famished,” or “starving” appear in a volume of poems.

- **Relational analysis.**

Relational analysis goes one step further by examining the relationships among concepts in a text. Returning to the “hunger” example, with relational analysis, you could identify what other words or phrases “hunger” or “famished” appear next to and then determine what different meanings emerge as a result of these groupings.

2

*Literature Survey*

---

---

---

## LITERATURE SURVEY

### 2.1 Hypertext Induced Topic Search

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. Second, 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, 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. How to get 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. How to generate Base Set from Root Set is shown in Figure 2.2.

- **Hub Page:**

A page that doesn't provide information, but tell you where to find the information. 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 Figure2.3(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. 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. Co-reference measures the similarity between WebPages. Thus hub matrix is the sum of co-reference and out degree

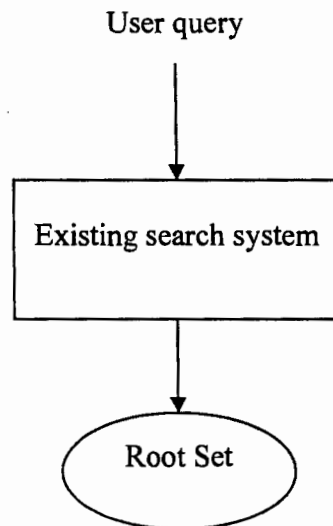


Fig2.1: How to get Root Set

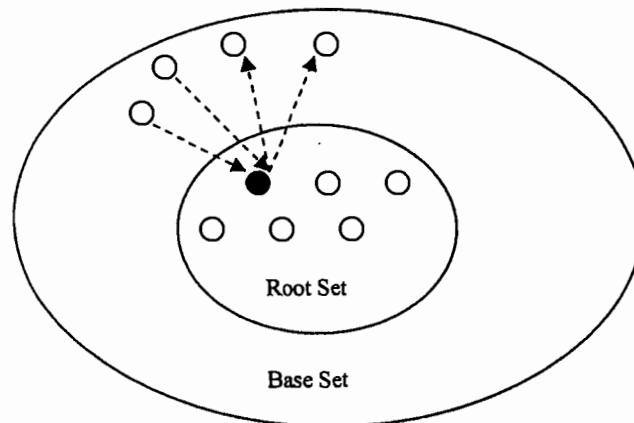


Fig2.2: Generating Base Set of Root Set

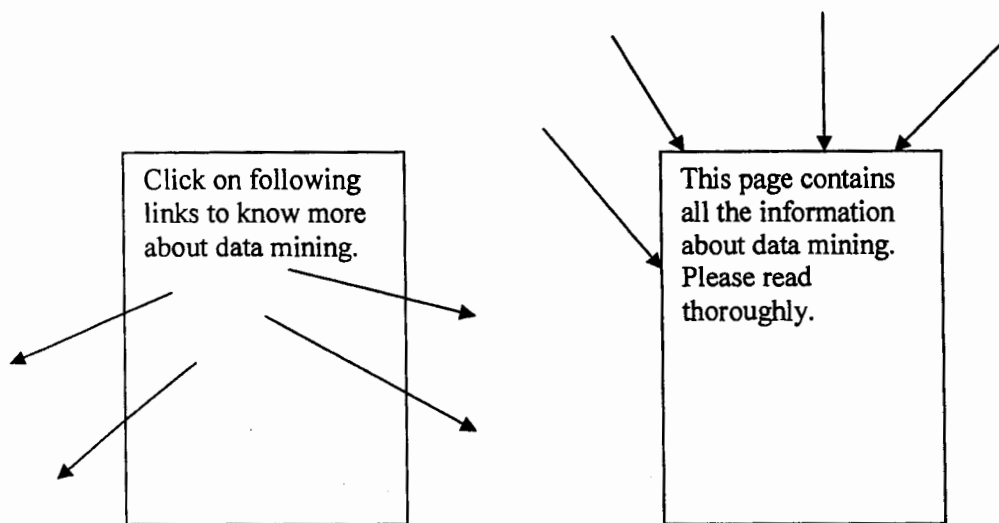


Figure2.3(a): Hub Page

Figure2.3(b): Authority Page

## 2.2 Page Rank, HITS and a Unified Framework for Link Analysis

Parry *et al* [12] discuss Page Rank and Hypertext Induced Topic Search (HITS) with mutual reinforcement of hub and authorities. Concept of Co-citation and Co-Reference is discussed. If two distinct web pages  $p_i$ ,  $p_j$  are co-cited by many other web pages  $p_k$  as in Figure2.4,  $p_i$ ,  $p_j$  are likely to be related in some sense. The fact that two distinct

webpage's  $p_i$ ;  $p_j$  co-reference several other webpage's  $p_k$ ) indicates that  $p_i$ ;  $p_j$  have certain commonality [11].

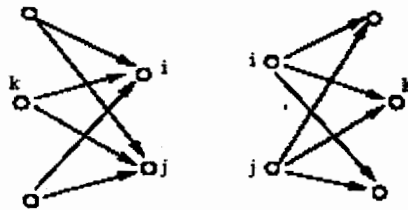


Figure 2.4: Left: web pages  $p_i$ ;  $p_j$  are co-cited by webpage  $p_k$ . Right: web pages  $p_i$ ;  $p_j$  Co-reference webpage  $p_k$ .

The most important feature of Hypertext Induced Topic Search (HITS) algorithm by Kleinberg *et al* (1998)[6] is the mutual reinforcement between hubs and authorities, while the most important feature of PageRank is the hyperlink weight normalization.

This paper combine concepts of mutual reinforcement and hyperlink weight normalization into a unified framework and introduce three new normalized ranking algorithms within this framework

- 1) **INORM Rank:** In this case inlinks are normalized using norm.
- 2) **ONORM Rank:** In this case outlinks are normalized by taking norm.
- 3) **SNORM Rank:** In this case in-links and out-links are normalized in symmetric fashion.

### 2.3 The Connectivity Server: fast access to Linkage information on the Web

Monika *et al* [11] describe a system called Connectivity Server that provides linkage information for all pages indexed by the AltaVista search engine. In its basic operation, the server accepts a query consisting of a set  $L$  of one or more URLs and returns a list of all pages that point to pages in  $L$  (predecessors) and a list of all pages that are pointed from pages in  $L$  (successors). More generally the server can produce the entire neighborhood (in the graph theory sense) of  $L$  up to a given distance and can include

information about all links that exist among pages in the neighborhood. Although some of this information can be retrieved directly from Alta Vista or other search engines, these engines are not optimized for this purpose and the process of constructing the neighbourhood of a given set of pages is slow and laborious. In contrast Connectivity Server needs less than 0.1 ms per result URL. Two applications that use the Connectivity Server: a direct interface that permits fast navigation of the Web via the predecessor/successor relation, and a visualization tool for the neighbourhood of a given set of pages.

### 2.3.1 Internal organization

- **Initial data structures**

Representing a small graph is trivial. Representing a graph with 100 millions nodes and close to a billion edges is an engineering challenge.

We represent the Web as a graph consisting of nodes and directed edges. Each node represents a page and a directed edge from node  $A$  to node  $B$  means that page  $A$  contains a link to page  $B$ . The set of nodes is stored in an array, each element of the array representing a node. The array index of a node element is the node's  $ID$ . We represent the set of edges emanating from a node as an adjacency list that is for each node we maintain a list of its successors. In addition, for each node we also maintain an inverted adjacency list that is a list of nodes from which this node is directly accessible, namely its predecessors. Therefore a directed edge from node  $A$  to node  $B$  appears twice in our graph representation, in the adjacency list of  $A$  and the inverted adjacency

list of  $B$ . This redundancy in representing edges simplifies both forward and backward traversal of edges. To minimize fragmentation, elements of all adjacency lists are stored together in one array called the Outlist. Similarly elements of all inverted adjacency lists are stored in another array called the Inlist. The adjacency and inverted adjacency lists stored in each node are represented as offsets into the Outlist and Inlist arrays respectively. The end of the adjacency list for a node is marked by an entry whose high order bit is set Figure 2.5. Thus we can determine the predecessors and the successors of any node very quickly.



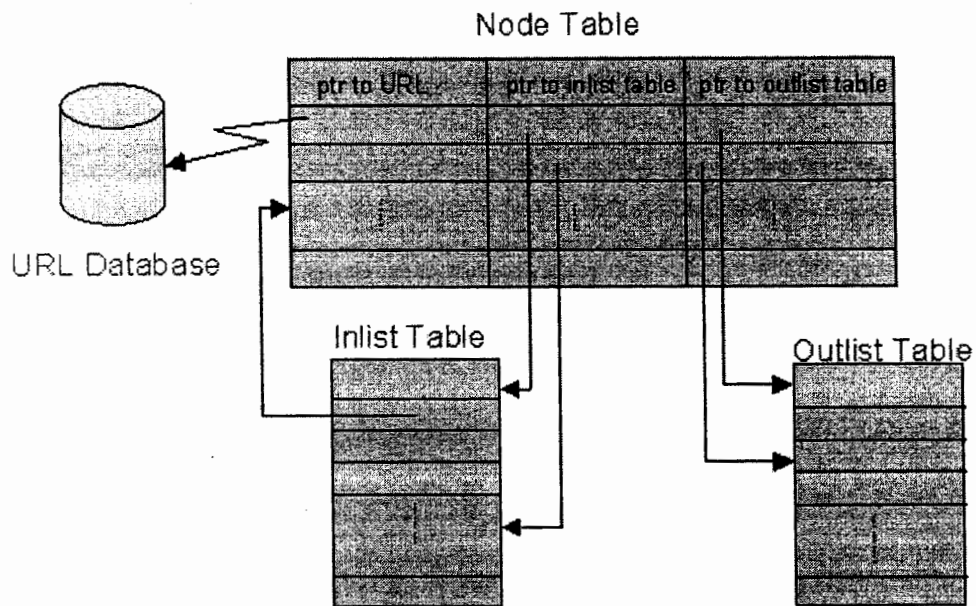


Figure2.5.: Representation of the graph.

A node in the Web-graph has an attached URL. Since URLs are rather long (about 80 bytes on average), storing the full URL within every node in the graph would be quite wasteful. (The storage requirement of a naive implementation would be about 8 gigabytes for 100 million URLs!) Instead the server maintains data structures that represent the *ID* to URL and URL to *ID* mappings.

After a full crawl of the Web, all the URLs that are to be represented in the server are sorted lexicographically. The index of a URL in this sorted list is its initial *ID* (see the discussion of updates below). Then the list of sorted URLs is stored as a delta-encoded text file, that is, each entry is stored as the difference (delta) between the current and previous URL. Since the common prefix between two URLs from the same server is often quite long, this scheme reduces the storage requirements significantly. With the 100 million URLs in our prototype we have seen a 70% reduction in size Figure2.6.

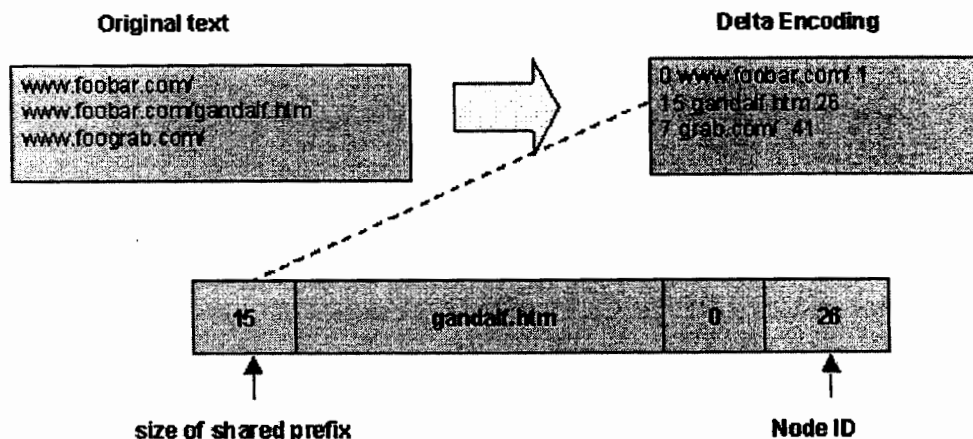


Figure 2.6: Delta Encoding the URL's

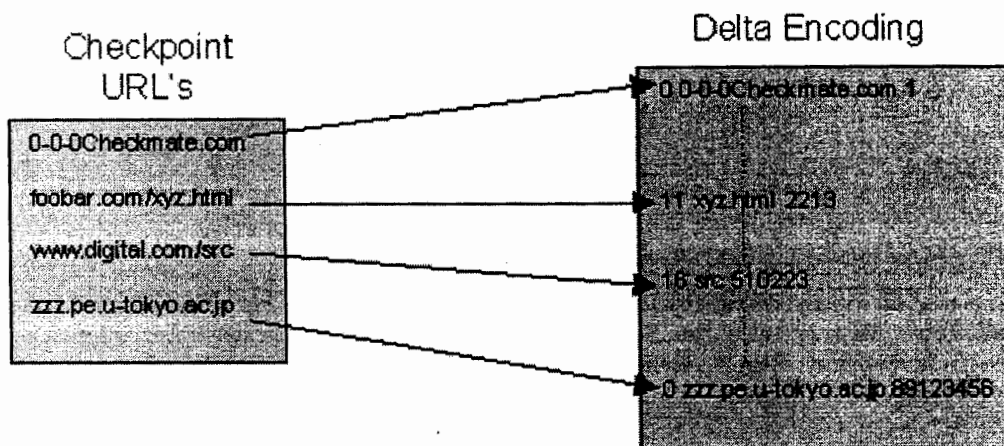


Figure 2.7: Indexing the Delta Encoding

This reduction in storage requirements comes at a price, namely the slowdown of the translation. In order to convert a delta encoded entry back to its complete URL, one needs to start at the first URL and apply all the deltas in sequence until arriving at the URL in question. Author avoids this problem by periodically storing the entire URL instead of the delta encoding. This entry is called a *checkpoint* URL. Therefore to translate a delta encoded URL, we need to apply the deltas starting from the last checkpoint URL rather than the first URL. The cost of the translation can be reduced by increasing the checkpoint frequency Figure 2.7. To translate a URL to an internal ID we first search the

sorted list of checkpoint URLs to find the closest checkpoint. Then the delta encoded list is searched linearly from that checkpoint URL until the relevant URL is reached. To speed up the reverse translation from internal ID to an URL, the relevant node points directly to the closest checkpoint URL. As before the URL is computed by searching linearly from the checkpoint URL.

- **Updates**

Since their structure is very tight, updates are not simple. Currently their design is to batch all the updates for a day. They view all the updates as a collection of nodes and edges to be added or deleted. All deletions can be done by marking the deleted edges and nodes in a straightforward manner. This requires an extra bit per edge and node. Additions are done as follows.

To allow for additions, They allocate initially larger tables than immediately necessary. For newly added nodes, they maintain a separate structure for the URL to id translation, organized as a string search tree. This tree contains all the newly added nodes and their assigned ID's in the main data structure. To update the Outlist table, the list of new edges is grouped by source. If the new Outlist associated to a node is longer than the old Outlist, space is allocated at the end of the current Outlist table. The update of the Inlist table is done similarly, except that edges are sorted by destination. Eventually the wasted gaps in tables consume too much space, and/or the additional node tree becomes too large and then the entire structure is rebuilt.

### **2.3.2 Performance**

The Connectivity Server performs three steps to process queries: translate the URLs in the query to node IDs, explore the Web graph around these nodes and translate the IDs in the result set back to URLs. Thus the time needed to process queries is proportional to the size of the result set. On a 300 MHz Digital Alpha with 4 GB memory, the processing time is approximately 0.1 ms/URL in the result set. Figure 4 shows the timings for 15 different queries where the answer size varies from 1192 to 5734 URLs. As the third step takes up most of the processing time, i.e. 80%. The remainder time is shared equally between steps one and two. Therefore, applications

that can work with internal IDs can expect an even faster processing time of about 0.01 ms/URL.

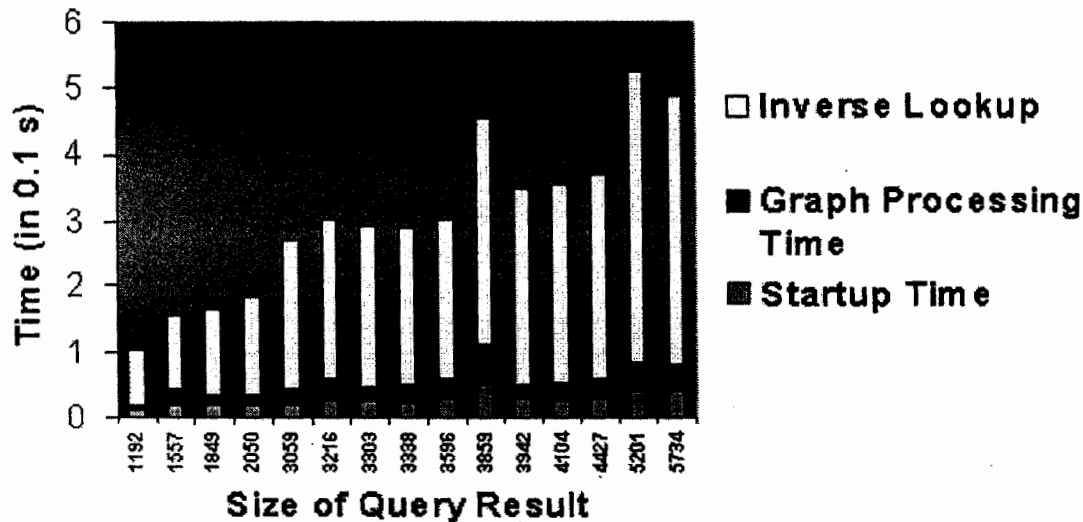


Figure 2.8 Query Processing Time

## 2.4 Selective Hypertext Induced Topic Search

Mitra *et al* [7] discussed SELHIT algorithm Figure 2.10. That is an improvement over Kleinberg *et al* [6] Hypertext Induced Topic Search algorithm (see Figure 2.9) for answering broad-topic queries and addresses some Hypertext Induced Topic Search (HITS) algorithm problems for example topic drift, Topic Contamination by selectively expanding the root set .

Basically the SELHIT algorithm first calculates the hub and authority scores on root set returned by the search engine and then selects top hubs and authorities for further expansion to get base set. This selective expansion procedure drastically reduces size of the base set to avoid topic drift, as irrelevant pages are not added to the root set. Therefore SELHITS algorithm indeed distills the most important and relevant pages for broad-topic queries.

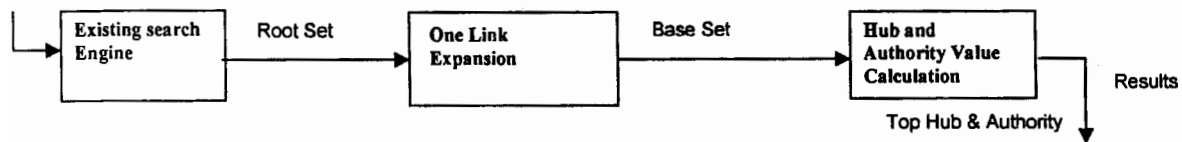


Figure 2.9: HITS Algorithm.

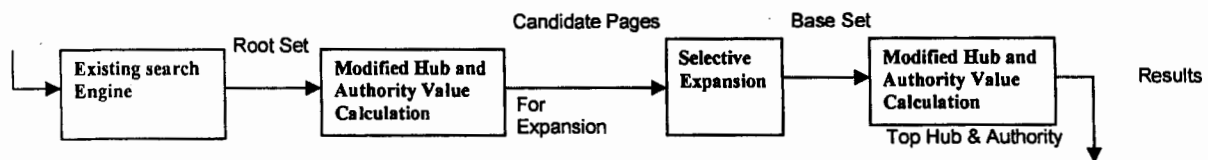


Figure 2.10: SelHit Algorithm.

## 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)[1] connectivity analysis algorithm and presented various algorithms to address those problems. Problems in Hypertext Induced Topic Search (HITS) connectivity analysis algorithm are as follows [1].

### 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 Hypernews 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 imp to address the first problem by giving fractional weights to each edge, basically the imp algorithms is an improvement over Hypertext Induced Topic Search (HITS) algorithm by Kleinberg *et al* (1997)[6] and described some other algorithms to address other two problems. These algorithms are the combination of content analysis using traditional Information Retrieval techniques with improved connectivity analysis algorithm "imp".

## 2.6 Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text

Chakrabarti *et al* [3] defines Automatic Resource Compilation (ARC). ARC is based on a combination of text and link analysis for distilling authoritative Web resources. The ARC algorithm also extends Hypertext Induced Topic Search (HITS) algorithm with textual analysis. ARC computes a distance-2 neighborhood graph and weights edges. The weight of each edge is based on the match between the query terms and the text surrounding the hyperlink in the source document. Both Chakrabarti *et al* [3] and Bharat *et al* [1] studies are different in three way . Bharat *et al*[1] uses an expanded query while Chakrabarti *et al* [3] uses the original query. Bharat *et al* [1] computed the relevancy using the whole document while Chakrabarti *et al* [3] computed the relevancy using the hyperlink text. The weight of an edge in Bharat *et al* [1] is either the relevance of the source document or the target document depending on whether authority or hub scores are being computed while in Chakrabarti *et al* [3] weight of each edge is based on the match between the query terms and the text surrounding the hyperlink .ARC is based on a combination of text and link analysis for distilling authoritative Web resources.

Chakrabarti *et.al*[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 our 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 are shown in Figure 2.12.

### 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*  $w(p,q)$  that increases with the amount of topic-related text in the vicinity of the href from  $p$  to  $q$ .

### 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.

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

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

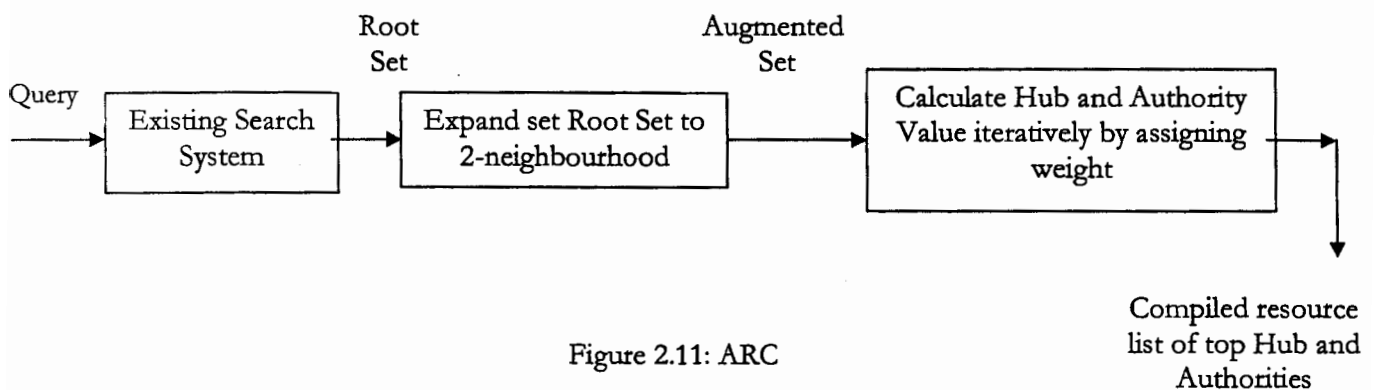


Figure 2.11: ARC



3

*Problem Definition*



### **Problem Statement.**

Bharat *et al* [1] discovered three problems with connectivity analysis as suggested by Kleinberg *et al* [6]

- 1) Mutually Reinforcing Relationships Between hosts,
- 2) Automatically Generated Links,
- 3) Non-relevant Documents and discussed the several techniques for tackling these three scenarios and achieved considerable improvements. However these techniques still contain the following problems.

#### **3.1 Blindly Expansion of Root set**

Existing search systems return thousands of results for broad queries, only top few are directly relevant and important for the topic of the query. After adding all pages in one link neighborhood, the size of the base set becomes of the order of a few thousand pages. Most of the pages added are useless and including them in the base set causes the **extra time consumption** because pages in the base are used for further processing for example content analysis.

#### **3.2 Distilling Pure Topic.**

Sometime users type ambiguous queries and search engine returns results from multiple topics, that causing **topic contamination** but the aim of topic distillation process is to deliver results for a single topic only. For example user fires the query “mouse”. This query is ambiguous and has multiple meanings. Meanings of mouse can be device of computer or animal. So depending on meaning there will be different topics for the query.

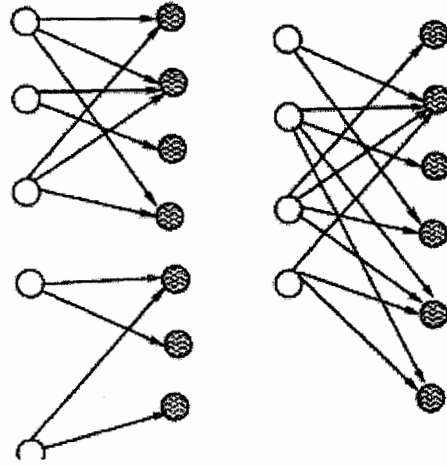


Fig3.1: Multiple Graphs Relating to Different Aspects Of A Single Broad Topic Query

4

*Design*



## DESIGN

Bharat *et al* [1] discussed one purely connectivity based algorithms “imp” and nine content and connectivity based algorithms. We have considered the three content and connectivity base algorithms.

- 1) med.
- 2) startmed,
- 3) maxby10.

These algorithms pruned the irrelevant nodes by computing the Relevance Weights of the nodes in the Neighborhood graph.

The relevance weight of a node equals the similarity of document to the query topic. These algorithms use the documents in the root set to define a query and match every document in the Neighborhood graph against this query, and consider the concatenation of the first 1000 words from each document to be the query,  $Q$  and compute *similarity* ( $Q, D$ ) by using following formula.

$$\text{similarity}(Q, D_j) = \frac{\sum_{i=1}^t (w_{iq} \times w_{ij})}{\sqrt{\sum_{i=1}^t (w_{iq})^2 \times \sum_{i=1}^t (w_{ij})^2}}$$

$$w_{iq} = \text{freq}_{iq} \times \text{IDF}_i.$$

$$w_{ij} = \text{freq}_{ij} \times \text{IDF}_i.$$

$\text{freq}_{iq}$  = the frequency of the term  $i$  in query  $Q$ .

$\text{freq}_{ij}$  = the frequency of the term  $i$  in document  $D_j$ .

$\text{IDF}_i$  = an estimate of the inverse document frequency of term  $i$  on the World Wide Web.

After getting the relevance weights threshold value is calculated by using relevance weights. In **med** algorithm threshold is the median of the relevance weights of the nodes in Neighborhood graph/base set. In **startmed** threshold is the median of the relevance weights of the nodes in the start/root set. and in **maxby10** threshold is a fixed fraction of the maximum weight of the nodes in Neighborhood graph/base set.. This threshold value is used to pruned irrelevant documents.

#### 4.1 Content and Connectivity based Algorithms.

These algorithms first construct a query specific graph against the user query whose nodes are documents. The graph is constructed as follows Figure 4.1. A start/root set of documents matching the query is fetched from a search engine (say the top 200 matches), This set is blindly expanded by its neighborhood, which is the set of documents that either point to or are pointed by the documents in the root set. The documents in the root set and its neighborhood together form the nodes of the neighborhood graph, the number of nodes in neighborhood graph is called base set. After getting the base set, the algorithms perform content analysis to Pruned irrelevant Nodes from the Neighborhood Graph. Pruning is performed by Computing the Relevance Weights of the Nodes in neighborhood graph and use the relevance weight of a node to decide if it should be eliminated from the graph. This decision is dependent on the thresholds of the relevance weights. Thresholds are picked in one of three ways.

1. **med(Median Weight):** The threshold is the median of the relevance weights of node in Neighborhood graph/base set,
2. **startmed (Start Set Median Weight):** The threshold is the median of the relevance weights of the nodes in the start/root set.
3. **maxby10(Fraction of Maximum Weight):** The threshold is a fixed fraction of the maximum weight. Bharat *et al* [12] used  $max/10$ . The relevance weight of a node equals the similarity of its node/document to the query topic.

All nodes whose Weights are below a threshold are pruned from base set and resultant pruned set is used for further processing. On the pruned set/graph the connectivity based algorithm “imp” is applied to computes the hub and authority scores for all the nodes in pruned set/graph and call the corresponding algorithms: *med*, *startmed*, and *maxby10*. These algorithms report top hub and authority pages to the user.

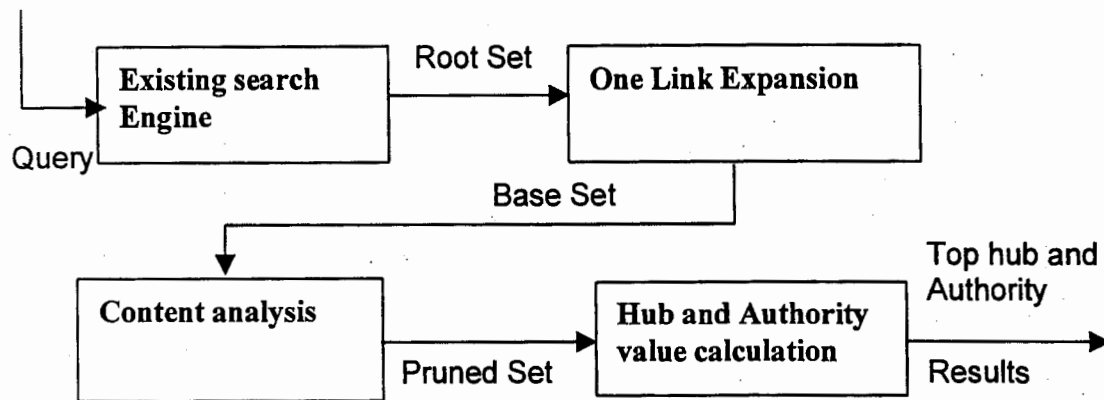


Figure 4.1: Architecture of three Content and Connectivity based Algorithms

#### Drawbacks:

- The blindly “One Link Expansion” procedure to get base set drastically increases size of the base set. Most of the pages added are useless and including them in the base set causes the **extra time consumption**, as the content analysis is also performed on useless pages as well as useful pages.
- In case of ambiguous queries and queries that have multiple meaning, Search Engine can return the results from multiple topics. Therefore blindly “one link expansion” procedure may causes the **topic contamination**, as one link expansion procedure is performed equally on multiple topic pages and resultant base set contains pages that are from multiple topic. Therefore content analysis is performed on multiple topic pages but the aim of topic distillation is to deliver the results from single topic only.

#### 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.

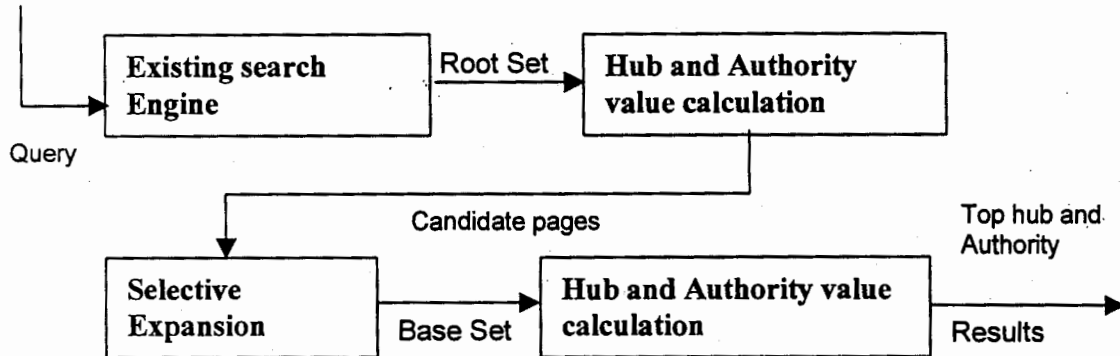


Figure4.2: Architecture of SelHit Algorithm

- The “Selective Expansion ” procedure achieves considerable improvement over blindly “one link expansion” procedure, as only relevant pages are expanded instead of expanded all relevant and irrelevant pages Figure 4.1 to get base set. Therefore size of base set decreases. Most of pages added in base set are useful and about to user query topic.
- The “Selective Expansion ”Figure 4.2 procedure also addresses problem **topic contamination that accrues due to blindly” One Link Expansion”** procedure to get base, as single topic pages are expanded instead of expanded multiple topic pages to get base set.



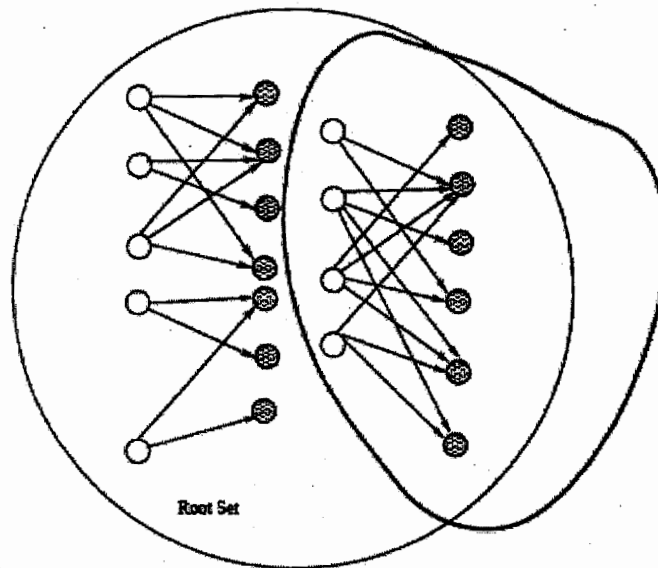


Fig 4.3: Most predominant community selected for expansion

### 4.3 System Architecture

Our aim is to blend method “SELHIT” algorithm by Mitra et al [7] with three content and connectivity based algorithms of Bharat et al [1]

- 1) med
- 2) startmed
- 3) maxby10 and hope that proposed system will further help to get better results.

In its basic operation, we have interchanged phase “one link expansion” with the two phases “Hub and Authority value calculation” and “Selective Expansion” (Fig 4.2) to address problems **topic contamination** and **extra time consumption**.

The “Hub and Authority value calculation” phase have calculated Hub and Authority values for all pages in root set and have selected top Hub and authority pages as candidate pages and “Selective Expansion” phase have selectively expanded the “candidate pages” to get the base set. This Selective Expansion procedure have decreased size of base set, as irrelevant pages have not been added in base set. Now the

base set contains pages related to user query topic and content analysis have been performed only on query related pages that have removed the **extra time consumption**.

In case of ambiguous queries, “Selective Expansion” procedure have expanded candidate pages (user query related pages) from single topic to avoid **topic contamination** because “Hub and Authority calculation“ (Figure 4.3) phase have returned the candidate pages that are from the single topic.

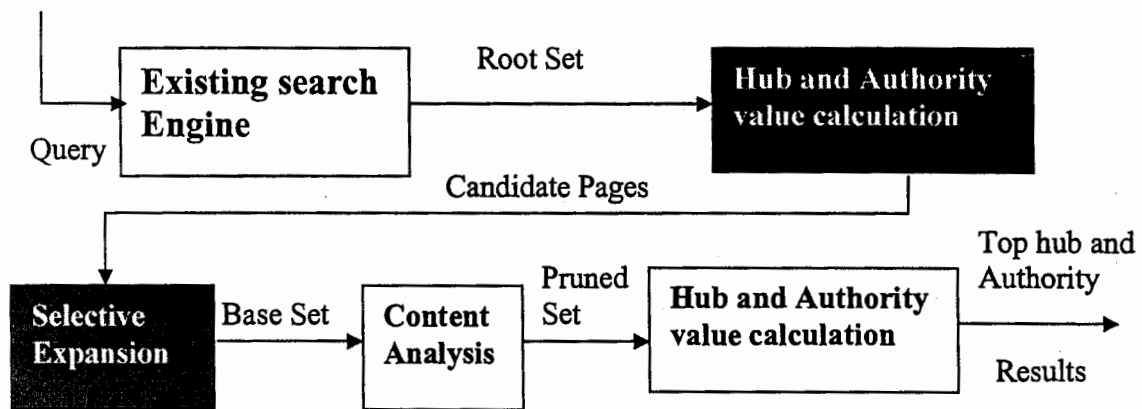


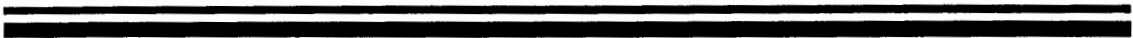
Figure4.4: Architecture of Proposed System.

Main phases of our system will be

- **Root Set:** Implemented system gets root set from existing search engine against user query.
- **Candidate Pages:** Implemented system calculates hub and authority values for each page in root set and selects top hub and authority pages as candidate pages.
- **Base Set:** Implemented system gets base set by the selective expansion of candidate pages. So instead of expanding all the pages in one link neighborhoods, Implemented system expands selective candidate pages only. Thus the resultant base set is much smaller and contains more relevant pages than total expanded root set.

5

*Implementation*



## **IMPLEMENTATION**

Implementation is an important stage and phase of software lifecycle where the thoughts and ideas are given physical shape. Implementation is a summary description of the noteworthy organization of deliverables. A good implementation approach and strategy leads to successful application or system.

### **5.1 Technology**

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

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

#### **5.1.1 Matlab**

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include Math and computation Algorithm development Data acquisition Modeling, simulation, and prototyping Data analysis, exploration, and visualization Scientific and engineering graphics Application development, including graphical user interface building MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it

would take to write a program in a scalar no interactive language such as C or Fortran. The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis. MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

We have used matlab for connecting search engine google and for extracting URI's from web pages and for designing the different functions that are helpful in getting results.

### 5.1.2 C#.Net C#.net

C# is an object-oriented programming language developed by Microsoft as part of the .NET initiative and later approved as a standard by ECMA. C# is intended to be a simple, modern, general-purpose, object-oriented programming language. C# is intended to be suitable for writing applications for both hosted and embedded systems, ranging from the very large that use sophisticated operating systems, down to the very small having dedicated functions. By design, C# is the programming language that most directly reflects the underlying Common Language Infrastructure (CLI).

We have used C#.net for text analysis of web pages.

## 5.2 Implementation

Following are the main phases of the system:

- Root Set.
- Candidate Pages.
- Base Set.
- Pruned Set
- Top hub and Authority.

Important functions used in each phase are as follows:

### 5.2.1 Root set:

The root set is of order of at least two hundred pages related to query topic. The root set is fetched from existing search engine (google) against user query.

- **Webbot()**.

This function extracts links related to user query from a existing search engine Google. These links are used for further processing.

### 5.2.2 Candidate Pages:

The top hub and authority value pages in root set are called candidate pages.

- **match()**

This function finds out the link relation among root set pages.

- **auth()**

This function finds out authority value of each page in root set.

- **hub()**

This function finds out hub value of each page in root set.

- **Cond()**

This function selects top hub and authority pages as candidate pages.

### 5.2.3 Base Set:

Base set is the one link expansion of candidate pages.

- **root\_exp\_out()**

This function finds out the out links of candidate pages.

- **root\_exp\_in()**

This function finds out the in links of candidate pages.

- **Base()**

This function combines the candidate pages and the pages resulted from the one link expansion of candidate pages as a base set.

### 5.2.4 Pruned Set

Pruned set is found by pruning irrelevant pages through the content analysis of pages in base set. Content analysis include the following steps.

- **Extraction of plain text from web pages.**
- **Finding the relevance weight of web pages**

$$\text{similarity}(Q, D_j) = \frac{\sum_{i=1}^t (w_{iq} \times w_{ij})}{\sqrt{\sum_{i=1}^t (w_{iq})^2 \times \sum_{i=1}^t (w_{ij})^2}}$$

- **Med()**

This function finds out median of base set pages on the bases of relevance weights and pruned the pages whose relevance weight less than median. The remaining set is a pruned set.

- **Fraction()**

This function find out fraction/10 of pages on the bases of relevance weights and pruned the pages whose relevance weight less than fraction/10. The remaining set is a pruned set.

- **Start\_Med()**

This function finds out median of root set pages on the bases of relevance weights and pruned the pages whose relevance weight less than median. The remaining set is a pruned set.

### 5.2.5 Top hub and Authority:

Finally top hub and authority pages in pruned are reported to the user.

#### 5.2.5.1 Top hub and Authority through *med* algorithm

- **Med match()**

This function finds out the link relation among pruned set pages.

- **Med\_auth()**

This function finds out authority value of each page in pruned set and reports the top authority pages to user.

- **Med\_hub()**

This function finds out hub value of each page in pruned set and reports the top hub pages to user.

#### 5.2.5.2 Top hub and Authority through *Start Set Median* algorithm

- **start\_match()**

This function finds out the link relation among pruned set pages.

- **start\_auth()**

This function finds out authority value of each page in pruned set and reports the top authority pages to user.

- **start\_hub()**

This function finds out hub value of each page in pruned set and reports the top hub pages to user.



### 5.2.5.3 Top hub and Authority through *Fraction of Maximum Weight* algorithm.

- **frac\_match()**  
This function finds out the link relation among pruned set pages.
- **frac\_auth()**  
This function finds out authority value of each page in pruned set and reports the top authority pages to user.
- **frac\_hub()**  
This function finds out hub value of each page in pruned set and reports the top hub pages to user.

# 6

## *Testing and Results*

---

---

---

## **TESTING AND RESULTS**

Testing is the process of executing software and comparing the observed behavior to the desired behavior and uncovering issues so that they can be addressed to validate that the team is doing the right thing.

Well-performed tests, initiated early in the software lifecycle, will significantly lower the cost of completing and maintaining the software. It will also greatly reduce the risks or liabilities associated with deploying poor quality software, such as poor user productivity, data entry and calculations errors, and unacceptable functional behavior.

### **6.1 Purpose**

Testing is a process of executing a program with the intent of finding an error. A good test case is the one that has a high probability of finding an as-yet-undiscovered error. The objective should be to design tests that systematically uncover different classes of errors and do so with a minimum amount of time and effort. The purpose of testing is to achieve the following goals:

- **Primary Goal:** To discover errors in the software
- **Secondary Goal:** Building confidence in the proper operation of the software when testing does not discover errors.
- To verify the interaction between the objects.
- To verify the proper integration of all components of the software.
- To verify that all requirements have been implemented.

## 6.2 Testing Principles

Following are some of the principles to be kept in mind before doing the testing:

- All tests should be traceable to the requirements.
- Tests should be planned long before testing begins.
- Testing should gradually cover the whole project.
- Exhaustive testing is not possible.
- To be most effective, an independent third party should conduct testing.

## 6.3 Testing Specification Plan:

The purpose of Test Specification Plan is to give complete instructions on how to perform tests on the software so that they correspond to the requirements of an application.

For quality control to be effective, testing should follow the same pattern throughout. When test cases are changed the result becomes inconsistent with functionality of the software.

A test plan is simply a high level summary of the areas (functionality, elements, regions etc) to be tested, how often these areas are tested, and where in the development or publication process one will test them. A test plan also states the duration testing and list of required resources.

The purpose of software and the constraints under which it has been developed should be understood. The software should illustrate all the characteristics that were initially visualized before its creation and should follow the hoped-for “path” for its success.

### **6.4 Testing during Design:**

It is very important that design document be tested and reviewed in order to develop a clear picture of how the system will work. Following issues were reviewed during design phase:

- Is the design healthy?
- Does the design meet the requirements?
- Is the design complete?
- Is the design implement able?
- How well the design handles error handling?

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

### **6.5 Testing during Coding**

The testing strategy that would be used is that first the software as a whole will be tested against the specification to discover the “faults of omission”, indicating the part of specification that has not been fulfilled. Then the software would be tested against the implementation to discover “faults of commission”, indicating that part of implementation that is faulty.

Some programmers do it as they code, and others wait until the end. Either way, testing is a necessary part of any software development project. Without it, one cannot determine that the software functions correctly.

### 6.6 Testing Of “*Improved algorithms for topic distillation using SelHITS*”:

The testing of “Improved algorithms for topic distillation using SelHITS” is undergone through all stages of black box testing and to extent white box testing. The system is reviewed to see whether the objectives of the system are accomplished or not. A major factor considered during system evaluation is to evaluate the system with the perspective of queries entered by users.

The sample tests performed on our project “Improved algorithms for topic distillation using SelHITS” are performed on the following queries:

- 1) Mouse
- 2) Windows

## 6.6.1 Mouse

Table 6.1: Top Hub and Authority for Med algorithm

Hub	Authority
uri	uri
<a href="http://io.wikipedia.org/wiki/Mapa_Mouse_mouse.jpg">http://io.wikipedia.org/wiki/Mapa_Mouse_mouse.jpg</a>	<a href="http://en.wikipedia.org/wiki/Mouse">http://en.wikipedia.org/wiki/Mouse</a>
<a href="http://en.wikipedia.org/wiki/Mouse">http://en.wikipedia.org/wiki/Mouse</a>	<a href="http://www.nih.gov/science/models/mouse/">http://www.nih.gov/science/models/mouse/</a>
<a href="http://lv.wikipedia.org/wiki/Peles">http://lv.wikipedia.org/wiki/Peles</a>	<a href="http://io.wikipedia.org/wiki/Muso">http://io.wikipedia.org/wiki/Muso</a>
<a href="http://mouseblast.informatics.jax.org/">http://mouseblast.informatics.jax.org/</a>	<a href="http://www.mgu.har.mrc.ac.uk/">http://www.mgu.har.mrc.ac.uk/</a>
<a href="http://simple.wikipedia.org/wiki/Mouse">http://simple.wikipedia.org/wiki/Mouse</a>	<a href="http://lv.wikipedia.org/wiki/Peles">http://lv.wikipedia.org/wiki/Peles</a>
<a href="http://da.wikipedia.org/wiki/Mus">http://da.wikipedia.org/wiki/Mus</a>	<a href="http://simple.wikipedia.org/wiki/Mouse">http://simple.wikipedia.org/wiki/Mouse</a>
<a href="http://www.informatics.jax.org/reports/homologymap/mouse_human.shtml">http://www.informatics.jax.org/reports/homologymap/mouse_human.shtml</a>	<a href="http://da.wikipedia.org/wiki/Mus">http://da.wikipedia.org/wiki/Mus</a>
<a href="http://www.informatics.jax.org/mgihome/other/citation.shtml">http://www.informatics.jax.org/mgihome/other/citation.shtml</a>	<a href="http://www.informatics.jax.org/orthology.shtml">http://www.informatics.jax.org/orthology.shtml</a>
<a href="http://www.informatics.jax.org/mgihome/other/copyright.shtml">http://www.informatics.jax.org/mgihome/other/copyright.shtml</a>	<a href="http://www.informatics.jax.org/mgihome/lists/lists.shtml">http://www.informatics.jax.org/mgihome/lists/lists.shtml</a>
<a href="http://www.informatics.jax.org/mgihome/other/link_instructions.shtml">http://www.informatics.jax.org/mgihome/other/link_instructions.shtml</a>	<a href="http://www.informatics.jax.org/mgihome/other/citation.shtml">http://www.informatics.jax.org/mgihome/other/citation.shtml</a>
<a href="http://www.informatics.jax.org/mgihome/other/mouse_facts1.shtml">http://www.informatics.jax.org/mgihome/other/mouse_facts1.shtml</a>	<a href="http://www.informatics.jax.org/mgihome/other/copyright.shtml">http://www.informatics.jax.org/mgihome/other/copyright.shtml</a>
<a href="http://www.informatics.jax.org/mgihome/lists/lists.shtml">http://www.informatics.jax.org/mgihome/lists/lists.shtml</a>	<a href="http://www.informatics.jax.org/mgihome/other/link_instructions.shtml">http://www.informatics.jax.org/mgihome/other/link_instructions.shtml</a>
<a href="http://www.informatics.jax.org/orthology.shtml">http://www.informatics.jax.org/orthology.shtml</a>	<a href="http://www.informatics.jax.org/mgihome/other/mgi_funding.shtml">http://www.informatics.jax.org/mgihome/other/mgi_funding.shtml</a>
<a href="http://www.informatics.jax.org/genes.shtml">http://www.informatics.jax.org/genes.shtml</a>	<a href="http://www.informatics.jax.org/mgihome/homepages/browser_compatibility.shtml">http://www.informatics.jax.org/mgihome/homepages/browser_compatibility.shtml</a>
<a href="http://www.informatics.jax.org/reports/homologymap/mouse_rat.shtml">http://www.informatics.jax.org/reports/homologymap/mouse_rat.shtml</a>	<a href="http://www.informatics.jax.org/mgihome/support/tjl_inbox.shtml">http://www.informatics.jax.org/mgihome/support/tjl_inbox.shtml</a>
<a href="http://www.informatics.jax.org/mgihome/support/tjl_inbox.shtml">http://www.informatics.jax.org/mgihome/support/tjl_inbox.shtml</a>	<a href="http://www.informatics.jax.org/genes.shtml">http://www.informatics.jax.org/genes.shtml</a>
<a href="http://www.informatics.jax.org/mgihome/homepages/browser_compatibility.shtml">http://www.informatics.jax.org/mgihome/homepages/browser_compatibility.shtml</a>	<a href="http://www.informatics.jax.org/mgihome/other/mouse_facts4.shtml">http://www.informatics.jax.org/mgihome/other/mouse_facts4.shtml</a>
<a href="http://www.informatics.jax.org/reports/mitmap/">http://www.informatics.jax.org/reports/mitmap/</a>	<a href="http://www.informatics.jax.org/mgihome/GXD/GEN/">http://www.informatics.jax.org/mgihome/GXD/GEN/</a>
<a href="http://www.informatics.jax.org/mgihome/homepages/browser_compatibility.shtml">http://www.informatics.jax.org/mgihome/homepages/browser_compatibility.shtml</a>	<a href="http://www.informatics.jax.org/reports/homologymap/mouse_human.shtml">http://www.informatics.jax.org/reports/homologymap/mouse_human.shtml</a>
<a href="http://www.informatics.jax.org/reports/mitmap/">http://www.informatics.jax.org/reports/mitmap/</a>	<a href="http://www.informatics.jax.org/imstr/index.jsp">http://www.informatics.jax.org/imstr/index.jsp</a>
<a href="http://www.informatics.jax.org/mgihome/homepages/browser_compatibility.shtml">http://www.informatics.jax.org/mgihome/homepages/browser_compatibility.shtml</a>	<a href="http://www.informatics.jax.org/mgihome/other/web_service.shtml">http://www.informatics.jax.org/mgihome/other/web_service.shtml</a>
<a href="http://www.informatics.jax.org/mgihome/homepages/browser_compatibility.shtml">http://www.informatics.jax.org/mgihome/homepages/browser_compatibility.shtml</a>	<a href="http://www.informatics.jax.org/function.shtml">http://www.informatics.jax.org/function.shtml</a>
<a href="http://www.informatics.jax.org/mgihome/homepages/browser_compatibility.shtml">http://www.informatics.jax.org/mgihome/homepages/browser_compatibility.shtml</a>	<a href="http://www.informatics.jax.org/external/ko/">http://www.informatics.jax.org/external/ko/</a>

The table 6.1 shows the top hub and authority URL's against the user query "Mouse" by using Med algorithm. These top hub and authority pages contains majority of the URL's related to the single aspect of "Mouse" query.

Table 6.2: Top Hub and Authority for Max by 10 algorithm

Hub	Authority
url	url
<a href="http://en.wikipedia.org/wiki/Mouse">http://en.wikipedia.org/wiki/Mouse</a>	<a href="http://en.wikipedia.org/wiki/Mouse">http://en.wikipedia.org/wiki/Mouse</a>
<a href="http://commons.wikimedia.org/wiki/Mus">http://commons.wikimedia.org/wiki/Mus</a>	<a href="http://en.wikipedia.org/wiki/Mouse">http://en.wikipedia.org/wiki/Mouse</a>
<a href="http://su.wikipedia.org/wiki/Beurit">http://su.wikipedia.org/wiki/Beurit</a>	<a href="http://da.wikipedia.org/wiki/Mus">http://da.wikipedia.org/wiki/Mus</a>
<a href="http://lv.wikipedia.org/wiki/Peles">http://lv.wikipedia.org/wiki/Peles</a>	<a href="http://gl.wikipedia.org/wiki/Rato">http://gl.wikipedia.org/wiki/Rato</a>
<a href="http://nah.wikipedia.org/wiki/Quimichin">http://nah.wikipedia.org/wiki/Quimichin</a>	<a href="http://io.wikipedia.org/wiki/Muso">http://io.wikipedia.org/wiki/Muso</a>
<a href="http://af.wikipedia.org/wiki/Muis">http://af.wikipedia.org/wiki/Muis</a>	<a href="http://lv.wikipedia.org/wiki/Peles">http://lv.wikipedia.org/wiki/Peles</a>
<a href="http://simple.wikipedia.org/wiki/Mouse">http://simple.wikipedia.org/wiki/Mouse</a>	<a href="http://nah.wikipedia.org/wiki/Quimichin">http://nah.wikipedia.org/wiki/Quimichin</a>
<a href="http://fo.wikipedia.org/wiki/Muso">http://fo.wikipedia.org/wiki/Muso</a>	<a href="http://simple.wikipedia.org/wiki/Mouse">http://simple.wikipedia.org/wiki/Mouse</a>
<a href="http://gl.wikipedia.org/wiki/Rato">http://gl.wikipedia.org/wiki/Rato</a>	<a href="http://su.wikipedia.org/wiki/Beurit">http://su.wikipedia.org/wiki/Beurit</a>
<a href="http://da.wikipedia.org/wiki/Mus">http://da.wikipedia.org/wiki/Mus</a>	<a href="http://af.wikipedia.org/wiki/Muis">http://af.wikipedia.org/wiki/Muis</a>
<a href="http://www.informatics.jax.org/mgihome/other/mouse">http://www.informatics.jax.org/mgihome/other/mouse</a>	<a href="http://www.informatics.jax.org/mgihome/other/copyright.shtml">http://www.informatics.jax.org/mgihome/other/copyright.shtml</a>
<a href="http://mouseblast.informatics.jax.org/">http://mouseblast.informatics.jax.org/</a>	<a href="http://www.informatics.jax.org/mgihome/other/link_instructions.shtml">http://www.informatics.jax.org/mgihome/other/link_instructions.shtml</a>
<a href="http://www.informatics.jax.org/mgihome/other/web_s">http://www.informatics.jax.org/mgihome/other/web_s</a>	<a href="http://www.informatics.jax.org/mgihome/other/mouse_facts1.shtml">http://www.informatics.jax.org/mgihome/other/mouse_facts1.shtml</a>
<a href="http://www.informatics.jax.org/mgihome/other/link_ir">http://www.informatics.jax.org/mgihome/other/link_ir</a>	<a href="http://www.informatics.jax.org/mgihome/support/tj_inbox.shtml">http://www.informatics.jax.org/mgihome/support/tj_inbox.shtml</a>
<a href="http://www.informatics.jax.org/mgihome/other/mgi_fu">http://www.informatics.jax.org/mgihome/other/mgi_fu</a>	<a href="http://www.informatics.jax.org/orthology.shtml">http://www.informatics.jax.org/orthology.shtml</a>
<a href="http://www.informatics.jax.org/mgihome/other/mouse">http://www.informatics.jax.org/mgihome/other/mouse</a>	<a href="http://www.informatics.jax.org/reports/homologymap/mouse_human.shln">http://www.informatics.jax.org/reports/homologymap/mouse_human.shln</a>
<a href="http://www.informatics.jax.org/mgihome/support/tj_i">http://www.informatics.jax.org/mgihome/support/tj_i</a>	<a href="http://www.informatics.jax.org/reports/homologymap/mouse_rat.shtml">http://www.informatics.jax.org/reports/homologymap/mouse_rat.shtml</a>
<a href="http://www.informatics.jax.org/orthology.shtml">http://www.informatics.jax.org/orthology.shtml</a>	<a href="http://www.informatics.jax.org/mgihome/other/citation.shtml">http://www.informatics.jax.org/mgihome/other/citation.shtml</a>
<a href="http://www.informatics.jax.org/reports/homologymap">http://www.informatics.jax.org/reports/homologymap</a>	<a href="http://www.informatics.jax.org/reports/snpSummary.shtml">http://www.informatics.jax.org/reports/snpSummary.shtml</a>
<a href="http://www.informatics.iax.org/maihome/other/coovri">http://www.informatics.iax.org/maihome/other/coovri</a>	<a href="http://www.informatics.jax.org/genes.shtml">http://www.informatics.jax.org/genes.shtml</a>

The table 6.2 shows the top hub and authority URL's against the user query "Mouse" by using Max by 10 algorithms. These top hub and authority pages contains majority of the URL's related to the single aspect of "Mouse" query.



**Table 6.3: Top Hub and Authority for Start Med algorithm**

Hub	Authority
url	url
<a href="http://www.genome.gov/10001859">http://www.genome.gov/10001859</a>	<a href="http://www.informatics.jax.org/">http://www.informatics.jax.org/</a>
<a href="http://www.informatics.jax.org/">http://www.informatics.jax.org/</a>	<a href="http://www.eucomm.org/">http://www.eucomm.org/</a>
<a href="http://www.informatics.jax.org/mgihome/nomen/">http://www.informatics.jax.org/mgihome/nomen/</a>	<a href="http://phenome.jax.org/pub-cgi/phenome/mpdcgi">http://phenome.jax.org/pub-cgi/phenome/mpdcgi</a>

The table 6.3 shows the top hub and authority URL's against the user query "Mouse" by using Start Med algorithms. These top hub and authority pages contains majority of the URL's related to the single aspect of "Mouse" query.

## 6.6.2 Windows

Table 6.4: Top Hub and Authority for med algorithm

Hub	Authority
url	url
<a href="http://tr.wikipedia.org/wiki/Microsoft_Windows">http://tr.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://es.wikipedia.org/wiki/Microsoft_Windows">http://es.wikipedia.org/wiki/Microsoft_Windows</a>
<a href="http://tl.wikipedia.org/wiki/Microsoft_Windows">http://tl.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://et.wikipedia.org/wiki/Microsoft_Windows">http://et.wikipedia.org/wiki/Microsoft_Windows</a>
<a href="http://vi.wikipedia.org/wiki/Microsoft_Windows">http://vi.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://fr.wikipedia.org/wiki/Microsoft_Windows">http://fr.wikipedia.org/wiki/Microsoft_Windows</a>
<a href="http://fi.wikipedia.org/wiki/Microsoft_Windows">http://fi.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://it.wikipedia.org/wiki/Microsoft_Windows">http://it.wikipedia.org/wiki/Microsoft_Windows</a>
<a href="http://uk.wikipedia.org/wiki/Microsoft_Windows">http://uk.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://hsb.wikipedia.org/wiki/Windows">http://hsb.wikipedia.org/wiki/Windows</a>
<a href="http://sv.wikipedia.org/wiki/Microsoft_Windows">http://sv.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://fr.wikipedia.org/wiki/Microsoft_Windows">http://fr.wikipedia.org/wiki/Microsoft_Windows</a>
<a href="http://zh.wikipedia.org/wiki/Microsoft_Windows">http://zh.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://he.wikipedia.org/wiki/Microsoft_Windows">http://he.wikipedia.org/wiki/Microsoft_Windows</a>
<a href="http://sl.wikipedia.org/wiki/Microsoft_Windows">http://sl.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://www.windowstpro.com">http://www.windowstpro.com</a>
<a href="http://simple.wikipedia.org/wiki/Microsoft_Windows">http://simple.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://www.winexcavator.com/">http://www.winexcavator.com/</a>
<a href="http://ca.wikipedia.org/wiki/Microsoft_Windows">http://ca.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://www.windowstlibrary.com/">http://www.windowstlibrary.com/</a>
<a href="http://bs.wikipedia.org/wiki/Microsoft_Windows">http://bs.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://bs.wikipedia.org/wiki/Microsoft_Windows">http://bs.wikipedia.org/wiki/Microsoft_Windows</a>
<a href="http://bg.wikipedia.org/wiki/Microsoft_Windows">http://bg.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://windowsdevpro.com/">http://windowsdevpro.com/</a>
<a href="http://ceb.wikipedia.org/wiki/Microsoft_Windows">http://ceb.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://www.microsoft.com/windowsxp/default.asp">http://www.microsoft.com/windowsxp/default.asp</a>
<a href="http://ms.wikipedia.org/wiki/Microsoft_Windows">http://ms.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://windowsitpro.com/windowsnt20002003faq/">http://windowsitpro.com/windowsnt20002003faq/</a>
<a href="http://ro.wikipedia.org/wiki/Microsoft_Windows">http://ro.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://www.microsoft.com/windows2000/default.asp">http://www.microsoft.com/windows2000/default.asp</a>
<a href="http://ru.wikipedia.org/wiki/Microsoft_Windows">http://ru.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://www.opensourcewindows.org">http://www.opensourcewindows.org</a>
<a href="http://lt.wikipedia.org/wiki/Microsoft_Windows">http://lt.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://www.microsoft.com/Windows/default.mspx">http://www.microsoft.com/Windows/default.mspx</a>
<a href="http://hu.wikipedia.org/wiki/Microsoft_Windows">http://hu.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://www.connectedhomemedia.com/">http://www.connectedhomemedia.com/</a>
<a href="http://pt.wikipedia.org/wiki/Microsoft_Windows">http://pt.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://eu.wikipedia.org/wiki/Microsoft_Windows">http://eu.wikipedia.org/wiki/Microsoft_Windows</a>
<a href="http://eu.wikipedia.org/wiki/Microsoft_Windows">http://eu.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://www.wininformant.com">http://www.wininformant.com</a>

The table 6.4 shows the top hub and authority URL's against the user query "Windows" by using Start Med algorithms. These top hub and authority pages contains majority of the URL's related to the single aspect of "Windows" query.

Table 6.5: Top Hub and Authority for Max by 10 algorithm

Hub	Authority
url	url
<a href="http://windowsitpro.com/windowsnt20002003faq/">http://windowsitpro.com/windowsnt20002003faq/</a>	<a href="http://www.microsoft.com/windows">http://www.microsoft.com/windows</a>
<a href="http://community.winsupersite.com/blogs/itprotips/archive/">http://community.winsupersite.com/blogs/itprotips/archive/</a>	<a href="http://www.microsoft.com/windows/products/winfamily/virtualpc/default.msx">http://www.microsoft.com/windows/products/winfamily/virtualpc/default.msx</a>
<a href="http://community.winsupersite.com/blogs/paul/archive/200">http://community.winsupersite.com/blogs/paul/archive/200</a>	<a href="http://shop.internet.com/">http://shop.internet.com/</a>
<a href="http://community.winsupersite.com/blogs/paul/archive/200">http://community.winsupersite.com/blogs/paul/archive/200</a>	<a href="http://www.windowsitpro.com">http://www.windowsitpro.com</a>
<a href="http://windowsitpro.com/article/articleid/85057/jsi-tip-10082">http://windowsitpro.com/article/articleid/85057/jsi-tip-10082</a>	<a href="http://www.winexcavator.com/">http://www.winexcavator.com/</a>
<a href="http://windowsitpro.com/Windows/article/articleid/95024/wi">http://windowsitpro.com/Windows/article/articleid/95024/wi</a>	<a href="http://www.internet.com">http://www.internet.com</a>
<a href="http://windowsitpro.com/article/articleid/98780/opera-927-s">http://windowsitpro.com/article/articleid/98780/opera-927-s</a>	<a href="http://www.windowslibrary.com/">http://www.windowslibrary.com/</a>
<a href="http://windowsitpro.com/article/articleid/94380/where-in-the">http://windowsitpro.com/article/articleid/94380/where-in-the</a>	<a href="http://windowsdevpro.com/">http://windowsdevpro.com/</a>
<a href="http://windowsitpro.com/article/articleid/93915/availability-a">http://windowsitpro.com/article/articleid/93915/availability-a</a>	<a href="http://www.winsupersite.com/">http://www.winsupersite.com/</a>
<a href="http://windowsitpro.com/article/articleid/95862/exchange-e">http://windowsitpro.com/article/articleid/95862/exchange-e</a>	<a href="http://windowsvistablog.com/">http://windowsvistablog.com/</a>
<a href="http://windowsitpro.com/article/articleid/41546/how-can-i-ea">http://windowsitpro.com/article/articleid/41546/how-can-i-ea</a>	<a href="http://windowsvistablog.com/blogs/windowsvista/archive/2007/01/23/secureit">http://windowsvistablog.com/blogs/windowsvista/archive/2007/01/23/secureit</a>
<a href="http://windowsitpro.com/article/articleid/23057/does-windov">http://windowsitpro.com/article/articleid/23057/does-windov</a>	<a href="http://www.webvideouniverse.com/">http://www.webvideouniverse.com/</a>
<a href="http://windowsitpro.com/article/articleid/15557/how-can-i-c">http://windowsitpro.com/article/articleid/15557/how-can-i-c</a>	<a href="http://technet.microsoft.com/en-us/windowsvista/aa906021.aspx">http://technet.microsoft.com/en-us/windowsvista/aa906021.aspx</a>
<a href="http://windowsitpro.com/article/articleid/14006/can-nt-act-a">http://windowsitpro.com/article/articleid/14006/can-nt-act-a</a>	<a href="http://www.connectedhomemedia.com/">http://www.connectedhomemedia.com/</a>
<a href="http://www.connectedhomemedia.com/">http://www.connectedhomemedia.com/</a>	<a href="http://www.wininformant.com">http://www.wininformant.com</a>
<a href="http://windowsitpro.com/article/articleid/93959/zero-day-vul">http://windowsitpro.com/article/articleid/93959/zero-day-vul</a>	<a href="http://www.windrivers.com/faq.asp">http://www.windrivers.com/faq.asp</a>
<a href="http://windowsitpro.com/article/articleid/94826/preventing-d">http://windowsitpro.com/article/articleid/94826/preventing-d</a>	<a href="http://www.windrivers.com/benefits.asp">http://www.windrivers.com/benefits.asp</a>
<a href="http://windowsitpro.com/article/articleid/49499/customizing">http://windowsitpro.com/article/articleid/49499/customizing</a>	<a href="http://www.windrivers.com/beginner/index.htm">http://www.windrivers.com/beginner/index.htm</a>
<a href="http://windowsitpro.com/article/articleid/49289/neon-lansur">http://windowsitpro.com/article/articleid/49289/neon-lansur</a>	<a href="http://www.windrivers.com/">http://www.windrivers.com/</a>
<a href="http://community.winsupersite.com/blogs/itrotips/archive/">http://community.winsupersite.com/blogs/itrotips/archive/</a>	<a href="http://technet.microsoft.com/en-us/updatesmanagement/default.aspx">http://technet.microsoft.com/en-us/updatesmanagement/default.aspx</a>

The table 6.5 shows the top hub and authority URL's against the user query "Windows" by using Start Med algorithms. These top hub and authority pages contains majority of the URL's related to the single aspect of "Windows" query.

Hub	Authority
url	url
<a href="http://en.wikipedia.org/wiki/Microsoft_Windows">http://en.wikipedia.org/wiki/Microsoft_Windows</a>	<a href="http://www.winusersite.com/">http://www.winusersite.com/</a>
<a href="http://windowsitpro.com/windowsnt20002003faq/">http://windowsitpro.com/windowsnt20002003faq/</a>	<a href="http://www.levenez.com/windows/">http://www.levenez.com/windows/</a>
<a href="http://www.levenez.com/windows/">http://www.levenez.com/windows/</a>	

**Table 6.6: Top Hub and Authority for Start Med algorithm**

The table 6.6 shows the top hub and authority URL's against the user query "Windows" by using Start Med algorithms. These top hub and authority pages contains majority of the URL's related to the single aspect of "Windows" query.

## 6.7 Analysis of Results

From the obtained results, we have successfully achieved improved results. The analysis of results show that in case we blindly expand the root set of content and connectivity based algorithms, we get large amount of irrelevant pages .The majority of the pages obtained donot fulfill user needs.

In the implemented system we have selectively expand the root set, the selective expansion helps us in giving most appropriate and relevant pages.The selective expansion help us in solving the problem of topic drift in broad queries.

The first query that we select to test our system is Mouse. There are two interpretations for this broad query one is, animal mouse and other is, computer mouse. Our system has improved the results by giving top hub and top authorities as compare to previous algorithms, related to one aspect of mouse i.e animal mouse and remove the problem of topic contamination and topic drift to some extent than the previous ones. By running the same query using previous algorithm one will get the top hub and top authority related to different interpretations of the same query. Some top hubs and authorities are animal mouse and some are of computer mouse. The one who is firing the query 'Mouse'; it has more probability that he requires the informative pages related animal mouse. The existing search engine algorithms mostly consider it as computer mouse, which is not the Required interpretation for this query as the device use by computer users is basically computer mouse.

The achieved results have reduced the time consumption because we have selectively expanding the base set. When we have blindly expanded the root set, the base set drastically increased to 7000 pages, majority of which are irrelevant pages and give rise to the problem of topic contamination and topic drift.

Similar is the case , when we test 'Windows' and 'Gates' query.

7

*Conclusion and Future Works*

---

---

---

## **CONCLUSION AND FUTURE WORKS**

### **7.1 Conclusion**

In this thesis we have blended the “SELHIT” algorithm by Mitra et al[7] with three content and connectivity based algorithms by Bharat et al [11]:

- 1) Med
- 2) Startmed
- 3) Maxby10

The implemented system is successful in giving the improved desired results.

Previously the blind “One Link Expansion” procedure to get base set drastically increases size of the base set. Most of the pages added are useless and including them in the base set causes the extra time consumption, as the content analysis is also performed on useless pages as well as useful pages. By the blending of SelHITS with content and connectivity based algorithm we have distilled pure topic related to query to some extent as shown from our results. Selective expansion has also reduced extra time consumption 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.

### **7.2 Future work**

- 1) In future we want to design a search engine that is based on the improved algorithms and SelHITS. We hope that the proposed search engine will give better results than existing search engines.
- 2) We also want to further improve these connectivity and content based algorithms by expanding 2-neighbour hood. This is helpful in getting more number of relevant pages according to query.
- 3) We want to improve the remaining six algorithms by Bharat et.al [1] by applying SelHITS as we have done in these algorithms.

# *Appendix-A*

## *List of Abbreviations*



---

## Appendix A

### DEFINITION OF TERMS

<b>Abbreviations</b>	<b>Full Form</b>
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

---

---

# *Appendix-B*

*Screen Shots*

# Appendix B

## SCREEN SHOTS

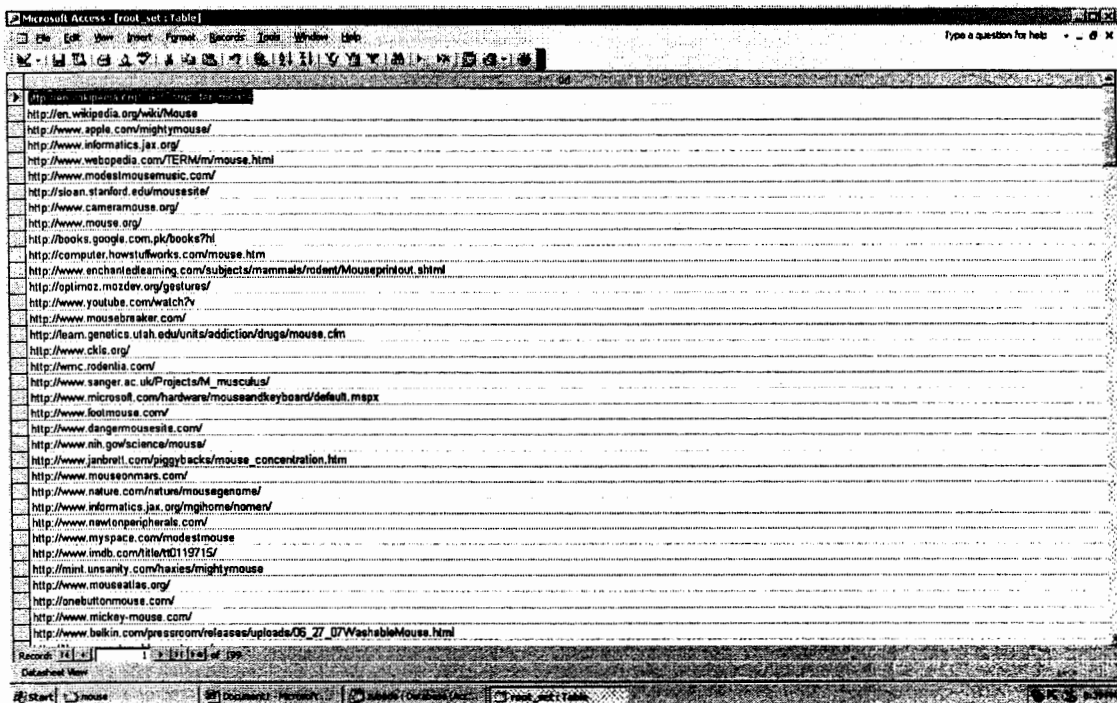


Fig B-1: Root Set

Microsoft Access - [root\_hub - Table]

URL	Count
http://www.informatics.jax.org/mghome/homen/	10
http://www.5mra.org/	10
http://www.nature.com/nature/mousegenomel	9
http://www.bcgsc.ca/platform/mapping/mouse	8
http://www.tigr.org/db/bac_ends/mouse/bac_end_int	6
http://www.informatics.jax.org/	6
http://www.myspace.com/modestmouse	5
http://management.sarthweb.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.evokent.com/vm3.html	1
http://www.animax.na/	1
http://tech.ln.lib.mi.us/tutor/welcome.htm	1
http://optimoz.mozdev.org/gestures/installation.html	1
http://optimoz.mozdev.org/gestures/	1
http://nono.learnrubber.com/rubber_client_work/mou	1
http://www.montrossescam.com/	1
http://en.wikipedia.org/wiki/Mouse	1
http://www.emmanet.org/	1

Records: 34 of 34  
 Deleted View  
 Start mouse Document1 - Microsoft Access View - JMS

Fig B-2: Root\_Hub

Microsoft Access - [root\_auth - Table]

URL	Count
http://en.wikipedia.org/wiki/Mouse	6
http://phenome.jax.org/pub-cgi/phenome/mpcdgi	4
http://www.sucomm.org/	2
http://www.sanger.ac.uk/Projects/M_musculus/	2
http://www.informatics.jax.org/mghome/homen/	2
http://www.mic.eastnets.com/	1
http://www.modestmousemusic.com/	1
http://computer.howstuffworks.com/mouse.htm	1
http://optimoz.mozdev.org/gestures/	1
http://www.ckts.org/	1
http://www.newtonperipherals.com/	1
http://mousesng.rocha.com/	1
http://www.5mra.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/db/bac_ends/mouse/bac_end_intro.h	1
http://www.sri.com/about/timeline/mouse.html	1
http://solutions.3m.com/wps/portal/3M/en_US/ergonomic	1
http://www.emmanet.org/	1
http://optimoz.mozdev.org/gestures/installation.html	1
http://www.highrez.co.uk/downloads/9MouseButtonCont	1
http://www.mousemailer.org/	1
http://www.hgsc.bcm.tmc.edu/projects/mouse/	1
http://domino.research.ibm.com/comm/pr.nsf/pages/new	1
http://www.nervenet.org/main/dictionary.html	1

Records: 26 of 27  
 Deleted View  
 Start mouse Document1 - Microsoft Access View - JMS

Fig B-3: Root\_Auth

id	url
10	http://www.wininformant.com
100	http://community.winsuperite.com/blogs/paul/default.aspx
1000	http://msdn.microsoft.com/library/default.asp?url=
10000	http://www.utdallas.edu/r/security/ViewNews.htm
10001	http://www.us-cert.gov/cas/alerts/SADS-165A.html
10002	http://www.microsoft.com/technet/security/bulletin/ms01-021.mspx
10003	http://www.geocities.com/Tokyo/Market/2376/index.html
10004	http://www.sjsci.edu.cn/
10005	http://blogs.msdn.com/gdthie/archives/2007/11.aspx
10006	http://www.spamfighter.com/sem_email_server_spam_filter.asp
10007	http://www.nbt.com/webapp/security/news/supportnews_detail.jsp?newsid
10008	http://us.trendmicro.com/us/threats/home-usa/preventing-intrusions/safe-computing_guide/windows_server_2003/
10009	http://www.microsoft.com/technet/security/bulletin/ms01-051.mspx
101	http://msdn.microsoft.com/library/default.asp?url=
1010	http://www.auburn.edu/ok/hardware_software/os/common_topics.php
10101	http://www.ctinel.nu/support.asp
10102	http://www.sab-online.com/Security/SecurityGuide.html
10103	http://www.microsoft.com/japan/security/bulletins/feb2006.mspx
10104	http://www.colby-sawyer.edu/about/web/site/help.html
10105	http://www.lsa.ac.uk/itservices/help/Windowsvulnerabilities.htm
10106	http://www.stardriver.org/wallpapersync/
10107	http://gameknot.com/faq.pl
10108	http://www.whoi.edu/cis/security/news/platform_security.html
10109	http://www.screenit.com/subscribers/win98winME.asp
102	http://support.microsoft.com/kb/896358
1020	http://hr123.any2000.com/
10201	http://learning.bankofshanghai.com.cn/html/client_check/zh-cn/index.htm
10202	http://ctaar.rutgers.edu/workshops/
10203	http://www.microsoft.com/spain/athome/security/protect/windowsmeupdates.mspx
10204	http://www.microsoft.com/france/windows/xp/sp2/technologiesoverview.mspx
10205	http://www.microsoft.com/middleeast/
10206	http://www.microsoft.com/technet/security/bulletin/ms01-043.mspx
10207	http://www.spamfighter.com/lang_n/product_sem.asp
10208	http://judi.kuleuven.be/software/bewilliging/windowsupdates.html
10209	http://www.cis.kokushikan.ac.jp/

Fig B-4: Base\_Set



The screenshot shows a Microsoft Access database window titled 'frac\_auth - Microsoft Access'. The table 'authority' is displayed in Datasheet View. The table contains 25 rows of data, each with a URL in the first column and an authority value in the second column. The authority values range from 4 to 12. The table is sorted by authority in descending order.

URL	Authority
http://www.jax.org/	12
http://en.wikipedia.org/wiki/Mouse	11
http://da.wikipedia.org/wiki/Mus	9
http://gl.wikipedia.org/wiki/Rato	9
http://io.wikipedia.org/wiki/Muso	9
http://lv.wikipedia.org/wiki/Peles	9
http://nah.wikipedia.org/wiki/Quimichin	9
http://simple.wikipedia.org/wiki/Mouse	9
http://su.wikipedia.org/wiki/Beurit	9
http://af.wikipedia.org/wiki/Muis	9
http://www.informatics.jax.org/mgi/home/other/copyright.shtml	7
http://www.informatics.jax.org/mgi/home/other/link_instructions.shtml	7
http://www.informatics.jax.org/mgi/home/other/mouse_facts1.shtml	7
http://www.informatics.jax.org/mgi/home/support/tjt_inbox.shtml	7
http://www.informatics.jax.org/orthology.shtml	7
http://www.informatics.jax.org/reports/homologymap/mouse_human.shtml	7
http://www.informatics.jax.org/reports/homologymap/mouse_rat.shtml	7
http://www.informatics.jax.org/mgi/home/other/citation.shtml	7
http://www.informatics.jax.org/reports/snpSummary.shtml	7
http://www.informatics.jax.org/genes.shtml	7
http://www.informatics.jax.org/reports/mitomap/	7
http://www.informatics.jax.org/mgi/home/lists/lstst.shtml	7
http://www.informatics.jax.org/mgi/home/homepages/browser_compatibility.shtml	7
http://www.informatics.jax.org/mgi/home/GXD/GEN/	7
http://www.informatics.jax.org/resources.shtml	7
http://www.informatics.jax.org/mnr/index.jsp	7

Record: 1 of 82 | No Filter | Search

Fig B-7: Max by 10\_auth

	url	
1	http://www.windowsmail.com	
1001	http://www.zc-net.gov.tw/seris/SADS-165A.html	
1004	http://www.gact.wa.civ	
1005	http://blogs.msdn.com/gdthia/archive/2007/11.aspx	
1008	http://www.trendmicro.com/usa/usa/home_www/pressing_situations/6-computing_guide/windows_serve_2007	
1010	http://www.adam.roberts.harvard.edu/roberts/uk/summer_topics.php	
1013	http://www.microsoft.com/press/secure/yuldes/04/2004_msp3	
1015	http://www.iss.ac.uk/iss/securehelp/HFidmsec/04/03/05.htm	
1016	http://www.what.edu/wh/secu/04/03/05.htm	
1019	http://www.screenit.com/subscribe/04/03/05.htm	
102	http://support.microsoft.com/kb/953528	
1026	http://www.microsoft.com/technet/security/yuldes/04/03/05_msp3	
1028	http://msd.lukeaven.be/04/03/05/windowsupdate.html	
1031	http://www.cdn.com/blog/	
1033	http://www.windowsnewsletter.com/html/04/03/05_msp3.htm	
1037	http://www.microsoft.com/usa/usa/windowsupdate/04/03/05.htm	
1047	http://windowsupdate.microsoft.com/	
105	http://forums.windowsupdate.com	
1051	http://www.level1.com/windows/	
1054	http://can1.project.org/04/03/05/windows/FAQ.html	
1057	http://www.milest.com/windows.html	
1059	http://ipgipdoc.mozdev.org/windows.html	
1061	http://en.wikipedia.org/wiki/Microsoft_Windows	
1062	http://windowsupdate.microsoft.com/	
1063	http://www.windowsupdate.com/	
11	http://www.windows.com/	
12	http://blogs.msdn.com/windowsupdateplace	
122	http://www.windows@bary.com/	
13	http://www.computerhope.com	
132	http://support.microsoft.com/default.asp?scid	
134	http://www.winsider.com/	
136	http://www.microsoft.com/windowsupdate/default.asp	
137	http://www.hwexp.com/hwexp/	
138	http://www.win95central.com/	
139	http://www.techwarlabs.com/04/03/05/windowsupdate-data/04/03/05.htm	

Fig B-8: Median

	url	hub
17	http://www.wiki-tracker.com/wiki-tracker/04/03/05/windowsupdate	
15	http://en.wikipedia.org/wiki/Mouse	
13	http://fr.wikipedia.org/wiki/Peles	
13	http://mouseblast.informatics.jax.org/	
13	http://simple.wikipedia.org/wiki/Mouse	
13	http://da.wikipedia.org/wiki/Mus	
11	http://www.informatics.jax.org/reports/homologymap/mouse_human.shtml	
11	http://www.informatics.jax.org/mgihome/other/citation.shtml	
11	http://www.informatics.jax.org/mgihome/other/copyright.shtml	
11	http://www.informatics.jax.org/mgihome/other/link_instructions.shtml	
11	http://www.informatics.jax.org/mgihome/other/mouse_facts1.shtml	
11	http://www.informatics.jax.org/mgihome/lists/lists.shtml	
11	http://www.informatics.jax.org/orthology.shtml	
11	http://www.informatics.jax.org/genes.shtml	
11	http://www.informatics.jax.org/reports/homologymap/mouse_rat.shtml	
11	http://www.informatics.jax.org/mgihome/support/t/ inbox.shtml	
11	http://www.informatics.jax.org/mgihome/homepages/browser_compatibility.shtml	
11	http://www.informatics.jax.org/reports/mitmap/	
11	http://www.informatics.jax.org/mgihome/genealogy/	
11	http://www.informatics.jax.org/mgihome/other/web_service.shtml	
11	http://www.informatics.jax.org/function.shtml	
11	http://www.informatics.jax.org/external/co/	
11	http://www.informatics.jax.org/external/feeding/search_form.cgi	
11	http://www.informatics.jax.org/external/feeding/mouse/docs/1.shtml	

Fig B-9: Med\_Hub



Microsoft Access - [med\_auth : Table]

url	authority
http://www.nih.gov/science/models/mouse/	5
http://www.nih.gov/science/models/mouse/	4
http://www.nih.gov/science/models/mouse/	4
http://www.mgu.har.mrc.ac.uk/	4
http://www.mgu.har.mrc.ac.uk/	4
http://www.mgu.har.mrc.ac.uk/	4
http://www.mgu.har.mrc.ac.uk/	4
http://www.mgu.har.mrc.ac.uk/	4
http://www.mgu.har.mrc.ac.uk/	4
http://www.informatics.jax.org/orthology.shtml	4
http://www.informatics.jax.org/mgihome/lists/lists.shtml	4
http://www.informatics.jax.org/mgihome/other/citation.shtml	3
http://www.informatics.jax.org/mgihome/other/copyright.shtml	3
http://www.informatics.jax.org/mgihome/other/link_instructions.shtml	3
http://www.informatics.jax.org/mgihome/other/mgi_funding.shtml	3
http://www.informatics.jax.org/mgihome/homepages/browser_compatibility.shtml	3
http://www.informatics.jax.org/mgihome/support/tij_inbox.shtml	3
http://www.informatics.jax.org/genes.shtml	3
http://www.informatics.jax.org/mgihome/other/mouse_facts4.shtml	3
http://www.informatics.jax.org/mgihome/GXD/GENE	3
http://www.informatics.jax.org/reports/homologymap/mouse_human.shtml	3
http://www.informatics.jax.org/imsr/index.jsp	3
http://www.informatics.jax.org/mgihome/other/web_service.shtml	3
http://www.informatics.jax.org/function.shtml	3
http://www.informatics.jax.org/external/ko/	3
http://www.informatics.jax.org/external/feeding/search_form.cgi	3

Record: 14 of 64

Datasheet View

Fig B-10:Med\_Auth

Microsoft Access - [start : Table]

id	url
321	http://www.informatics.jax.org/
324	http://www.informatics.jax.org/mgihome/nomen/
325	http://www.eucomm.org/
326	http://phenome.jax.org/pub-cgi/phenome/mpdcgi
332	http://www.genome.gov/10001859
334	http://www.informatics.jax.org/mgihome/nomen/
*	

Fig B-11:Start

start_hub : Table		
	url	hub
▶	http://www.genome.gov/10001859	2
	http://www.informatics.jax.org/	4
	http://www.informatics.jax.org/mgihome/nomen/	4
*		0

Fig B-12:Start\_hub

start_auth : Table		
	url	authority
▶	http://www.informatics.jax.org/	2
	http://www.eucomm.org/	2
	http://phenome.jax.org/pub-cgi/phenome/mpdcgi	2
*		0

Fig B-13:Start\_auth

## 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 Authoritative Sources in a Hyperlinked Environment Cornell University, Ithaca, New York 1997
- [7] Selective Hypertext Induced Topic Search Amit C. Awekar NC State University Raleigh, NC 27695, USA [acawekar@ncsu.edu](mailto:acawekar@ncsu.edu) Pabitra Mitra Indian Institute of Technology Kharagpur, India – 721302 [pabitra@cse.iitkgp.ernet.in](mailto:pabitra@cse.iitkgp.ernet.in) Jaewoo Kang NC State University Raleigh, NC 27695, USA [kang@csc.ncsu.edu](mailto: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] Bharata K., Brodera A., Henzinger M., Kumara P., and Suresh,(1998), The Connectivity Server: fast access to linkage information on the Web, Elsevier Science Publishers B. V,30, 469 – 477.

[12] Dingy C., Hey X., Husbandsy P., Zhaz H., Simony H.,(2001), PageRank, HITS and a Unified Framework for Link Analysis, ACM Press New York, 353 – 354.

