DESIGN OF INTEREST BASED SEARCH SYSTEM USING QUERY STRUCTURING

THESIS

submitted in fulfilment of the requirement of the degree of

DOCTOR OF PHILOSOPHY

to

YMCA UNIVERSITY OF SCIENCE & TECHNOLOGY

by

SHILPA SETHI

Registration No: YMCAUST/PH60/2011

Under the Supervision of

Dr. ASHUTOSH DIXIT

ASSOCIATE PROFESSOR

Department of Computer Engineering
Faculty of Engineering and Technology
YMCA University of Science &Technology
Sector-6, Mathura Road, Faridabad, Haryana, INDIA

August, 2018

DECLARATION

I hereby declare that this thesis entitled "**DESGIN OF INTEREST BASED SEARCH SYS**TEM **USING QUERY STRUCTURING"** by **SHILPA SETHI,** being submitted in fulfilment of the requirements for the Degree of Doctor of Philosophy in Department of Computer Engineering under Faculty of Engineering and Technology of YMCA University of Science & Technology, Faridabad, during the academic year 2017-2018, is a bona fide record of my original work carried out under the guidance and supervision of **Dr. ASHUTOSH DIXIT, ASSOCIATE PROFESSOR, DEPARTMENT OF COMPUTER ENGINEERING, YMCA UNIVERSITY OF SCIENCE & TECHNOLOGY, FARIDABAD** and has not been presented elsewhere.

I further declare that the thesis does not contain any part of any work which has been submitted for the award of any degree either in this university or in any other university.

(SHILPA SETHI)

Registration No: YMCAUST/PH60/2011

CERTIFICATE

This is to certify that this thesis entitled "**DESGIN OF INTEREST BASED SEARCH SYSTEM USING QUERY STRUCTURING"** by **SHILPA SETHI,** submitted in fulfillment of the requirements for the award of Degree of Doctor of Philosophy in Department of Computer Engineering, under Faculty of Engineering and Technology of YMCA University of Science and Technology, Faridabad, during the academic year 2017-18, is a bona fide record of work carried out under my guidance and supervision.

I further declare that to the best of my knowledge, the thesis does not contain any part of any work which has been submitted for the award of any degree either in this university or in any other university.

Dr. ASHUTOSH DIXIT
Associate Professor
Department of Computer Engineering
Faculty of Engineering and Technology
YMCA University of Science & Technology, Faridabad

Date:

ACKNOWLEDGEMENT

I express my gratitude to almighty God for giving me strength and courage to complete this thesis.

I would like to express my sincere and deep gratitude to my Ph.D. supervisor, **Dr. Ashutosh Dixit,** Associate Professor, Department of Computer Engineering, YMCA University of Science & Technology, Faridabad for giving me the opportunity to work in this area. It would never be possible for me to take this thesis to this level without his continuous guidance, innovative ideas, constructive criticism and valuable advice. His knowledge in various perspectives of research provided me the opportunity to make significant work in this direction. His continuous support and encouragement keep me sailing through tough times of my Ph.D. pursuit.

I am thankful to Dr. Atul Mishra, Chairman, Department of Information Technology & Computer Application and Dr. Komal Bhatia, Chairman, Department of Computer Engineering, YMCA University of Science and Technology for their constant encouragement and moral support during the course of this research work. I want to express my special thanks to Dr. Neelam Duhan for her motivation and support for doing research in better way. I would like to thank my colleagues Dr. Naresh Chauhan, Dr. C.K. Nagpal, Dr. Sapna Gambhir, Dr. Parul Tomar, Dr. Deepika, Ms. Mamta, Mr. Umesh and others for their cooperation.

A special thanks to my parents Late Sh. Rajender Sethi and Smt. Santosh Sethi, and my brother Sandeep for all their love and encouragement throughout my education. I am also thankful to my Father-in- law Sh. S.S. Munjal, my mother-in-law Smt. Usha Munjal, and my friends Ms. Sandhya and Ms. Amita Nahar for their blessings and support.

A word of special thanks to my dear husband Sh. Ashish Munjal and my son Master Saransh Munjal for the utmost patience, love and faithful support without which it would never be possible to finish this thesis.

Thank you all**!**

(Shilpa Sethi)

DEDICATION

to

My Son

Saransh Munjal, for his endless patience

ABSTRACT

World Wide Web is interlinked collection of HTML documents spread over millions of disjoint web sites and accessed via Internet. The common standards of Internet suits supported the exponential growth of web. In fact, it has expanded about 10^8 times since its inception. Search engines are the automated tools for retrieving the information from such a huge collection. They apply sophisticated retrieval techniques to help the user in finding require information from the web. But in spite of using sophisticated searching techniques, the result set returned by search engine in response to user query is flooded with magnitude of irrelevant documents due to many reasons: Web is huge, dynamic and expanding at staggering rate. The users of web are heterogeneous in the sense, that they possess varying interest in different areas. They often submit short, ambiguous and instant queries that are not enough to precisely interpret its information need. In order to retrieve the information available in large repositories of web as per user's need, the concept of web search personalization come into picture.

The process of web search personalization involves tracking the activities of individual/group of user(s) during the information seeking hours and utilizing them in providing the required information to the user. Although the modern search engines perform well in e-commerce domain for recommending products to their customers by application of web personalization but they are still lacking in general topic search. The web personalization in general topic search may be applied at the level of query submission or at the level of search result formation. This can be achieved by the application of web mining techniques at both levels in context of web search personalization. Several researches available in literature demonstrate the utility of web mining in the field of Query suggestion, Query expansion and Page ranking separately. But no unified system is reported that has applied techniques of web mining at various phases of search process i.e. query processing, ranking and indexing.

So thrust of the thesis is to design a unified framework of search system that can optimize the entire process of information searching, beginning from submission of the query till search result presentation, in accordance with user's information need.

Towards this goal, firstly a novel query suggestion technique based on user browsing patterns has been proposed. The technique suggests personalised queries to each user based on context and click through data. The contextually similar queries are identified using WordNet dictionary. The user interest is discovered by capturing the clicks made by user on web pages belonging to different domains. For this purpose, definition repository, query log and profile database is maintained using SQL server. The comparison is made between proposed query suggestion technique and conventional query suggestion method. With the help of implementation results, a significant reduction in search space is observed.

The next module of the thesis is motivated by utilising the user behaviour in ranking process. For this purpose, a customized browser is developed to capture the number of clicks, time spent and action performed by the user on a particular web page .In addition to this, the spatial properties of web document in N-dimensional vector space are utilized to find content relevancy of page with respect to query. The structural summary of page is also used in finding its relevancy within the web. The technique produces highly relevant result set as compared to conventional ranking mechanism.

The work has also put forward a solution to deal with dynamic nature of web. An arithmetic progression based crawling mechanism for providing up to date information to the user is developed. The technique computes the re-crawl interval for each page based on its change frequency. A significant improvement in freshness of downloaded collection is achieved without incurring extra load on network.

Further, bi-layer architecture of interest based search system using query structuring has been developed which includes a novel domain specific repository that helps in computation of degree of user's interest in different domains. The repository contains contextual, structural and usage information about each page. The system is tested for five domains viz. education, travelling & tourism, sports, food & beverages and fashion & shopping using Dmoz directory. The number of domains is easily extendible. The implementation results show a significant improvement in result relevancy as compared to conventional search system.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

WWW	World Wide Web
HTTP	Hypertext Transfer Protocol
URL	Uniform Resource Locator
BFS	Breadth First Search
DFS	Depth First Search
HITS	Hyperlink Induced Topic Search
WPR	Weighted Page Rank
PR	Page Rank
IR	Information Retrieval
DOC	Document
MIT	Massachusetts Institute of Technology
ISI	Information Science Institute
NCSA	National Center for Supercomputing application
SQL	Structured Query Language
Aliweb	Archie Like Indexing for Web
PRLV	Page Rank based on Link Visit
PPC	Page Probability Calculator
PPF	Page Probability Factor
SRS	Suggestion Relevance Score
APCM	Arithmetic Progression based Crawling Mechanism
RRS	Result Relevancy Score
PR-PPF	Page Ranking based on Page Probability Factor
DSCD	Domain Specific Classified Database

CHAPTER I

INTRODUCTION

1.1 GENEREAL

Internet [90] is network of networks that connects millions of computers globally through electronic or optical medium. Whereas *World Wide Web* (WWW) is a global information resource that contains trillions of documents interconnected by hyperlinks, identified by uniform resource locator and accessed via Internet. It is the most widely and frequently available largest repository used by millions of people to share information with others. With the help of web browser such as Internet Explorer, Firefox, Mozilla, one can easily view the web pages on its computer. A *web page* is primarily a text document formatted and annotated by hypertext markup language. It may contain graphics, audio, video and even hyper links to other web pages. Multiple web pages with a common domain name are stored under a web site. Statistics of authoritative web sites show that from one website in 1991, it is expected to have more than one billion of websites at the end of 2017-2018 [4].

In order to view the web pages on World Wide Web, two well - established models are used: *browsing* and *searching*. A user browses the web to look around with open mind to discover something new. It searches the web when he or she has specific topic in mind to look for. The distinction between the two models can be easily understood by a simple example, when a user reads the headline of newspaper (either hard or electronic) in the morning, he is basically browsing the information but on finding certain news interesting, he may start exploring it on other web sites too. At that time he is basically searching the web.

Due to abundance of decentralized information available on web, it is not prudent for a user to sift through millions of web sites to search for a specific piece of information. To achieve effective web searching, the automated programs such as *"Search engines"* come into appearance [8]. They help the user to find the information by making available the sorted links of URLs related to its query.

The modern search engines like Google, AltaVista, Bing etc. are applying sophisticated searching algorithms to better serve the information need of user. The goal of recent search engines is to provide sociable interface to the users that can precisely infer their information need and render the latest results in least time.

1.2 MOTIVATION

The motivating factors behind this research are listed below:

1. Completeness

The key challenge for search engine is how to harness the entire image of web in its search repositories as no crawler can download the complete web [82]. Even a most inclusive crawler can store only a portion of web. Owing to this fact, it becomes essential for the web crawler to carefully populate its queue with important URLs so as to reflect near to true image of web.

2. Freshness

User often browses the web to obtain latest information through search engines [86]. So after downloading a significant portion of web, next issue to be considered is how to keep the downloaded collection up to date. Owing to the fact that all web pages do not change at same frequency, so it is important for the web crawler to decide the re-crawl interval for each web page.

3. Information Overkill

Search engine generally returns a long list of web pages in response to the user query with possible duplicated pages contained therein. But most of these search results are of no interest to the user and ends up as low precision searches. The huge list of documents coupled with unwanted results forces the user to either change the keywords of the submitted query or sift through long list of documents resulting in the problem of information overkill [77] [79].

4. Lack of proper personalization

Personalization means observing and recording certain facts about user browsing behavior and utilizing them in providing desired information to the user [83]. Although the modern search engines count the number of user visits on a web page [80] and perform well in e-commerce domain for recommending products to their customers but they are still lacking in general topic search. So there is need to excavate other parameters of user behavior whose inclusion may result in better response list [85][116].

1.3 PROBLEM IDENTIFICATION

With the exponential growth of WWW, more and more people relay on search engine for exploring information from macrocosmic to microcosmic level. Search engine has become the most powerful automated tool for information retrieval. But the information on web is unstructured, diverse and dynamic [76]. Although the current search engines are applying the state of art techniques to make the information available to the user but one of the crucial problem they face is to find the relevant information the user is seeking for. There are several reasons as to why finding the information through search engines is not always successful like:

- ***Huge Size of web***

The first difficulty in finding the relevant information is the huge size of web. Today the web contains information on almost all the topics ranging from education, trade, entertainment, policies, social networking etc. It has been observed in a survey that there are 4.8 billion pages (as reported on 10 May 2017) in the index able web and many more are lying in the hidden web [17]. Moreover this collection keeps on growing at a rate of one million pages per day [8]. Also it cannot be denied that quality of a search engine is greatly influenced by completeness of its downloaded collection of web pages. In fact, no search engine can index the whole image of web. Even the most comprehensive search engine can index only 16% of entire web due to space constraint [88]

- ***Dynamic nature of web***

Another factor that influences the quality of search engine is the dynamic nature of web. The information on web is volatile with new pages added, old pages removed and existing pages modified. The new information is pouring from Twitter, blog, news sites and other source of social networks every other second [89][100]. Even after a search engine had stored significant portion of web, it has to send out its crawling program to revisit the same page in order to detect any change, explore discrepancies and correspondingly update it. Moreover the rate of change in all the pages is not uniform. Some pages are less dynamic and change once a year whereas some are so dynamic that change in less than a day. With this fact, it becomes essential for the crawler to design efficient re-crawl policies among the URLs so as to maintain fresh database.

- ***Length of query***

Retrieval of information from search repositories is highly dependent on keywords supplied by the user in its query. The pages are retrieved by matching the query keywords with the term weight vector of the documents archived in search engine repositories. The scheme performs outstanding when the user transforms its thoughts into well-defined set of keywords. But it is observed in a study done on Alta Vista query log that average query length issued by the user is 2.31 words with less than 4% of the queries having six terms and more than 70 % of the queries having only single term [91]. Speculating the information based on such a condensed representation is analogues to searching a needle in haystack. So it is highly important for the query processor to automatically decipher user real intent of search and map it to appropriate query structure at database level.

- ***Context of query***

One of the biggest problems of information retrieval is difficulty in determining the relevancy of document with respect to user query. Infact, finding the relevancy according to user query is subjective and interest varying concept. Some words may have different meaning for different people and in different context. This is the problem of synonymy (different word having the same meaning) and polysemy (same word having the different meaning in different context) [84]. So, search engine must be able to retrieve the document depending upon the context of user query.

- ***Delay in information filtering***

Although the modern search engines are applying personalized techniques to better serve the information need of the user but this require either the participation of user (in which user is not generally interested) or implicitly deducing the browsing patterns of the user which is quite cumbersome or violates privacy concerns [117]. So, it is required for the search engine to design optimized personalization technique that provides the relevant results without violating the privacy concerns. In addition to this, the current personalization techniques are applied on the query results. However, in addition to it, personalization at the level of query submission may produce more relevant results.

1.4 OBJECTIVES OF THE PROPOSED WORK

The information seeker wants to fulfill their information need in less and less time with minimum efforts at their end. So, it becomes important for the search engine to infer real intent of user's search automatically in response to its queries. The specific objectives of the proposed work are as follows:

- ***To design unified interest based search system:*** *The foremost objective of the research is to develop a unified technique for the search system that can deal with the interest of user both at front and back end level so as to provide personalized results to each user.*

Solution: In this thesis, a unified framework for interest based search system has been proposed. All the query specific processes are put together in data presentation layer and all the processes related to storage of web pages at the database level are tied to data collection layer. Both layers contain the processes that specifically deal in deciphering user interest.

- ***Query assistance:*** *As the relative effectiveness of search result is highly affected by the extent to which the query keywords map to the actual need of user. So, in order to precisely fulfill the information need of the user, an important objective of the proposed work is to assist user in query formation phase.*

Solution: To accomplish this objective number of techniques for query suggestion based on web context and usage mining has been proposed and implemented. In order to understand the context of query and usage trends in information retrieval, various web resources such as definition_ repository, query log, profile database and web dictionary has been employed. The technique presents the personalized queries to each individual thereby reducing the search space and provides relevant information to the user.

- ***Develop efficient ranking algorithm:*** *Traditional search engine employs only link structure or/and content oriented approach to rank documents in the result list. And simply ignores the user browsing behavior for which the sorted list is being prepared. The objective of the proposed system is to utilize usage trends during the ranking process.*

Solution: A novel page ranking mechanism based on user browsing behavior had been proposed. It assigns the page probability factor to each page based on the number of clicks, time spent and actions performed on the page relative to other pages in the same domain. Besides this, the technique also utilizes the link and content information of page to compute its rank.

- ***Maintaining up to date information:*** *There is a need to gradually update the contents of search repositories with latest information. So another objective of the proposed search system is to design re-crawl policy for each URL separately***.**

Solution: An arithmetic progression based model for computing the data collection cycle for each web page based on page updation frequency has been designed and implemented.

- ***Maintaining priorities among the URLs:*** *In order to enrich the search engine repositories with important information, another objective of research is to assign priorities among the URLs with same data collection cycle.*

Solution: At present, the document that is referred by multiple sites is treated as duplicate/redundant and discarded by the crawler. However the document being referred by the multiple sites gives an idea about its importance i.e. it is expected that the document is containing very useful information as far as latest user information need is concerned. So this information can be used to assign priorities among the URLs.

- ***Creating domain specific classes:*** *Since the user of web search engine possesses varying degree of interest in different domains, another objective of the proposed system is to create various domain specific classes that facilitate in computing the interest factor of each user in different areas.*

Solution: In the proposed system, the existing data structure of search engine database is modified for the sake of better interpretability, efficiency and personalization. The existing database is classified in various domain specific classes such as education, travelling & tourism, food & beverages, digital products and entertainment. Further, each class contains the structural and usage information about downloaded pages which facilitates in query formation as well as result set construction process.

1.5 ORGANIZATION OF RESEARCH WORK

This work is divided into eight chapters. The content of each chapter is outlined below:

Chapter 1 covers the introduction about WWW, Internet and search engine. It discusses the motivation behind the research, problems faced in retrieval of relevant information from WWW, objectives of proposed work and organization of thesis.

Chapter 2 presents the taxonomy used in the field of information retrieval. The chapter describes the technology behind the general search engine and provides the detailed review of prevalent query suggestion, crawling, and page ranking mechanisms used by the search engines. It also discusses web personalization, its need and applications in the area of user characterization. Chapter also highlights various web mining techniques and their utility in information retrieval. Based on the literature review, major challenges and limitations of existing approaches are identified.

Chapter 3 In this chapter, necessity to assist the user during query submission has been identified. Two successive proposed models of query suggestion: one based on wordnet similarity and other based on user browsing history are presented.

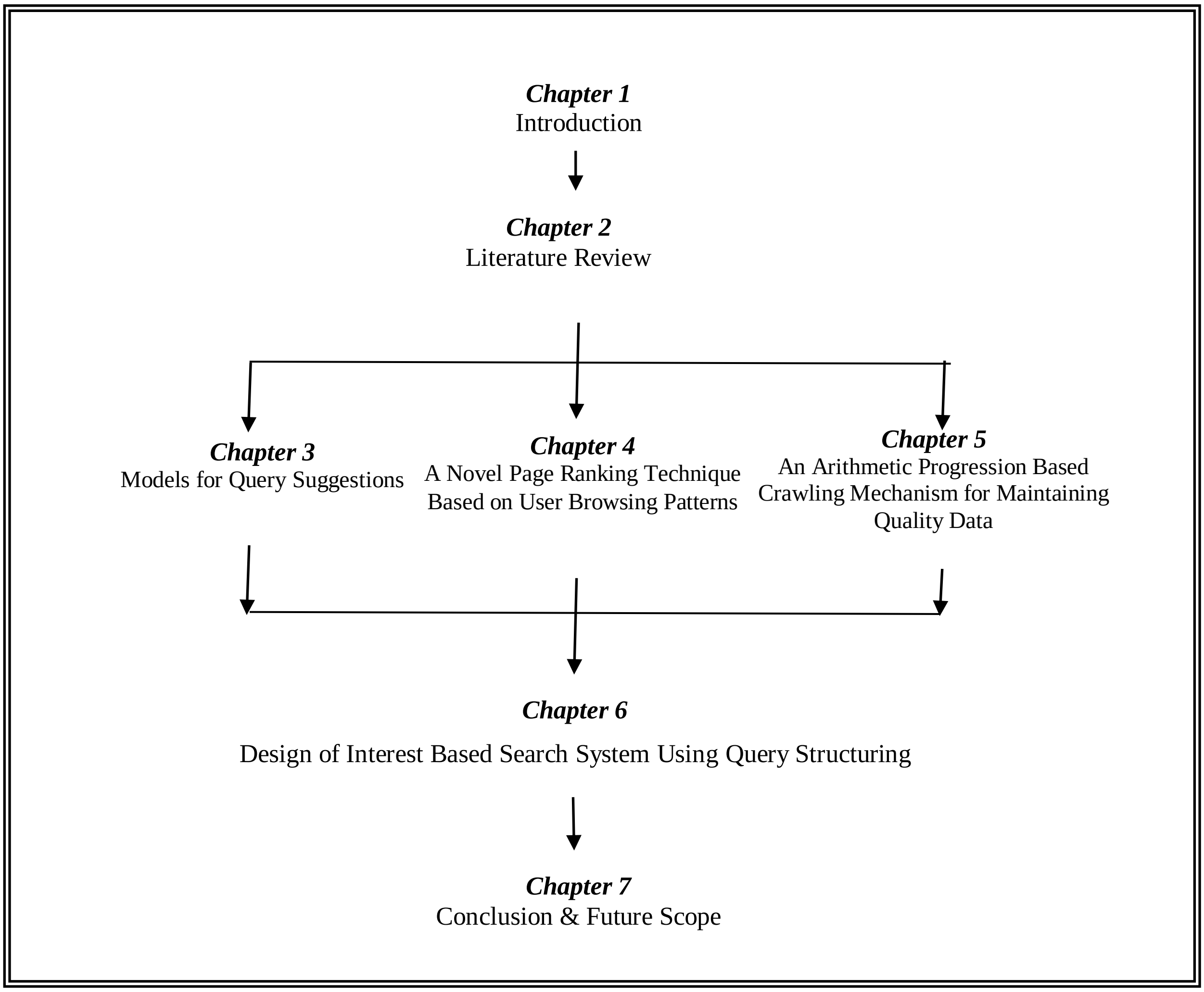

Chapter 1
Introduction
Chapter 2
Literature Review
Chapter 3
Models for Query Suggestions
Chapter 4
A Novel Page Ranking Technique Based on User Browsing Patterns
Chapter 5
An Arithmetic Progression Based Crawling Mechanism for Maintaining Quality Data
Chapter 6
Design of Interest Based Search System Using Query Structuring
Chapter 7
Conclusion & Future Scope

Fig1.1: Organization of thesis in chapters

Chapter 4 covers the detailed discussion on the design of novel page ranking mechanism based on page probability factor. The mechanism prepares sorted list of documents by applying web content mining, web structure mining and web usage mining at various level of search process.

Chapter 5 provides the detailed discussion on arithmetic progression based crawling mechanism for maintaining latest collection of web data. Modified data structures for URL Frontier and Page Repository have also been proposed in this chapter.

Chapter 6 presents how the various proposed techniques discussed in chapter 3, 4 and 5 works together to provide the desired functionality of interest based search system. The framework designed for interest based search system based on query structuring consists of two main layers: Data Presentation and Data Collection Layer. Data collection layer downloads the web documents, stores in a local repository and maintains the domain specific search engine database. The data presentation layer assists the user in query formation, computes the relevance score of each matched document retrieved by query processor and returns the ranked list to the user. Detail of modified data structure for indexing and query log is also provided.

Chapter 7 summarizes the contributions of the proposed work and provides future possibilities for extending the research in this direction.

Finally, the bibliography includes references to publications in this area. Appendix A includes query log utilized in the experiment. Appendix B to Appendix E includes snapshots of implementation.

A survey on existing search engines, web personalization and web mining techniques are given in next chapter.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

Internet is global network of networks that serve billions of users worldwide. Today's Internet is result of visionary thinking of some people in early 1960's to share military information. The following proponents contributed towards development of Internet from time to time:

- In 1962, Leonard Kleinrock of MIT (Massachusetts Institute of Technology) developed the theory of packet switching that subsequently formed the basis of data transfer over Internet.
- In 1965, Lawrence Robert of MIT connected the computer in Massachusetts with the computer located in California through dial-up- telephone lines.
- In late 1965, Ted Nelson, an American scientist proposed the idea of cross referenced documents which allow the user to sift from one document to other very easily.
- In 1966, Lawrence Robert developed ARPANET for Defense Advanced Research Project Agency.
- In 1969, ARPANET was brought online by connecting computers in four geographically distant universities of America: University of California, University of Utah, Stamford Research Institute and University of Santa Barbara.
- In 1984, Paul. V. Mockapetris of ISI (Information Science Institute) invented domain name system to map complex IP address with easy to remember extensions such as .mil, .com, gov., .edu etc. Out of these .com domain is most popular today.
- In 1989, Timothy Berners Lee of MIT implemented the idea of Ted Nelson on large scale to enable information sharing among the research team of European laboratory, Genewa which was dispersed globally and working on particle physics. It subsequently became the platform for the development of World Wide Web (WWW).

- In 1993, Marc Andresseen of NCSA (National Center for Supercomputing application) developed first graphical browser Mosaic. Later, The development of Mosaic became the foundation for Netscape Navigator. After that many more web browsers came into existence like Internet Explorer, FireFox, Mozilla, Google Crome, Opra, Bing, Safari etc. Today Google Crome occupies 62.7% of web browser market share [18].
- In 1998, U.S Department of Commerce formed ICANN (Internet Corporation for Assigned Name and Number) to privatize the operation and registration of domain names.

Thus WWW emerged as massive collection of globally distributed, highly heterogeneous, semi structured hypertext information repository that is accessed via Internet. The major difference between Internet and WWW is Internet is basically hardware part (network of computers connected via wired or wireless technology, where as WWW is software component (collection of cross referenced web pages). Another difference between them is base of their protocol suit. Internet is based on internet protocol (IP) that deals with physical transmission of data in bits whereas WWW is based on Hypertext Transfer Protocol (HTTP) that deals with transmission of information in packets.

2.1.1 Evolution of Internet and WWW

Internet has made distances shorter and world smaller. Over the years, the number of computers connected via Internet has grown exponentially [2]. Beginning with only four universities computers connected for research purpose, today the Internet connects millions of computers worldwide for diverse purposes. The explosion of the Internet has transformed not only the discipline related to computer science but the lifestyle of the people and economies of the countries. The Internet statistics in terms of millions of users from 1995 to 2016 is depicted in Fig 2.1 [3].

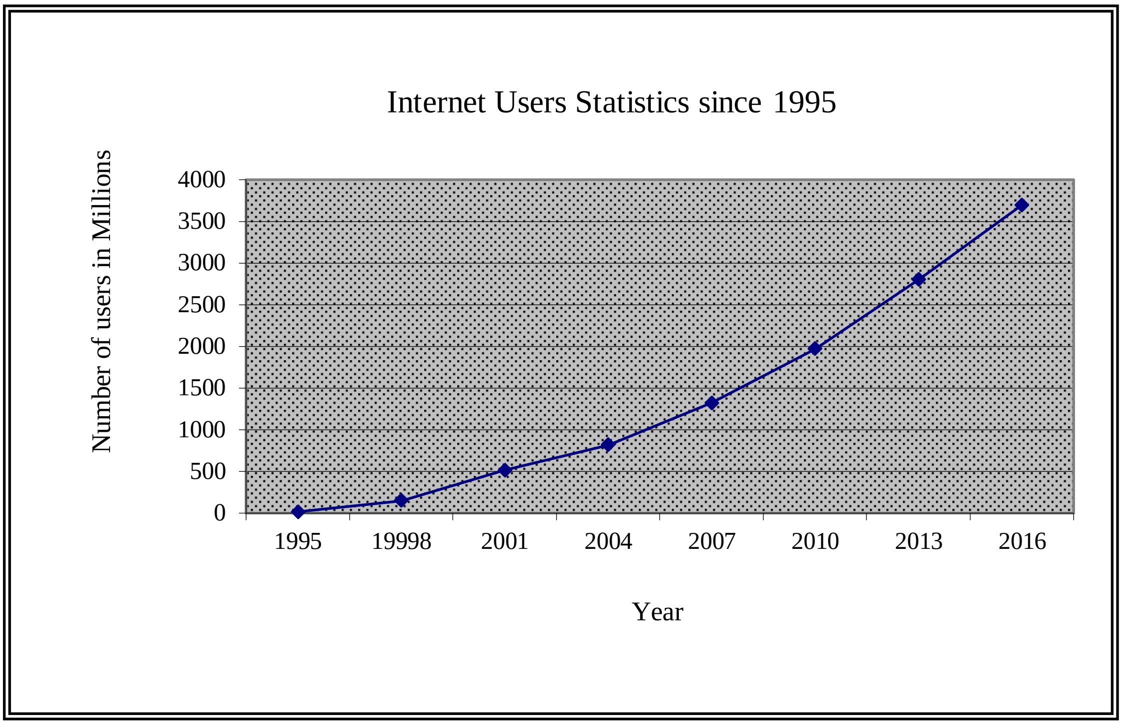


Fig 2.1: Internet Statistics in Last 20 Years

Graph shows that number of users on Internet has increased 230% since 1995 and the number is continuously growing.

With the increasing demand of Internet, WWW has also evolved incredibly from simple information sharing of text and images to intense assortment of dynamic and interactive multimedia services like audio/video conferencing, E-learning, E-business and social networking. In fact, WWW has experienced three growth stages [5] since its inception. They are:

- **Explosive growth (Stage 1):** From 1991-1997, the number of websites on WWW grew at the rate of 85% per year
- **Rapid growth (Stage 2):** From 1998 -2001, the number of websites on WWW grew at the rate of 15% per year.
- **Mature growth (Stage 3):** From 2001 onwards, the number of websites on WWW grew at the rate of 25% per year

Growth curve of WWW in terms websites registered to various domains are shown in Fig 2.2.

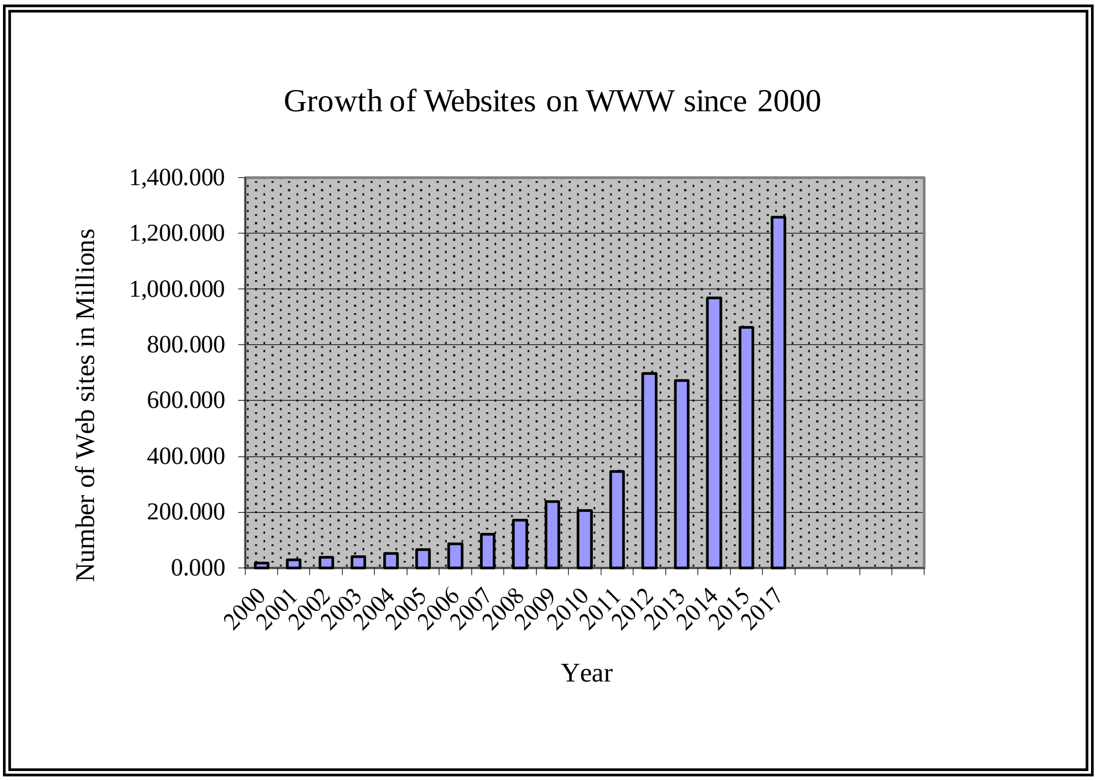


Fig 2.2: Growth of Websites on WWW in Last 15 Years

It may be observed from the graph that number of web sites increased exponentially since 2000. The number reached near to 1 billion in 2014, but then declined to a level below 1 billion (due to fluctuation in the count of inactive websites) before increasing again and stabilizing to level of above 1 billion in 2016 [3].

Out of these web sites, not all are alive. Some are registered under parked domains whereas some are not updated for long. If only half of these are maintained, still there exist half billion of web sites that people keep on surfing throughout the world [4]. Fig 2.3 shows growth curve of WWW in terms of average density of web pages and average web page size

It is observed that average size of web page has increased quintuple since 2010 (from 312.5KB in 2010 to 1600KB in 2017) and in same 7 years period average number of web pages grew four folds (from 1 billion in 2010 to 4.53 billion in 2017) [4]. Long-

tstudies reveal that average webpage density on WWW has increased 50 times and average web page size has grown 80 times since 1995. Besides the growth of page density, the pages are continuously updated and removed at a rate of 23% per day [10]. So to retrieve the information from such huge repositories of WWW,

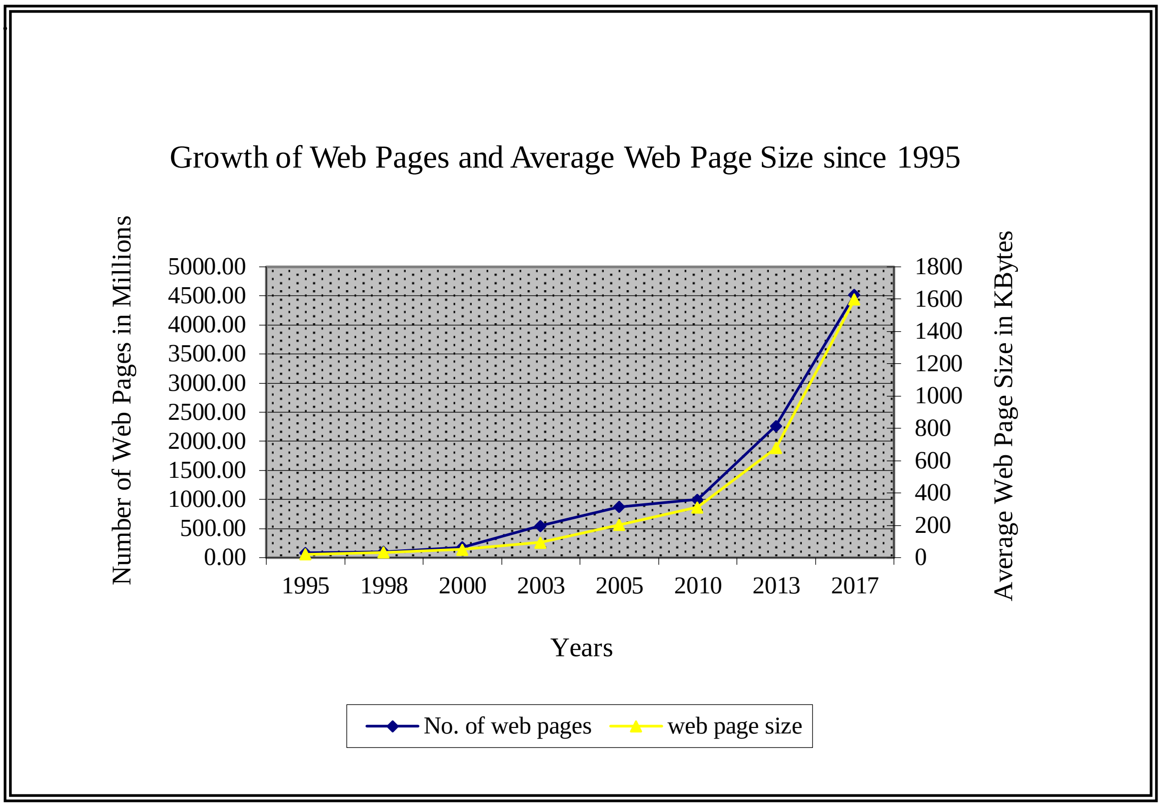


Fig 2.3: Growth of WWW in Terms of Web Pages and Page Size

Information retrieval systems came into appearance. The following sections aim to survey information retrieval system, their basic working and applications.

2.2 INFORMATION RETRIEVAL

Information retrieval means the process of obtaining data objects from large repositories relevant to user information need [6]. The data objects are generally of unstructured nature and may comprise of text, pictures, audio, video or Google maps. Often the objects are not stored directly in the repository but are in the form of metadata.

So, in a more elaborated sense, information retrieval (IR) is the field of computer science that deals with acquisition, storage, organization and access of data objects of usually unstructured nature from large source of web data based on user specific need [40]. User specifies his search need in the form of string, formally called as *query*.

Information retrieval activity begins when the user submits its query at information system. In IR, a query does not necessarily map to a single object instead many objects can be mapped to single query with different degree of relevancy. That's why the IR systems are equipped with query engines. It is the responsibility of query engine to retrieve the data objects from large repositories and compute a numeric score for each object indicating how well the object matches with the user query. The top ‘N’ results are then presented back to the user. The general working of IR system is depicted in Fig 2.4 [8].

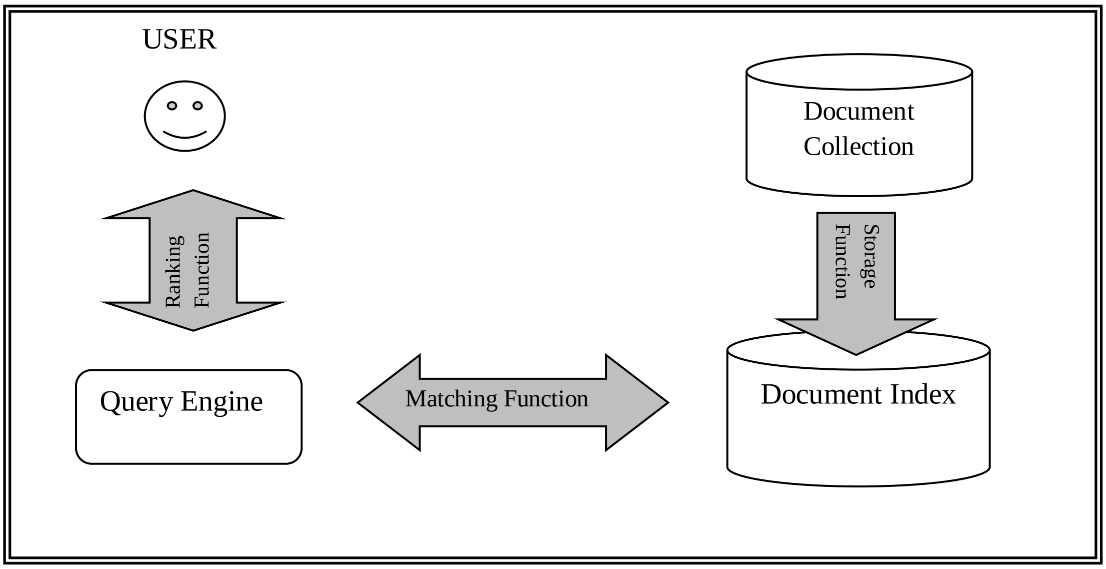


Fig 2.4: General Working of IR System

Many people apply the term Data retrieval (DR) interchangeably with Information retrieval (IR) but these two are different from IR point of view [7]. Table 2.1 lists some of the important differences between data retrieval and information retrieval.

Table 2.1: Comparison between Data Retrieval and Information Retrieval

Sr. No.	Feature	Information Retrieval (IR)	Data Retrieval (DR)
1.	Model	Probabilistic	Deterministic
2.	Classification	Polythetic	Monothetic
3.	Mapping	Relevant	Exact
4.	Query Language	Natural (Free form)	Artificial (Regular expression, relational algebra)
5.	Query Specification	Semi structured or unstructured	Structured
6.	Error Response	Insensitive to small errors	Sensitive to small errors

7.	Example	Google Search Engine	DBMS

2.2.1 Classification of Information Retrieval Systems

Information retrieval systems can also be broadly classified into three prominent classes on the basis of kind of objects they store [11][68] and scale at which they operate [9]. They are:

i) Web Information Retrieval System

ii) Desktop Information Retrieval System

iii) Enterprise Information Retrieval System

Web Information Retrieval Systems, collectively referred as *Public Search Systems* are designed to search over trillions of documents archived on millions of web servers to fulfill the information need of billions of people. Major issues to be consider here are how to gather widely spread documents and build powerful indices to efficiently operate at such an enormous scale. Web search engine is popular application of web information retrieval system.

On the other extreme, Desktop Information Retrieval System also referred to as *Personal Search System* creates index for all the files on a single computer so that user can quickly access the wide variety of objects including documents, images, audio, video etc. [11]. In recent years, consumer operating systems are coming with built-in desktop search features such as spotlight in Apple’s Mac and instant search in windows operating system [13][14]. These types of system built full-text indexes of all the files on the computer. Once the contents are populated in the index, searches can be performed not only on file names, but also on the contents of file, keywords, comments and all sort of metadata. Email search programs also come under personal information retrieval system. Major issues to be considered here include handling the broad range of document types (such as word documents, excel spreadsheet, power point presentations, HTML files, JPEG, BMP, AVI, PDF etc.), making the search system maintenance free and lightweight in terms of startup, processing, and disk space usage [22] [28].

The third category, enterprise information retrieval system lies in between the web and document retrieval system in terms of size of source of information and audience. Enterprise systems build their index from multiple enterprise sources (such as research articles, patents, emails or enterprise internal files maintained on multiple machines connected by Intranet) to serve the need of defined audience. Many enterprise information retrieval systems integrate structured and unstructured data in their collection. These systems provide search of information within the enterprise/institute (usually covering domain specific information in which enterprise deals) through a search function but their results may still be public [15]. Major issues include how to seamlessly and scalably harness structured (e.g. relational database) and unstructured data in a document for search as well as for organization purposes (such as clustering and classification) and enforcing the security policies for access control to their users [16] [17]. Some of the popular enterprise search softwares available in market are Viaworks, Lookeen Server, Integrator, EXALEAD Cloudview, Datafari etc [95].

Some of the important differences between web, desktop and enterprise information retrieval systems are summarized in table 2.2.

Table 2.2: Difference between Various Kinds of Information Retrieval Systems

Sr. No.	Parameters	Web IR	Desktop IR	Enterprise IR
1.	Source of data	World Wide Web	Personal device	Devices connected to Intranet
2.	Nature of data	Unstructured	Semi structured	Semi structured
3.	Access control	Open	Restricted	Restricted
4.	User Interface	Friendly	Simple	Personalized
5.	Search relevance factor	Popularity of document in link structure of web	Text classification	Document relevancy
6.	Size of audience	Enormous	Single user	Limited to Enterprise

7.	Work vicinity	Online	Offline	Offline
8.	Examples	Search Engines	Instant, Spotlight	Viaworks, Lookeen

The next section describes the history and working of search engines to better understand the concept of web IR.

2.3 SEARCH ENGINE

Before the invention of search engines, users were confined to visit only those web sites that they already knew in order to fulfill their information need. This might be adequate when only few people used Internet and web had just begun. But the web continued to grow at staggering rate and became the source of all kinds of data so it became extremely difficult to manually search information from such a large reservoir [21] [22]. This is where the need of automated tools came into picture.

"Search engines are the automated programs that help the user to quickly locate the information based on the issued query".

At first, search engines were quite rudimentary, but over the years they have grown sophisticated. Searching the web using search engine is perhaps the most frequent activity among the youngsters. As stated by *Comscore* in a press release in 2017 total worldwide search market has boasted by 67% in last three years [23]. The next section highlights some of the important milestones in the development of search engines.

2.3.1 History of Search Engine

1. Early Search Tools (1990-1993)

The first automated program created for searching information on web as opposed to manual search was Archie. The name stands for "archive" without the "v". It was created in 1990 by *Alan, Bill* and *Peter Deutsch,* computer science students at McGill University [19]. They built database of Archie by downloading the directory listings of all the files located on public anonymous FTP sites. However Archie did not index the contents of these sites.

In 1991, two new search tools, Veronica and Jughead were launched to search the file names and titles stored on Gopher system. So what the Archie did for FTP sites, Veronica and Jughead did for Gopher sites. Veronica (**V**ery **E**asy **R**odent-**O**riented **N**et-wide **I**ndex to **C**omputerized **A**rchives) was developed by *Mark McCahill* at the University of Minnesota . In the same year, another search tool named Jughead came into existence with the same purpose as Veronica.

2. Web Robot (1993-1995)

Though many specialized indices were maintained for particular sites but no unified index was created for whole web. In 1993, Oscar Nierctrasz at the University of Genewa wrote Perl script that could periodically copy the pages from WWW and transform them into a standard format. It subsequently formed the base of world's first primitive search engine W3Catalog, In June 1995, Matthew Gray, at MIT, developed first web robot, called Wandex. The purpose of the Wandex was to measure the size of the World Wide Web [20].

The world's second search engine Aliweb (**A**rchie **L**ike **I**ndexing for **Web**) appeared in late 1993. Aliweb required the web site administrator to submit their file with URL in a specified format instead of maintaining the index through web robot. Unfortunately, the application file was difficult to submit so many websites were never registered with Aliweb.

Next important milestone developed in the history of search engine was by Jonathon Fletcher. He deployed the search tool named as JumpStaon that could collect web documents from WWW, built its index, and offered a web form as the interface to its user to submit their query. It was thus the first search tool that possessed three essential features of a web search engine i.e. crawling, indexing, and query processing. But due to limited resources available on this platform, searching was limited to only titles and headings in the web page [30].

In 1994, Brian Pinkerton resolved the limitation JumpStaonfirs and developed "all text" public crawler based search engine named WebCrawler (still exists). It allowed the users to search for any string in any webpage and subsequently became the standard for later search engines Soon after, many search engines such as Magellan, Excite, Lycos, Inktomi, Northern Light, Infoseek, and AltaVista came into appearance

3. Search Directories (1994)

The first browsable search directory, Tradeware, was launched in 1994. It helped the user in narrowing its search by browsing information in layer of categories and sub categories instead of searching through keyword-based search engines.

Also in 1994, Yahoo (Yet Another Hierarchical Officious Oracle) was developed on the same concept by two electrical students, David Filo and Jerry Yang at Stanford University as the way to record their favorite links on the web page. Soon it became popular among people. Yahoo database was manually maintained in subject categories by human editors. Typically, it required website administrator to submit a brief description of its website which was further reviewed, edited and classified under subject categories by Yahoo editors. But such a directory based search engine could index only small portion of web.

4. Meta Search Engines (1995)

The next milestone in the history of search engines was development of Meta Search Engines. In 1995, first metasearch engine, *MetaCrawler* was developed by Eric Selburg, a student at University of Washington. The idea was to simultaneously fetch the search results for a query from multiple search engines and compile them into a single list according to their collective relevancy.

5. Google (1998):

Google was initially begun as a research project by two Ph.D scholars Larry Page and Sergey Brim in 1996 [78]. The idea was to rank the web pages based on their popularity in the link structure of web. With this idea they launched a small search engine company named goto.com. This move led a significant boom in search engine market [27]. .

Today, Google is most popular search engine covering 62.7% share of global search market [31].

6. MSN Search (1998):

MSN Search was launched by Microsoft in 1998. It began to display combined listings obtained from Looksmart and Inktomi. In 2004, Microsoft began a transition to its own search technology, called *msnbot* [29].

Later in 2009, Microsoft released its new search engine, *Bing* in SE market. On July 29, 2009, Yahoo! and Microsoft finalized a deal in which Yahoo! Search would be powered by Microsoft Bing technology.

7. Page hit (2003):

Till 1998, there were only three major kinds of search tools namely crawler based search engine. Human powered directories and Meta search engines. Besides these major search tools, a new kind of user controlled search tool named ***Direct Hit*** was introduced by Grey Culliss in 2003. It is based on the assumption that more hits received on a particular page in result list truly reflects its relevancy for the user and thus contributes towards ranking of page [33][38].

8. Semantic Search engine (2006)

Semantic search engines came into appearance in late 2006 [82]. They seek to improve search accuracy for specific area by understanding the contextual meaning of terms issued by the user. They may also consider other parameters such as location, variation of words, fuzzy concepts, generalized and specialized queries, natural language queries to provide more relevant results. For example Linkedin, Evi, Yummily Swoogle [84].

As the web and its user with diverse information need continue to grow, thus emergence of new search tools also continue.

Since the research is centered on crawler based search engine therefore the detailed architecture of crawler based search engine is presented in section 2.3.2.

2.3.2 Search Engine Architecture

A search engine is coordinated set of programs that contact various web sites on the internet, pursuit for specific information, archive the information it finds in its database, compare the information with user search request and eventually returns a sorted list of documents to the user [64][139] . A web search engine consists of five major components as depicted in Fig 2.5.

A. Crawler

B. Indexer

C. Query Processor

D. Ranker and

E. Search Interface

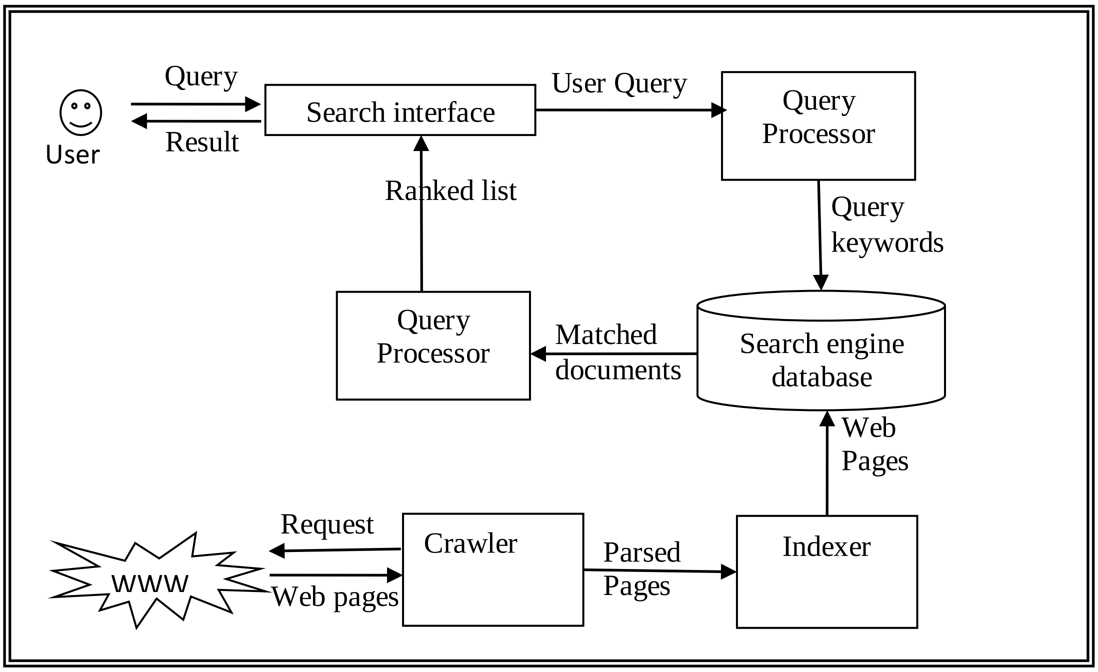


Fig 2.5: General Architecture of Crawler Based Search Engine

Role of each of the component is described below.

A. Crawler

Crawler is the program that traverses the web on behalf of search engine in order to collect up to date data in the form of web pages and stores them in page repository. The basic operation of any crawler starts by picking up the seed URL form URL frontier (URL frontier is basically a list of URLs to be crawled and initially populated by search engine's designer) [69]. The crawler then visits each web page, extracts the links embedded in it and inserts them into URL frontier. The working of a typical web crawler is shown by the flowchart given in Fig 2.6 [103].

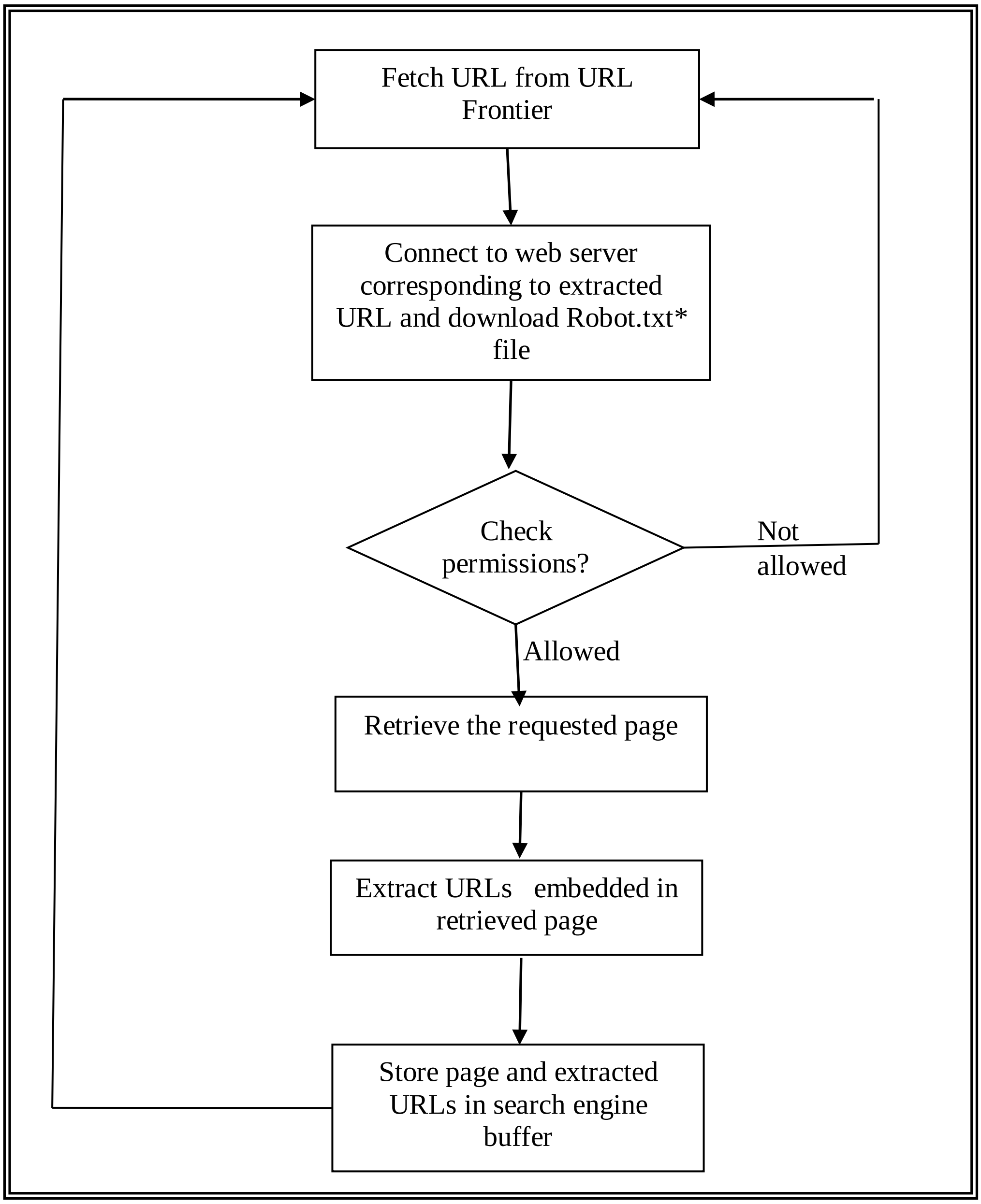

Fetch URL from URL Frontier
Connect to web server corresponding to extracted URL and download Robot.txt* file
Check permissions?
Not allowed
Allowed
Retrieve the requested page
Extract URLs embedded in retrieved page
Store page and extracted URLs in search engine buffer

Fig 2.6: Working of Typical Crawler

**Robot. txt* carries the downloading permission and also specifies the file to be excluded by crawler

In order to cope up with ever changing nature of web, different search engines adopt different policies to crawl the web. Some of the important policies that all crawler programs must opt are summarized below [86]

- **Selection Policy:** This policy identifies the URL to be crawled.
- **Re-Crawl Policy:** This policy determines revisit interval for each page in order to

maintain its updated image in database

- **Parallelization Policy:** This policy designs the techniques to avoid redundant downloading of same web page.
- **Politeness Policy:** This policy formulates the guidelines to avoid overloading of web sites by crawler.

The pages downloaded by crawler are then organized by Indexer in search engine database. Next subsection covers basics of indexing.

B. Indexer

Indexer is a program that distills the data contained in collection of documents into a format which can be quickly retrieved by query processor [40]. Typically, it carries out some or all of the following steps:

i. Convert the document to a standard format.

ii. Identifies potential index able elements in documents.

iii. Extract query independent information related to document.

iv. Assign doc_ ID to document being indexed.

v. Delete stop words.

vi. Perform Stemming and Lemmatization.

vii. Compute weight.

viii. Create index.

Steps (i –iv): Parsing – Since the documents downloaded by crawler from various web sites may be written in different formats so first step in indexing is to convert them in a predefined format so that later steps of indexing can be easily carried out. Second step involves extracting document features from document by breaking it into constitute terms known as tokens. For example: friendship, uniqueness, life etc. can be tokens appearing in a document. Third step stores query independent evidences of document such as its in links or out links information. The result of this step is generally a very large lookup table that lists all the URLs pointed to and pointed by the document. In the fourth step, every document which is being parsed, assigned a doc_ID. A file is also maintained to store the URL checksums with their corresponding doc_ID’s.

Steps (v -vi): Linguistic Preprocessing- These steps identify the retrievable terms in the document. The process involves deleting extremely common words (such as ‘a’, ‘an’, ‘and’, ‘are’, ‘as’, ‘at’, ‘be’, ‘by’, ‘for’, ‘from’ etc) also called as stop words from the document. Since these words contribute very less towards finding relevant documents in response to a user's query Next process called Stemming and Lemmatization is carried out to reduce inflectional terms (such as operate, operated, operating) and derivational terms (such as social, society, socialism, socially) of a word to a base form so as to obtain unique words from the document. For example, following conversions can be carried out in Stemming and Lemmatization.

is, are, am→ be

book, books → book

operate, operates, operated, operating, operation, operational → Operat

To clearly distinguish between stemming and lemmatization, stemming usually involves crude heuristic process that drops trailing symbols from the end of word in hope to obtain unique terms for indexing whereas lemmatization is done with the help of morphological analysis of terms in order to return the term to its dictionary form, which is known as lemma.

Step vii: Compute Weight- The step involves computing weight of each index able term identified during linguistic preprocessing. Early indexer applied binary weight scheme- 1 for presence and 0 for absence with a term for convenience. Later complex scheme such as *tf/ idf* are introduced that computes relative frequency of term in the whole document/corpus.

Step vii: Create Index- Create inverted index of documents in accordance with their terms occurrence by creating a dictionary and document posting.

C. Query Processor

The goal of this component [45] is to parse user queries to identify its search intent, execute the parsed query on search engine database to fetch matched documents and hand over them to ranking module for further processing. The concrete steps involved in query processing are as follow:

i. **Tokenization**: It is the process of breaking a query into cohesive logical units called as tokens. A token is basically an alpha-numeric segment of string that occurs between two delimiters such as white spaces, punctuations symbols etc. Further tokenization is followed by parsing. Parsing is employed only in the scenarios where user query contains special operators such as Boolean, adjacency, or proximity operators. The output of this step is the stream of tokens and stop words which is further processed.

ii. **Stem generatio**n: This step involves deletion of stop words and generation of stems from the token stream for the obvious reasons explained earlier in indexing process. Their removal helps to reduce processing time required in matching potential documents from search engine database. However, in today's era, when memory is so much cheap and systems are equipped with fast processor, stop words are no longer an issue [75]. But as an average document contains 40 percent of stop words, it is still wise to ignore them from the query as well as from the document as far as processing and matching time with the document is concerned [47]. After removal of stop words, next task is to create stems of each leftover token. Stemming of token is done for two purposes:

- First, stemming is done to obtain unique keywords which helps in speeding up of searching process.
- Stemming increases recall of search results by reducing all forms of the word to its base form. Matching the query with the document is analogous to pattern recognition system. Recall of pattern recognition system can be defined as the fraction of exact instances among all instances that actually belong to the relevant subset. For example, if a user seeks for word operation, they may also want documents that contain operate, operated, operating, and operator or operate. Therefore, the query processor stems query term to operat- so that documents which include various forms of operat- will have equal probability of being matched.

iii. **Query expansion**: In order to retrieve relevant documents from search engine database, some sophisticated query processing system forms the alternate queries using synonyms of terms present in user query [12]. Some other systems also help the users to iterate i.e. offers some relevant suggestions to modify their query. For example, Google started automated query suggestion system in 2008 that offers similar queries issued by other users in the past.

iv. Query term weighting: This step is required only when the user query comprises of more than one term. Some system adopts implicit weighing scheme where first term in the query is considered to be highly important and thus given high weight whereas some other systems assign weight to query term based on their relative frequency in the query and still other system exist that allow the user to control how much to weight each term or simply which term/concept in the query matters must appear in each retrieved document to ensure relevance. But later scheme is rarely preferred because users are not particularly good in determining the relative importance of terms in their queries due to following reasons [46].

- As most user search for an unfamiliar concept so they may not use correct terminology [102].
- They may not know what else can exist in the vicinity of particular term in the document.

v. Searching & Matching: The final step in query processing involves consulting the search engine database for the processed query and retrieving the matched documents from it. The raw results of search process are reported to ranking module for further processing. The next subsection describes the function ranking module in detail.

D. Ranker

Once the matched documents are issued by the query processor, next step is to compute similarity score of each document with the processed query. The score may be based on document content or document popularity or even past retrieval history [85].

Content similarity ranking generally covers position and frequency of query terms in the document. For instance, the documents that contain the query terms near the top of the document such as in heading or first paragraph are preferred over others.

Term frequency is other important factor for measuring the relevancy of document with respect to query. Ranking module determines how often the query term appears in the document in relation to other terms.

As the data gathering and organizing scheme of one search engine vary from other so no two search engines have same collection of web pages to search through and henceforth adopt different criteria for ranking of their pages [78]. For instance, some search engines may opt to collect more web pages after a long period of time and later index them whereas others may frequently index their web pages with lesser web pages each time These factors naturally create differences while comparing their adventures results.

Unfortunately, search engine designers have to take caution while deciding the ranking factors for the web pages because some web masters may try to misuse these ranking factors in order to promote the ranking of certain web pages for their own interests Therefore many search engines keep changing their ranking algorithms from time to time. Following are some of the ways in which the web masters and search engines counteract each other.

- **Misuse:** In order to increase the frequency of terms present in certain web pages, web masters may insert ***spam*** in the web page. [36] Spamming is a technique that makes a word to repeat thousands of time on a page so as to boost its rank in the final listing.

  Action: Search engine may black list these web sites and exclude them permanently from their database.

- **Misuse:** They may apply reverse- engineering on certain ranking criteria (such as frequency or location) to boost the rank of page in search engine listing.

  Action: Ranking algorithm may embed some *off-the-page* ranking factors that webmasters can’t easily crack. Some of them are listed below:

  - One solution to this act is to rank the page based on its link popularity. For instance the *pageRank* algorithm designed by Larry page and Sergey computes the reliability of a page by analyzing how many other web pages mentioned the page in their out links and how trustworthy the linking pages are. In addition, sophisticated algorithms can be employed to avoid any artificial link imposed by webmasters to propel ranking of page.

- Another solution to this act is to record number of times a web page gets selected by the user in ranked listing [49] and accordingly drop/boost the ranking of the page.

E. Search Interface

It provides the interface to user to find information on web. User can submit his information need in the form of query and get the response in the form of list of documents with corresponding URLs. For example: Google home page.

In the light of above discussion, it may be noted that major challenges faced by search engines are : handling volatile data spread over enormous web sites operating on diverse platforms , discovering relevant information based on short and ambiguous queries and variation in use's search need especially of temporal nature. Many methods have been suggested in the literature to address information retrieval problems. A list of some of these problems and their solution are listed in table 2.3.

Table 2.3: Major Information Retrieval Problems Faced by Search Engine

Problem	Possible Solution(s)
Large volumes of distributed data	Parallel and distributed crawling
Poorly designed queries	Query recommendation, web personalization, Search result clustering
Discovering relevant information	Ranking , web personalization , Data mining
Variation in user's search need	Web personalization
Providing up to date information	Ranking, dynamic information retrieval

Since existing methods/ techniques for improving search performance are broadly centered around web personalization so its design assumes utmost importance. The detail discussion on web personalization is given in next section.

2.4 WEB PERSONALIZATION

Although search engines have significantly applied sophisticated methods to reduce the transaction cost of obtaining the information from web but still human ability to search relevant information is not expanded much [34][32]. When facing overwhelming amount of information, user still needs some automated tools to prune undesirable results from search listings. This is where the need and role of web personalization comes into picture.

Web personalization refers to search experience that tailors the individual interests by incorporating certain facts obtained from user past searches beyond the submitted query [33][26]. For example, if a user submits the query "information on BP" at search engine interface, it will get results for the oil company if the past searches were related to fuel for transportation or energy for heat and light, but it will get results for the business investment if past searches were related to say shares or investments plans.

The process of personalization involves tracking the activities of individual/group of user(s) during the information seeking hours, transforming the tracked information into set of preferences and utilizing them in ranking the search results [31][38]. Most publically accessible search system like Google, Altavista, Bing offers personalized services to their users,

Google pioneered personalized search in 2005 [25]. Some of the factors that Google utilizes in personalizing the search results include search location, user language, and browsing history. When a user has enabled web history in its browser, the Google automatically keeps record of the pages the user visits. If the user clicks the same page again, Google considers it as a vote towards that page and boosts the ranking of page in search list [24]. Even if the user is signed out, Google keeps user browsing record of 180 days and may provide personalized results [87][27]. By applying personalization, user can benefit in following ways:

1. It reduces the work that is no longer needed.
2. It provides more satisfactory results to user.
3. It improves the quality of decision customers make.
4. Search results can be efficiently obtained using personalization.
5. Through personalization, the search time is considerably reduced by eliminating repetitive task [118].

There are two general approaches to improve search effectiveness through personalization. They are: i) Refining the user query to avoid irrelevant results ii) Re–ranking the search results as per user preferences. In fact both the approaches can be carried out by the application of data mining- web mining. So before going to concrete approach in detail, discussion on web mining is covered in subsequent sections.

2.5 WEB MINING

When techniques of data mining are applied to discover and retrieve interesting, non trivial, previously unknown and potentially useful pattern on web data, then it is known

as web mining [41]. Basically, web mining is amalgam of data mining, artificial intelligence and information retrieval techniques. In fact it has been evolved as most powerful technique to track and analyze the usage pattern in information retrieval. It creates server side and client side intelligent system that discover suitable target data from large repositories of web. Technically, the process of web mining can be carried out in four steps as shown in Fig 2.7.

i. **Data collection:** This step involves retrieving the target data for knowledge discovery from web. The target data may involve web documents downloaded by search engine or simply by crawler, queries issued by user at search interfaces and stored in query log by query processor, or usage browsing data stored in web server logs. Nature of data collection differs not only in location of data source but also in methods and implementations used for a particular segment of users [42].

ii. **Pre-processing:** This step transforms the raw data into formal representation. For example, content and hyperlink information extracted from web documents are stored in separate tables for further processing. RapidMiner, Orange, Knime and R-Programming are some of the popular readymade open source pre- processing tools available online for multifaceted data [35][44].

iii. **Pattern generation :** This step carries out automatic discovery of patterns from the processed data through either of these techniques:

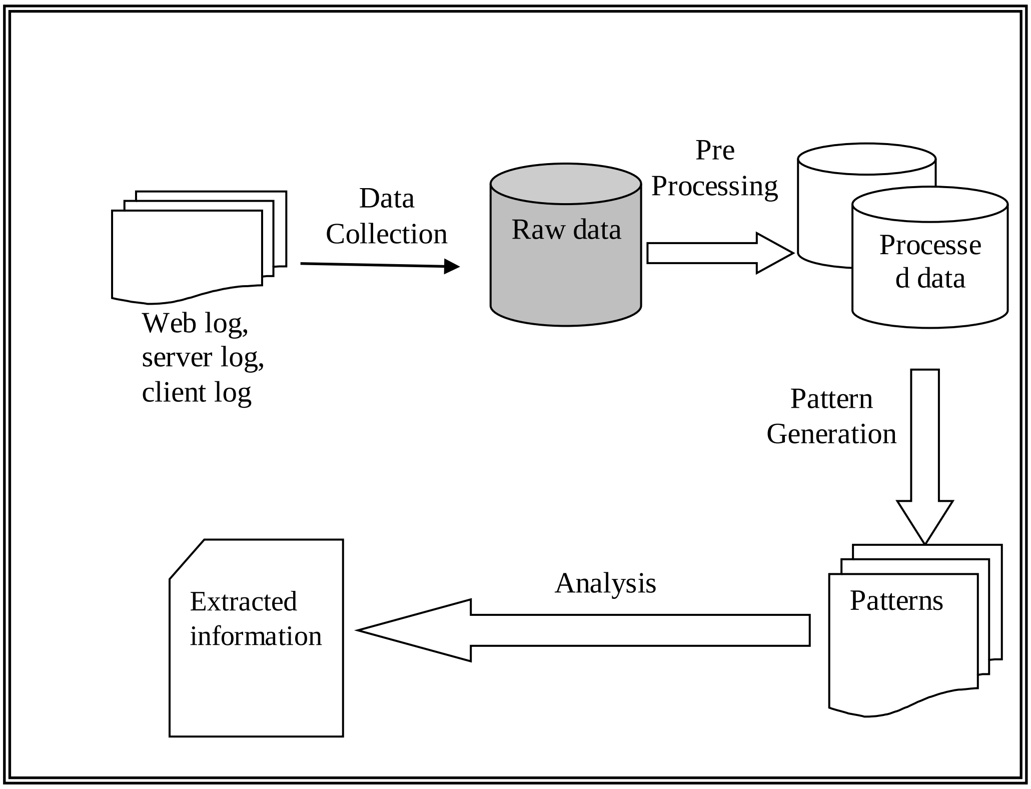


Fig 2.7: Major Steps in Web Mining

- *Genetic algorithm:* Set of non linear optimization techniques based on mutation, generic combination and natural evaluation. For instance, R.Feldman proposed the method to classify web documents based on genetic algorithm. [37]

- *Classification:* It classifies each record available in data set under a cluster such that members of a particular cluster share similar properties. For instance Bolsus & Pazanni proposed the techniques for classifying the web sites based on their contents using classification method [38].

- *Decision tree:* This structure classifies the data set in hierarchy. For instance Heinonen & Ahomen proposed a technique to predict patterns that reflect economic growth of companies by automatically collecting their data from web [36].

- *Association rules:* Discovering usage pattern that mainly reflect web site visitor's behavior by applying rules on web log for the purpose of improving web search experience fall under this category [127]. For instance, D. Mademic suggested a technique to recognize products that are brought together by most of the customers by applying association rule mining [39].

- *Visualization:* The technique discovers the complex relations among multidimensional data using graphical tools such as WEKA and RapidMinner [40] [41].

iv. **Analysis:** Finally the patterns obtained are validated and interpreted to get useful information in tabular or graphical form according to kind of source data, web mining can be broadly divided into three distinct categories as depicted in Fig 2.8,

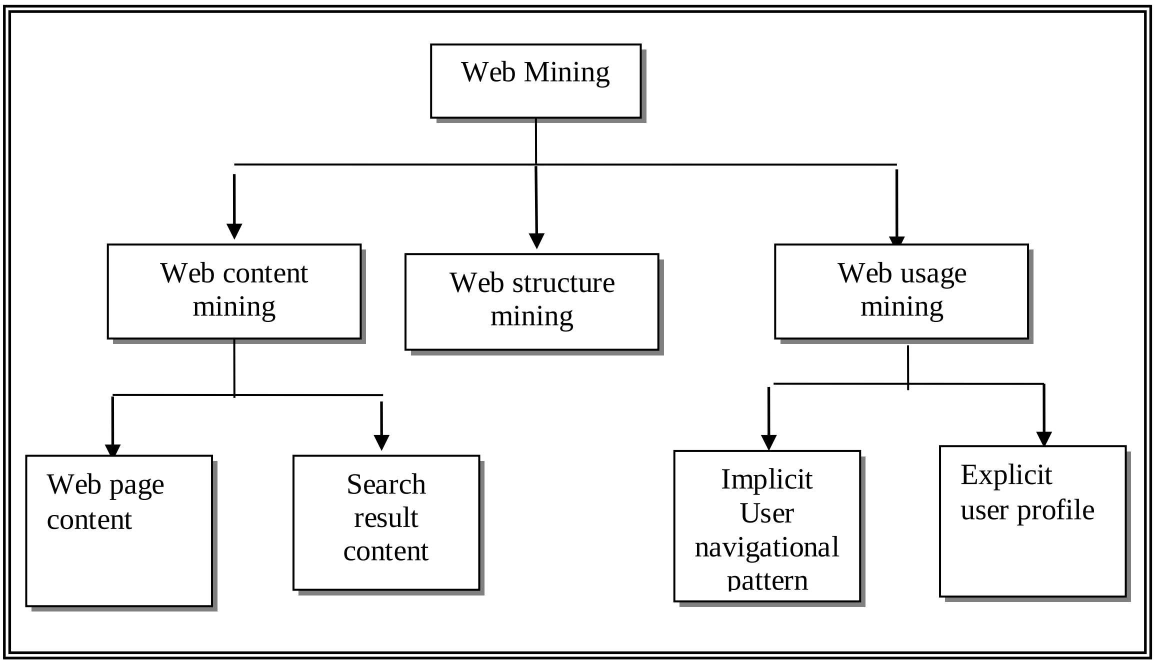


Fig 2.8: Categories of Web Mining

- **Web content mining (WCM):** It is automatic extraction of useful patterns from contents of web document.

- **Web Structure mining: (WSM):** It refers to generation of structural summary of links between web servers or web documents.

- **Web usage mining (WUM):** It refers to identification of user access pattern from web logs.

A detailed discussion on each category and its applications in the field of personalization is covered in next subsections.

2.5.1 Web Content Mining

Web content mining can be regarded as process of discovering useful information from the content of web pages such as plain text, images, audio, video, list and tables. There can be two views to consider the contents of web documents for mining purpose.

i) IR view

ii) DB view

In IR view, a web page is considered as bag of words/phrases and feature of single word in isolation are used as the basis of pattern extraction. Single word in isolation means the technique ignores the exact sequence in which the word occurs and considers only statistics about its existence in isolation. The feature can be boolean (either word exists or does not exist) or frequency based (number of occurrences of word). These features can be further reduced by using different feature selection parameters like odd ratio, relative entropy, mutual information [39] etc. This approach is applicable to both structured as well as unstructured data. It utilizes the internal structure of HTML or XML web documents to carry out its operation efficiently. A summary of various web content mining techniques proposed in past for unstructured documents is given in table 2.4.

Table 2.4: Summary of Related Work on Web Content Mining In IR View for Unstructured Data

Sr. No.	Author(s)	Content consideration	Method Used	Applications
1.	D. Billsus & M. Pazzani [38]	-Bag of words	-Naïve bayes and TF/ IDF	-Text classification
2.	Dagan & Feldman [37]	-Concept categories	-Relative entropy	-Discovering patterns between concepts
3.	Paynter & Frank [42]	-Phrases features	-Naïve Bayes	-Identify key phrases from textual data
4.	Hamzaoglu & Kargupta	-N- gram words	-Decision tree	-Hierarchal clustering and text

	[44]			classification
5.	S. Soderland [43]	-Sentences and clauses	-Association rules	-Learning extraction rule
6.	Yang Willmus [45]	-Bag of words and phrases	-Decision tree	-Event detection
7.	Heinone & Ahonen [36]	-Bag of Words and their position	-Episode rules	-Discovering grammatical rules key phrases

Table 2.4 summarizes some of the research done in the field of web content mining for unstructured data. Most of these research used bag of words to represent the web document. Table 2.5 summarizes the related work in the area of web content mining for structured / semi structured data. It may be observed that work surveyed in table 2.5 use richer representations (mostly hyperlink information) of web documents as compared to the work surveyed in table 2.4.

Table 2.5: Summary of Related Work on Web Content Mining in IR View for Semi Structure /Structured Data

Sr No.	Author(s)	Content consideration	Method Used	Applications
1.	H. Kim, S. Chen [48]	-Relational -Ontology	-Inductive logic programming -Naïve Bayes	-Learning extraction rule -Learning relations between web pages -Hypertext classification
2.	Chaung Haung Lee, Hasin Chang Yang [49]	-Phrases -URLs	-Unsupervised and supervised classification	-Hierarchal and graphical classification
3.	S. Jusoh & S. Osman [47]	-hyperlinks	-Association rules	-Hypertext classification -Clustering
4.	W. Jhang, T. Yoshida [51]	-Bag of words -Hyperlinks	-Reinforcement learning	-Hypertext prediction and

				classification
5.	B. Yu, Z. Xu, C. Li. [50]	- Bag of words -Relational tables	-Neural Network with reinforcement learning	-Hypertext classification
6.	A. Khan, B. Baharudin & L. H. Lee [46]	-concept -Named entity	-Machine association rule	-Pattern identification in semi structured text

In contrast to IR vies, DB view infers the structure of web site so as to transform it into a database for the purpose of better queering and management. It only models and integrates the structured and semi structured data on web into sophisticated queries other than keyword based queries that are able to produce more relevant results when executed on search repositories. This can be achieved by studying schema of web pages and creating virtual database or web knowledge base. Summary of some of the research done in DB view for web content mining is presented in table 2.6.

Table 2.6: Summary of Related Work on Web Content Mining In DB View for Semi Structure /Structured Data

Sr No.	Author(s)	Content consideration	Method Used	Applications
1.	Johnson & Faustina [54]	-Relational	-Attribute induction algorithm	-Hierarchal databases
2.	B. Singh, H. Kumar [53]	-Edge labeled graph	-Association rules mining	-Opinion extraction
3.	R. Malarvizhi & K. Saraswathi [52]	-Strings -Relational	-Proprietary algorithms	-Identifying schema of semi structured data
4.	A. Kumar & P. C. Gupta [56]	-Edge labeled graph	-Proprietary algorithms	-Finding data guide in structured data
5.	Kavita, G. Shrivastava	-Edge labeled	-Upgraded	-Knowledge

	and V. Kumar [57]	graph	-Association rules	synthesis
6.	M. Srividya et.al [58]	-Relational	-k-mean algorithm	-Text classification

From table 2.6, the differences in DB and IR view can be noted easily. DB view mainly deals with Object exchange model (OEM) of web documents that represents semi structured data in the form of labeled graph. Most of techniques that are surveyed above perform the task of schema extraction from semi structured documents and building Data Guide [119][121]. A Data Guide is kind of structural summary of web documents obtained for practical and computational applications in multi-layered databases (MLDB). In MLDB, each layer is constructed by generalizing / specializing the lower layer and use special query language for pattern identification [122]. The method has a great utility in the area of query suggestion.

2.5.2 Web Structure Mining

Web structural mining (WSM) can be regarded as collection of techniques for extracting structural summary of web sites. Based on the topology of web graph, WSM tries to discover the similarity and relationship between websites and web pages from inter and intra structure of web documents. A web graph is basically network of web pages connected by hyperlinks. A sampled web graph is depicted in Fig 2.9. Some popular algorithm such as PageRank [93], WPR [77], HITS[106] based on web structure mining had been proposed in past . They are mainly used to model web topology and calculate the relevancy of web page. A detailed discussion on these algorithms is covered in section 2.7.

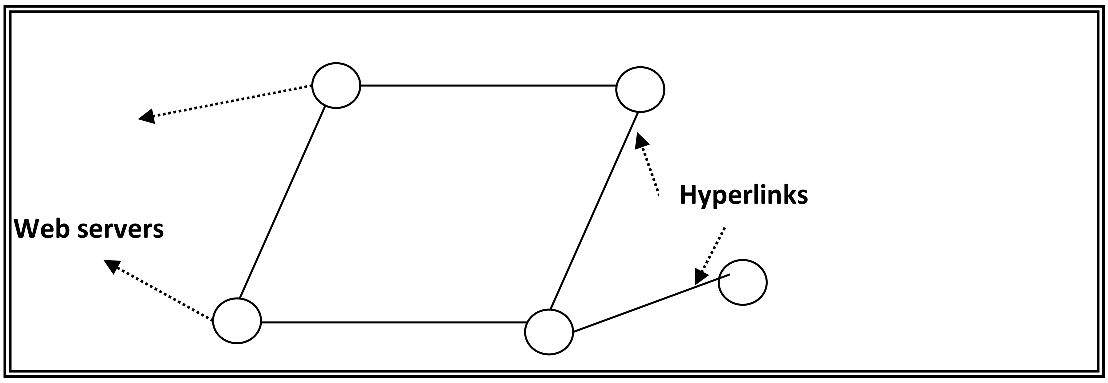


Fig 2.9: Web Graph

Further web mining can be divided into two types based on kind of data used.. They are:

- ***Hyperlink analysis:*** A hyperlink can be viewed as a linking unit that connects different parts of same document (Inter- document links) or connects the different web pages (intra documents links) on the web. These are basic unit of analysis in hyperlink based web structure mining. It can be applied on social networks to model the underlying link structure of web. A significant amount of survey on hyperlink analysis is documented by Bing Lu [10].

- ***Content analysis:*** In addition to link summary, some WSM techniques also utilize the HTML and XML tags to organize the content of web pages in tree-form generally known as DOM objects (Document Object Model) [111] [134].This type of research model is mainly inspired by social network and citation analysis [96] [112] For instance, the research carried out by Yung [126] utilizes the network of people to discover the structural summary of AI researcher. The research tries to find out the author entity in close proximity of any web page such as co-author entity, organization chart, citation of research papers and any exchange of information found in net activities.

In fact most of search system adds content information to link structure during knowledge extraction e.g Goggle [125] and Clever System [127].

The major applications of web structure mining include web page categorization, rank computation, discovering micro communities on web and elimination of mirrored web sites [124].

2.5.3 Web Usage Mining

Web usages mining aims to predict user behavior when the user interacts with web. The usage data can be navigation template, user profile, site content, web topology, concept hierarchy and syntactic constraint [50][80]. The data can be obtained by processing user profile or adapting user modeling interfaces (personalized method) or learning user navigational patterns (impersonalized method). The usages data is stored and maintained in server logs, proxy logs and client side logs [106]. Further, Web usages mining studies can be classified into two main approaches.

The first approach directly processes the log data to discover useful patterns whereas the second approach transforms the server log data into relational tables before applying any data mining technique. The major challenges faced in web usages mining are distinguishing among server sessions, episode, maintaining and updating user profiles [136].

The major applications of web usages mining include search result personalization, site modification, system improvement, usages characterization and business intelligence. The more detailed study on web usage mining is covered in [128].

The overall summary of web mining techniques is given in table 2.7. It may be noted from the table 2.7, web usage mining is most useful technique among various categories of web mining because of the following reasons:

- Determine the life time value of the user
- Improve business strategies
- Improve web searching environment
- Provide personalized results

Table 2.7: Overall Summary of Web Categories

Features	**Web Mining Categories**			
	Web Content Mining		**Web structure mining**	**Web Usage Mining**
Mining focus	Within the document	Within the document	Within as well as between the documents	User navigational patterns
View of Data	Unstructured , semi structured	Semi-structured, web site as database	Link structures	User interaction
Input Data	Text document , HTML, XML documents	HTML, XML documents	Web graph	Server log, client log, proxy log
Representation of Data	Bags of words, phrases , concept hierarchy	OEM, Relational tables	DOM objects, web Graph	Relational tables

Method	TF/IDF, association rules, NLP methods	Proprietary algorithm, association rules (modified)	Proprietary algorithm	Machine learning, statistical (NLP), personalization algorithm
Applications	Text classification and clustering, pattern recognition, extraction rules	Web site schema discovery, building data guide. opinion extraction	Categorization, clustering, page ranking, Business intelligence	User modeling, web personalization

Since goal of various web mining techniques is to capture, model and analyze the web data so as to improve the user searching experiences. So the next section attempts to survey the role of web mining in Query Suggestion.

2.6 QUERY SUGGESTION

User searches information on web by submitting query at search engine interface. The keywords of query play vital role in evaluating the relevancy between document and query. Web search queries are distinctive from SQL queries in the sense that they are simple, unstructured (does not follow any syntax), and ambiguous in nature. These queries can be broadly classified into four categories:

1. ***Navigational Queries:*** Queries that seek a particular webpage/website such as “Facebook”, “Linkedin india”, “Amazon” rather than entering specific URL of web site in browser navigational bar fall under this category.

2. ***Informational Queries:*** Queries that target quite broader topic such as “car”, “animal” for which there may be millions of matching pages are classified under this category.

3. ***Transactional Queries:*** Queries that implies the intent of a user to perform a action such as purchase of a mobile or downloading of some online software (e.g. price of Samsung S8 or Hp laser printer 1020 etc.) fall under this category.

4. ***Statistical Queries:*** Queries that relates to some survey conducted for particular

time period (e.g. indexed web size or growth of reliance share etc.) fall under this category.

User of search engine generally issue short, imprecise and ambitious query that often leads to the inclusion of non desirable documents in the search results. Query recommendation has evolved as powerful method to assist user in query formation phase. Popular search systems like Google and Bing offer query recommendations to their users through *"search related to"* section given at the bottom of search result page. Query recommendation methods can be broadly grouped into four categories. They are as follows:

A. Query expansion

B. Query flow graph

C. Query association

D. Query clustering

The first method, query expansion helps the user by adding related terms to original query for effective retrieval but it may sometime adversely decreases the precision of search results [119]. Second method, query flow graph stores the past queries submitted by the user in single session in the form of graph and helps the user by offering these past queries, Keyword ambiguity and session segments are major issue with this approach. In the third approach, association rule mining is applied to discover the related queries in different sessions. Here also session segmentation is problem. Fourth method query clustering group the queries based on their click – through data. For this purpose, Query logs are maintained at search engine sites. Query logs contain the information about query issued by the user, list of URLs clicked by user for particular query, time, number of clicks etc. Many different approaches have been proposed in past to discover essential knowledge from query log. For instance, Query clusters can be formed by extracting knowledge about term-term pair, term-document pair or query-document pair from query log. Since the query clustering methods are developed in the present work therefore a detailed review of query clustering is presented in the next subsection.

2.6.1 Query Clustering

Classifying the set of objects into defined groups (called clusters) in such a way that objects in the same group share similar properties with each other than to those in other group is known as “clustering” it is very popular technique of data mining and can be applied in many areas including pattern recognitions, information retrieval, image processing, .data compression, statistical data analysis and bio informatics. In information retrieval field, it can be used to group similar documents, queries, users or even sessions in potential cluster. The major clustering method can be grouped in three categories [129]:

- ***Portioning based clustering:*** This method aims to divide the data space containing ‘n’ objects (that may be queries or documents) into ‘k’ partitions or clusters where each cluster satisfy the following conditions [120]:

  - Empty cluster is not allowed. It must contain at least one object.
  - An object can belong to only one cluster.
  - Objects in one cluster must be very closely related whereas objects belonging to different clusters must be very far away means share very less, near to null properties in common.

  The clustering is computationally NP hard problem however there are some popular heuristic algorithms such as k-mean clustering, k- medoids, and nearest centroids that can be employed to cover local optimum. The k-means algorithm is given in Fig 2.10. In k-mean algorithm, initially the number of clusters i.e. value of k is decided. Then the algorithm randomly picks center of these k clusters. If ‘k’ is greater than the no. of objects then each object constitutes a separate cluster and act as centroid of its cluster. Otherwise the algorithm computes Euclidean distance between each object and centroid of cluster.

  The object is allocated to the cluster having minimum Euclidean distance .As the location of centroid is chosen randomly, so the algorithm needs to revise the centroid location with respect to updated information. The process is terminated when the centroids move by negligible distance in successive iterations

- ***Connectivity based clustering:*** Connectivity based clustering also called as Hierarchal clustering groups the objects in tree- form structures also known as dendrogram [125]. Each node of the dendrogram represents the object (query or document) and edge represents the similarity between the objects as shown in Fig

2.11. Two approaches can be considered to carry out hierarchal clustering. They are:

```
k-mean ()
Input: A set of n objects, value of  k
Output: k-clusters
Method:
Step1: Randomly pick cluster centroid;
Step 2: while (!convergent ) {
        2.1. For each object ‘o’ € D do
            2.1.1. Find the closest cluster ‘c’ whose centroid matches with ‘o’
            2.1.2. Allocate ‘o’ to ‘c’
            2.1.3. Recomputed centroid of ‘c’}
```

Fig 2.10: K-Mean Algorithm for Object Clustering

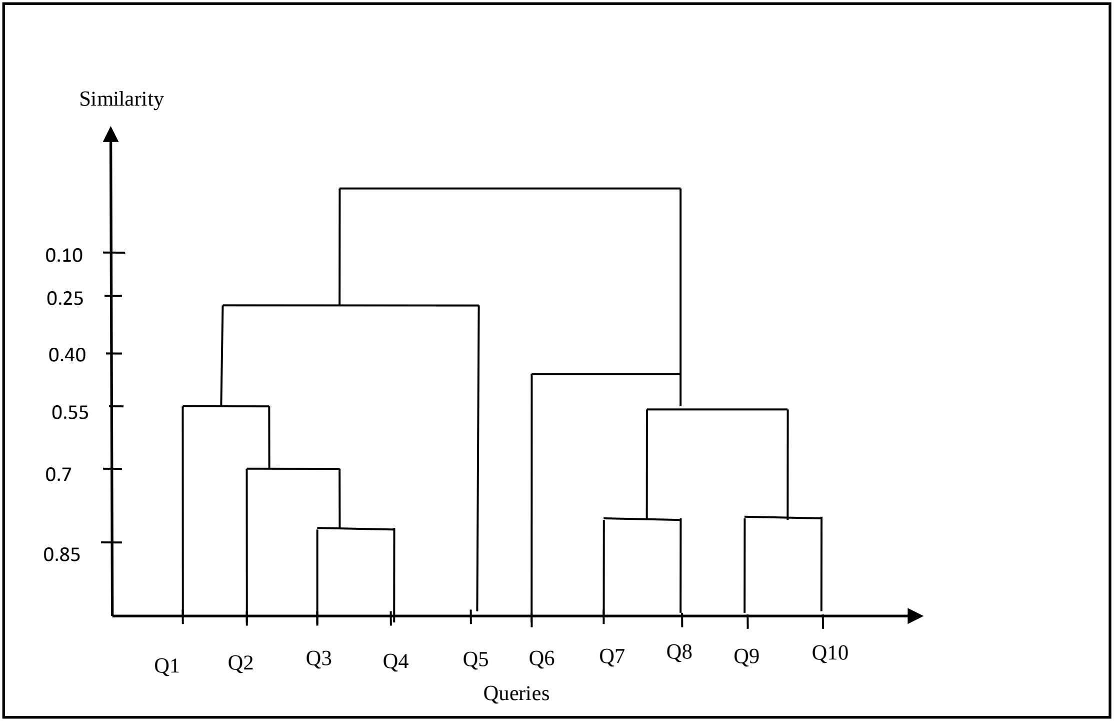


Fig 2.11: A Sample Dendrogram with 10 Objects (Queries)

Top down approach: The top down also called as divisive approach begins by initially putting all the objects in same cluster and successively splitting into smaller clusters until each cluster contains only one object or termination condition is achieved.

Bottom up approach: The bottom up approach also called as agglomerative clustering begins by building separate cluster for each object and successively merging the clusters until a single cluster is formed or termination condition is achieved. The algorithm for agglomerative clustering is given in Fig 2.12 [118].

- ***Density based clustering*****:** he previous two clustering methods only discover the clusters of spherical shape but main focus of density based method is to form clusters of objects of arbitrary shape. Here, the clusters are defined in area of high density. For each object, if the objects in it neighborhood exceeds a thresh hold value, then they are cluttered together. The most popular density based clustering method is DBSCAN [128].

```
Agglomerative Clustering ()
Input: A set of n objects
Output:  A hierarchy of clusters
Method:
Step 1: Create n clusters with each cluster having single object
Step 2:  Clust={$c_1$,$c_2$,.....$c_n$}                    //set of clusters
Step 3:  while |Clust| >1 {
        3.1 For all clusters i, j $\in$ Clust {
                3.1.1    If sim(i,j) < threshold{
                    k= i $\cup$ j;
                    Delete i , j from Clust;
                    Clust = Clust $\cup$ k;
}}}
```

Fig 2.12: Agglomerative Hierarchal Clustering Algorithm

Some of the researches conducted in recent years in the area of query clustering have been reviewed and presented here.

Beeferrman and Berger [1] applied hierarchical agglomerative clustering on click-through data archived in server logs to find clusters of queries that lead to selection of same URL. For this purpose, bipartite graph is constructed by taking one set of nodes as queries and other set of nodes as URLs. An edge between a query node and URL node is introduced whenever the URL get selected corresponding to a query by some user. This method is quite impressive in the sense two queries can be clustered in same group even if they do not share common keywords.

K.Khan. analyzed query keywords as well as click through data and applied DBSCAN clustering method to form cluster of similar queries. The high computation cost is major drawback of this method [108].

In [114] query flow graph based method is proposed to cluster similar queries in one group. An edge is introduced in query flow graph if the queries are part of same search mission. For this purpose, time and contextual properties of queries are analyzed.

Ji Rong Wen utilizes query content and user feedback to group the similar queries in single cluster [145]. The approach performed well but had two limitations: i) It is difficult to decide the parameters for user feedback ii) It is difficult to set parameters for combining the two metrics together.

Yuan Hang et. al. [123] found the similarity between two queries by analyzing the ranks of clicked URL in result set. The advantage of this method is its scalable nature and disadvantage is high computational cost.

Qiazhu & Dengyo proposed the method for personalized query suggestion using click time and semantics of current query. But generation of personalized suggestion is still a problem. Xiaochum & Muyum proposed incremental clustering algorithm as a solution to this problem [118]. It considers the browsing history of each user by streaming on-line sources and presents a personalized search model that keeps on changing with every new interaction of user with search system using incremental approach. The algorithm performs well as user latest interest and query semantics were taken into account but the algorithm is not scalable and failed to deal with long term interest of user.

Unlike previous methods, Hang Cui proposed a new approach for query suggestion that utilizes the session data and click through data to group the similar queries in same cluster. It not only considers the current query but also the recent queries in same sessions to offer more meaningful query suggestions to user. Moreover it also extracts the concept behind the query and also forms query suggestion using extracted concepts.

A novel query suggestion model based on user re-querying activities is presented in [122]. The approach specifically analyzed query reformation data such as adding of new terms, deleting of previous terms or modifying of existing terms and constructed the scalable transition graph.

An entirely new method of query expansion based on correlation between query terms and document terms is presented in [93]. These correlation factors are used to select high quality expansion terms for future queries. The method is effective for short queries only. The summary of various query suggestion method is laid down in table 2.8.

Table 2.8: Summary of Query Suggestion Techniques

Author	**Method Applied**	**Advantages**	**Disadvantages**	**Application categories**
DougBeefer Man, AdamBerger [1]	Hierarchal agglomerative clustering	Queries with dissimilar content but similar intent are identified	Less data available for analysis, Query contents are totally ignored	Query suggestion
Ji-Rong Wen, JianYun Nie, HongJiang Zhang [145]	Proprietary algorithm	Queries with low frequency are easily filtered out and rest of data is used to produce FAQs, Nanual setting to form clusters are not required.	Clustering is based on keyword similarity and produce accurate results for short queries.	FAQs
Paolo Boldi,	Query Flow	More satisfied	Not enough	Query

France Bonchi, Carl Castillo [114]	graph	results by mining user behavior, Improved server log analysis	parameters for user characterization	suggestion
Ji-Rong Wen, JianYun Nie, HongJiang Zhang [145]	User feedback, Content analysis	More relevant query suggestion as combined approach of web content and usage mining is adopted.	Query semantics are not considered	Query suggestion
Yuan Hung,J aidee p V aidya, Haibin g Lu [126]	Modified k-mean clustering	Scalable	No user behavior characterization	Query suggestion, search result rank optimization
Qiaozhu Mei, Dengyo & Zhou [123]	User characterization	Personalized query suggestions	More computational cost	Query suggestion
Xiaochun Wang, Muyun Yung [93]	Incremental clustering, Content analysis	User specific results	User long term interest is not considered	Personalized search
Sharma et.al. [124]	Association rule mining	Seek time to obtain search result is reduced	Only link analysis is done	Search result optimization
Hang Cui, Wei-Ying Ma [36]	Query log analysis	Narrow gap between query and document	Not applicable to long queries	Query expansion

The next section covers the detailed review on prevalent ranking algorithms.

2.7 WEB PAGE RANKING

With the vast amount of information and exponential growth of web, it become increasingly difficult to provide relevant information to the user based on their precise queries. Some of the reasons behind the problem include non descriptive nature of web page content, ambiguous queries and fake embedded links on web documents that are purely created for navigational purpose. So searching the appropriate pages of high quality through search systems that rely on page content and link popularity is tedious task.

To address the problem mentioned above, various algorithm based of web mining had been proposed in past. Some of these algorithms utilize the content inside the page to compute relevance score whereas some assign rank score based on link popularity of web page in overall structure of web by extracting link information using web structure mining techniques and still other are based on user behavior characterization [80].

Some popular page ranking algorithms include PageRank (PR), Weighted page rank (WPR), Hypertext induced topic search (HITS), Page content rank (PCR), Page rank based on number of visits (PRLV) etc. The discussion on these prevalent algorithms is covered in next subsections.

2.7.1 Pagerank Algorithm

The pageRank algorithm was developed by Larry Page and S. Brim. It is based upon citation analysis of a web page to find its importance in the corpus [78]. According to this algorithm, if the incoming links of a page are important then its outgoing links also become important. So page rank of any page ‘n’ is equally divided among its outgoing links that also further, propagated to their corresponding outgoing links. The page rank of a page n can be calculated by eq^{n} (2.1) as given below.

$$PR(n) = (1 - d) + d \sum_{m \epsilon I(n)} \frac{PR(m)}{N_m} \tag{2.1}$$

Where:

- *PR(m)* and *PR(n)* represent the page rank of page *m* and page *n* respectively.
- *I(n)* is set of incoming links of page *n*.
- N_m represents the no. of outgoing links of page *m*.
- *d* is the damping factor that measures probability of user following direct link. Its value is usually set to 0.85.

Example Illustrating Working of PR

To explain the working of page rank algorithm, let us take an small hyperlinked structure shown in Fig 2.13, consisting of three pages X, Y and Z. where page X links to the page Y and Z, page Y links to page Z and X, page Z links to page X and Y.

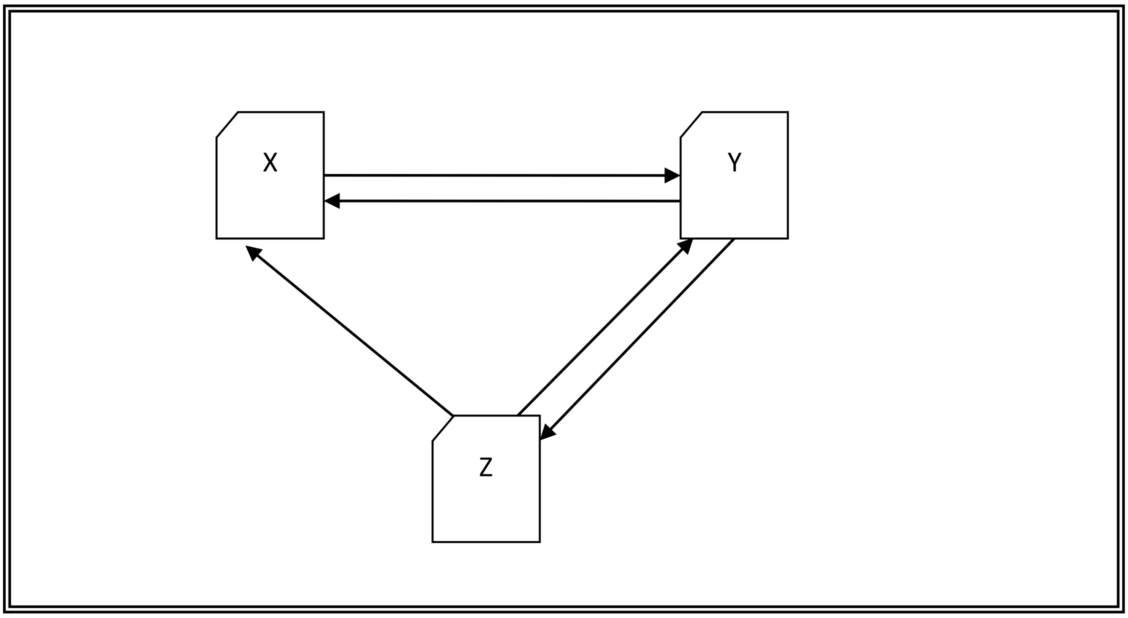


Fig 2.13: Example - Hyperlinked Structure

According to equation (2.1), Pagerank of page X, Y , Z can be computed as follows:

PR(X) = [(1-d) +d (PR (Z)/2)+PR(Y/2))] *2.1(a)*

PR(Y) = [(1-d) +d (PR(X)/1+PR (Z)/2)] *2.1(b)*

PR (Z) = [(1-d) +d (PR(Y)/2)] *2.1(c)*

Initially, considering the page rank of each page equal to 1 and taking the value of d=0.5,, the new page rank of pages can be computed as follows:

$$PR(X) = 0.5 + 0.5\left(\frac{1}{2} + \frac{1}{2}\right) = 1.0$$

$$PR(Y) = 0.5 + 0.5\left(1 + \frac{1}{2}\right) = 1.25$$

$$PR(Z) = 0.5 + 0.5\left(\frac{1.25}{2}\right) = 0.75$$

Calculating page rank of each page by using iteration method as shown in table 2.9..

Table 2.9: Calculation of Page Rank by PR Method

(k)	**PR(X)**	**PR(Y)**	**PR(Z)**
1	1.0	1.0	1.0
2	1.0	1.25	0.75
3	1.0	1.19	0.81
4	1.0	1.2	0.8
…	…	…	…

From the above table, it may be noted that PR(Y) > PR(X) > PR (Z). These PR values are extracted by crawler while downloading a page from web server and these values will remain same till the web link structure will not change. In order to obtain the overall page score of a page , the query processor add the pre-computed pagerank(PR) value associated with the page with text matching score of page with the user query before presenting the results to the user.

2.7.2 Weighted Page Rank

Weipu Xing et.al [77] proposed an algorithm which is extension of basic page rank algorithm. It assigns the page rank on the basis of link popularity of both incoming and outgoing links. The page rank of page n is computed by eq^{n} (2.2) given below:

$$PR(n) = (1-d) + d \sum_{m\epsilon I(n)} PR \times \frac{I_n}{\sum_{p\in R(m)} I_p} \times \frac{O_n}{\sum_{p\in R(m)} O_p} \qquad (2.2)$$

Where:

- *PR(m) and PR(n)* are page rank of page m and n respectively

- *d* is damping factor as discussed earlier
- *R(m)* denotes the reference list of page m
- I_n and I_p denote the no. of incoming links to page *n* and page *p* respectively.
- O_n and O_P denote the no. of outgoing links of page *n* and page *p* respectively.

Example Illustrating Working of WPR

By considering the same hyperlinked structure as shown in Fig 2.13 and initially taking weighted page rank of each page equal to 1 and d=0.5, the new weighted page rank of pages X. Y and Z can be computed by using eqn. 2.2 as follows:

$$PR(X) = 0.5 + 0.5\left(\left(1 \times \frac{2}{3} \times \frac{2}{3}\right) + \left(1 \times \frac{1}{2} \times \frac{1}{3}\right)\right) = 0.5 \qquad 2.2(a)$$

$$PR(Y) = 0.5 + 0.5\left((1 \times 1 \times 1) + \left(1 \times \frac{1}{2} \times \frac{1}{3}\right)\right) = 1.0 \qquad 2.2(b)$$

$$PR(Z) = 0.5 + 0.5\left(1 \times \frac{1}{2} \times \frac{1}{3}\right) = 1.1 \qquad 2.2(c)$$

Calculating weighted page rank of each page by iteration method as shown in table 2.10.

Table 2.10: Calculation of Page Rank by WPR Method

Iteration(k)	**PR(X)**	**PR(Y)**	**PR(Z)**
1	0.5	1.0	1.1
2	0.3	0.9	0.6
3	0.7	0.9	0.6
4	0.7	0.9	0.7
…	…	…	…

From the above table, it may be noted that PR(Y) > PR (X) = PR (Z). The order of page rank is different from PR method.

2.7.3 Page Ranking Based on Link Visit

Duhan et al [79] identified the limitation of traditional PR method that it evenly distributes the page rank of page among its outgoing links whereas it may not be always the case that all the outgoing links of a page holds equal importance. S0, they proposed a method which assigns more rank to an outgoing link that is more visited by the user. For this purpose a client side agent is used to send the page visit information to server side agent. A database of log files is maintained on the server side that stores the URLs of the visited pages, its hyperlinks and IP addresses of the users visiting these hyperlinks. The visit weight of a hyperlink is calculated by counting the distinct IP addresses who clicked the corresponding page. The page rank of page 'm' based upon visits of link is computed by the eq^{n} (2.3).

$$\boxed{PR(n) = (1-d) + d\sum_{m\epsilon I(n)} \frac{PR(m) \times LV(m)}{TV(m, O(m))} \qquad (2.3)}$$

Where:

- *PR (m)* and *PR (n)* are page rank of page *m* and *n* respectively.
- *I (n)* denotes set of incoming links of page *n*.
- *LV (m, n)* is no. of link visits from *m* to *n*.
- *TV (m, O (m))* total no. of user visits on all the outgoing links of page *m*.

Example Illustrating the Working of PRLV

Consider the hyperlinked structure as shown in Fig 2.14. Let the no. of visits from page X to page Y are 100; no. of visits from page Y to X are 45 and the no. of visits from page Y to Z are 15; the no. of visits from page Z to Y are 50 and the no. of visits from page from Z to X are 25. The Page rank based on link visit can be easily calculated using eq^{n} (2.3). Initially taking page rank of each page equal to 1 and d = 0.5.

$$\boldsymbol{PR(X) = 0.5 + 0.5\left(\left(1 \times \frac{45}{45+15}\right) + \left(1 \times \frac{25}{25+50}\right)\right) = 1.0 \qquad 2.3(a)}$$

$$\boldsymbol{PR(Y) = 0.5 + 0.5\left(\left(1 \times \frac{100}{100}\right) + \left(1 \times \frac{50}{25+50}\right)\right) = 1.3} \qquad \mathbf{2.3(}\boldsymbol{b}\mathbf{)}$$

$$\boldsymbol{PR(Z) = 0.5 + 0.5\left(1 \times \frac{15}{15+45}\right) = 0.6} \qquad \mathbf{2.2(}\boldsymbol{c}\mathbf{)}$$

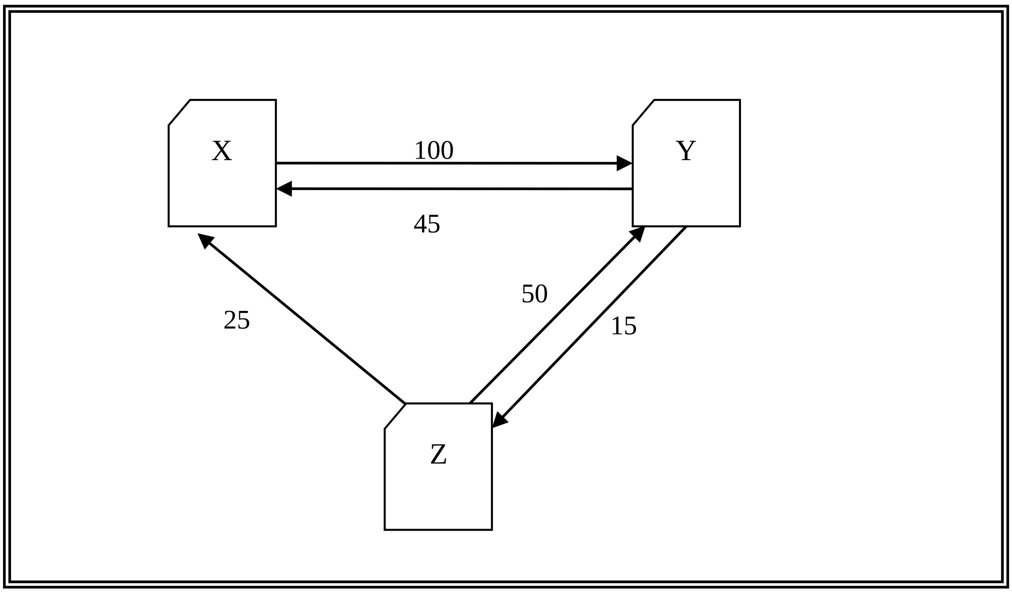


Fig 2.14: Example- Hyperlinked Structure with Visits of Link

Calculating page rank based on link visit of each page by iteration method as shown in table 2.11.

Table 2.11: Calculation of Page Rank by PRLV Method

Iteration(k)	**PR(X)**	**PR(Y)**	**PR(Z)**
1	1.0	1.0	1.0
2	1.0	1.3	0.6
3	1.1	1.2	0.6
4	1.0	1.2	0.6
…	…	…	…

From the above table, it may be noted that PR(Y) > PR(X) > PR (Z) .The order of page rank is different from above two methods. By comparing the above methods results, it is found that page Y is obtaining highest precedence over other pages.

2.7.4 Hypertext Induced Topic Search

Hypertext Induced Topic Search, a precursor to *PageRank* algorithm, was developed by Jon Kleinberg in 1998. It is based on both web content as well as web structure mining. It divides the whole web into hub and authority pages. The idea behind hub and authority originated from a particular insight into web page creations process. Hubs are those web pages that do not actually contain useful information but serve as large directories that point to other useful pages called as authority. Hubs and authorities are illustrated in Fig 2.15.

Step 1: Formation of Base Set

The first step is to create base set of most relevant pages related to search query. The reason behind constructing the base set is to ensure that most of the strongest authorities

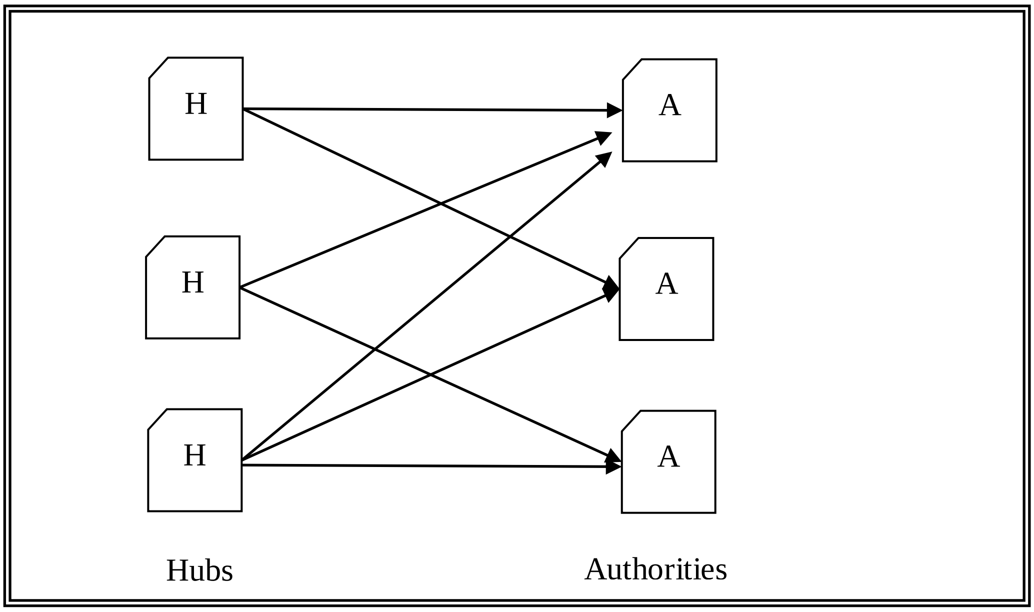


Fig 2.15: Hubs and Authorities

are considered [105]. The base set can be obtained by taking the top n pages returned by a general text based search engine in response to a query (also called as root set) and augmenting the root set with some of the pages that are linked to and from it as shown in Fig 2.16

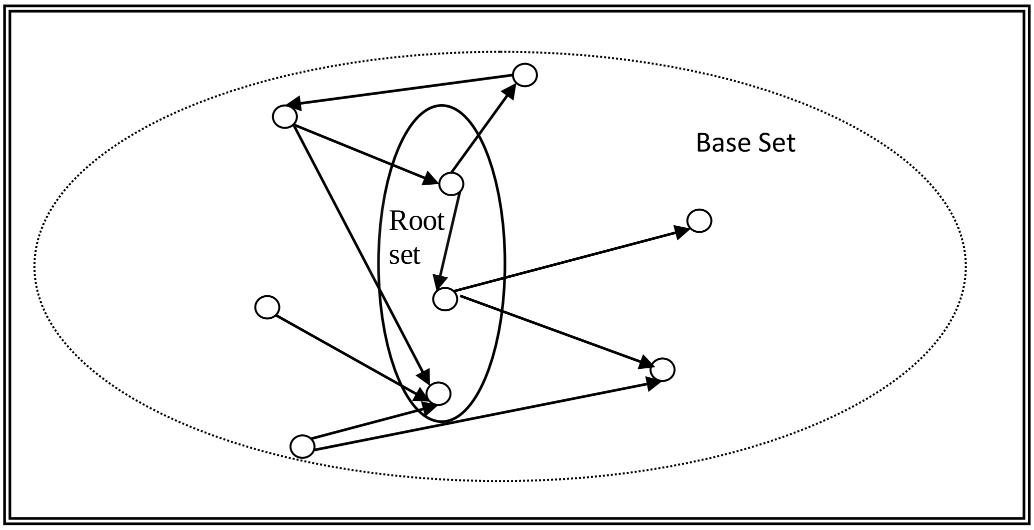


Fig2.16: Creation of Base Set

Then a web graph is constructed by taking web pages from base set. This type of web graph is termed as focused web graph. Further computation will only be performed on participants of focused graph [133]. The algorithm for constructing Focused graph is given in Fig 2.17.

Step 2: Finding hubs and authority

Initially hub and authority value of each node in focused graph is set to some initial value say H_p and A_p respectively. For convenience, it is set to 1. Then authority and hub value is updated iteratively as follows:

Update Authority: Updates each node's authority value equals to sum of hub values of all the nodes that points to it. It is based on the assumption that a node is more authoritative if it is pointed by many hub pages. The formula to calculate authority

value A_p of page P is given by eq^n (2.4).

$$A_p = \sum_{q \in B(P)} H_q \tag{2.4}$$

```
Focused Graph Construction ()
Input: Root set, R
Output:  Focused graph, F
Method:
Step 1: Set B = R;
Step 2: Repeat step 3 to 5 for all P € B
Step 3:  Find set 'O' of all pages pointed by P
Step 4: Find set 'I' of all the pages that points P
Step 5: Update B = B + O + I;
Step 6: Delete all duplicate links
Step 7: Construct the focused graph for base set B
```

Fig 2.17: Algorithm for Base Set Construction

Where:

- *Ap* denotes a authority page and *Hq* denotes the hub page.
- *B(P)* denotes set of referrer pages of page *P*.

Update Hub: Update each node's hub value equals to sum of authority values of all the nodes pointing by it. That is a node is assigned a high hub value if it points to pages that are authoritative the subject. The formula to calculate hub value H_p of page *p* is given by eq^n (2.5).

$$H_p = \sum_{q \in R(P)} A_q \tag{2.5}$$

Where:

- A_q denotes a authority page and H_p denotes the hub page.
- *R (P)* denotes set of pages referenced by page *P*.

Step 3: Normalize hub and authority value

Normalize the updated value of hub and authority by dividing each hub value by square root of sum of squares of all hub values and dividing each authority value by square root of sum of squares of all authority values. The normalization step is repeated till termination condition is achieved. The termination condition for a page is achieved when sum of squares of its hub and authority value comes out to be 1. The algorithm for finding hub and authority is given in Fig2.18.

Finding hub and authority ( )

Input: Focused graph, F

Output: Set of hubs and authorities

Method:

Step 1: Set initial value of H_p and A_p of each page, P in focused graph equals to 1

Step 2: Repeat step 3 and 4 till termination condition is achieved

Step 3: Update H_p and A_p of each page by using eq^n (2.4) and (2.5)

Step 4: Normalize H_p and A_p of each page so that their squared sum equals to 1

Step 5: Pages with relatively higher value of Ap are classified as authorities and pages with relatively higher value of Hp are classified as hubs

Step 7: Construct the focused graph for base set B

Fig 2.18: Algorithm for Finding Hubs and Authority

HITS like PageRank algorithm is a recursive algorithm that is based on web structure mining but it suffers from certain limitations as listed below:

- ***High seek time:*** The algorithm takes quite long time to give response to user query because identification of hubs and authorities gets initiated only after recievng the user query and then rest of the ranking computation is done. So the performance of tHITS is not good in real time.

- ***Lack of trnasperency:*** There is no well defined boundaries between hubs and authorities since same web site can act as hub as well as authority for the same query . This lead to inaccuracy in rank computation.

- ***Irrelevant Hubs/ Authorities:*** Since the selection of root set is dependent on the top *n* results provided by text based search engine which further affects the final identification of hubs and authorities. So in the scenarios, where root set obtained is irrelevat for some queries leads to identification of irrelevant hubs and authorities and hence wrong rank computaion.

- ***Topic drift:*** Topic drift occurs when the root set contains non relevant pages and that too are strongly connected with each other. Since the root set itself is not of good quality so same is reflected in base set and thus in focused graph.
- ***Less feasibility:*** As mentioned earlier , the algorithm computes rank values at query time so it is not a feasible solution in the scenarios where a search engine has to handle millions of query per day.

Many variants of above discussed page ranking algorithms have been proposed by researches in recent years such as weighted page rank based on link visit [78], time based ranking, [116], distance based ranking [120] , probabilistic HIITs [105] etc. The comparison between prevalent page ranking algorithm is given in table 2.12.

Table 2.12: Comparison of Page Raking Algorithms

Algorithm → **Features ↓**	**PageRank (PR)**	**Weighted page rank (WPR)**	**Hypertext induced topic search (HITS)**	**Page rank based on link visits (PRLV)**
Technique applied	Web structure mining	Web structure mining	Web structure mining, web content mining	Web structure mining, web usage mining
Input parameters	Hyperlinks	Hyperlinks	Content, hyperlinks	Hyperlinks, user navigational history
Rank computation stage	At indexing time	At indexing time	At query time	At indexing time
Rank Distribution	Equally distribute to outgoing links	Unequal	-	Unequal
Relevancy	Less	Less (higher than PR)	More (higher than WPR)	More (higher than PR)

Quality of search results	Medium	More than PR	Less than PR	More than PR
Complexity	O(Log n)	< O(Log n)	< O(Log n)	> O(Log n)
Limitations	No focus on user query , evenly distribution of rank to outbound links	No focus on user query, Extra effort of client	Topic drift, relevancy problem, High seek time	Biased click weight, high computational cost

Beside query suggestion and page ranking, other important task that a search engine needs to focus is development of efficient crawling strategies to download the documents from widely spread heterogeneous web sites containing volatile information. A detailed discussion on web crawling strategies is covered in next section.

2.8 WEB CRAWLING STRATEGIES

As already discussed in section 2.3.2 that a crawling process starts with a seed URL and attempts to precede the embedded links in the downloaded page. A web page can be viewed as an arrangement of hyperlinks in tree structure form as depicted in Fig 2.19. The root is the seed URL and links embedded in the HTML page corresponding to the seed URL act as children to the root node.

There are many ways in which a crawler can traverse the links and collects the information from web servers [66]. Some basic traversing strategies are:

Breadth First Search (BFS): In breadth first search, the crawler starts with the root page and follows all the links embedded in it. After exploring all the links present in the root page, it proceeds with the next level and explore all the siblings present in it and so on [19]. since the method does not consider the relevancy of path while traversing so it is also called as Blind search method [72].

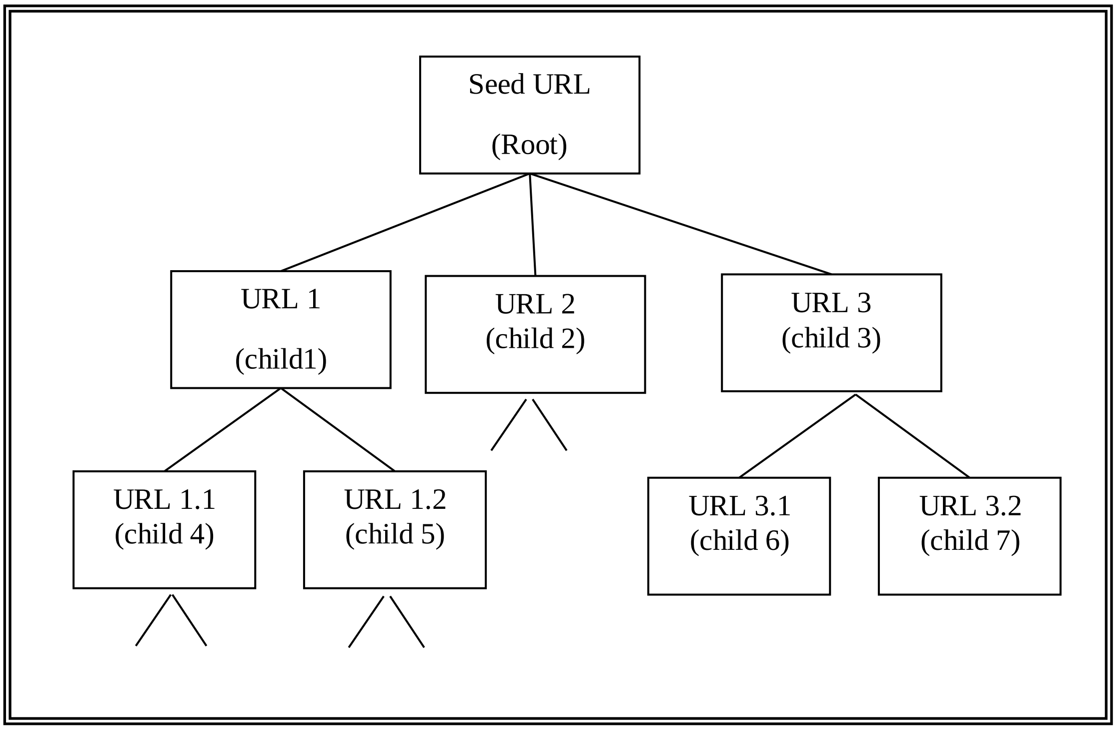


Fig 2.19: Tree Structure Arrangement of Hyperlinks in a Web Page

- ***Depth First Search (DFS):*** In depth first search, a web crawler starts from the root node and follows the left most child at first level to last level. After reaching at the end for first left most children, it tracks to the next unvisited node and continues the process till children of each node visited once. The method performs well in case of sparse web graph because there is no certainty about its termination in dense graph having huge number of nodes. In comparison to DFS, BFS performs well for both sparse and dense graph but consumes more time and memory for dense graph [100].

- ***Best First Search:*** Best first search is a heuristic based link traversing strategy that ensures that crawler preferentially pursue promising crawling path. At each level, it attempts to calculate the relevancy of links (in most cases rank score) and proceeds towards the node with highest relevancy value [71]. Thus every time best available node is selected for traversal. A* is example of best first search algorithm. The algorithm for A* algorithm is given in Fig 2.20.

```
A*()
Input: Seed URL
Output: URL with highest relevancy value
Method:
Step 1:  Start with the given seed URL as input
Step 2: Repeat step 3and 4 till Frontier is not empty
Step 3:  Remove front URL from Frontier
Step 4: Download the web page related to fetched URL
Step 5: Repeat for every child node of downloaded page
        5.1 Calculate relevancy value till the downloaded page
        Relevancy_val (downloaded_node) = Relevancy_val (topic,
        donloaded_node, web page)
        5.2 Calculate total relevancy value of the path to goal node
        Relevancy_val (child_node) = Relevancy_val (topic, goal web page) +
        Relevancy_val _X (child_node)
Step 6: Add the link with maximum relevancy value to Frontier
Step 7: Construct the focused graph for base set B
```

Fig 2.20: A* Algorithm

- ***Fish Search Algorithm:*** Fish search is dynamic heuristic web traversal algorithm that works on the principle that relevant pages are often surrounded by relevant neighbors [20]. Here relevancy is determined on the basis of user query. It starts with a relevant link and goes deeper down the link until an irrelevant link is identified. To achieve this, it analyzes relevancy of document with the query and if the document is found to be relevant it sets its relevancy score as 1 otherwise 0. The relevant page is further explored for its children and the same process repeats for them. When relevancy value of some node comes out to be 0, the search direction drops at that node and none of its children is inserted in the frontier.

- ***Shark Search***

  Shark search is successor of fish search algorithm. It improves fish search algorithm in three ways [103]:

  - It uses TF/IDF measure along with cosine similarity measure to determine relevancy of a page.
  - Besides, it also considers meta information and anchor text in calculating the relevancy score.
  - It changes binary score of relevancy as 0 and 1 to fuzzy score defined in [0,1] Fuzzy relevance score has a direct impact on the priority list. Because the parent score is also propagated to its descendent nodes thus automatically increase in fuzzy score linked to grandchildren of a relevant mode than to the grandchildren of an irrelevant mode [125].

- ***Tunneling***

  Shokouhi et al. [63] assumed that sometimes web pages related to specific topic do not directly link to each other and thus it is necessary to go through several off-topic pages to get to the topic related page. The weakness of method is its lack of ability to accurately model pages that can tunnel to on-topic pages. Two remarkable projects based on tunneling are context-graph-based crawler [61] and Cora’s focused crawler [131]. Based on how the crawler retrieve the web pages from web servers and maintain its up to date copy, they can be classified into several categories. Some of the prevalent web crawler categories are discussed in next section.

2.9 TYPES OF WEB CRAWLER

Several web crawling techniques [61] are in use that differ in their mechanism, implementation and objective. Some important web crawling techniques are given below:

2.9.1 Parallel and Distributed Crawler

As web is dynamic and growing at a staggering rate, the biggest challenge for search engine is how to crawl the complete web. The single crawling process even with multithreading is not sufficient to download large volume of data rapidly because only single physical layer is available to fetch and pass the data. So many search engines often employ multiple crawler instances to maximize the web coverage. The technique not only helps in downloading of significant portion of web but also speed up the crawling process with reduced hardware requirements. Executing the crawling process via multiple instances also results in development of scalable, easily configurable, and robust search system. Fig 2.21 shows general architecture of parallel crawler proposed by Jungo Cho [121].

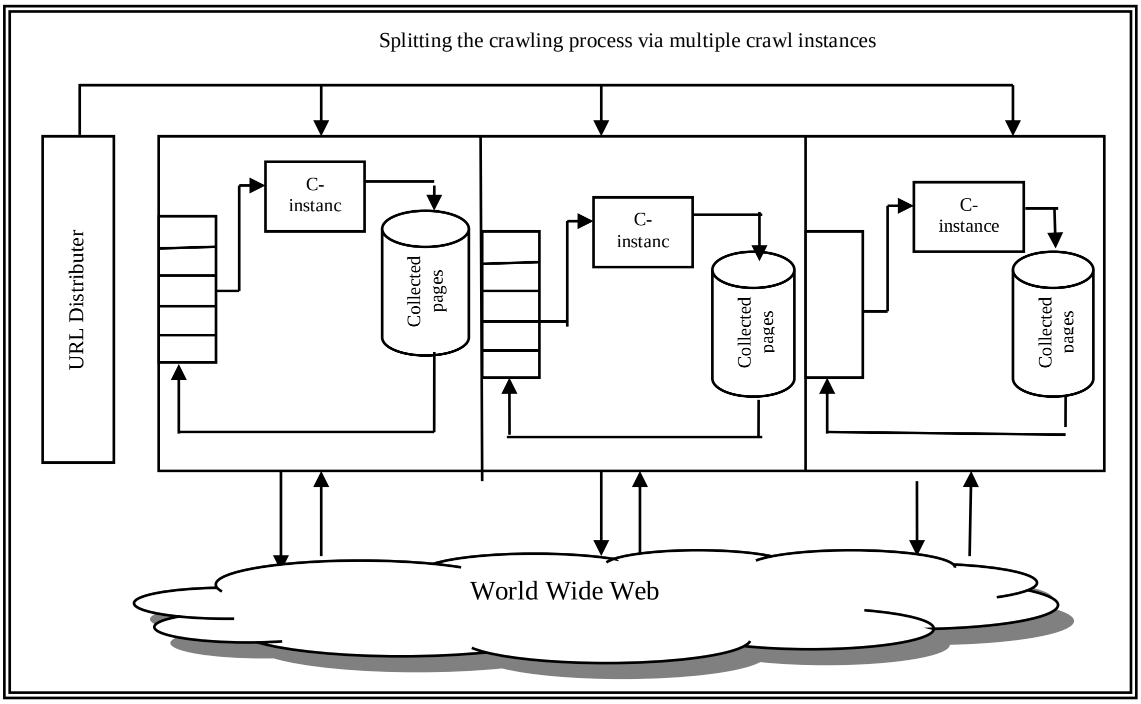


Fig 2.21: General Architecture of Parallel Crawler

It consists of multiple crawling processes referred to as C-instances. Each C-instance performs the basic task that a single crawler conducts. It downloads the web pages from web server, stores the pages in a local repository, extracts the links from the downloaded pages and follows the extracted links [108].

These multiple crawl instances may be located locally or globally [98]. When they run on same local network connected through high speed LAN, it is called as ***Intra-site parallel crawler.*** When crawl instances run at geographically distant locations and communicate via Internet, it is called as ***Distributed crawler***. The general architecture of distributed parallel crawler is depicted in Fig 2.22.

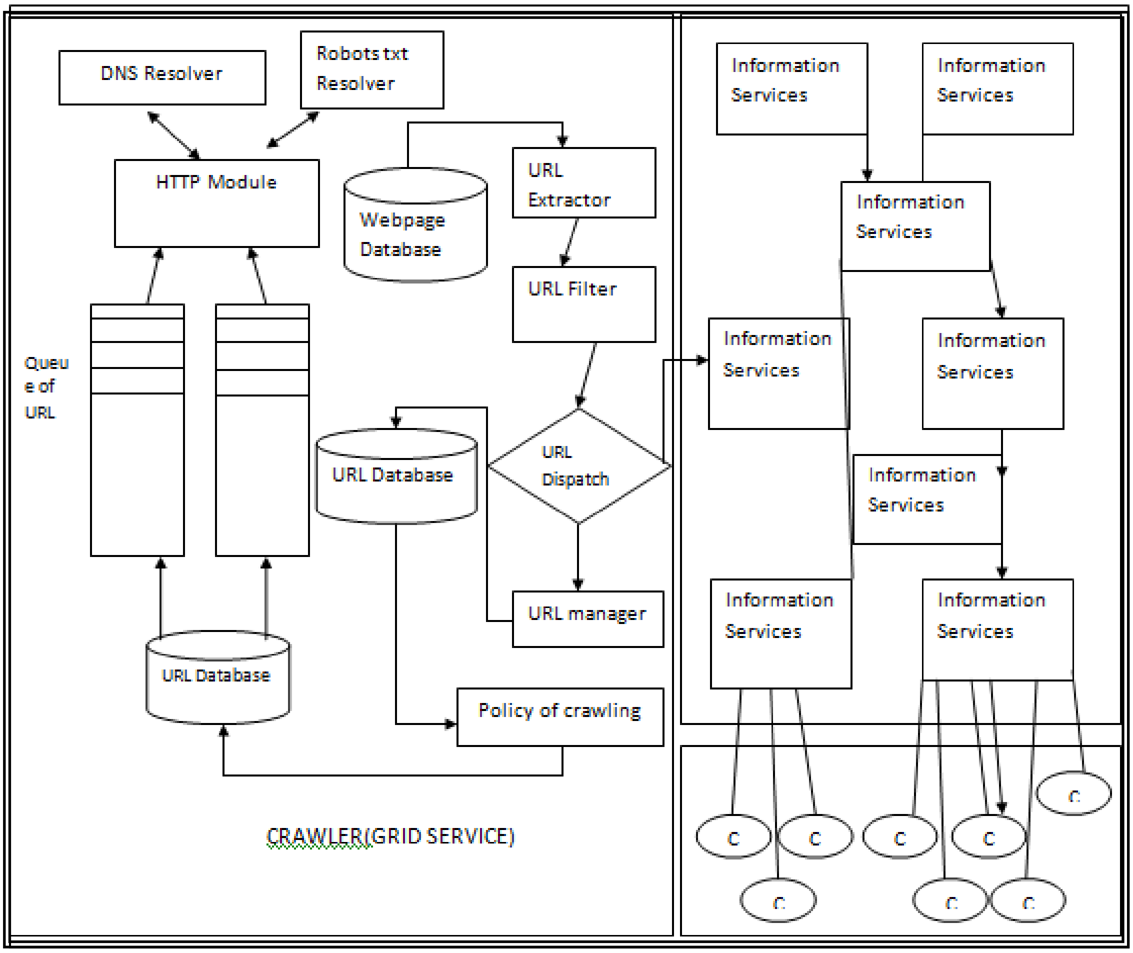


Fig 2.22: General Architecture of Distributed Parallel Crawler

The distributed crawler is preferred to intra site parallel crawlers because it can disperse the load on multiple networks and thus reduction in overall network is achieved.

The main challenge for parallel crawler is to opt proper coordination policy among C-instances because when multiple instances run in parallel they may download the same page multiple times. To reduce the overlap in downloading, three different ways are prescribed below [96]:

Independent: Each C-instance begins with its own set of seed URL and follows link without coordinating with each other [94].

Dynamic Assignment: In this method, a centralized coordinator also called as crawl manager logically divides the web into some partitions and dynamically allots each partition to a C-instance for downloading the pages [109].

Static Assignment: In contrast to dynamic assignment, there exists no central coordinator. In fact each C-instance knows about the tertiary of other C-instance before the actual start of crawl process.

2.9.2 Focused Crawler

The focused crawler seeks, gathers, stores and maintains web pages on a specific set of topics that represent a relatively small portion of the web so as to reduce the amount of network traffic caused by irrelevant downloads [107] [111]. Thus unlike conventional crawler that follows each and every link on the page, the focused crawler gives priority to links that belong to pages classified as relevant. For this purpose, it uses a special component called as classifier as shown in Fig 2.23 Classifier performs three major tasks:

i) It is used to find whether the page belongs to the topic taxonomy or not.

ii) It identifies whether the links points to page that contain relevant topic or not.

iii) It filters out irrelevant pages and store the relevant pages in database if they are not already present.

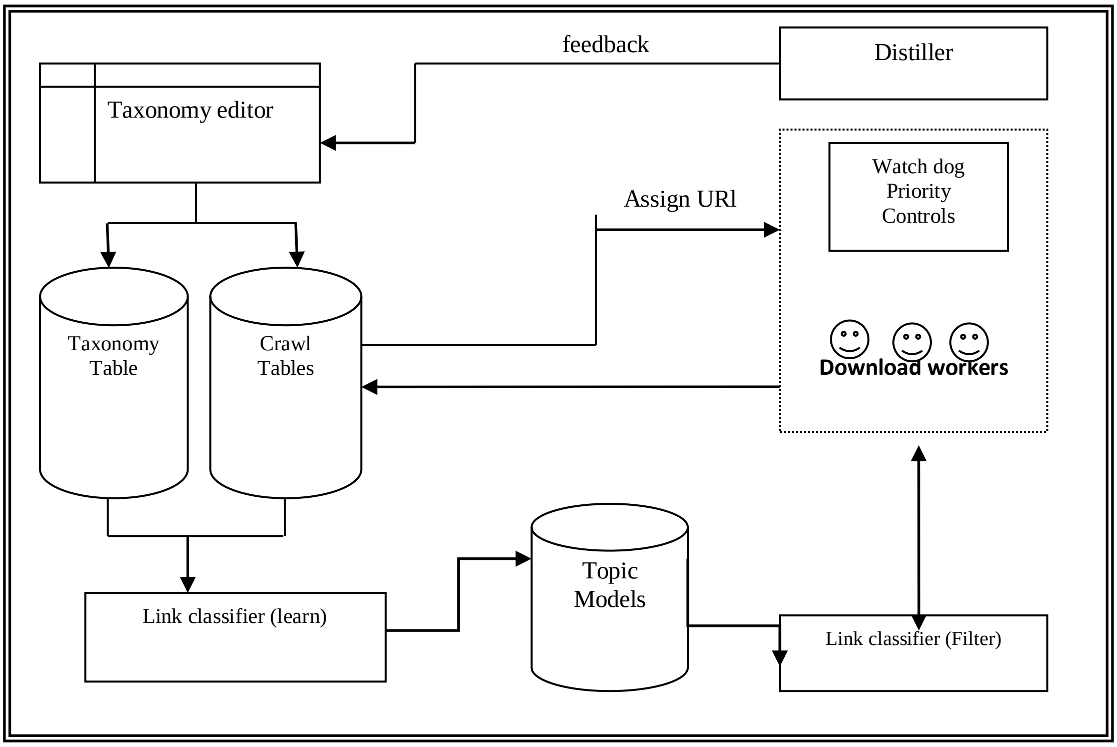


Fig 2.23: General Architecture of Focused Crawler

Besides, the other important components are distiller and download workers. Distiller determines the visit priorities among the links to be crawled. Download workers: are dynamically reconfigured and controlled crawl instances that are governed by both classifier and distiller. The crawled pages are further classified into topic taxonomy. The topic taxonomy is maintained with the help of user feedback by asking interestingpages as they browse [107].

The most crucial evaluation of focused crawling is to measure the harvest ratio. It is basically the rate at which relevant pages are obtained and irrelevant pages are discarded. This ratio must be high, otherwise the focused crawler would spend a lot of time merely eliminating irrelevant pages, and it would better to use an ordinary crawler instead [109].

2.9.3 Hidden Web Crawler

Some of the information on web is not explicitly contained in web documents instead they are implied from other web pages and can only be obtained by special type of crawler known as hidden web crawler [98]. As traditional crawler follows up the link present on web to discover pages so they can't index the hidden web. Search forms act as the entry-points into the hidden Web. Specifically, hidden web crawler takes a sequence of actions for each form i.e. form analysis, value assignment, submission, response analysis and response navigation [132]. The architecture of hidden web crawler is depicted in Fig 2.24.

Architecture includes following modules / data structures:

- ***URL List:*** It contains all the URLs that the crawler has discovered so far.
- ***The Crawl Manager***: It controls the entire crawling process i.e. it decides which link to visit next, makes the network connection to retrieve page from the web, and handover the downloaded page to the Parser module.
- ***Parser:*** it extracts the links from downloaded pages and inserts them in URL list.
- ***Form Analyzer:*** This component is used to parse and build internal representation of form.

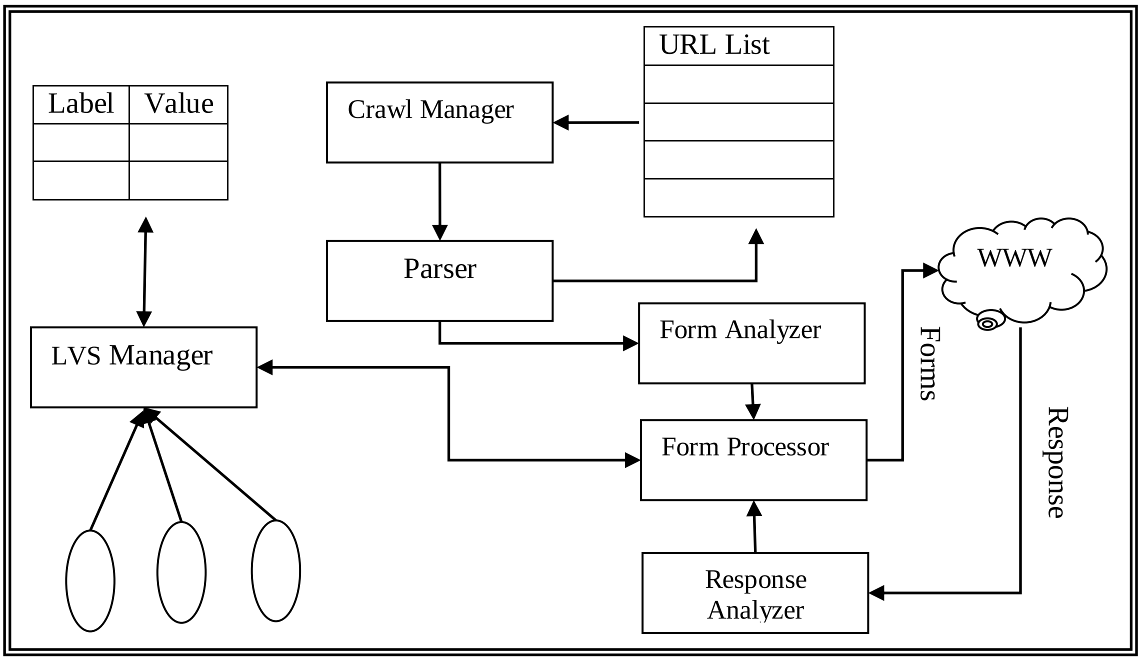


Fig 2.24: Architecture of Hidden Web Crawler

- ***Form Processor:*** it assigns the values to various form elements and submits the form to get corresponding response pages.
- ***Response Analyzer:*** It analyzes the response pages to check the validity of result.
- ***LVS (Label-value set) Manager:*** It provides the interface for various application, receives the values for various form elements and store them in LVS table. The Parser extracts links from the downloaded page and adds them to the URL List. This sequence of operations is repeated until some termination condition is satisfied.

There are certain limitations of hidden web crawler. First, they do not perform well if the size of its crawl area is large. Second, they are not able to process the all types of formats such as .jpeg, .pdf etc.

2.9.4. Migrating Crawler

To reduce the network load caused by crawling and downloading the large amount of documents in uncompressed form using HTTP across the web, Odysseas et al proposed an efficient crawler program called migrating crawler [99]. The main idea behind the migrating crawler is to move the computation unit i.e. crawl instances to the data source rather than moving the data to computation [81] as shown in Fig 2.25.

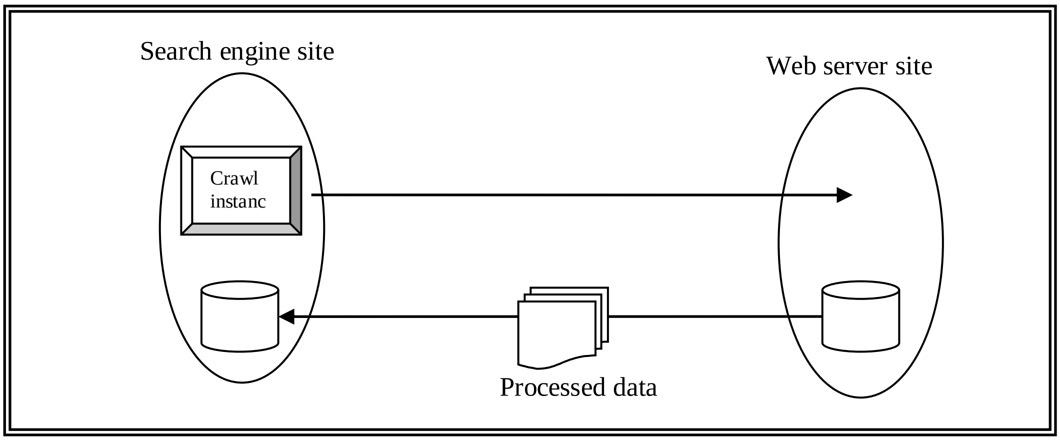


Fig 2.25: Concept behind Migrating Crawler

By migrating the crawl instances to the different data sources help in speeding up the task of downloading the web pages as compared to traditional crawlers. The architecture of UCYMicra [97] developed by Odysseas to catch up with dynamic web is depicted in Fig 2.26. It consists of three subsystems: Coordinator Subsystem, Mobile Agents Subsystem and Public Search Engine as shown in Fig 2.26.

The purpose of each of these components is explained below:

- ***Coordinator Subsystem:*** The coordinator subsystem resides at the search engine side and performs three main task:

   i. It administers Mobile Agent Subsystem.
   ii. It is responsible for maintaining search engine database.
   iii. It facilitates online registration for new websites to participate in UCYMicra.

- ***Mobile Agent Subsystem:*** This component is specifically designed to crawl the web. It is composed of two types of mobile agents:

   Migrating Crawlers. They are responsible for on-site monitoring and crawling web servers. The number of tasks carried out by migrating crawler is outlined below:

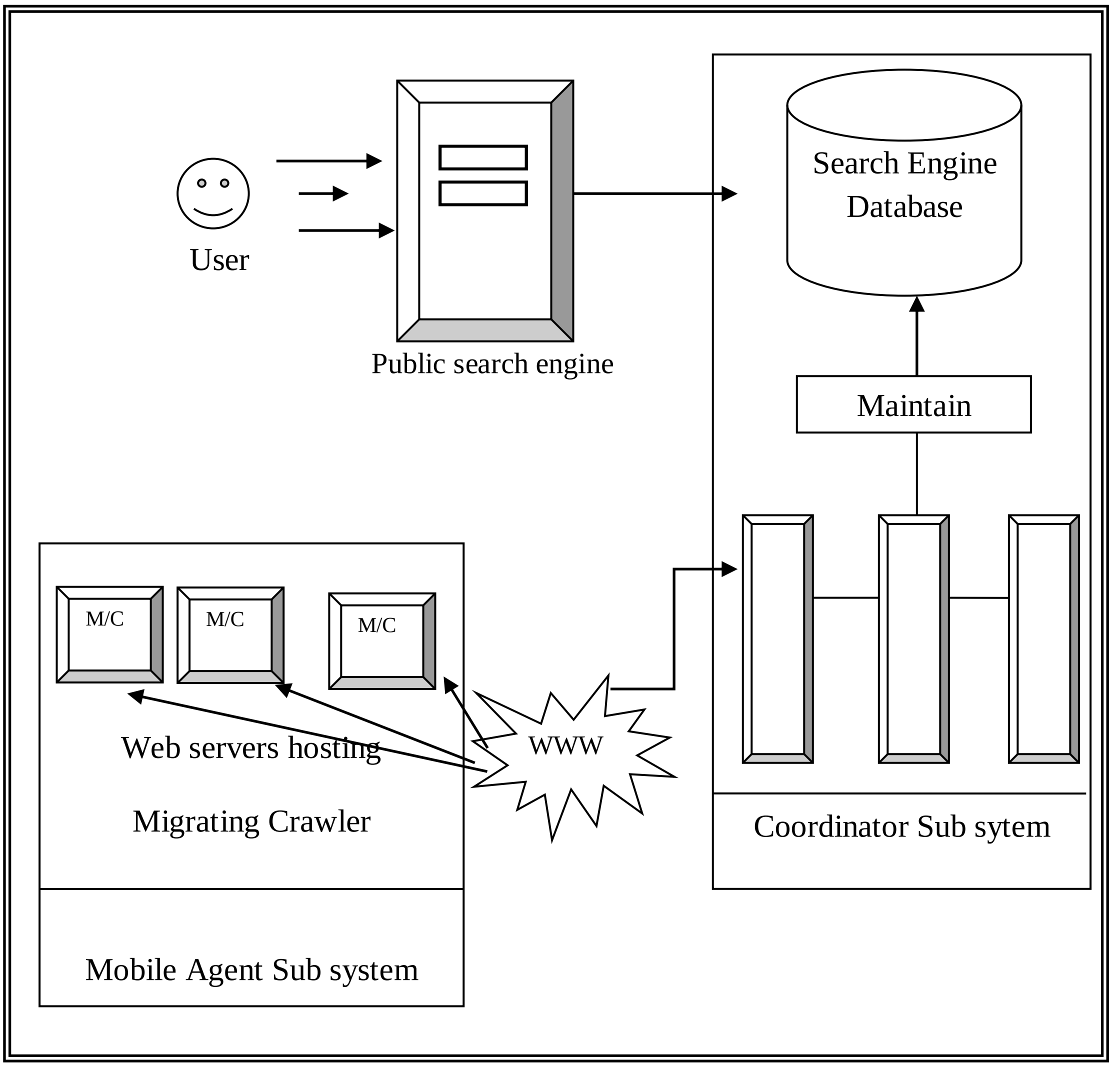


Fig 2.26: Architecture of Migrating Crawler

- Get dispatched to web servers registered with UCYMicra.
- It performs a complete local crawling (either through HTTP or the file sub system).
- it process the documents locally at web server site to extract keywords from the crawled pages and rank them based on some pre defined visual properties such as font, color, position frequency etc. It then locally creates the index of processed data. The Migrating Crawler can detect changes on the Web server content. Detected changes are instantly processed and transmitted to the Coordinator subsystem through data caries.

i. Data Carries: It transmits the compressed and processed data maintained at server site back to coordinator subsystem.

- ***Public Search Engine***: Its main task is to execute the user queries on the search engine database maintained by the coordinator subsystem.

The power of UCYMicra lies in its capabilities to carry real time upgrades because any of the above task can be easily deployed because UCYMicra was implemented in JAVA [130].

2.9.5 Incremental Crawler

As web is dynamic in nature, a crawler needs to maintain up to date collection of documents in its local repository. The document freshness can be carried out in two ways:

Batch Mode: In batch mode, a crawler updates all its documents in local collection after a fixed interval, say in a month as shown in Fig 2.27. The graph in Fig 2.28 shows change in freshness of documents over time.

It may be noted in Fig 2.26 that the documents get updated in grey region i.e. during the re-crawl of pages and the documents gets decayed in white region i.e. when the crawler is sitting idle, then freshness decreases. Moreover, the freshness never equals to 1 even when the visit is complete because some pages are so volatile that again changes in the time the crawler just finished with re-crawl. Also, the freshness declines exponentially in the time crawler is not running [101].

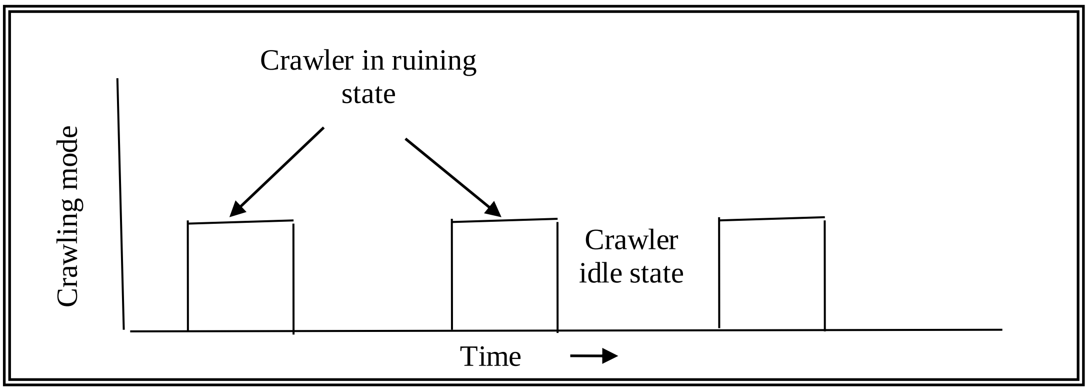


Fig 2.27: Crawler Running in Batch Mode

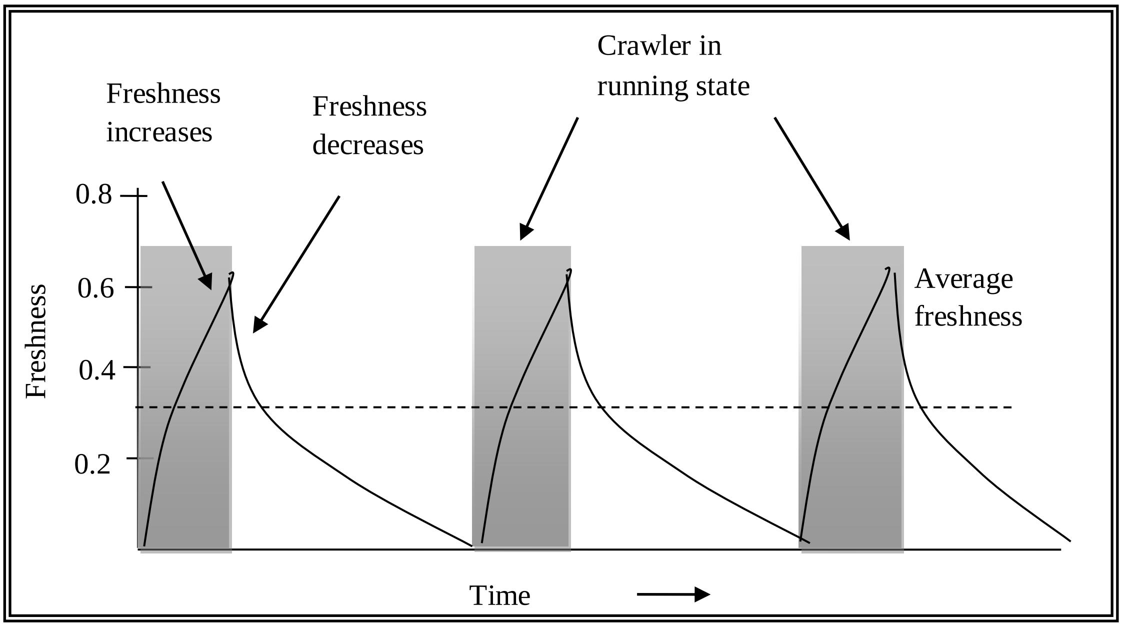


Fig 2.28: Freshness in Batch Mode

Steady Mode: In contrast, the crawler runs continuously in steady mode to update its collection as shown in Fig 2.29. As the collection is continuously and incrementally updated, document freshness in steady mode is stable over time as shown in Fig 2.30 [110]. It may be observed that the average freshness is same in both cases. However, to achieve same freshness level, a batch mode crawler needs to revisit pages at a higher speed as compared to a steady crawler that continually run at a lower speed. It is always better to run crawler continuously at average speed rather than periodically with high speed because high speed leads to increases in unnecessary congestion on busy network [104].

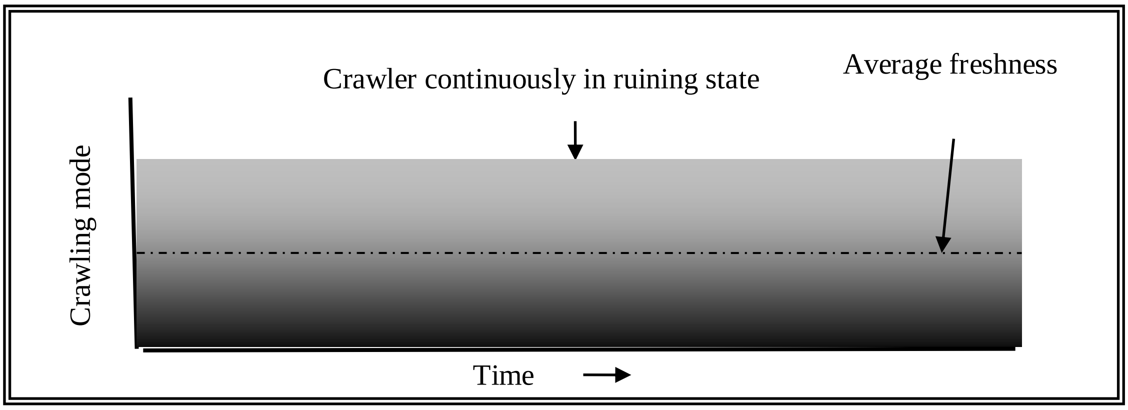


Fig 2.29: Crawler Running in Steady Mode

Further, a crawler may update the old version of a page with new version in two ways:

In-place updation: In this case, the old page get replace by new version immediately, providing fresh information to user at all times.

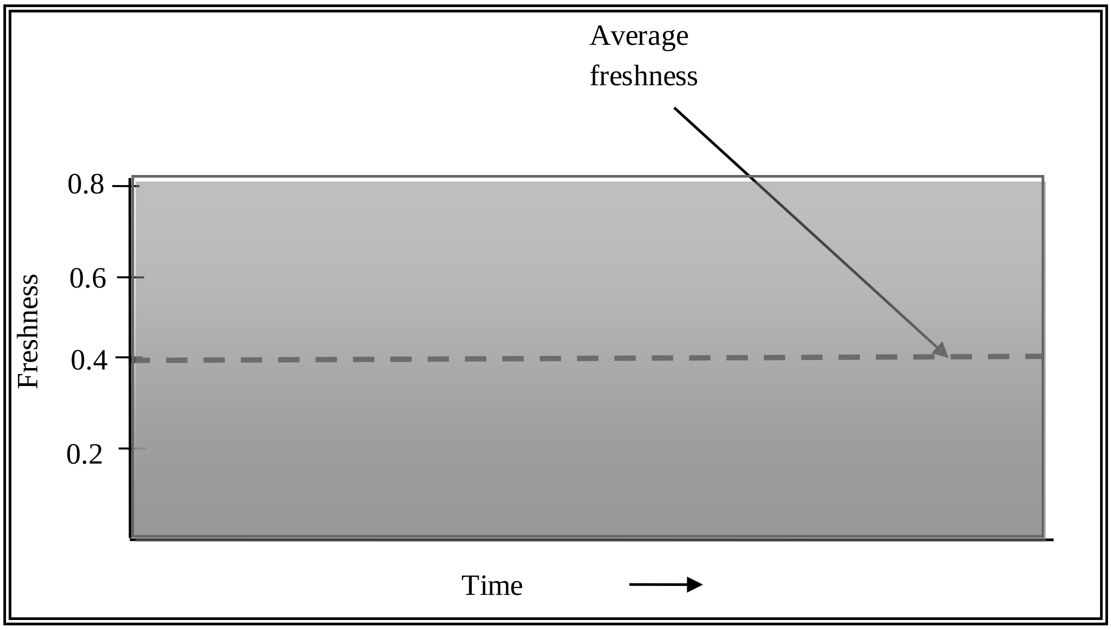


Fig 2.30: Freshness in Steady Mode

Shadowing: In contrast, shadowing allows collecting and storing new set of pages in separate location, and replaces the old version with new only after re-crawling is completed.When batch mode crawling is supplemented with shadowing updation and fixed revisit interval, it is called a ***periodic web crawling*** [112]. In contrast, a steady crawling supplemented with in-place updation with variable revisit interval, it is called an ***incremental web crawling***.

The ***periodic crawler*** visits the websites until its collection has a desirable number of pages. Whenever it needs to refresh its collection, it revisits the sites, creates a new collection and replaces the old collection with the new [135]. Whereas an ***incremental crawler*** replaces less important existing pages with more important new pages and refreshes its collection [107]. High level of freshness can only be guaranteed by revisiting all pages at rapid rate while optimized network load The design of an incremental crawler needs to address the following issues:

- **Keep the local collection fresh:** The crawler should keep the collection up to date. The up to dateness of pages in local collection depends on the policy adored such as adjusting the revisit frequency for a page based on its estimated change frequency [81].

- **Improve quality of the local collection:** The crawler should replace less important pages with more important pages to improve the quality of its local collection because of the following reasons:
  - Firstly, web pages are constantly created and destroyed. Some of these freshly created pages may possesses high importance than existing pages in local collection. So, the crawler needs to replace less important existing pages with more important new pages.
  - Secondly, the importance of existing pages may changes with time or become obsolete. So these pages must be removed from the collection.

The architecture of incremental crawler proposed by Jungo Cho includes three basic functional components and three internal data structures as shown in Fig 2.31.

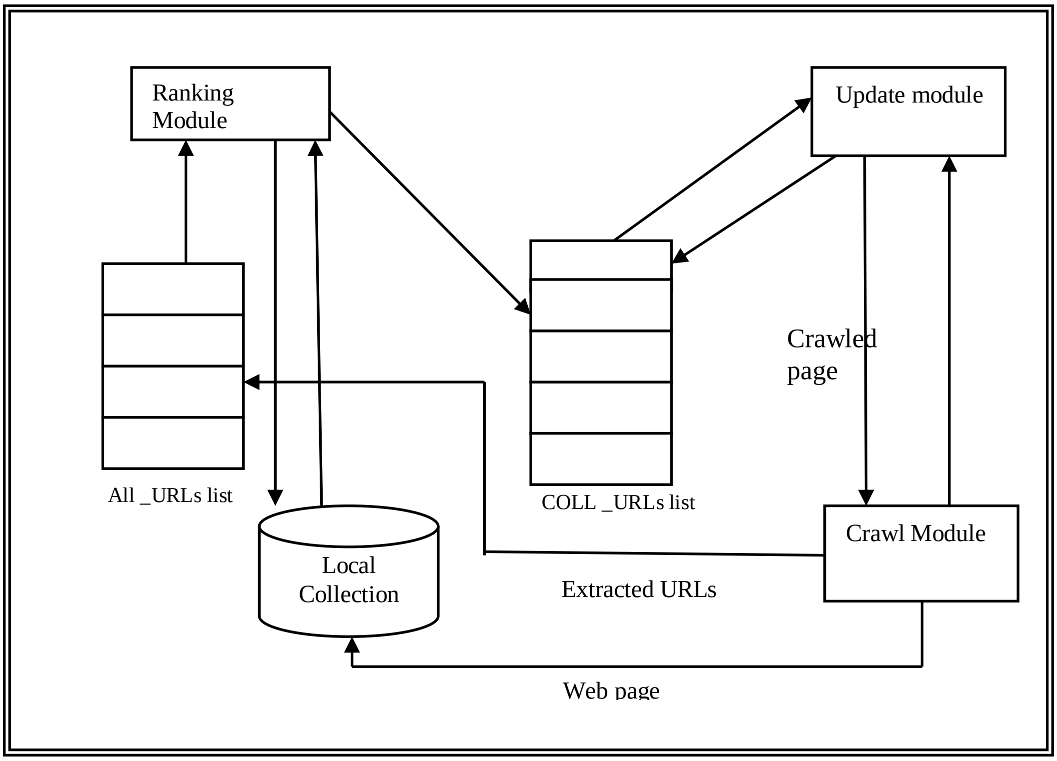


Fig 2.31: Architecture of Incremental Crawler

The description of each of them is given below:

- ***ALL_URLs list:*** ALL_URLs records all URLs that the crawler has discovered so far.
- ***COLL_URLs list***: It records the URLs that are in the local Collection.
- ***Local Collection:*** Database of documents downloaded corresponding to URLs in COLL_URLs list.
- ***Ranking module:*** It picks the URL from ALL_URLs list and assign a numeric score based on some previous history and insert the ranked URL in COLL_URL list.
- ***Update module:*** it picks the URL from COLL_URLs list and crawl the corresponding page and update it in local collection.
- ***Crawl Module:*** It extract the links embedded in downloaded document and insert them in ALL_URLs list.

Based on objectives, working model, mechanism, and crawling policies, comparison between the various aforementioned web crawlers is summarized in table 2.13

Table 2.13: Comparisons between Prevalent Web Crawling Techniques

Characteristics	**Parallel**	**Focused**	**Incremental**	**Hidden Web**	**Migrating**
Objective	To speed up crawling process	To crawl topic specific web pages	To maintain freshness of search database	To crawl information hidden behind web pages	To reduce network load
Mechanism	Crawling done in parallel by multiple crawler instances	Crawling done by crawler in a specific Field	Re-Crawling is only done	High quality search forms	Crawling is done by migrants
Revisit Policy	Revisit on restart crawling	-	Revisit is based on page rank	-	Revisit is based on change frequency
Crawling Strategy	BFS	DFS	BFS	DFS	BFS

Crawling Speed	Fast	-	-	-	Fast
Network Load	High	High	High	High	Low
I/P parameter	Seed URLs	Topic Specific URLs	URLs governed by priority Queue	Search forms	Seed URLs
Scalable	Yes	No	No	No	Yes
Web Crawler Example	PARCHAYD [121]	S.Chakra Barti [107]	Jungoo Cho [101]	HIWE [98]	Odysseas [97]

The next section provides a review summary of limitations identified in the literature so forth.

2.10 REVIEW SUMMARY

A critical look at available literature indicates the following issues that need to be addressed towards development of efficient interest based search system based on personalization techniques:

- The traditional query suggestion system employed by most of search engine returns the query suggestions based on popularity of query keywords and ignores its context in all other possible areas, thus sometimes fail in narrowing down the search space. For instance, the topical query Java when submitted to search system leads to query suggestions that only belongs to computer field and do not give other context in which java may be applicable. This give rise to need for development of query suggestion technique that can narrow down user search need in right direction by the application of web mining techniques.

- The other possible limitation of traditional query suggestion systems is that they provide same types of query suggestions to all its users regardless of their actual interest. As the user of search systems belongs to variant communities, customs, location and educational backgrounds, so they may possess different degree of interest in different domains. So some more efficient query suggestion system based on user interest using personalization techniques are actually needed to move the search process in right direction.

- Most of the search systems depict low precision in their search results. A magnitude of irrelevant pages is contained in it. These pages are highlighted and included by the ranking system just because they contain query keywords. Even the most relevant pages with respect to user query terms are sometimes not positioned on the top. Hence the user spent more time to fulfill its information need.

- As most of the ranking techniques used by search engines are based on web structure mining and/or web content mining and pay less attention to methods inferring the user actual need .So mapping between web pages and requirement of user is not completely established. Hence amalgam of all web mining techniques must be adopted by ranking system for better formation of search results.

- Although there are many efficient crawling techniques to deal with dynamic nature of web but many of them either provide obsolete information or increase network load. So an optimized crawling mechanism based on change frequency of web pages need to be designed.

- There exist many mechanisms to detect redundant downloading of same document but they are not much scalable due to abundance of information available on web. So the mechanism need to be introduces that can cope up with exponential growth of web and restrict duplicate downloading.

The subsequent chapters present optimized solutions for query suggestions, ranking and crawling with a view to resolve aforementioned issues. The proposed query suggestion models are presented in next chapter.

CHAPTER III

MODELS FOR QUERY SUGGESTION

3.1 INTRODUCTION

The most commercial search engines return the search list by matching the user query terms with the documents terms available in its database. The relative effectiveness of search result is highly affected by the extent to which the query keywords map to the actual need of the user. But unfortunately, user generally forms the short, ambiguous and instant queries which are not always enough to clearly interpret its information need, thereby lead to inclusion of irrelevant documents in search results. Moreover the different users use the same keyword to retrieve contextually varied information which further increases the complexity of information retrieval process. Consider a scenario where a user tends to search by issuing any of the term like java, mouse, net, cluster, mining, spider on search interface. Typically, the search engine returns the results which are most popular and particularly in these cases from computer field. But it may be possible that user wants to retrieve the information from other than computer field. So he/she has to modify the query and resubmit it. But even the resubmission of query does not guarantee the retrieval of desired information. So, making search engine responsive to user's need requires understanding the semantics of submitted query terms and mining user behavior characterization. The mined knowledge can be used to construct alternate queries that not only help the user in refining its need clearly but it also reduces the search space, thereby providing fast and relevant information to the user.

In the light of above discussion, two novel techniques focused on query structuring under the data presentation layer have been developed. In order to understand the context of query and user trends in information retrieval, various data resources such as definition_ repository, query log, profile database web dictionary have also been proposed.

The detailed discussion on both the techniques is given in the next section:

3.2 QUERY SUGGESTION BASED ON SEMANTIC SIMILARITY (QSSS)

A dynamic query suggestion approach based on semantic similarity measure is basically a meta search layer between the user and general search engine that aim to predict all the possible semantic meanings related to a submitted query terms, and accordingly suggests the best alternate queries to the user.

As a conventional search engine relies on keyword matching mechanism for retrieving documents from its database, it means a poorly designed query may lead to inclusion of wrong documents in the result set. To reduce the gap between query formation and information retrieval, user assisting system is needed that may help him/her in better query formation. Although the existing systems are providing the query suggestions, but again the suggested queries are retrieved based on keyword similarity with the queries submitted in past regardless of the fact that a term may have several synonyms that are applicable in different context. So, the system that considers the context of query terms is highly appreciated and is today’s need.

In many cases, the result set returned in response to a user query contains the topics of varying domains that not only increases the volume of search results and leads to information overkill problem but also increases the selection complexity at the user’s side. Traditional search engine does not provide different descriptions of query term to the user so as to move the search in particular direction.

In the abstract form, the approach to be used by the proposed system is depicted in Fig 3.1. The dashed line represents the proposed meta search layer between the user and keyword based search engine. In order to assist the user in query formation phase, the systems carries out the four steps as listed below:

- Building the Definition Repository
- Query Normalization
- Equivalent Query Formation
- Best Alternate Query Generation

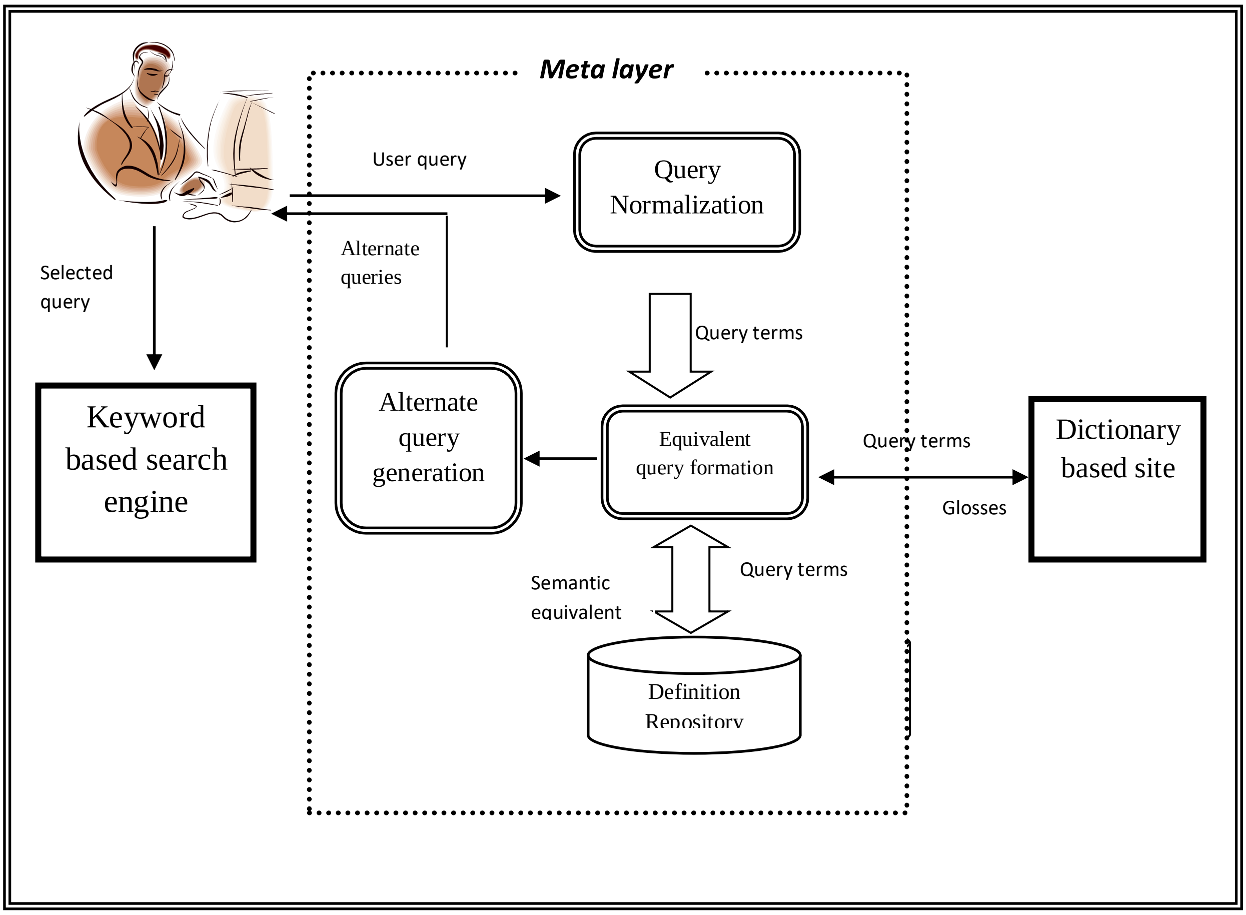


Fig 3.1: Query Suggestion based on Semantic Similarity

3.2.1 Building the Definition Repository

A publically available English lexical dictionary such as wordnet 3.1 [134] is used to populate the *Definition_ Repository*. It stores the terms and their related semantic definitions. Here the tem *definition* can be defined as a phrase or set of terms that describes the meaning of term in different contexts. In the proposed approach, initially the definition repository is populated with seed set of keywords and incrementally enhanced with the occurrence of new terms in submitted query. The schema for definition repository is depicted in Fig 3.2.

Term_id	Term	Definition	Semantic equivalent

Fig 3.2: Schema for Definition _ Repository

The description of each field of definition repository is given in table 3.1.

Table 3.1: Description of Various Fields of Definition Repository

Field	Description
Term _ id	An alphanumeric number assigned to each term encountered in user query
Term	The keyword occurring in original query submitted by the user
Definition	The gloss specifying the various contextual definition of term returned by online dictionary.
Semantic equivalent	The noun phrases extracted from definition returned by online dictionary

The organization of information in definition _repository can easily be understood by the examples summarized in table 3.2.

Table 3.2: Example illustration of Definition _ Repository

Term _ id	Term	Definition	Alternate Term
Java#n#1	Java	An island in Indonessia to the south of Borneo; one of the world's most densely populated region	Java Island
Java#n#2	Java	A beverage consisting of infusion of ground coffee beans	Java Coffee
Java#n#3	Java	A platform independent object-oriented programming language	Java Programming language
Mouse#n#1	mouse	Any of the numerous small rodent resembling diminutive rat having pointed snouts and small ears on	Rodent mouse

		elongated bodied with slender usually hairless tail	
Mouse#n2	mouse	A swollen bruise caused by blow of eyes	Shiner mouse
Mouse#n3	mouse	Person who is quite or timid	Person mouse
Mouse#n4	mouse	A hand operated electronic device that controls the coordinates of a cursor on computer screen as you move it around on a pad	Computer mouse

For instance, the term *'java'* posses three different meaning: Java Island, java coffee, java programming language in different contexts as depicted by row 1, 2, 3 in table 3.1. These semantic equivalents along with its brief description are stored in definition _ repository.

3.2.2 Query Normalization

Query normalization is most basic and indispensible step in refining the submitted query. It normalizes the query by removing the stop words from the query as these terms do not contribute towards retrieval of relevant information. After removal of stop words, the spelling of each candidate term is checked and if there is any mistake made by the user, it is corrected .At last stemming is done on the corrected query terms using Porter's algorithm to produce normalized query terms [75]. The algorithm for query normalization is given in Fig 3.3.

```
Query Normalization ()
Input: User query
Output: Normalized query
Method: Do for each user query{
    1. Remove stop words from the user query
    2. Check spelling of each candidate term in partially processed query
    3. Generate stem of each candidate term in partially processed query using
       Porter.s algorithm}
```

Fig 3.3: Algorithm for Query Normalization

The algorithm takes user query as input, processes it and produces normalized query as output.

3.2.3 Equivalent Query Construction

This step involves searching of semantic equivalents S_1,S_2,…S_k of normalized term (say t_n) in the definition repository and forms the equivalent queries. The existence of term in definition_ repository ensures the fast retrieval of various semantic meaning related to a query. But if the term does not exist in definition_ repository, the search is directed to publically available online dictionary websites. After identification of semantic equivalent of various normalized terms belonging to query q_i, they are arranged to form alternate queries.

If the submitted query contains only one normalized term, then its semantic equivalents are treated as alternate queries and presented back to the user. The algorithm for constructing equivalent queries is given in Fig 3.4.

```
Equivalent _ query (normalized query)
Input: Normalized query
Output: Set of equivalent query
Method:
Do {
    1. Repeat for each normalized term belongs to query {
        1.1 Search the term in definition_repository;
        1.2 If match is found
                Fetch (semantic equivalence);
        1.3 Else(Get _ semantic _ wordnet ( normalized term)}
    2. if ( no. of normalized terms > 1) {
        2.1 Arrange the semantic equivalent of each normalized term in the
            same order as that of query
        Return (arranged semantic equivalence)}
        2.2  else
         Return (semantic equivalence)}
}
```

Fig 3.4: Algorithm for Constructing Equivalent Queries

To better understand the process, let us consider the scenario 1 given below:

Scenario 1: "A user 'U' wants to search about the term crane .He is new to this term and has no idea about it".

The various semantic equivalents related to term *'crane'* retrieved from online dictionary are shown in Fig 3.5. The user 'U' selects the definition which is of his interest and proceeds with the general search engine. In this way, the unwanted results from the other contexts which were earlier increasing the overhead in finding the desired information are filtered out and thus lead to reduction in information overkill.

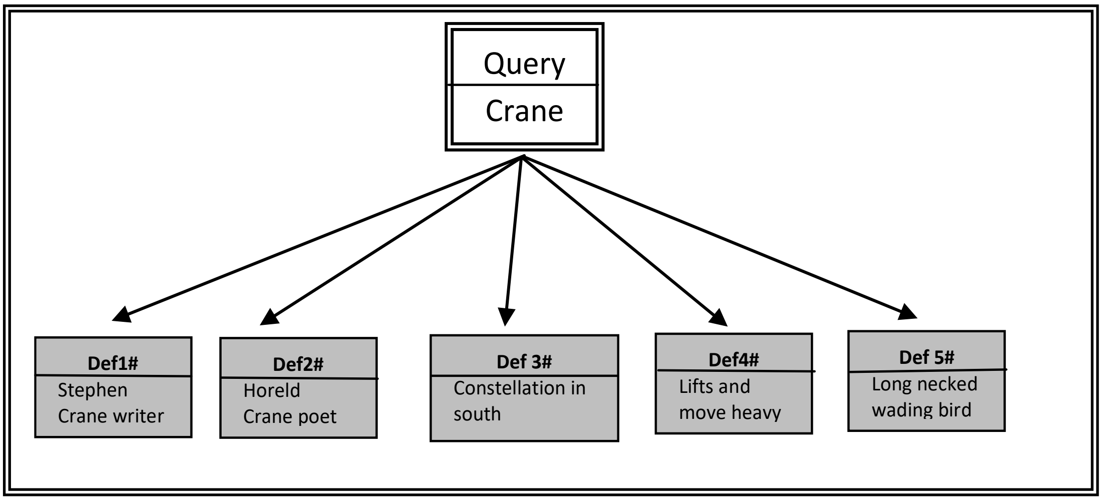


Fig 3.5 Various Semantic Equivalents of Term 'Crane' Obtained Through Wordnet Dictionary

But there may be many cases where user query contains more than one normalized word. In such cases, the process of alternate query generation is handled differently as explained with the help of scenario 2 given below:

Scenario 2: "A user U has an interest in cosmology and wants to know about the "shapes of crane".

As there exist two normalized terms in this query: i) shape ii) crane. So, semantic definitions of both the normalized terms need to be search in definition repository. The various definitions available in definition _repository are listed in table.3.3.

Table 3.3 Semantic Definitions for Terms *Crane* and *Shape*

Term _ id	Term	Definition	Semantic equivalent

Shape#n#1	Shape	Any spatial attribute (especially defined by outline)	Configuration, contour, conformation
Shape#n#2		The spatial arrangement of something that is distinct from its substances	Form
Shape#n#3		Alternative name for body of human being	Human body, physical body, material body, soma , built, figure, physique, anatomy, bod, chassis, frame, flesh
Shape#n#4		A concrete representation of otherwise nebulous concept	Embodiment
Shape#n#5		A visual appearance of something or someone	Cast
Shape#n#6		The state of (good) health (especially in the phrase 'in condition' or 'in shape' or 'out of condition' or 'out of shape'	Condition
Shape#n#7		The supreme headquarter that advises NATO on military matters or oversees all aspects of Allied Command Europe	Supreme headquarters allied power Europe
Shape#n#8		A perceptual structure	Pattern
Crane#n#1	Crane	Stephen Crane(United State Writer(1871-1900))	Stephen Crane
Crane#n#2		Hart Crane(United State (1899-1932))	Hart Crane
Crane#n#3		A small constellation in the south hemisphere near Phoenix	Grus constellation
Crane#n#4		Lifts and moves heavy object, lifting tackle is suspended from a pivoted boom that rotates about a vertical axis	Crane lifts
Crane#n#5		Large long necked wading bird of marshes and plains in many part of the world	Crane bird

As listed in table 3.3, there exist eight different definitions related to term *shape* and five definitions related to term *crane* that can further be combined to form many different alternate queries. To simplify the process, the equivalent queries are formed by arranging the synonyms of each normalized term in the same order as that of original query. The set containing the alternate queries for scenario 2 includes the queries like configuration of Stephen crane. Configuration of hart Crane, configuration of grus, configuration of crane lift, configuration of crane bird and so on. In this way 40 different alternate queries are generated as shown in table 3.4.

Table 3.4: Generation of Equivalent Queries for the Sampled Query *Shapes of Crane*

S.no.	Term 1	Term 2	Equivalent Query
1	Shape#n#1	Crane#n#1	Configuration of Stephen crane
2		Crane#n#2	Configuration of Hart Crane
3		Crane#n#3	Configuration of Grus constellation
4		Crane#n#4	Configuration of crane lift
5		Crane#n#5	Configuration of crane bird
6	Shape#n#2	Crane#n#1	Form of Stephen Crane
7		Crane#n#2	Form of Hart Crane
8		Crane#n#3	Form of Grus constellation
9		Crane#n#4	Form of crane lift
10		Crane#n#5	Form of crane bird
11	Shape#n#3	Crane#n#1	Physical body of Stephen crane
12		Crane#n#2	Physical body of Hart Crane
13		Crane#n#3	Physical body of Grus constellation
14		Crane#n#4	Physical body of crane lift
15		Crane#n#5	Physical body of crane bird
16	Shape#n#4	Crane#n#1	Embodiment of Stephen crane
17		Crane#n#2	Embodiment of Hart Crane
18		Crane#n#3	Embodiment of Grus constellation
19		Crane#n#4	Embodiment of crane lift
20		Crane#n#5	Embodiment of crane bird
21	Shape#n#5	Crane#n#1	Cast of Stephen Crane
22		Crane#n#2	Cast of Hart Crane
23		Crane#n#3	Cast of Grus constellation
24		Crane#n#4	Cast of crane lift
25		Crane#n#5	Cast of crane bird
26	Shape#n#6	Crane#n#1	Condition of Stephen Crane
27		Crane#n#2	Condition of Hart Crane
28		Crane#n#3	Condition of Grus

29		Crane#n#4	Condition of crane lift
30		Crane#n#5	Condition of crane bird
31	Shape#n#7	Crane#n#1	Head quarter of Stephen Crane
32		Crane#n#2	Head quarter of Hart Crane
33		Crane#n#3	Head quarter of Grus constellation
34		Crane#n#4	Head quarter of crane lift
35		Crane#n#5	Head quarter of crane bird
36	Shape#n#8	Crane#n#1	Pattern of Stephen Crane
37		Crane#n#2	Pattern of Hart Crane
38		Crane#n#3	Pattern of Grus constellation
39		Crane#n#4	Pattern of crane lift
40		Crane#n#5	Pattern of crane bird

So in total 40 different alternate queries for scenario 2 are obtained but not all of them are meaningful. Thus the next step aims to find meaningful queries.

3.2.4 Best Alternate Query Generation

Out of these 40 queries, not all of them are meaningful as paraphrasing using synonyms of terms does not necessarily generate valid sentences at least for a given context. So this step is focused on identifying relevant alternate queries based on relatedness score among every pair of synonyms generated by normalized terms. For this purpose, the shortest path between the pair wise senses of normalized terms stored in *is-a* hierarchy of WordNet dictionary is identified and number of nodes along the shortest path is counted. More number of nodes along the path implies that senses are less related to each other and vice-versa .Based on shortest path, the formula for relatedness score among the pair wise senses can be designed as shown by the eq^n 3.1.

$$\boxed{\boldsymbol{Relatedness\ Score}(\boldsymbol{S_1},\boldsymbol{S_2}) = {}^{\boldsymbol{1}}/_{\boldsymbol{Path\ length}} \qquad (\boldsymbol{3.1})}$$

For the scenario 2, the relatedness score among the various senses of normalized terms is evaluated by eqn (3.1) and summarized in table 3.5.

Table 3.5: Relatedness Score between Various Sensets

Senses pair (S1,S2)	Path length	Relatedness Score(S1,S2)	Process done (in msec)	Alternate Query
(Shape#n#1,Crane#n#3)	5	0.2	0.16	Configuration of Grus constellation
(Shape#n#2,Crane#n#3)	9	0.111	0.15	Form of Grus constellation
(Shape#n#3,Crane#n#1)	10	0.1	0.18	Physical body of Stephen crane
(Shape#n#3,Crane#n#2)	10	0.1	0.21	Physical body of Hart Crane
(Shape#n#8,Crane#n#3)	11	0.09	0.19	Pattern of Grus constellation
(Shape#n#7,Crane#n#3)	11	0.09	0.2	Head quarter of Grus constellation
(Shape#n#6,Crane#n#4)	12	0.083	0.16	Condition of crane lift
(Shape#n#2,Crane#n#4)	12	0.083	0.18	Form of crane lift
(Shape#n#1,Crane#n#1)	12	0.083	0.19	Configuration of Stephen crane
(Shape#n#2,Crane#n#1)	12	0.083	0.19	Form of stephen crane
(Shape#n#5,Crane#n#1)	12	0.083	0.21	Cast of Stephen Crane
(Shape#n#6,Crane#n#5)	12	0.083	0.25	Condition of crane bird
(Shape#n#3,Crane#n#4)	12	0.083	0,16	Physical body of crane lift
(Shape#n#3,Crane#n#5)	13	0.076	0.18	Physical body of crane bird
(Shape#n#1,Crane#n#2)	13	0.076	0.21	Configuration of Hart Crane
(Shape#n#8,Crane#n#2)	13	0.076	0.26	Pattern of Hart Crane
(Shape#n#5,Crane#n#2)	13	0.076	0/22	Cast of Hart Crane
(Shape#n#6,Crane#n#3)	14	0.0714	0.18	Condition of Grus
(Shape#n#4,Crane#n#1)	14	0.0714	0.23	Embodiment of Stephen crane
(Shape#n#7,Crane#n#1)	14	0.0714	0.28	Head quarter of Stephen Crane
(Shape#n#5,Crane#n#4)	14	0.071	0.17	Cast of crane lift
(Shape#n#8,Crane#n#4)	14	0.071	0.18	Pattern of crane lift
(Shape#n#1,Crane#n#4)	14	0.071	0.19	Configuration of crane lift
(Shape#n#4,Crane#n#3)	14	0.071	0.27	Embodiment of Grus constellation
(Shape#n#2,Crane#n#2)	14	0.071	0.29	Form of Hart Crane
(Shape#n#6,Crane#n#1)	14	0.071	0.32	Condition of Stephen Crane
(Shape#n#3,Crane#n#3)	15	0.066	0.27	Physical body of Grus constellation

(Shape#n#7,Crane#n#2)	15	0.066	0.27	Head quarter of Hart Crane
(Shape#n#6,Crane#n#2)	15	0.066	0.28	Condition of Hart Crane
(Shape#n#4,Crane#n#2)	15	0.066	0.31	Embodiment of Hart Crane
(Shape#n#2,Crane#n#5)	16	0.062	0.19	Form of crane bird
(Shape#n#8,Crane#n#5)	16	0.062	0.23	Pattern of crane bird
(Shape#n#4,Crane#n#4)	16	0.062	0.25	Embodiment of crane lift
(Shape#n#7,Crane#n#4)	16	0.062	0.25	Head quarter of crane lift
(Shape#n#5,Crane#n#3)	18	0.055	0.26	Cast of Grus constellation
(Shape#n#1,Crane#n#5)	18	0.055	0.3	Configuration of crane bird
(Shape#n#5,Crane#n#5)	18	0.055	28	Cast of crane bird
(Shape#n#4,Crane#n#5)	20	0.05	0.22	Embodiment of crane bird
(Shape#n#7,Crane#n#5)	20	0.05	0.26	Head quarter of crane bird
(Shape#n#8,Crane#n#1)	20	0.05	0.27	Pattern of Stephen Crane

It may be noted from table 3.5 that more relevant sensets or conversely more relevant alternate queries occupy top positions with high relatedness scores by applying proposed technique and vice-versa. For instance, more meaningful queries such as Configuration of Grus constellation, Form of Grus constellation, Physical body of Stephen Crane, Physical body of Hart Crane, Pattern of Grus constellation scored high relatedness score and placed at top few positions whereas less meaningful queries such as cast of crane bird, embodiment of crane bird, head quarter of crane bird, patterns of Stephen crane had scored less relatedness score.

Based on the semantic relatedness between the different senses of query terms, alternate queries are sorted and up to a top n threshold value alternate queries are presented back to the user. User may select option pertaining to its interest and proceeds with general search engine.

3.2.5 Performance Evaluation of QSSS

Query Suggestion based on semantic similarity had been implemented using C#.Net at front end and SQL Server at the back end. To support definition _repository English lexical dictionary WordNet 3.1 is used to derive the various semantic meaning of term .WS4Jdemo tool is used to find the number of nodes between two terms based on the position in is-a hierarchy of WordNet. The snapshots of implementation are included in appendix B.

Procedure: To check the performance of proposed technique, a seed set of 100 queries (given in appendix E) are submitted to proposed system as well as two popular systems: Google and Bing. For the sake of comparison, the relative position of common URLs occurring in their search results is noted. Some of the queries and their analytical results are summarized in Table 3.6.

Table 3.6: Result Analysis of Proposed System

Id	**User query**	**Number of alternate queries**	**Selected alternate query**	**Is same alternate query suggested by**		**No. of URLs moved to top 20 position from later positions**
				Google	**Bing**	
Q1	Birth of Pluto	15	Birth of Pluto Cartoon	No	No	9
Q2	Places of kingfisher	16	Places of kingfisher Bird	No	No	11
Q3	Shapes of crane	45	Shapes of crane constellation	No	No	19
Q4	Phases of moon	24	Stages moon	No	No	01
Q5	Famous tourist spot in India	28	Famous tourist spot in India	Yes	Yes	6
Q6	Cheap airfare from Delhi to Mumbai	4	Cheap airfare from Delhi to Mumbai	Yes	Yes	0
Q7	Tennis racket brands	20	Tennis racket make	No	No	4
Q8	Information about spider	15	Information about spider wanderer	No	No	10
Q9	List of thorn forest in south India	144	List of thorn forest in south India	Yes	Yes	0
Q10	Java	03	Java island	No	No	19

As it is widely accepted that user try to fulfill its information need by scanning only one or two pages in the result set so precision is calculated at a given cut off links in the result set . The formula for calculating the precision is given in eq^n (3.2)

$$\boldsymbol{Precision}(\%) = \frac{\boldsymbol{retrieved\ relevant\ documents\ with\ in\ cut\_off}}{\boldsymbol{total\ no\ of\ retrieved\ document\ upto\ given\ cut\_off}} \qquad (\mathbf{3.2})$$

In experiment, the number of cut off links is taken as 20. By applying the eq^n 3.2 , the precision of Google, Bing and QSSS is computed and summarized in table 3.7.

Table 3.7: Comparison of Proposed System with Existing Search System

Id	**Google**		**Bing**		**QSSS**		**Precision (%)**		
							Google	Bing	QSSS
	Matched docs	Unmatched docs	Matched Docs	Unmatched docs	Matched Docs	Unmatched docs			
Q1	2	18	1	19	15	5	10	5	75
Q2	2	18	3	17	17	3	10	15	85
Q3	1	19	1	19	16	4	5	5	80
Q4	19	1	18	2	19	1	95	90	95
Q5	17	03	16	4	17	3	85	80	85
Q6	18	2	18	2	18	2	90	90	90
Q7	18	2	17	03	19	1	90	85	95
Q8	2	18	1	19	3	17	10	5	15
Q9	18	2	18	2	18	2	90	90	90
Q10	1	19	2	18	17	3	10	5	85
Average							**48.5**	**47**	**79.5**

The comparison of proposed system with Google and Bing in terms of precision of search results is shown in Fig 3.6.

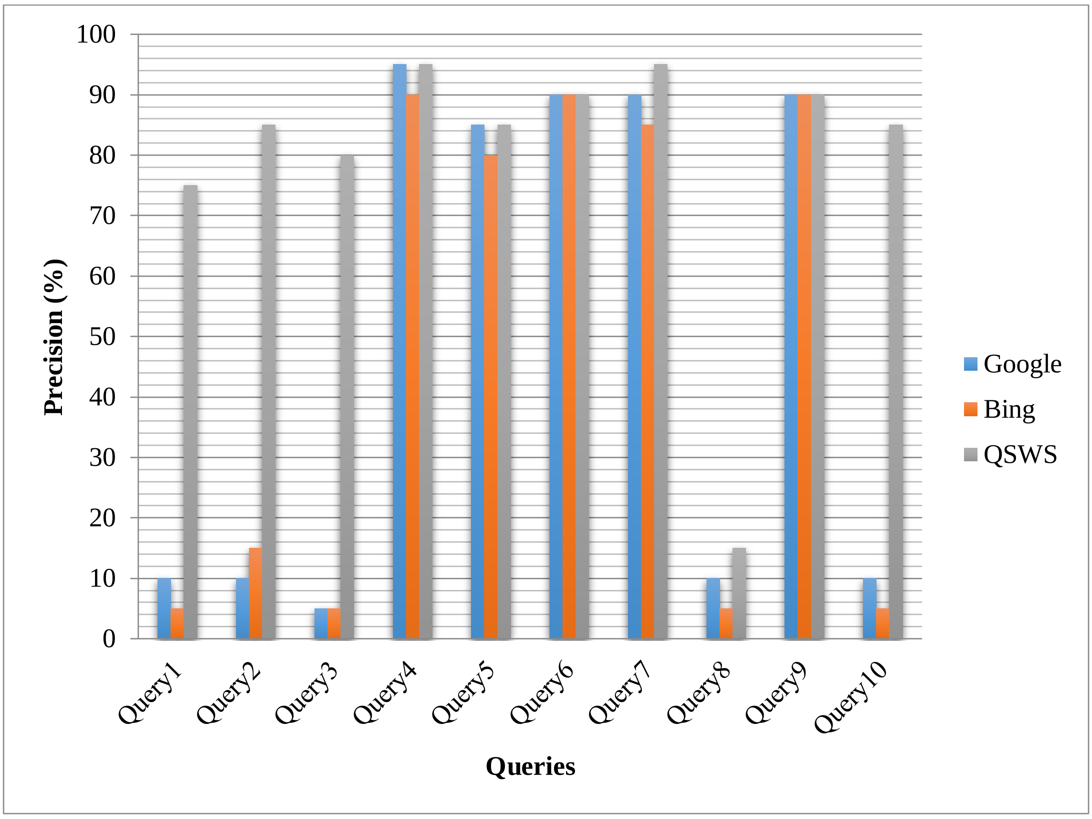


Fig 3.6: Comparison of Proposed System with Existing Search System

It may be observed from Fig 3.6 that proposed Query suggestion system based on semantic similarity achieved the following gains over conventional keyword based query suggestion systems:

- **Average precision of proposed QSSS is more than 79.5%.**
- **The net performance of QSSS in terms of quality of search results comes out to be around 32 % higher than the existing systems.**

3.3 QUERY SUGGESTION BASED ON USER BROWSING HISTORY (QSUB)

Query suggestion has been the inherent feature of information providing services on web since 2008. The aim of suggesting the alternate queries at the level of query submission was to refine the search need of each individual so as to minimize the consequences of keyword based information retrieval process running at the back end. As discussed earlier, the quality of search results can be significantly improved by disambiguating the context of user's search via maintaining the definition repository at the back end with the help of dictionary based sites, But it is observed that dictionary synonyms are not always enough to form query suggestions because user often submit colloquial queries instead of the actual words.

For example, consider the query *'Famous tourist spots in Rajasthan'* The query would map to "*Top tourist spots in Rajasthan'', "Top travel destination in Rajasthan", " Popular tourist places in Rajasthan'', "Popular travel spot in Rajasthan''* as shown in Fig 3.7..But the words such as *top* and *popular* are not the synonyms of the word *famous* so they will not constitute the part of alternate queries as far as the context of the queries are fetched from lexical dictionaries.

Over and above this, the information seekers are variant and have different degree of interest in different domains. For example, the query *'speed of jaguar'* is contextually applicable to car as well as animal. The user *(say user1)* may have the interest in car, so he would have high probability of searching for speed of jaguar as car whereas the other user *(say user 2)* having interest in zoology may mean speed of jaguar as animal. In these situations,

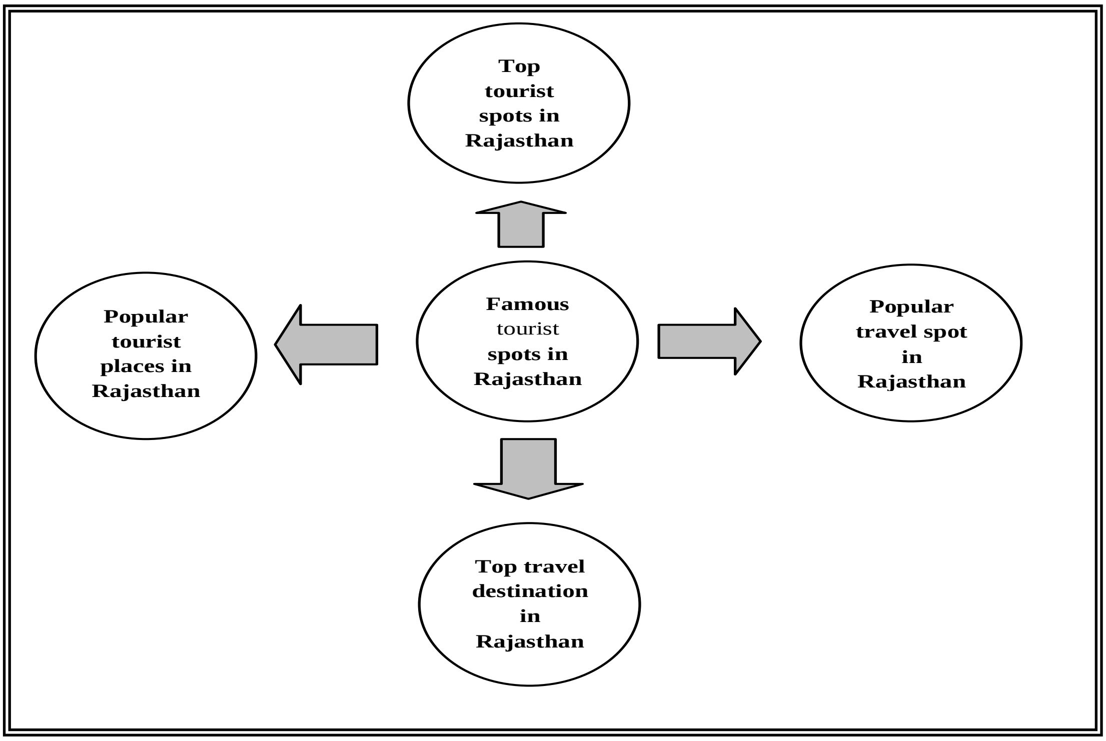

Top tourist spots in Rajasthan
Popular tourist places in Rajasthan
Famous tourist spots in Rajasthan
Popular travel spot in Rajasthan
Top travel destination in Rajasthan

Fig 3.7: Mapping Of Query *'Famous Tourist Places in Rajasthan'* to Different Alternatives

the query suggestion system must be intelligent enough to predict that alternate queries accordingly.

So in the light of above discussion , the mechanism for constructing alternate queries based on contextual senses need to be modified in such a way that after finding the synonyms of query terms, user navigational patterns are used to b form alternate queries thereby providing the personalized results to each user. It will not only help user to quickly define its information need but also results in reduction of search space admirably

The proposed framework for query suggestion based on user browsing history works at the three levels:

1) It maintains the query log to store the historical data about how user actually searched in past. For this, it not only stores the historic queries submitted by the user and their clicked URLs information, but also stores the domain to which the each clicked URL belongs. The aim of maintaining the query log is three folds. First, the queries which are contextually similar are identified. Second, the queries which are not contextually similar, but point to same concept can be identified by their related clicked URLs. Third storing the information about the domain of each assessed page help in forming the personalized query suggestions.
2) It groups the similar queries under one cluster based on context and clicked URL similarity score so as to quickly retrieve the alternate queries.
3) The profile database is maintained to store the degree of interest of each user based on their browsing history so as to issue personalized queries to each individual.

The main intention here is to apply the personalization at the early stage of search process rather than personalizing the ranking list as done by the previous techniques. The query clustering algorithm works offline by mining the query logs and definition repository at the back end. However, the personalized queries are identified online by the query suggestion module of the proposed technique. The amalgam of offline – online execution of various modules of proposed system contributes towards high performance in search process.

The detail discussion of the framework is given in the following section.

3.4 A FRAMEWORK FOR QUERY SUGGESTION BASED ON USER BROWSING HISTORY

The framework for query suggestion system based on user browsing history has been designed as depicted in Fig 3.8. It consists of following four major components:

- User interface
- Query Processor
- Profile generation module
- Query clustering module
- Query recommendation module

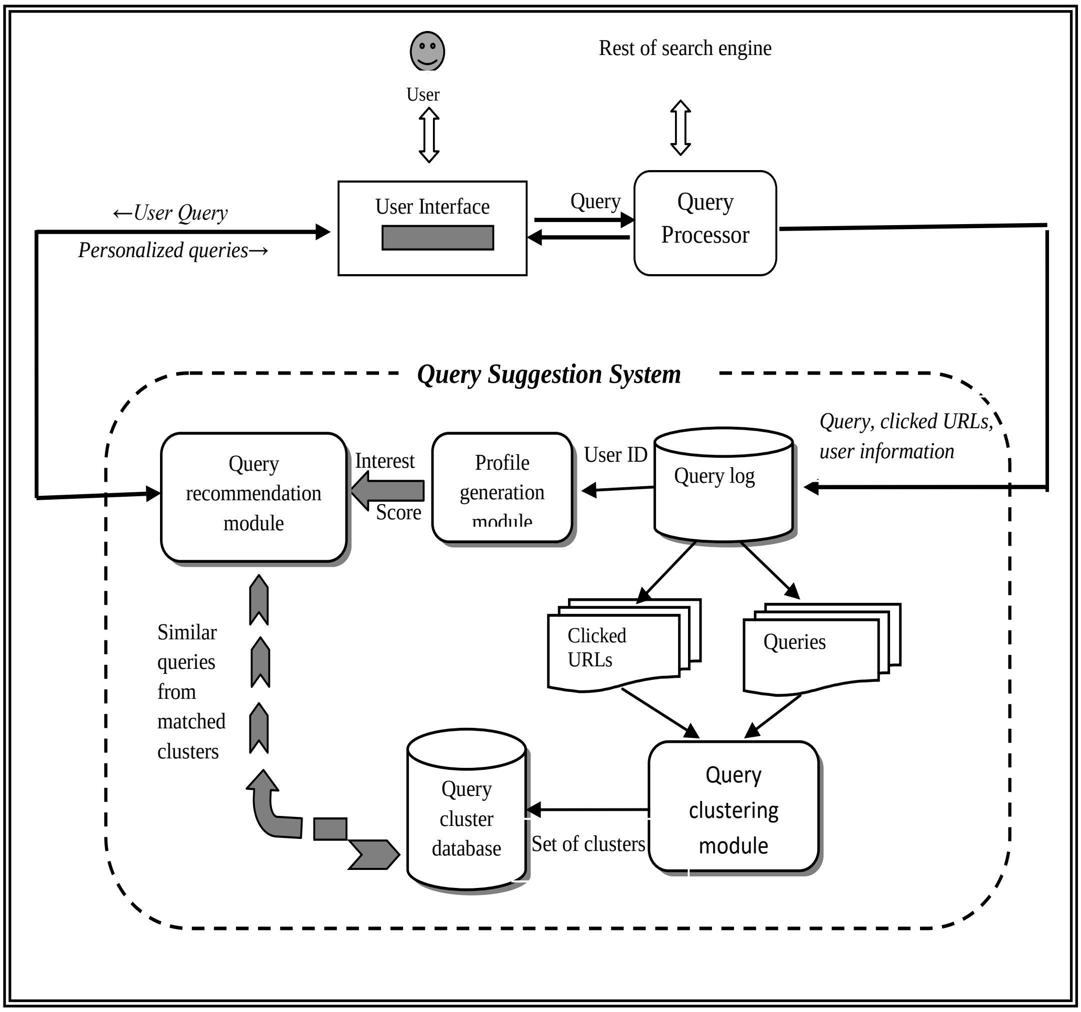


Fig 3.8: Proposed Query Suggestion System

The query recommendation module provides alternate queries to each individual user online by mining the personalized queries from clustered database by means of user interest factor. The information about the user interest is stored in profile database. The query clustering module works offline by grouping the similar queries from query log under one cluster using the concept of web content and web structure mining.

The detail description of each component is given in the following subsections.

3.4.1 User interface

It is an interface where the user specifies its information need in the form of query. It first creates the account for a novice user or verifies the existing user with the help of special module named as profile generation module. After creation/verification, it offers the set of personalized queries to the user with the help of query recommendation module. The user is expected to select one query out to offered queries. The selected query is then passed to query processor to obtain the sorted list of URLs. At last, the sorted list is presented back to the user.

3.4.2 Query Processor

Beside the conventional job of query processor to execute user query on search database, here it is also responsible to maintain the query log. It stores the user information such as user id along with its submitted query; clicked URLS and class to which these clicked URLs belong into query log. This information is further used by profile generation module.

3.4.3 Profile Generation Module

This module maintains the user's information (such as user id, and degree of user's interest) in profile database. In order to accomplish this, the search engine database is proposed to partitioned in different classes $C=\{C_1 , C_2 ....C_m\}$ based on different domains (For instance, in the current implementation the database is partitioned in five classes namely: education, travelling and tourism, sports, food & beverages and fashion & shopping, these classes are further extendible) discussed in more detail in section 6.2.6. The degree of user's interest in a specific class is denoted by *deg* (u_a,C_k) . It can be defined as follows:

Definition 3.1: The degree of user interest in a class is defined as the ratio of no. of pages accessed by user u_a in class C_k to the total no. of pages accessed by u_a in all the classes.

The eq^n (3.3) measures the degree of interest of each user based on page clicks.

$$\boldsymbol{deg}\ (\mathbf{u_a}, \mathbf{C_k}) = \frac{\boldsymbol{NC(u_a, C_k)}}{\sum_{i=1}^{m} \boldsymbol{NC(u_a, C_i)}} \qquad \mathbf{(3.3)}$$

Where:

- ***NC(u$_a$, C$_k$)*** denotes the no. of pages clicked by user u_a in class C_k,
- m is the no. of classes in search engine database.

For example if the user clicks 63 documents belonging to a particular class out of total 100 clicks in all classes Then degree of user interest is 0.63. The working of profile generation module is depicted in Fig 3.9.

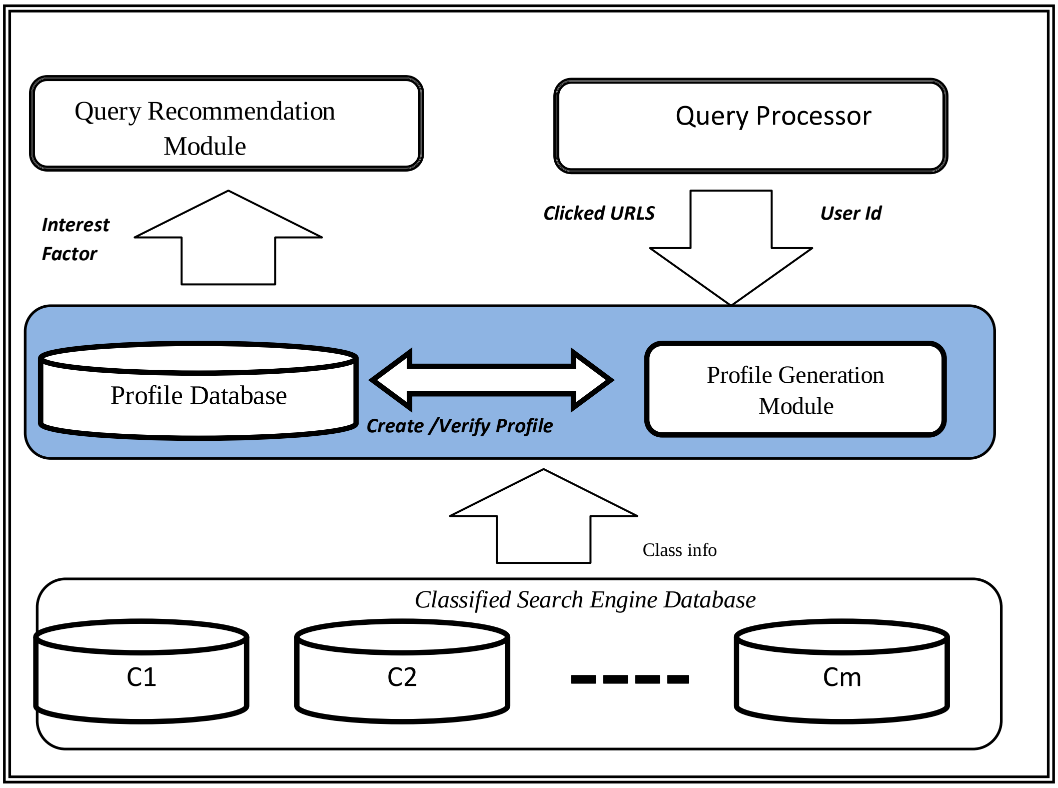


Fig 3.9: Role of Profile Generation Module in QSUB

It gets user identification information (user id) and browsing information (clicked URLs) from query processor as inputs, computes the degree of interest of user and updates the same in profile database. The algorithm for profile generation module is given in Fig 3.10. In the present work, the profile database is maintained in MS SQL Server 2012. The snapshot of the same is also included in Appendix C.

Profile_Generation( )

Input: uid, clicked pages, partitioned database containing Set of classes C= {C1, C2......Cm} ,

Output: Profile database containing degree of interest of each user, Profiler[n][m] ; where n is no. of users and m is no. of classes in search engine database

Method:

```
Step 1: Initialize n← 0;
        2. wait (uid);
        3. Do {
          Check uid  in Profile database.
            3.1  If (uid  ε Profiler[n][m] ){
                 3.1.1 Wait (click_uid);
                 3.1.2 Do for each user click on any page P ε Ck{
                 3.1.3 NC(uid, Ck) ←NC(uid,Ck)+1;
                 3.1.4 Do for each class Ci ,
                     3.1.4.1 Update
                         $deg(ua\,,CK) = \frac{NC(ua, Ck)}{\sum_{i=1}^{m} NC(ua, Ci)}$
}}
           3.2 else {
                 3.2.1 Create a new entry for uid in Profile database
                 3.2.2 Set for all class Ci, NC[uid,Ci] ←0}}
```

Fig 3.10: Algorithm for Profile Generation Module

The algorithm updates the degree of user interest every time a click is encountered on a page belonging to some class in current user session. This information is very useful in filtering out the alternate queries to be offered to the user. To better understand the working of Profile generation module, A small fragment of user profile database at any time 't' is presented here in table 3.8.

Table3.3.8: A Small Fragment of Profile Database at any time ‘t'

User	Classes				
	C1	C2	C3	C4	C5
User 1	0.4	025	0.15	0.2	0
User 2	0.22	0.05	0.5	0	0.22
User 3	0.63	0	0	0.36	0
User 4	0	0.7	0.2	0.09	0.01
User 5	0.63	0	0.1	0.07	0.2

The value in each cell of table 3.8 represents the degree of interest of a particular user in particular class. . The interest value in particular class is computed using eq^{n} 3.3. For instance, the value 0.4 represents user 1 has clicked 40 documents belonging to class C1 out of total 100 clicks in all classes .Likewise the other values are computed. So user 1 posses 40 % interest in class C1, 25 % interest in class C2, 15 % interest in class C3, 20% interest in class C4 and no interest in class C5 at time ‘t’. These clicks are updated in every user session. So the technique has also taken both short term and long term interest of user well into consideration.

It may be further analyzed from the table 3.8 that each user possesses different level of interest in different classes. This information is utilized by query recommendation module in finding personalized query for the user.

3.4.4 Query Clustering Module

This module is responsible to group the similar queries under a common cluster based on two main concepts as discussed below:

A. Evaluating similarity based on context of query terms: Two queries are said to be similar if query terms or synonym of query terms match above a threshold value say, $\boldsymbol{T_{context}}$**.** To compute the context similarity between two queries P and Q, the eq^{n} (3.4) is formulated.

$$\boldsymbol{Sim_{context(P,Q)}} = \boldsymbol{max}\left\lfloor \frac{|QT(P)\cap QT(Q)|}{max\{|QT(P)|,|QT(Q)|\}}, \frac{|QT(P)\cap QT(SQ)|}{max\{|QT(P)|,|QT(SQ)|\}}, \frac{|QT(SP)\cap QT(Q)|}{max\{|QT(SP)|,|QT(Q)|\}}, \frac{|QT(SP)\cap QT(SQ)}{max\{|QT(SP)|,|QT(SQ)|\}} \right\rfloor \quad (3.4)$$

Where:

- QT (P) and QT (SP) represent the terms in query P and synonym of terms in query P respectively. Similarly, QT (Q) and QT (SQ) represent the terms in query Q and synonym of terms in query Q respectively.
- |QT (P)| and |QT (Q)| measure the count of terms in query P and Q respectively.

B. Evaluating similarity based on common clicked URL: If two queries lead to the selection of same URL, then they may be considered as similar. In order to find the extent to which they are similar, the concept of no. of clicks on common URLs is introduced here. Formula for measuring the similarity between two queries based on no. of clicks on common URLs is given in eq^{n} (3.5).

$$\boldsymbol{Sim_{ClickedURL}(P,Q) = \sum_{i=1}^{n} \frac{min(NC(P,Li),NC(Q,Li))}{\max(NC(P,Li),NC(Q,Li))}} \; ;\forall \boldsymbol{Li} \in \boldsymbol{CL(P)} \cap \boldsymbol{CL(Q)} \qquad (\mathbf{3.5})$$

Where:

- *CL (P)*, *CL(Q)* are the sets containing the clicked URLs corresponding to query P and Q respectively.
- *NC (P, Li)* and *NC (Q, Li)* are no. of clicks on URL with respect to query P and Q respectively.
- n denotes the number of common clicked URLs with respect to query P and Q.

C. Combined similarity measure:

The two similarity concepts described above have their own benefits. On the one hand, the contextual similarity groups all those queries which share the similar composition of query terms or synonyms of query terms into one cluster. On the other hand, the common click based similarity takes the advantage of user feedback in identification of similar queries. But alone each of them can partially capture the similarity among the queries, so, it’s better to combine both the measures in a single measure as shown in eq^{n} (3.6).

$$Sim_{combined(P,Q)} = (1 - \mu)Sim_{context(P,Q)} + \mu\, Sim_{clickedURL(P,Q)} \qquad (3.6)$$

Where:

- $Sim_{combined(P,Q)}$ denotes the similarity between query P and Q based on both measures i.e context as well as clicked.
- μ is similarity constant such that $\mu \in [0,1]$.

In the current implementation value of μ is taken as 0.5. If the $\mathrm{Sim}_{\mathrm{combined}}$(P,Q) is greater than the pre- defined threshold value $T_{combined}$,they are grouped under the same cluster. The algorithm for query clustering module is given in Fig 3.11.

Query _Cluster( )

Input: Similarity constant , similarity threshold $\mathrm{T}_{\mathrm{combine}}$, Query log containing the following fields:

1. User ID of each user
2. Query ID assigned to each query
3. Query
4. URLs selected by user corresponding to Query ID
5. No. of clicks on selected URLs
6. Class ID of selected URL
7.

Output: Set of clusters Clust = {clust1,clust2 .... clustn}; each cluster contains the following information.

1 Cluster ID
2 A collection of similar queries with clicked URL and class ID
3 Keyword set of each cluster

Method:

```
1  Intialize n-0                                    //No. of clusters
2  Initialize i=0                                   // No. of queries
3  Do for every query qi € query log{
   Flag[qi]=0;
   Clust(qi)=null;}
```

```
    4.  Find cluster for new query
                Do for each query qi € query log
                {
                4.1 if ( Flag[qi]= 0){
                    n=n+1;
                    cluster(qi)=$clust_{n;;}$
                    $clust_{n;}$=(qi, clicked URLs, class id);
                    $keyword_{clustn}$= stem(qi);
                    Flag=1;
                    Do for each query, $q_{i+1}$ € query log such that qi ≠ $q_{i+1}${
                    Find sim $_{combined}$(qi, $q_{i+1}$) using eqn(3.6);
                    If (sim $_{combined}$(qi, $q_{i+1}$) $\geq T_{combined}$){
                    $clust_{n;=}$ $clust_{n;}$∪ {( $q_{i+1}$,clicked URLs, class id)};
                    $keyword_{clustn}$= $keyword_{clustn\ ;}$∪  stem($q_{i+1}$);
                    clust($q_{i+1)}$= $clust_{n;}$;}
                    }}

                4.2 else
                        i = i+1;
}}
```

Fig 3.11: Algorithm for Query Clustering Module

To explain this algorithm, let us first measure the context similarity among the four queries Q1, Q2, Q3 and Q4 given in table 3.9. Initially the queries does not belong to any cluster i.e. set of clusters denoted by Clust=φ.

Table 3.9: Sample Query Examples

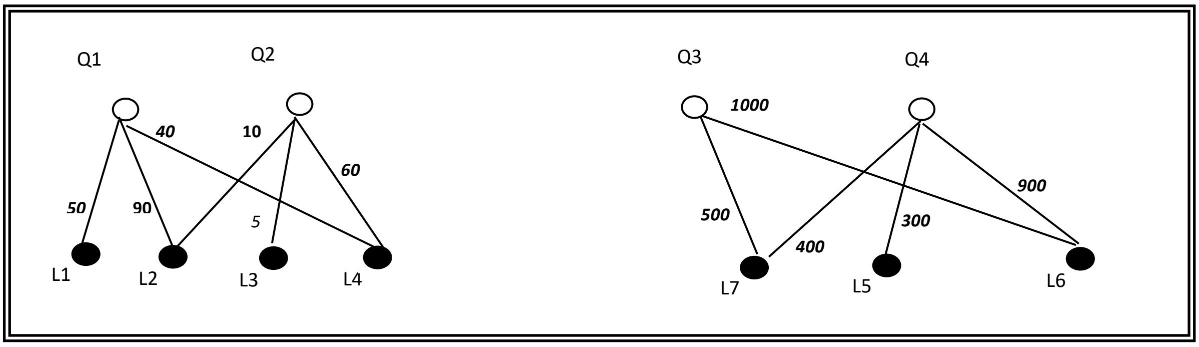
Query id	**Queries**
Q1	apple jams recipes
Q2	jam recipes
Q3	apple OS
Q4	Feature of mac

By applying eq^{n} (3.4), the context similarity between the queries represented by $Sim_{context(Qi,Qi+1)}$ is computed.

$$Sim_{context(Qi,Qi+1)} = \begin{pmatrix} 1 & 0.66 & 0.33 & 0 \\ 0.66 & 1 & 0.5 & 0 \\ 0.33 & 0.5 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Taking $\boldsymbol{T_{context}}$ = 0.65. The four queries can be grouped into three clusters i.e. Clust={ Clust$_1$, Clust$_{2,,}$ Clust$_3$} such that Clust$_1$={Q1, Q2,} and Clust$_2$={Q3}, Clust$_3$={Q4}.

For measuring the similarity based on formula (3.5), the query clustering module first constructs the bipartite graph in which one set of nodes corresponds to queries and other set of nodes corresponds to clicked URLS as shown in Fig 3.12 The no. mentioned on an edge e_i joining Qi and Li represents the no. of times the Li get selected w.r.t. Qi. For example, in Fig 3.12 the value 40 mentioned on edge joining Q1 and L4 implies that 40 clicks are received on URL L4 w.r.t query Q1. Further, it is considered that the user click on any URL w.r.t a query can be taken as a good source of user feedback.



(a) **(b)**

Fig 3.12: Sampled Bipartite Graph

In the Fig 3.12 (a) L1, L2, L4 are selected with respect to query Q1, which implies that they are relevant to query Q1. Similarly, L2, L3 and L4 are relevant to query Q2. In the Fig 3.12 (b), L6, are L7 are selected with respect to query Q3 which implies that they are relevant to query Q3. Similarly L5, L6 and L7 are relevant to query Q4. As Q1 shares common URLs with Q2 and Q3 shares common URLs with Q4. So they may be considered as similar. The extent to which Q1 is similar to Q2 and Q3 is s similar to Q4 can be measured using eqn (3.5) as follows:

$$Sim_{clickedURL(Q1,Q2)} = \frac{\min(90{,}10) + min(40{,}60)}{max(90{,}10) + \max(40{,}60)} = \frac{50}{150} = 0.33$$

$$Sim_{clickedURL(Q3,Q4)} = \frac{\min(500{,}400) + min(1000{,}900)}{max(500{,}400) + \max(1000{,}900)} = \frac{1300}{1500} = 0.87$$

$$Sim_{clicked(Qi,Qi+1)} = \begin{pmatrix} 0 & 0.33 & 0 & 0 \\ 0.33 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.87 \\ 0 & 0 & 0.87 & 0 \end{pmatrix}$$

By applying eq^{n} (3.6), finally the combined similarity between each pair of query is computed as follows:

$$Sim_{combined(Qi,Qi+1)} = 0.5\begin{pmatrix} 1 & 0.66 & 0.33 & 0 \\ 0.66 & 1 & 0.5 & 0 \\ 0.33 & 0.5 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} + 0.5\begin{pmatrix} 0 & 0.33 & 0 & 0 \\ 0.33 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.87 \\ 0 & 0 & 0.87 & 0 \end{pmatrix}$$

$$Sim_{combined\ (Qi,Qi+1)} = \begin{pmatrix} 0.5 & 0.49 & 0.165 & 0 \\ 0.49 & 0.5 & 0.25 & 0 \\ 0.165 & 0.25 & 0.5 & 0.43 \\ 0 & 0 & 00.43 & 0.5 \end{pmatrix}$$

So, Taking $T_{combined}$ = 0.3. The four queries are finally grouped into two clusters such that Clust$_1$={Q1, Q2,} and Clust$_2$={Q3,Q4} .It may be analyzed from the example that inspire of having no common keywords with any of query for Q4, it is identified as similar to Q3 based on clicked URLs. Once the query cluster is identified, they are saved under their respective cluster in clustered database.

Cluster database is further retrieved by query recommendation module for finding alternate query suggestions for the user.

3.4.5 Query Recommendation Module

It receives the user query from search engine interface and returns the set of alternate queries to be presented back to user. It applies two level of filtering process to construct the set of alternate queries. First, it retrieves the clusters from query cluster database whose keywords match with the query keywords. The four most popular queries are extracted from each matched cluster. It is assumed that the query which is submitted by more users is more popular. Second, the set of popular queries are filtered on the basis of user interest area. The profile generation module provides the interest score of each user in different domains. So, more personalized queries are returned to search engine interface to offer them to user. The algorithm for query recommendation module is given in Fig 3.13.

```
Query Recommend ( )

Input: user query Q, Query cluster database containing the following fields:
1. Cluster ID assigned to each cluster
3. Set of keywords associated with each cluster.
4. Set of queries with query weight
4. Set of clicked URLs for each query
5. Class ID of each URL

Output: Set of personalized queries $Q_{personalized}=\{q1,q2,q3,q4\}$

Method:
    1. Set $Clust_{matched}$=NULL for Q;
    2. Find matching cluster for user input query
       For i=1 to n{
               if $(Keyword(Q) \cap Keyword(clust_i)t > Threshold)$
                       $Clust_{matched}=Clust_{matched}\ \upsilon\ \{Clust_i\};$}
    3. Call the profile generation module to give Interest weight of UID in each class.
    4. Sort the $Q_{personalized}$ on the basis of interest weight;
    5. Pop the first four queries from $Q_{personalized}$ and return it to search engine
       interface,
    6. Fetch the four most popular queries from each matched cluster and store them
       in $Q_{personalized.}$ along with their class ID.
```

Fig 3.13 Algorithm for Query Recommendation Module

To better understand the Query recommendation process, let us consider the scenario where one user is having more interest in food and beverages class whereas other user is having more interest in fashion and shopping class (their degree of interest value in corresponding classes are maintained by profile generation module) . when these two user submit the same query “apple” at search interface, they receive the different set of alternate queries by the query recommendation module as listed in table 3.10.

Some snapshots related to working of query recommendation module for different users having different interest areas are also included in Appendix C.

Table 3.10: Personalized Query Recommendation

	USER ID	
	User 1	**User 2**
QUERY RECOMMENDATIONS	apple fruit price	apple iphone
	apple benefits	apple India
	apple jam recipes	apple OS
	jams recipes	Feature of mac

The performance evaluation of QSUB and their experimental results are given in following section.

3.4.6 Performance Evaluation

Query suggestion based on user browsing history (QSUB) is implemented using C#.NET and MS SQL Server 2012. The snapshots of implementation are included in Appendix C. The algorithm has been tested on the query log given in Appendix A. A profile database is also maintained to record the degree of user interest in each class. The profile database is accessed by profile generation module on every interaction of user with search system.

Procedure: To evaluate the relevance of query suggestions provided to the user, a volunteer group of 25 students from YMCA University are asked to submit queries at QSUB interface. Then a suggestion relevance score (SRS) is calculated for the alternate queries given by the proposed system using eq^n(3.7).

$$\textit{Suggestion Relevance Score (SRS) in \% =RS/TS*100} \quad \textbf{(3.7)}$$

Where:

- *RS* is the number of relevant query suggestions (according to user)

- *TS* is the total no. of query suggestions given by the system for submitted query

Table 3.11 to .3.15 list the set of queries submitted by five different users and the average SRS attained in each case. It may be observed from these tables that average SRS is more than 70 % which implicitly reflects the relevancy of proposed technique.

Table 3.11 SRS for Set of Queries (Set 1) Submitted by User 1

S.No.	**Queries**	**Query Type**	**No. of Relevant Query Suggestions(RS)**	**Total No. of Query Suggestions (TS)**	**SRS=RS/TS*100**
Q1	Best iphone price	Shopping	3	4	75.0
Q2	Samsung phone	Shopping	3	3	100.0
Q3	Best apple phone	Shopping	3	4	75.0
Q4	Tennis player in India	sport	2	4	50.0
Average SRS (%)					**75.0**

Table 3.12: SRS for Set of Queries (Set 2) Submitted by User 2

S.No	**Queries**	**Query Type**	**No. of Relevant Query Suggestions(RS)**	**Total No. of Query Suggestions (TS)**	**SRS=RS/TS*100**
Q1	Package for singapore	Travel	3	4	75.0
Q2	Singapoe dollar	Travel	1	2	50.0
Q3	Best place	Travel	3	4	75.0

	to live in India				
Q4	Engineering college	Education	3	4	75.0
Average SRS (%)					**68.75**

Table 3.13: SRS for Set of Queries (Set 3) Submitted by User 3

S.No.	**Queries**	**Query Type**	**No. of Relevant Query Suggestions(RS)**	**Total No. of Query Suggestions (TS)**	**SRS=RS/TS*100**
Q1	Places of kingfisher	Education	1	2	50.0
Q2	Oops tutorials	Education	3	4	75.0
Q3	Protein rich food	Food	4	4	100.0
Q4	Haldiram nearby	Food	3	3	100.0
Average SRS (%)					**81.25**

Table 3.14: SRS for Set of Queries (Set 4) Submitted by User 4

S.No.	**Queries**	**Query Type**	**No. of Relevant Query Suggestions(RS)**	**Total No. of Query Suggestions (TS)**	**SRS=RS/TS*100**
Q1	Food of Gujarat	Food	4	4	100.0
Q2	Pink city India	Travel	3	4	75.0
Q3	Clean city India	Food	4	4	100.0

Q4	Delhi to Ahmadabad airliners	Travel	3	3	100.0
Average SRS (%)					**93.75**

Table 3.15: SRS for Set of Queries (Set 5) Submitted by User 5

S.No.	Queries	Query Type	No. of Relevant Query Suggestions(RS)	Total No. of Query Suggestions (TS)	SRS=RS/TS*100
Q1	Babita geeta story	sport	2	2	100.0
Q2	Common wealth games 2010	sport	4	4	100.0
Q3	ymcaust	education	3	4	75.0
Q4	Ps4 price	Shopping	2	4	50.0
Average SRS (%)					**81.25**

Plot of average SRS attained for five different sets entered by five different users of varing interest is presented through graph given in Fig 3.14.

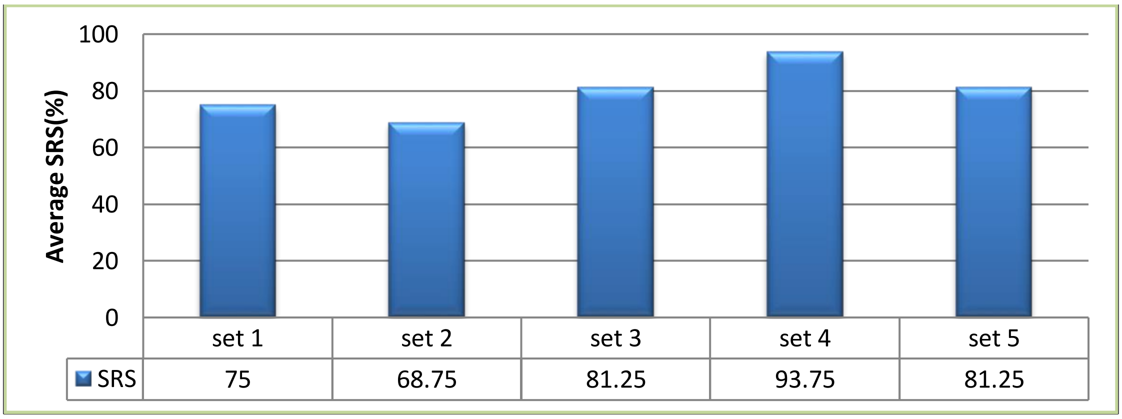


Fig 3.14: Plot of average SRS(%)

Therefore performance evaluation of QSUB indicates the following gains over conventional keyword based query suggestion systems:

- Average relevance score of proposed QSUB is more than 79%.
- The net performance of QSUB in terms of personalized query suggestion is remarkable as compared to existing systems

The technique proposed a novel query recommendation technique for implementing the efficient search engine .It suggests the personalized queries to individual user so that their diversified need can be fulfilled. The technique makes use of context similarity and click through data similarity among the queries to group them in relevant cluster. The user query is matched with query cluster to retrieve the relevant alternate queries from cluster database. The promising part of proposed system is that the alternate queries are further refined based on degree of interest of each user in different classes.

By refining the user search need at early stage results in reduction of time user spent for seeking out the desired information from search list. The result obtained from the experimental evaluation shows the increase in user satisfaction level with respect to query suggested by proposed system. Further more personalized techniques may be embedded in ranking phase which can provide more comprehensive ranked list to each individual user. The page ranking based on user browsing patterns is discussed in next chapter.

CHAPTER IV
A NOVEL PAGE RANKING TECHNIQUE BASED ON USER BROWSING PATTERNS

4.1 INTRODUCTION

The key challenge in front of search engine is to efficiently harness the web information and present relevant results to the user in easy accessible and fast way. A typical search engine generally encounters a massive collection of web documents from its database in response to a user query. To provide the user an easy way to navigate through such a colossal, a variety of page ranking techniques are applied. Some of these techniques assign the rank score to a page based on its in link and out link popularity in the web. Some other techniques rely on the content inside the page. Whereas rest are amalgam of both, to better evaluate the relevance of a page with respect to user query. But in spite of using these sophisticated ranking techniques, the results set obtained is still flooded with magnitude of irrelevant documents, rendering the required information a hard task. Consequently, the onus of finding the desired information still lies at the user's end. So the concept of precision and recall of search results come into play [22].

In context of information retrieval, Precision is defined as the ratio of relevant documents to the total number of retrieved documents whereas recall is defined as the ratio of relevant documents that are retrieved to total number of relevant documents [21]. The quality of search engine is highly affected by the precision and recall of its result set. To increase precision and recall of result set, it is significant to consider the user's feedback about the relevance of page for evaluating its rank score.

Further, the user's feedback can be obtained either explicitly asking the user or implicitly deducing from the interaction done by the user with the search engine [114]. As it is observed that user is less prone to give any explicit feedback [115]. So, an implicit way to capture user's feedback is need of today.

Based on the above discussion, a novel page ranking technique based on user browsing patterns has been developed for the interest based search system. Various flavours of web mining (i.e. web content mining, web structure mining and web usage mining) are merged to produce the highly relevant result list. A number of parameters reflecting the context and interest of user's search are identified and utilized in evaluating the selection probability of the page in comparison to other pages in the corpus. Based on page selection probability, a numeric score called page probability factor (PPF) is assigned to each page. The PPF associated with each page further contributes towards page rank computation. The following section provides the proposed page ranking technique in detail.

4.2 PROPOSED PAGE RANKING TECHNIQUE BASED ON USER BROWSING PATTERNS

Besides other routine functions, sorting the search results in accordance with user's need is an inherent task that a search engine has to deal with. Otherwise many irrelevant pages may appear in the result set to be presented back to the user in response to its query. The presence of irrelevant documents in the result set significantly decreases the quality of search engine.

Many sophisticated page ranking techniques such as *PR* (pageRanl), *WPR* (weighted page rank), *HTTS* (Hypertext induced topic search), *PRLV*(page rank based on link visit) etc have been proposed by researchers in past [133]. All of these techniques are based on the concept of web mining. PR, WPR and HTTS are purely based on web structure mining whereas PRLV is based in web usage as well as web structure mining both. None of the earlier approaches had used combination of all three mining techniques. Therefore a navel page ranking framework based on the concept of web content mining, web structure mining and web usage mining is being proposed here which enhances the search engine performance by utilizing the user browsing patterns along with content-match, and in links/out links information of the page while assigning the relevance score to a page thereby providing the satisfactory results to the user. The framework uses a unique technique in three ways:

i. It basically works towards improving the quality of search results.

ii. It utilizes the browsing patterns of all the users to compute the selection probability of a page thus avoiding the creation and storage of cumbersome profile of each individual which leads to reduction in the seek time to provide relevant results to the user.

iii. It uses various techniques of web mining to compute the overall rank score of the page.

The main intention here is to organize the search results in such a way that can better fulfill the user's information need rather than optimizing the crawling process to enrich the search engine repository. The algorithm to compute the relevance score of page assigns a numeric score to each page by capturing the user's actions (such as click, save , bookmark etc) online on the page with the aid of `proposed module called ***page probability calculator (PPC) module*** (discussed in section 4.3.1). However the relevance score based on content and link structure of the page is computed offline. The detail discussion of the proposed framework is given in the following sections.

4.3 PAGE RANK COMPUTATION FRAMEWORK

In order to accomplish the need of navigational tool, which can ensure the relevance of page in accordance with user's information need and present a sorted result list in quick and easily browsable way, A novel page ranking framework has been proposed and depicted in Fig 4.1. The proposed framework suggests two important components to the existing search engine architecture:

I. Page Ranker

II. Page probability calculator

The entire process of information searching beginning from submission of the query till sorted result presentation can be outlined in the following steps:

Step1: Retrieving the matched pages

When the user submits its information need in the form of query at search engine interface, it is propagated to query processer to extract the functional terms from it .The query processor matches the functional term with pages stored in search engine database. It fetches the matched page URLs and their associated page information and stores them in buffer.

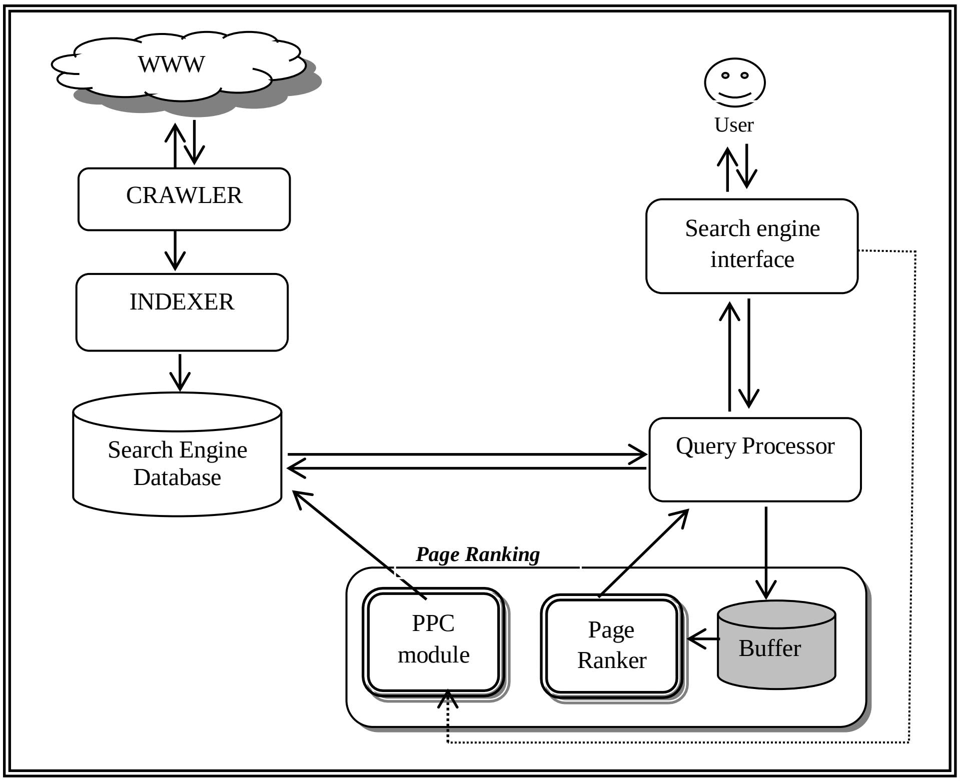


Fig 4.1: A Framework for Search Result Optimization

Step 2: Computing the document weight

The page information and query term information is taken by the page ranker from the buffer to compute the extent of content similarity between the matched pages and user query using the technique described later in section 4.3.2.1. Based on the content similarity, it assigns a numeric score termed as document weight, $\boldsymbol{weight_{doc}}$ to each page.

Step 3: Computing the link weight

Page ranker also retrieves the in links and out link information associated with the page and computes the link weight demoted by $\boldsymbol{weight_{link}}$ based on their popularity using the technique described in [141].

Step 4: Computing the PPF of the page

This Page probability calculator module becomes active as soon as the user submits the query at search engine interface. It analyzes each and every action done by the user on the clicked page and accordingly assigns the numeric score denoted by $\boldsymbol{PPF(P_i)}$ to a page $\boldsymbol{P_i}$, that actually reflects relevance of page from user's point of view.

Step 5: Prepare the sorted list

Finally the page ranker component computes the overall page rank to be assigned to each matched page $\boldsymbol{P_i}$, denoted by $\boldsymbol{Page_{Rank}(Pi)}$ by adding content weight , link weight and PPF. Based on this overall rank all the pages in the matched set are sorted and resultant list is presented back to the user.

To better understand the aforementioned steps, the workings of PPC and page ranker module are discussed in the following subsections.

4.3.1 Page Probability Calculator (PPC) Module

Owing to the fact that user's browsing patterns are implicit source of user's feedback, it computes the selection probability of page called Page Probability Factor denoted by ***PPF*** of each page by analyzing and mining the certain attributes during the information seeking process. Inferring the user behaviour from its browsing patterns is a tedious task. The privacy of user data is major concern while deciding the factors to be captured and utilized in page ranking.

Keeping in mind the aforementioned facts, the proposed approach identified the three most important factors that must be considered. They are: Click, time spent and action performed by the user on the page. Based on these factors, the working of ***PPC*** module is as follows:

When a user submits a query at search engine interface, ***PPC*** receives a signal '*something to monitor*' from search interface and become active. It observes and records three important facts about the user activity as shown in Fig 4.2. They are discussed below:

- **Click:** As the click made by the user on a particular page directly implies page relevance as compared to un-clicked pages so *PPC* observes each and every click made by the user in the result set and accordingly increases its selection probability.

- **Time spent:** Another important factor that is considered as implicit source of relevance of page is the amount of time it is viewed by the user. As more time spent by the user on a particular page entails that user is finding desired information in it. So the *PPF* of the page is increased.

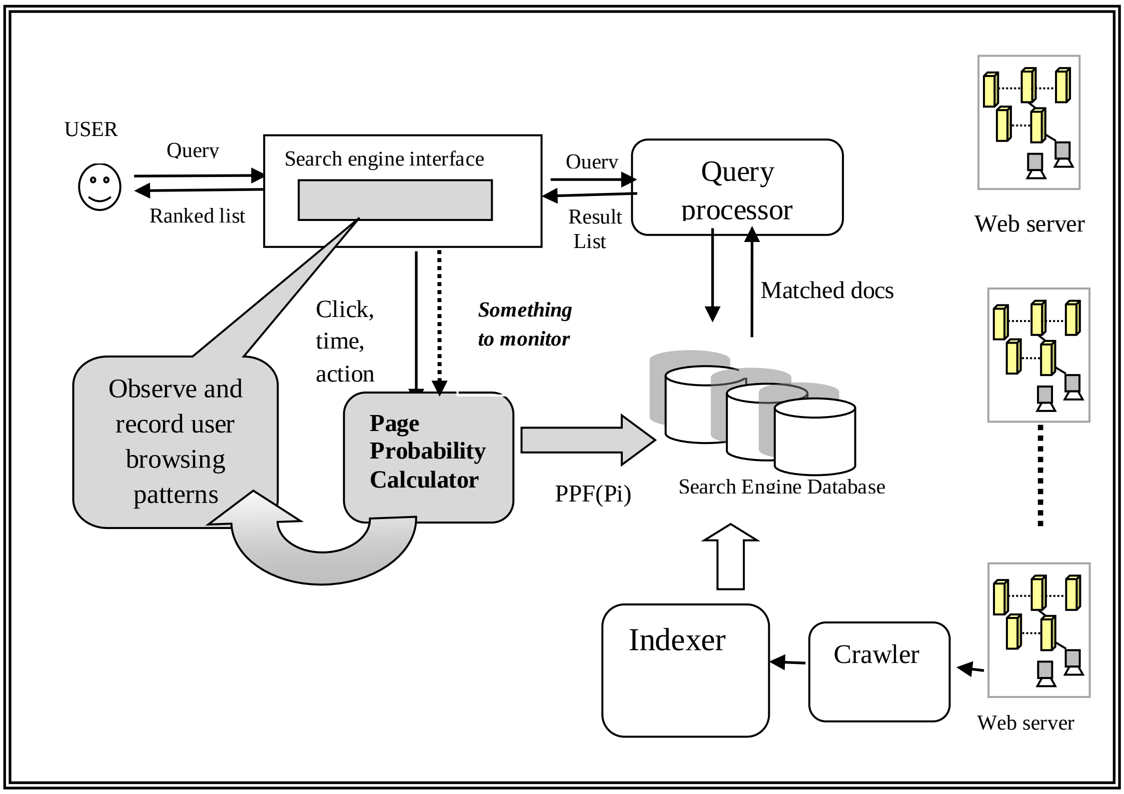


Fig4.2: Working of Page Probability Calculator

- **Action taken**: The actions taken by the user such as print, send, bookmarks, save are some of the important attributes reflecting the relevancy of page for user, so they are also mined and recorded to increase the selection probability of the page.

Hence in the view of the above discussion, the page probability factor *(PPF)* can be defined as follows:

Definition 4.1: it is a numeric score obtained by a page based on number of clicks, time spent and action taken by the user on the page in information seeking process that can further be used to find the relative importance of the page as compared to other pages in the alike domain.

It can be evaluated by the eq^{n} (4.1).

$$PPF(Pi) = \alpha\, CLICKwt(Pi) + \beta\, TIMEwt(Pi) + \gamma\, ACTIONwt(Pi) \qquad (4.1)$$

Where:

- *CLICK*$_{wt}$ *(P*$_i$*)* denotes the importance of page p$_i$ with respect to all the pages clicked by user u$_i$ for query q$_i$ in the current search session.
- *TIMESCORE(P*$_i$*)* denotes the relative time spent by user 'u$_i$' on the page P$_i$
- *ACTIONwt (P*$_i$*)* denotes the action performed on the page P$_i$

$\alpha, \beta\ and\ \gamma$ are the constants reflecting the relative importance of three attributes i.e. click, time and action respectively. Among these three browsing parameters of user behavior, action has given the highest weightage because if a user is taking some action on the clicked page it truly reflects that user is finding some/all desired information in that page; time comes next and click has given the lowest weightage. That is why the value of $\alpha, \beta\ and\ \gamma$ are taken in such a way $\alpha < \beta < \gamma$. The value of $\alpha \in [0.1,0.33), \beta \in [0.33, 0.66)$ and $\gamma \in [0.66,1)$.

The optimal values of $\alpha, \beta\ and\ \gamma$ in the defined intervals are computed by taking mean of centroids of area covered by every clicked page [142]. Their value obtained in the current implementation is 0.17, 0.49 and o.82 respectively. The detail regarding computation is given in section 4.3.1.4. Further the detail computation of *click*$_{wt}$, *Time*$_{wt}$ and *Action*$_{wt}$ is given in following sub sections.

4.3.1.1 Calculation of click weight on page P$_i$

When a user clicks a page, the click weight of page P increases as if the user votes for this page. For any more clicking by the same user, the click weight of page will not be affected in order to avoid unbiased increase in rank score. To find the importance of any arbitrary page ***P$_i$*** with respect to query q, the click weight can be evaluated by using eq^n (4.2).

$$\boldsymbol{Click_{wt}(P_i) = \frac{C}{|\ click(q,*,u)\ |}} \qquad \mathbf{(4.2)}$$

Where:

- ***Click (q, *, u)*** denotes the total no. of clicks made by user u on all the pages for query q in current session.
- ***C*** is no. of votes for a page. It is set to 1 for clicked pages and 0 otherwise.

4.3.1.2 Calculation of time weight on page P_i

Time is also an important factor as more time the user spent on something, more he/she is interesting in it. Time weight of document $\boldsymbol{P_i}$ is computed by analyzing its relevancy with respect to document ***p*** whose view time is maximum among all clicked pages in current session. The time weight of page $\boldsymbol{P_i}$ with respect to query $\boldsymbol{q_i}$ can be evaluated by using eq^n (4.3).

$$\boldsymbol{Time_{wt}(P_i) = \frac{time\ spent\ (P_i)}{highest\ time\ spent(P)}} \qquad (\mathbf{4.3})$$

Where:

- ***Time spent(P_i)*** is the time spent by user u_i on page P_i in minutes.
- ***Highest time spent (P)*** is the maximum time spent by user u_i on any arbitrary page in the result set in current session.

4.3.1.3 Calculation of action weight on page P_i

Actions that user may carry on any web document are listed in table 4.1 along with the weights. The weight is assigned according to the relevancy of the action where relevancy is determined based on user feedback in response to a survey. It is observed in the survey that if someone is printing the page means it has higher utility at present, Saving is less scored as the user will require it later on, bookmark come next and Sending comes at last in priority list as page is used by some other user. If a user performs more than one action then only the higher weight value is considered. For example, if user perfume printing as well as saving, then only the printing weight is assigned to the page.

Table 4.1: Action Weight

Action	Weight
Print	0.4
Save	0.3

Bookmark	0.2
Send	0.1
No action	0

The algorithm for Page probability calculator module in given in Fig 4.3.

```
PPC ()
Input: clicked URLS.
Output: PPF score of clicked URL
Method:
1: Wait( something to monitor);
2: do{
        2.1 for each clicked page Pi {
                2.1. 1 Record the time spent and action performed on page Pi}
        2.2 For each recorded URL {
                2.2.1 Calculate $CLICK_{wt}$ by using $eq^n$(4.2)
                2.2.2 Calculate $TIME_{wt}$ by using $eq^n$ (4.3)
                2.2.3 Assign $Action_{wt}$.as per table 4.1
                2.2.4 Calculate PPF score by using eqn(5) and update it in  classified
                      database
}
} While (session over)
```

Fig 4.3: Algorithm for *PPC* module

To better understand algorithm of *PPC* module, let us consider, a user clicked three pages P_1, P_2, P_5 in a list of ten pages (P_1, P_2……P_{10}) as shown in Fig 4.4.

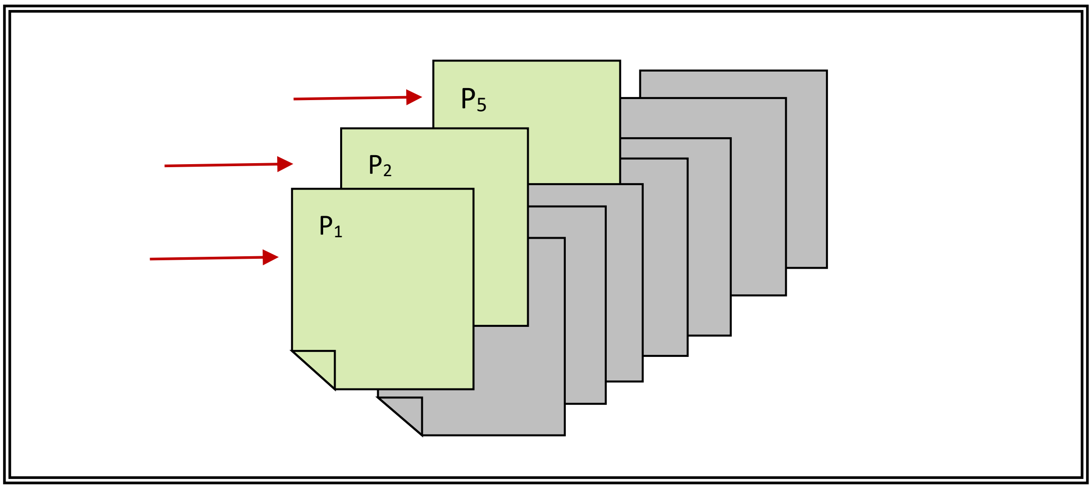


Fig 4.4: A Sample Example of User Click

Now, Click weight of each clicked page can be computed by eq^n (4.2) as shown below. The *CLICK$_{wt}$* of all other pages is set to zero as they did not get any hit from user.

$$Click_{wt}(P_1) = \frac{1}{1+1+1} = 0.33 \qquad (4.2.1)$$

$$Click_{wt}(P_2) = \frac{1}{1+1+1} = 0.33 \qquad (4.2.2)$$

$$Click_{wt}(P_5) = \frac{1}{1+1+1} = 0.33 \qquad (4.2.3)$$

Initially PPF of all pages is set to zero. It is computed and updated every time; a user selects the page in the result list. The updated click weight of all the selected pages in the example scenario is summarized in table 4.2.

Table 4.2: Updated Click Weight for Sampled Scenario

Page → Attribute↓	**P_i**	**P_2**	**P_3**	**P_4**	**P_5**	**P_6**	**P_7**	**P_8**	**P_9**	**P_{i0}**
Click$_{wt}$	0.33	0.33	0	0	0.33	0	0	0	0	0

Further, the user spent 3 minutes, 2 minutes and 9 minutes on page P_1, $P_2$and P_5 respectively in the above sampled scenario depicted in Fig 4.4. Thus in the set of 10 pages , only the time weight of pages P_1, P_2, P_5 get updated using eq^n (4.3).

$$Time_{wt}(P1) = \frac{3}{9} = 0.33 \qquad (4.3.1)$$

$$Time_{wt}(P2) = \frac{2}{9} = 0.22 \qquad (4.3.2)$$

$$Time_{wt}(P5) = \frac{9}{9} = 1 \qquad (4.3.3.)$$

The updation in time weight of all clicked pages is summarized below in table 4.3.

Table 4.3: Updated Click Weight and Time Weight for Sampled Scenario

Page → Attribute↓	**P_i**	**P_2**	**P_3**	**P_4**	**P_5**	**P_6**	**P_7**	**P_8**	**P_9**	**P_{i0}**
$Click_{wt}$	0.33	0.33	0	0	0.33	0	0	0	0	0
$Time_{wt}$	0.33	0.22	0	0	1	0	0	0	0	0

Next the user takes no action on page P_1, and P_2 but perform the save action on page P_5. So, *$ACTION_{wt}(P_5)$*=0.3; *$ACTION_{wt}(P_1)=0$* and *$ACTION_{wt}(P_2)=0$.* The updated PPF score for all the clicked pages P_1, P_2 and P_5 are listed in table 4.4.

Table 4.4: Updated PPF Score of Clicked Pages for Sampled Scenario

Selected pages	***$Click\ _{wt}$***	***$Time_{wt}$***	***$Action_{wt}$***	***PPF score***
P1	0.33	0.33	0	0.22
P2	0.33	0.22	0	0.16
P5	0.33	1	0.3	0.79

This PPF information related to each clicked page is updated in search engine database that is further utilized by page ranker module to evaluate the rank score of a page based on web usage mining.

4.3.1.4 Deciding the value of constants α, β and γ

Some practical work had been carried out to find the suitable values of α, β and γ**.** By taking the value of α from 0.1 to 0.32, value of β from 0.33 to 0.65, value of γ from 0.66 to 1. The PPF score of all the clicked pages can be computed by using the eq^{n} (4.1). For the sake of simplicity, value of α, β and γ is taken at regular interval as shown in table 4.5.

Table 4.5: Computation for Values Of α, β and γ

Test cases	α	β	γ	$PPF(P_1)$	$PPF(P_2)$	$PPF(P_5)$
Set 1	0.32	0.64	0.97	0.3168	1.2164	1.0366
Set 2	0.27	0.59	0.92	0.2838	1.1389	0.9551
Set 3	0.22	0.54	0.87	0.2508	1.0614	0.8736
Set 4	0.17	0.49	0.82	0.2178	0.9839	0.7921
Set 5	0.12	0.44	0.77	0.1848	0.9064	0.7106
Set 6	0.07	0.39	0.72	0.1518	0.8289	0.6291
Set 7	0.02	0.34	0.67	0.1188	0.7514	0.5476

The PPF score of all the clicked pages corresponding to the various values of α, β and γ has been plotted in Fig 4.5.

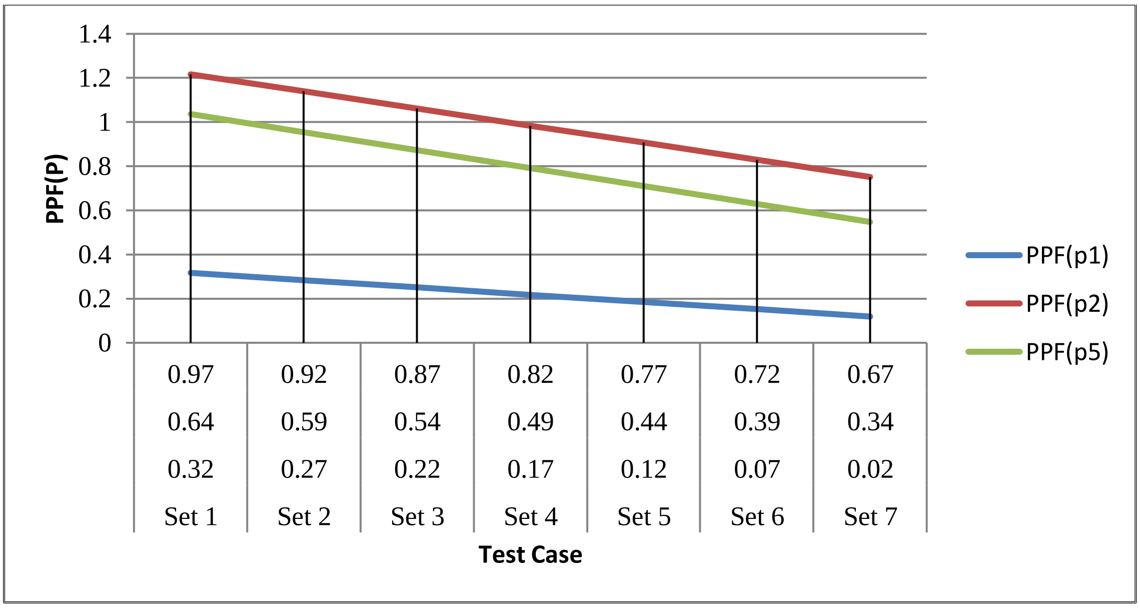


Fig 4.5: Computation of Value of α, β and γ

To find the optimized value of α, β *and* γ the mean of centroid of area covered by every clicked page is computed. For instance, computation of the centroid of the area covered by page p_2 is given on next page:

The area covered by page P2 is divided into two sub areas: rectangle and triangle as depicted in Fig 4.6.

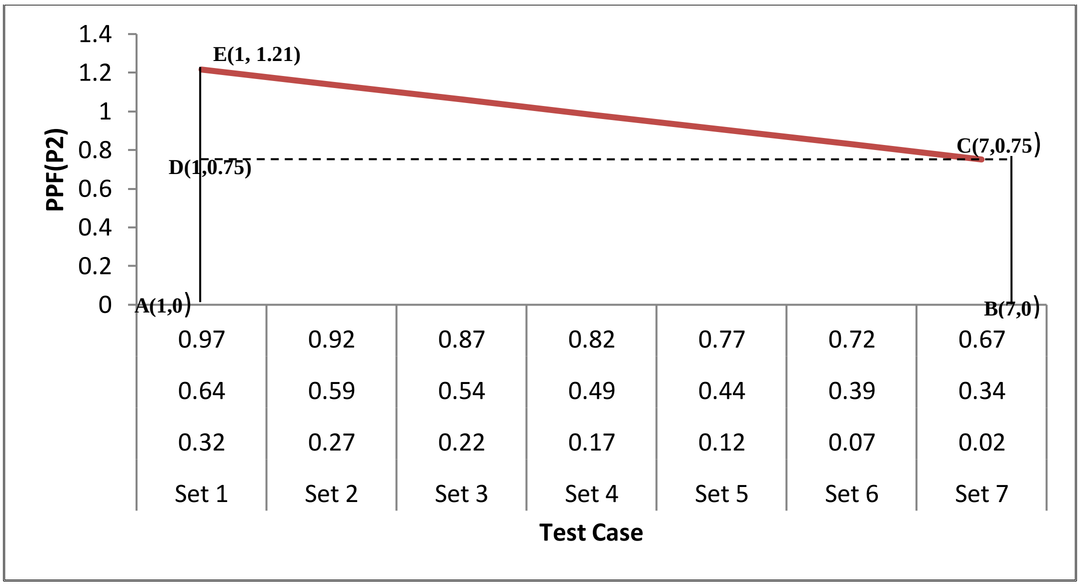


Fig 4.6: Computation of Centriod of Area Covered by Page P$_2$

For △ EDC:

$Area = \frac{1}{2} \times 6 \times 0.46 =$ 1.38 Sq. unit

$$Center = \left(\frac{1+1+7}{3}, \frac{1.21+0.75+0.75}{3}\right) = (3{,}0.9)$$

For□ ABCD:

$Area = 6 \times 0.75 = 4.5$ Sq. unit

$$Center = \left(\frac{1+7}{2}, \frac{0+0.75}{2}\right) = (4, 0.37)$$

To find centroid of whole area ABCE:

In X-direction :

$$1.38(3) + 4.5(4) = (1.38 + 4.5)x_{centroid}$$

$$x_{centroid} = 3.7$$

In Y-direction:

$$1.38(0.9) + 4.5(0.37) = (1.38 + 4.5)y_{centroid}$$

$$y_{centroid} = 0.49$$

So, $\boldsymbol{Page\ P_1\ (x_{centroid}, y_{centroid}) = (3.7, 0.49)}$

Likewise the centroid of area covered by page P1 and P5 are also calculated.

$\boldsymbol{Page\ P_1\ (x_{centroid}, y_{centroid}) = (3.52, 0.11)}$

$\boldsymbol{Page\ P_5(x_{centroid}, y_{centroid}) = (3.6, 0.4)}$

Optimum test case $(\alpha, \beta\ and\ \gamma)$ = mean of $x_{centroid}$ of all clicked pages =(3.7+3.52+3.6)/3=3.6 $\simeq$ 4 $\Longrightarrow$ 4th test case is optimum value of $\alpha, \beta\ and\ \gamma$. So $\alpha = 0.17$, $\beta = 0.49$ and $\gamma = 0.82$.

4.3.2 Page Ranker

This component is responsible for computing the overall page rank of a page based of web content mining, web structure mining and web usage mining. It receives the query terms and corresponding matched documents from query processor and prepares the ranked list. To accomplish the entire task from document characterization to rank computation, it has to perform the number of subtasks. They are listed as below:

1. *Content similarity* is computed by finding frequency and position of query terms with in the document.
2. *Link weight* computation is done based on forward as well as backward links of document with in the web graph.
3. The *page probability factor* associated with each page is retrieved from search engine database.
4. Final *rank computation* based on content, link and page probability factor is carried out.

5. The *sorted list* of documents based on their rank score is prepared and returned to query processor module of search engine.

Detail discussion is presented in the next subsection.

4.3.2.1. Content Similarity Computation

Content similarity of the document with the query means what query terms are present in the document, at which place and how many times they are present. There exist several possible measures such as coordination level match [116], jaccard's coefficient [137], Dice's coefficient [139], overlap coefficient [140] etc. that compute the similarity between the query and document. But the proposed approach is made to use a measure which considers the frequency of term in the document as well as in the query. Moreover the method assigns significant importance to position of term within the document. Thus the document containing the multiple occurrences of the query terms at different positions is likely to be more relevant than the document containing the single occurrence of term in the document. So in order to evaluate the numeric score of similarity between the document and query, the formulae 4.4 has been designed.

$$\mathbf{Sim_{content}(q, doc)} = \boldsymbol{W_{pos}} \times \boldsymbol{W_{freq}}\mathbf{(q, doc)} \qquad \mathbf{(4.4)}$$

Where:

- W_{pos} denotes the weight contributed due to the position of query term in the document *doc.*
- $W_{freq}(q,doc)$ denotes the weight of document based on frequency of query terms with in the document as well as in query.

Calculation of $W_{freq}(q,doc)$: In order to compute document weight based on frequency of query terms, let us consider the vector space model to represent the document and query.

$D_i=d_{i1},d_{i2}\ldots\ldots\ldots d_{in}$ represents a document vector in n-dimensional vector space.
$Q_j=q_{j1},q_{j2}\ldots\ldots\ldots q_{jn}$ represents a query vector in n-dimensional vector space.

Where *n* denotes different terms present in document and query. Fig 4.7 represents a document‘d’ in 3-dimensional term vector space.

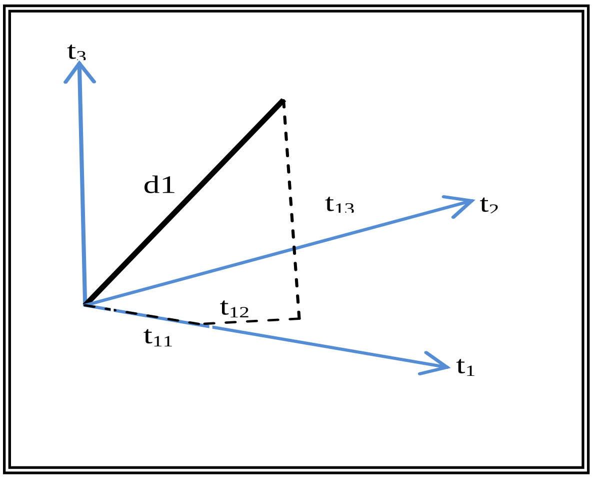


Fig 4.7: A Document Represented in a Three Dimensional Term Vector Space

Further, length of document vector/query vector can be computed by extension of Pythagoras’s theorem given in eqn (4.5).

$$|d_i|^2 = {d_{i1}}^2 + {d_{i2}}^2 + {d_{i3}}^2 \ldots\ldots {d_{in}}^2$$

$$\boxed{|\boldsymbol{d_i}| = \left(\boldsymbol{d_{i1}}^2 + \boldsymbol{d_{i2}}^2 + \boldsymbol{d_{i3}}^2 \ldots\ldots \boldsymbol{d_{in}}^2\right)^{\frac{1}{2}} \qquad (\mathbf{4.5})}$$

Similarly,

$$\boxed{|\boldsymbol{q_j}| = \left(\boldsymbol{q_{j1}}^2 + \boldsymbol{q_{j2}}^2 + \boldsymbol{q_{j3}}^2 \ldots. + \boldsymbol{q_{jn}}^2\right)^{\frac{1}{2}} \qquad (\mathbf{4.6})}$$

The magnitude of term $di_k \in di$ in dimension *k* can be represented by W_{ik} is given by eq^n (4.7).

$$\boxed{\boldsymbol{W_{ik}} = \begin{cases} \geq \mathbf{0} \;;\; \boldsymbol{if\ the\ query\ term\ present\ in\ the\ document} \\ \mathbf{0} \qquad\qquad \boldsymbol{otherwise} \end{cases} \qquad (\mathbf{4.7})}$$

Further, if di and qj are two vectors, then their inner product (dot product) is given by eqn (4.8).

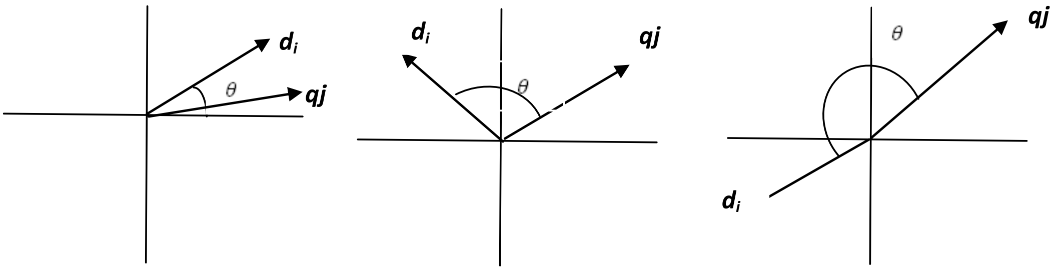
$$\boldsymbol{d_i . q_j = w_{i1} w_{j1} + w_{i2} w_{j2} + \cdots + w_{in} w_{jn}} \qquad (\mathbf{4, 8})$$

Now the similarity between the document and query can be defined as cosine of angle between their vectors in term vector space. Fig 4.8 depicts the possible scenario when all the query terms are present in the document to having none of query term in the document. The formula for computing the cosine similarity between document and query is given in eq^{n} (4.9) as follows:

$$\boldsymbol{W_{freq}(d_i, q_j) = \frac{d_i . q_j}{|d_i \; | \; |q_j \; |}} \qquad (\mathbf{4.9})$$

Where:

- W_{freq} (d$_i$,q$_j$) denotes the content similarity between document d_i and query q_j.
- $|d_i|$ and $|q_j|$ represents the length of vectors d_i and q_j respectively.



Both the vectors lies towards same direction; angle between is nearly equal to zero degree. Similarity is nearly equal to 1.

Vectors are nearly orthogonal to each other. Similarity is nearly equals to zero.

Vectors are nearly opposite to each other. Cosine of angle between them is -1. They are surly dissimilar.

Fig 4.8 Variation in Document Query Similarity

To illustrate the concept of cosine similarity more clearly, let us consider three documents d1, d2, d3 and query q1 as given in table 4.6.

Table 4.6 Simple Example for Illustration of Cosine Similarity

Query	Term	
Q	rat cat	
Document	Text	Term
d1	rat rat dog	rat, dog
d2	cat dog cat bat cat rat cat	cat, dog, bat, rat
d3	bee beer cat deer wolf	bee, beer ,cat, deer, wolf

Using eqn (4.5), the length of each document can be computed as follows:

$$|d_1| = (2^2 + 1^2)^{\frac{1}{2}} = \sqrt{5}$$

$$|d_2| = (1^2 + 1^2 + 4^2 + 1^2)^{\frac{1}{2}} = \sqrt{19}$$

$$|d_3| = (1^2 + 1^2 + 1^2 + 1^2 + 1^2)^{\frac{1}{2}} = \sqrt{5}$$

The length of query ‘q’ can be computed by eq$_n$ (4.6) as follows:

$$|q| = (1^2 + 1^2)^{\frac{1}{2}} = \sqrt{2}$$

Table 4.7 summarizes the weight of documents and query in 8-dimensional vector space.

Table 4.7: Computation of Weight Based on Term-Frequency

	rat	**dog**	**cat**	**bat**	**bee**	**beer**	**deer**	**wolf**	***Length***
q	1	0	1	0	0	0	0	0	*√2*
d1	2	1	0	0	0	0	0	0	*√5*
d2	1	1	4	1	0	0	0	0	*√19*
d3	0	0	1	0	1	1	1	1	*√5*

The similarity score of query ‘q’ with each document can be computed by eqn (4.9).

$$W_{freq}(q,d_1) = \frac{2+0+0+0+0+0+0+0}{\sqrt{2}x\sqrt{5}} \;=\; 0.63 \qquad 4.10(a)$$

$$W_{freq}(q,d_2) = \frac{1+0+4+0+0+0+0+0}{\sqrt{2}x\sqrt{19}} \;=\; 0.81 \qquad 4.10(b)$$

$$W_{freq}(q,d_3) = \frac{0+0+1+0+0+0+0+0}{\sqrt{2}x\sqrt{5}} \;=\; 0.32 \qquad 4.10(c)$$

Thus document d$_2$ is most similar and d$_3$ is least similar to query q as shown in table 4.8.

Table 4.8: Similarity of Query Q with Documents

	d1	d2	d3
Q	2/√10 = 0.63	5/√38 = 0.81	1/√10 =0.32

Let us also consider some real life example in which user submitted the query *Q= rectilinear propagation of light.* Number of documents is retrieved from the search database. For simplicity, the computation of similarity of three of such documents with user query is as follows:

Table 4.9 lists the URLs under consideration. For instance, number of different terms present in document D1 is 38, in document D2 is 158 and in document D3 is 80.

Table 4.9: Sample URLs for Frequency Weight Computation

Query		Term	
Q		Rectilinear propagation of light	
URL#	**Document#**	**Candidate URL**	Term
URL1	D1	http://www.tutorvista.com/content/physics/physics-ii/light-reflection/light-propagation.php	37
URL2	D2	http://study.com/academy/lesson/rectilinear-propagation-of-light-definition-examples.html	158
URL3	D3	https://en.wikipedia.org/wiki/Rectilinear_propagation	80

The document listed in table 4.9 is parsed using Porter's algorithm [74] and the obtained stems from three documents with their frequency is given in Appendix D. Using eq^{n} (4.5), the length of each document related to URL is computed as follows:

$$\mid \boldsymbol{D_1} \mid = \sqrt{\mathbf{141}}$$

$$\mid \boldsymbol{D_2} \mid == \sqrt{\mathbf{888}}$$

$$\mid \boldsymbol{D_3} \mid = \sqrt{\mathbf{154}}$$

The length of query 'q' can be computed by eq$_{n}$ (4.6) as follows:

$$\mid \boldsymbol{q} \mid = \left(\mathbf{1^2 + 1^2 + 1^2}\right)^{\frac{1}{2}} = \sqrt{\mathbf{3}}$$

The similarity score of query 'q' with each document can be computed by eqn (4.9).

$$\boldsymbol{W_{freq}}\left(\boldsymbol{Q}, \boldsymbol{D_{1_1}}\right) = \frac{\mathbf{4+2+2}}{\sqrt{\mathbf{3}}\sqrt{\mathbf{141}}} = \mathbf{0.389} \qquad \mathbf{4.11}(\boldsymbol{a})$$

$$\boldsymbol{W_{freq}}(\boldsymbol{Q}, \boldsymbol{D_2}) = \frac{\mathbf{19+6+5}}{\sqrt{\mathbf{3}}\sqrt{\mathbf{888}}} = \mathbf{0.581} \qquad \mathbf{4.11}(\boldsymbol{b})$$

$$\boldsymbol{W_{freq}}(\boldsymbol{Q}, \boldsymbol{D}\mathbf{3}) = \frac{\mathbf{4+1+1}}{\sqrt{\mathbf{3}}\sqrt{\mathbf{154}}} = \mathbf{0.27} \qquad \mathbf{4.11}(\boldsymbol{c})$$

Thus document D_2 is most similar and D_3 is least similar to query q.

Calculation of W$_{pos}$: The position of query term plays an important role while computing the weight of web document as the document containing query term in title tag is likely to be more important than the document having query term in body text. The weight corresponding to different positions are listed in table 4.10.

Table 4.10: Keyword Position Weight

Keyword position	Weight
<Title>	1
<H1><H2><H3>	0.75
<B><I><U>	0.5
<Body> ,<metatag>	0.25

Rules for assigning the position weight are as follow:

Rule1: If the query contains a single term and it is occurring at different positions in the web document, then the higher position weight is considered among all occurrences.

Rule2: If the query contains more than one functional term than the sum of highest position weight of all the terms are assigned to W_{pos}.

The algorithm for Page Ranker module for computation of overall rank score of a page is given in Fig 4.9.

```
Page Ranker ()
Input: User query, URL of matched documents
Output: sorted list of URLs
Method:
Do for each page in buffer {
    1. Compute content similarity using eqn (8);
    2. Compute Link Similarity using page rank algorithm();
    3. Fetch PPF of page from search engine database;
    4. Add three weights linearly to get overall rank of page;}
Sort the pages as per overall rank score;
Return the sorted list to query processor;
```

Fig 4.9: Algorithm for Page Ranker Module

Next section provide the comparison summary of proposed page ranking techniques with some of the prevalent page ranking technique discussed previously in chapter 2 under section 2,7.

4.4 PERFORMANCE EVALUATION OF PROPOSED SYSTEM

The proposed ranking technique is implemented in C# .net with MS SQL Server 2012 at the back end and test run was carried out on sampled seed URL set given in appendix A.

4.4.1 Procedure

For evaluating the performance of proposed system, a set of five hundred pages were archived in five selected domains and handed over to the users. The first user enters queries as per its interest in some/all selected domains and its browsing behaviour is captured in terms of relative clicks, time spent and action performed on the page. The feedback about user search experience for each user was also collected in terms of "satisfaction" or "dissatisfaction".

Then same process is followed for second user and so on. On the responses the performance metric *Result Relevancy* was applied for each user and its average was taken.

4.4.2 Performance Metric

Definition 4.2: *Result Relevancy* (*Rel*) is defined as a fraction of web pages found satisfactory by the user over all the pages returned by the proposed technique in response to its query. Mathematically, the *Result relevancy* is given by:

$$Rel = S/(S+Ns) \qquad \textbf{(4.11)}$$

Where:

- ***S*** is numbers of pages found satisfactory by the user and
- ***NS*** is number of pages not found to be satisfactory.

Based on the feedback of the first user, it was found that out of 10 web pages, user is satisfied with 7 pages So, using the terms defined, ***S***=7, ***NS***=3.

Rel=7/ (7+3)=70%.

Similarly the feedback of other users is also collected and summarized in table 4.11.

Table 4.11: Average Result Relevancy

User	Result relevancy (%)
1	70
2	75
3	80
4	88
5	84
Average	**79.4**

The average result relevancy of proposed PR-PPF is 79.4%.The net performance of PR-PPF in terms of personalized response is remarkable as compared to existing systems.

4.5 SUMMARY

In this chapter, three different aspects are applied to evaluate the relevancy of page i.e. web usage mining to compute PPF, web content mining to compute content similarity and web structure mining to compute link weight for a page. The special feature of PPF based ranking is action centric, content similarity based ranking is query centric and link based ranking is web page centric. Combination of all these has proven to be better in terms of quality of search results.

The comparison summary of proposed ranking technique with some of the prevalent ranking algorithm is presented in table 4.12.

Table 4.12: Comparison of proposed technique with prevalent page ranking techniques

Algorithm → **Parameters ↓**	**PageRank**	**Weighted Page Rank**	**Page Rank Based on Link Visit**	**PR-PPF**
Technique applied	Web structure mining	Web structure mining	Web structure as well as web	Web structure, web usage as

			usage mining	well as web content mining.
Input parameters	Back and forward links	Bank and forward links	Back links, forward links and number of clicks	Back links, forward links, query terms, clicks, time spent and action taken
Place of rank computation	Indexing	Indexing	Indexing and query submission	Indexing and query submission
Complexity	O(Log n)	< O(Log n)	>O(Log n)	>O(Log n)
Quality of search results	Low	Higher than PR	Higher than WPR	Higher than PRLV
Search engine	Own (Goggle)	Research model	Research model	Research model
Benefits	Simple to implement	Fast as run for single level in web graph	Relevancy of page based on user action	Proportionate clicking, time spent and action captured on page always avoids biased increase in rank
Limitations	Equally distributes ranks among outgoing links No consideration for user feedback	Pay equal importance to in links and outlines	Extra efforts for client and server side agents May results in biased relevancy as number of clicks directly leads to increase in rank	More efforts and complex.

The implementation snapshots of proposed technique are presented in Appendix D. Next chapter explains the proposed crawling mechanism for volatile data in detail.

CHAPTER V

AN ARITHMETIC PROGRESSION BASED CRAWLING MECHANISM FOR MAINTAINING QUALITY DATA

5.1 INTRODUCTION

The amount of information on web is exponentially growing with the emergence of 4G technology and day to day advancement in diverse smart devices such as i-phone, i-pad, tablet etc. It becomes highly important for a search engine to design optimal data collection policies that can lead to efficient extraction of information from such a huge collection of web contents. The characteristics of web contents are different from the conventional content stored in online/offline databases. The main differences are listed as below:

i. Web contents are composed of complex, non-hierarchical HTML structures that can be accessed by hundred millions of users.

ii. They can be created by anyone having internet access and often possess dynamic creation and updation cycle.

iii. Their physical boundaries are not defined.

iv. Much of same contents are repackaged on diverse web sites.

Keeping in mind the aforementioned characteristics, a self- adjusting arithmetic progression based crawling technique based on page changing frequency for enriching the search engine repository with fresh updated data has been proposed and developed. To avoid repeated downloading of same document, the necessary parameters are identified and embedded in the data structure of URL frontier. The detail discussion on the techniques is given in the next section.

5.2 PROPOSED CRAWLING FRAMEWORK BASED ON PAGE UPDATION FREQUENCY

In order to accomplish the need of self-adjusting crawler for maintaining quality data , two new components named visit scheduler and URL prioritizer are added to existing architecture of web crawler The working and structure of existing components are also modified in order to sink them with the newly added components. The proposed framework of arithmetic progression based crawler is depicted in Fig 5.1.

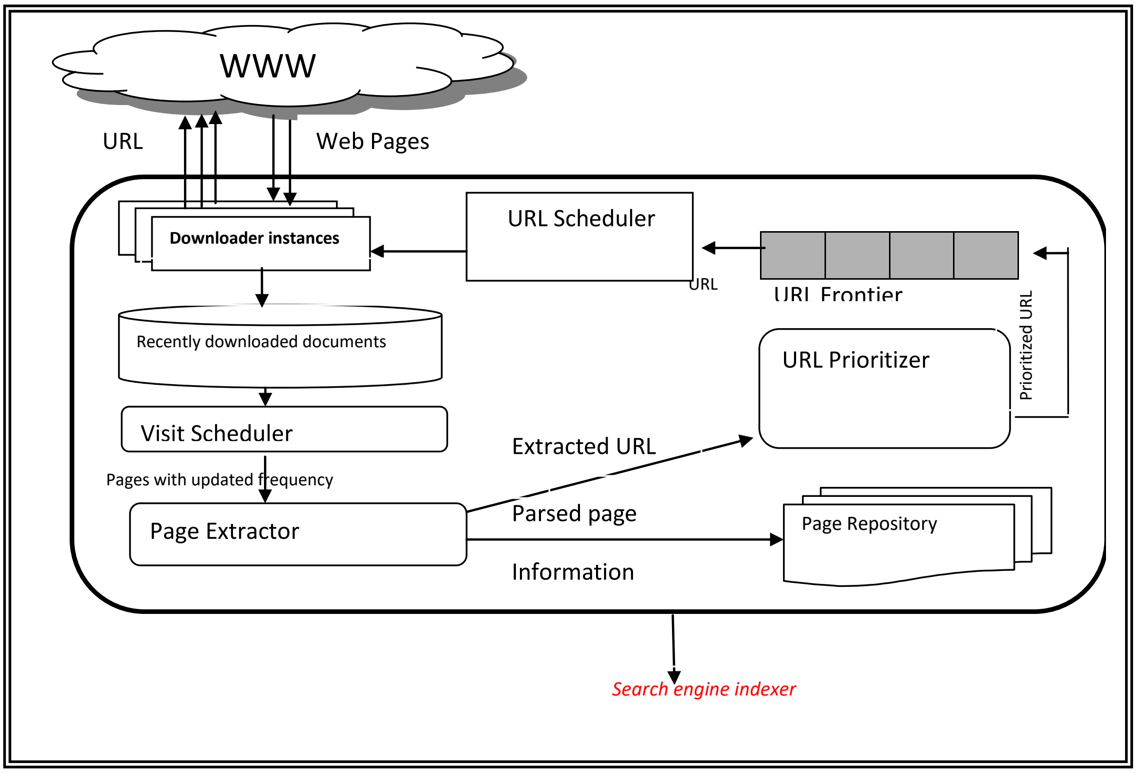


Fig 5.1: Component Diagram for Web Page Collection and Analysis

The process of web data collection is started by gathering the set of seed URLs in URL Frontier in ascending order of their visit interval computed by Visit Scheduler module. These URLs are then assigned to multi -threaded document downloader instances by URL scheduler module. The downloaded documents are examined to check any updation and accordingly their visit interval is rest. After updating their visit interval, documents are then parsed to extract embedded links and terms. The terms are stored in Page repository and extracted links are further processed by URL Prioritizer module to remove any duplicity. The priority bit is also assigned to each URL based of their out ink information by URL priritizer. At last they are inserted in URL frontier for recursive crawling as per their visit interval and link information. The detail working of each component is given in following subsections.

5.2.1 URL Frontier

URL Frontier (queue) is initially populated with set of seed URLs from five different domains (viz. education, travelling & tourism, sports, food & beverages and fashion & shopping; these domains are further extensible) using classification hierarchy offered by DMOZ (Presently owned by ResourceZone) [74]. Since such a hierarchy contain collection of selected web sites. Expert of each domain is asked to select top N relevant URLs from the classified hierarchy. These URLs serve as the entering point for data collection process. The structure of proposed URL Frontier is depicted in Fig 5.2.

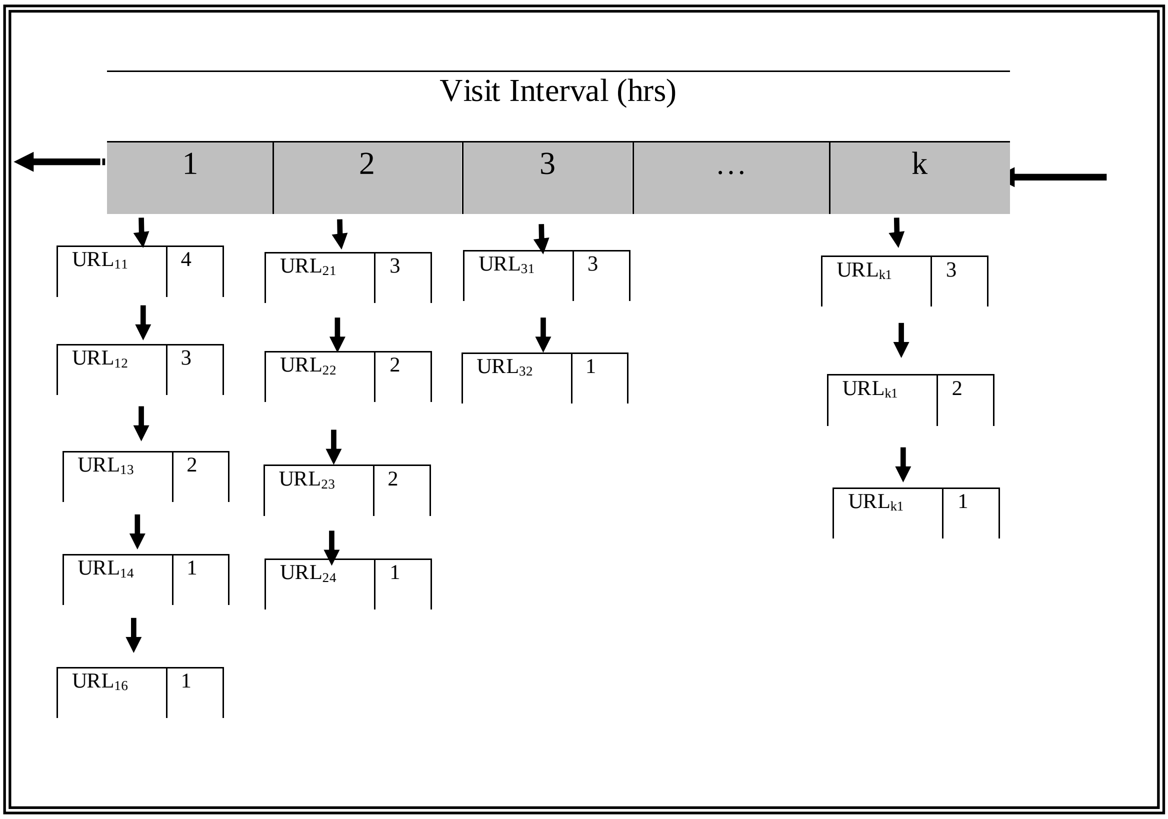


Fig 5.2: Structure of URL Frontier

The URLs are organized in ascending order of their visit interval (*VI*) The visit interval has been indicated by the positional marker 1 to k in the queue; where each position is further linked to a list of URLs having the same *VI*. It may be noted that a numeric value is associated with each URL. This is termed as Priority bit. It indicates the importance of URL among the members of the same list. Initially, the priority bit for all URLs is set to 1, which indicates that URL is seen for the first time.

As initially, the URLs have been taken from DMOZ, so visit interval for all the URLs in URL frontier is set to a default interval r by visit scheduler (in the experiment, the default visit interval is taken as 12 hours) and thus URLs are inserted in the URL frontier in accordance with their occurrence in the classification hierarchy. To support this work, some new fields are incorporated in the existing data structure of URL record as shown in Fig 5.3.

URL RECORD

URL	Visit_ interval	Priority_bit	FP_key	θ	θ_l	θ_g	Server name

Fig 5.3: Modified Structure of URL Record

The description regarding the various fields of modified URL record is given in Table 5.1.

Table 5.1: URL Record Description

Field	Description
URL	URL of the page yet to be downloaded by crawl instances. e.g. http://www.snapdeal.com/ domain name system.
Visit_ interval	It is the interval to visit a particular URL. After initializing it to a default value, the value is dynamically updated by visit scheduler.
Priority_ bit	The numeric value assigned to each URL based upon no. of documents from which the URL is extracted. Initially, it is set to 1 by page extractor and dynamically updated by URL prioritizer module.
FP_ key	This field stores the fingerprint key value of the crawled page used for detecting the change in the page.
θ, θ_l, θ_g	θ represents the probability of change in document assigned by visit scheduler. θ_l and θ_g are lower and upper bound values for θ.
Server name	The web server name of the URL. For example, in the above mentioned URL, the server name is snapdeal.com

The following section discusses the working of visit scheduler module in detail.

5.2.2 Visit Scheduler

The proposed module is responsible for computing the next visit time based on current visit interval for each downloaded document in order to maintain fresh collection of documents in search engine database. The mechanism to evaluate the next visit interval of a page has to deal with the following issues:

- ***How to detect the changes between old and new version of document to verify its updation?***

The Finger print scheme proposed by Sharma et.al [73] is used to check the similarity between two documents.

- ***What are the other factors that must be considered to compute revisit time of a web page?***

At once, it sounds appealing that a web page must be visited as frequent as the updations are occurring in it. But revisiting the web page directly at the rate of updation frequency, not only increase the network traffic but also the crawler will not be able to download the requested document on time. Owing to the above circumstances, it becomes essential for the search engine to design the optimized revisit policy for the crawler. To solve these issues, the visit scheduler has to perform following tasks:

- Deduce the probability of change of a document based upon the rate at which the visit to the document encounters an updation.
- Decide the lower and upper bound of change probability so that the optimization between network congestion and page freshness can be maintained.
- Compute the amount by which the current visit time may be increased /decreased based upon boundary condition of change probability.

Definition 5.1: The probability of change in document 'θ' can be defined as the ratio of no. of visits that encountered change in document to the total no. of visits to the document.

E.g. If a change in document is found 8 times out of 10 visits, then its probability of change is 80% i.e. 0.8.

So, in the worst scenario, either the document always contains a change on every visit or exhibits no change at all. Thus, the value of θ always lies between 0 and 1. Based upon the characteristics of web site, the visit scheduler may assign the lower and upper bound of θ.

Further, the relationship between the i^{th} visit time, VI_i and $i+1^{th}$ visit time, VI_{i+1} can be defined by the eq^n (5.1).

$$\boxed{VI_{i+1} = VI_i + \partial t \qquad (5.1)}$$

Where: ∂t denotes the change in visit time. The value of ∂t may be positive, negative or zero as there exist three different cases depending upon the relationship of current change probability with respect to its boundary conditions.

Let change probability at i^{th} visit is represented by θ_c and expected lower and upper bound of change probability is represented by θ_l and θ_g respectively.

Case 1: When the page is changing less than the expected change probability i.e. $\theta_c < \theta_l$, then the visit interval to the page must be increased by the amount ∂t as given in eq^n 5.2.

$$\boxed{\partial t = (1 - \theta_{c/}\, \theta_i)\, VI_i \qquad (5.2)}$$

Case 2: When the page is changing at a rate greater than the expected change probability, i.e. $\theta_c > \theta_g$, then the visit interval to the page must be decreased by the amount ∂t as given in eq^n 5.3.

$$\boxed{\partial t = (1 - \theta_{c/}\, \theta_g)\, VI_i \qquad (5.3)}$$

Case 3: When the page is changing at a rate as expected i.e. $\theta_l \leq \theta_c \leq \theta_g$, then it is not required to change the visit time as given in eqn (5,4).

$$\partial t = 0 \qquad (5.4)$$

Thus, the amount by which the visit interval VI_i must be modified is summarized below in Fig 5.4.

$$\partial t = \begin{cases} \left(1 - \frac{\theta_c}{\theta_g}\right) VI_i & if \theta_c > \theta_g \\ \left(1 - \frac{\theta_c}{\theta_l}\right) VT_i & if\ \theta_c < \theta_l \\ 0 & if \theta_l \leq \theta_c \leq \theta_g \end{cases}$$

Fig 5.4: Equation to Evaluate Next Visit Interval

Initially, the values of *VI*, θ_l, θ_c *and* θ_g are set to threshold values for all the URLs in the seed set θ_c as 0.5, θ_l as 0.2 and θ_g as 0.8. The algorithm of visit scheduler is given in Fig 5.5.

Visit scheduler ()

Input: Downloaded document D, current probability of change in document θ_i, lower threshold θ_l; upper threshold θ_g , Page address;

Output: Updated visit time of URL$_{D.}$

Method:

```
Do Forever
Step 1: wait (Document D);
Step2. Detect the change in D using finger printing scheme;
        3. Calculate time of revisit using eqn (1);
        4. Update URL record;
```

Fig 5.5: Algorithm for visit Scheduler

The algorithm checks each downloaded page for any updation and compute next visit interval according to eq^n(5.1). After updating the visit interval of page, the page is passed to page extractor module for further processing.

5.2.3 Page Extractor

The web page sent by the visit scheduler along with updated URL record is parsed by Page Extractor module to get the text and links contained in the page wherein each link may point to another link. The extracted text is in turn tokenized to get a list of normalized keywords termed as *indexing terms*. The tokenization involves removing of stop words as they contribute nothing in retrieval of relevant information and applying stemming algorithm to convert inflectional form of word into a common base form for example *wishes, wished, wishing→ wish(v)*. Further, the frequency and position of each stem in the document is determined and stored in page repository. The extracted URLs are passed to URL prioritizer as well as stored in Page repository to be used by search engine indexer. The initial value of visit interval for all the extracted URLs is set to threshold value and priority bit is set to 1. The algorithm of page extractor module is given in Fig 5.6.

```
Page Extractor ()
Input: Web Document D with updated visit time VT.
Output: Indexing terms of documents D with link information.
Method:
Step 1:  wait (document D);
        2. Parse the document D to extract token and hyperlinks. Store the hyperlinks in
        Next_URL[i];
              2.5 Set i=0;
              2.6 set Next_URL[i] ←$URL_D$;
              2.7 Set n ← no. of out links of $URL_D$;
              2.8 For i=1 to n do
                  Next_URL[i] ←outlink($URL_D$);
      3. Find the frequency and position of each token in the document
       4. Store the tokens and next URLs information in page repository.
       5.  Set Priority_bit of all URLs in Next_URL[n] to 1
        6. Send the next _URL list to URL prioritizer
```

Fig 5.6: Algorithm for Page Extractor Module

Once the embedded links are extracted from the downloaded documents, they are further processed to check any duplicacy by URL prioritizer module and inserted in

URL Frontier. The detail working of URL prioritizer is discussed in following section.

5.2.4. URL Prioritizer

The proposed URL prioritizer module is responsible for determining the importance of each URL among the group of URLs having the same visit time. It is based upon the fact that existence of a URL in the out link of other pages truly reflects its importance within the web graph. Thus, after receiving the Next_URL list from page extractor module, it assigns a priority bit to each URL. Initially the priority bit for all URLs is set to 1 which indicates that URLs are appearing for the first time as shown above in Fig 5.2. Afterwards, the priority bit of URL is incremented by one with every new existence of URL as an out link. To support this, it searches the URL contained in Next_URL list into URL frontier. If a match is found then the associated priority bit is updated and existing URL is repositioned in accordance to its updated priority as shown in Fig 5.7.

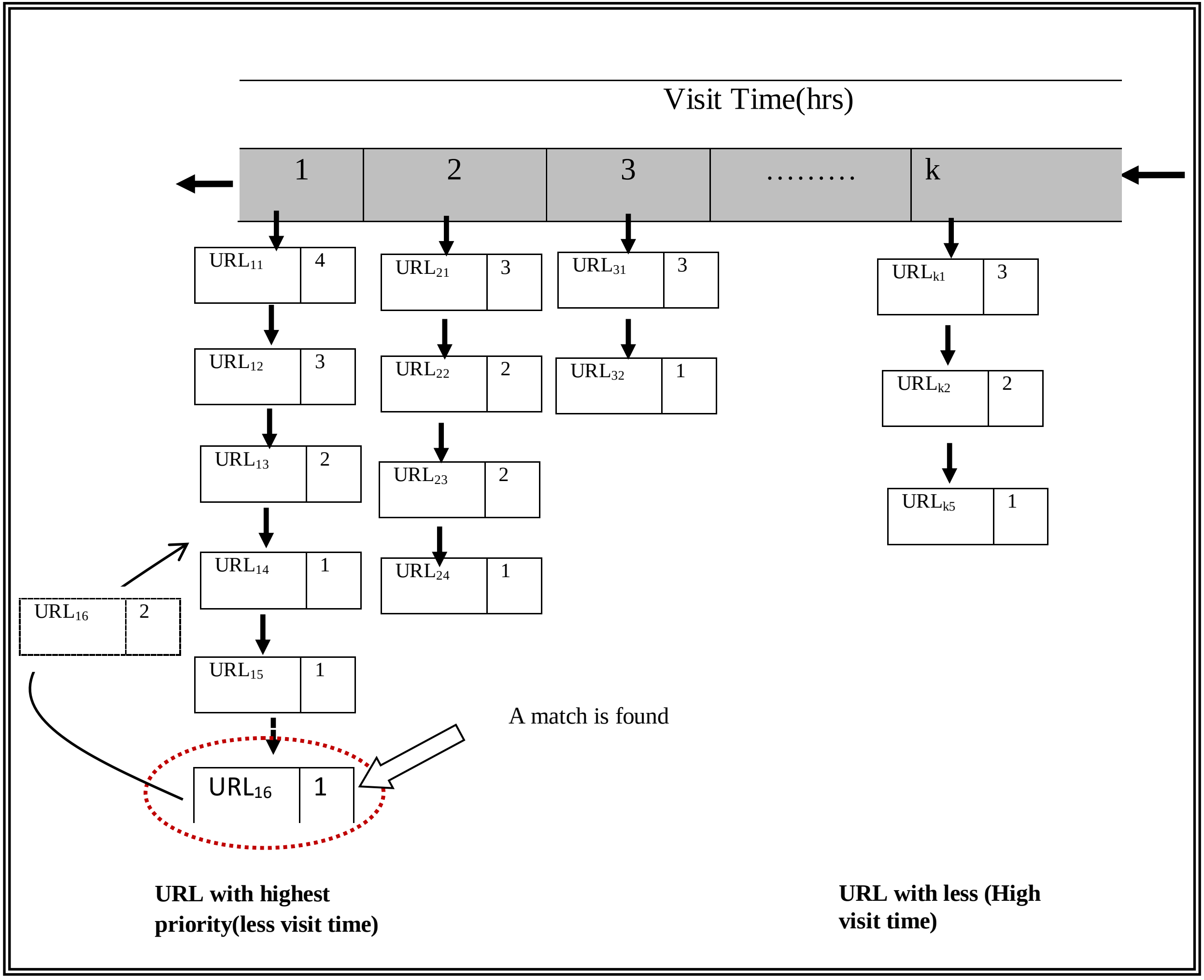


Fig 5.7: Repositioning the URL based on its priority bit

Otherwise the URL is inserted at the end of linked list marked by the same visit interval as that of URL.

For the sake of understanding its working in a better way, let a URL named as URL$_{156}$ with *VI* equal to 1 hr. is the next URL to be inserted in URL frontier. URL prioritizer module found a match at the position 6th in the list attached to location *VI* =1.

So, it first changes its associated priority bit and inserts the URL at new position i.e. position. 4th. The algorithm for URL prioritizer module is outlined in Fig 5.8.

URL _Prioritizer ()

Input: Next_ URL list with each URL associated with following fields:

1. URL of document to be inserted in URL Frontier
2. Visit time of URL in hrs.
3. Priority_Bit of URL .
4. FP_Key , Fingerprint key.
5. Server name
6. **Output:** Insert URL in URL Frontier at the proper position.

Method:

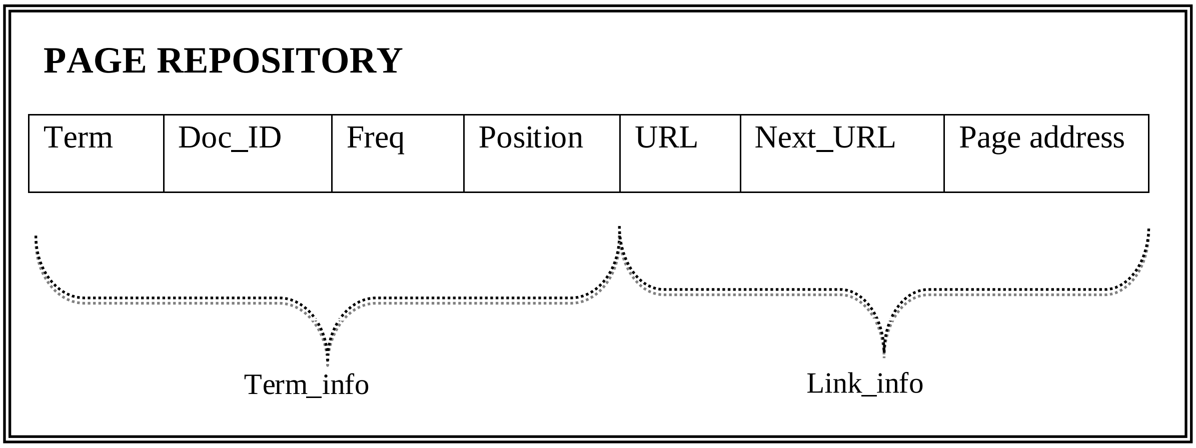
```
Step 1: Repeat for every URL(i) in Next _ URL list
Step 2: Repeat for every URL (j) in UR frontierL
            2.1 If (URL(j) = = URL(i))
                Delete URL(i) from Next_URL list;
                 Priority_Bit(URL(j)←Priority_Bit(URL(j)+1;
                Store Pos← Position(URL(j))
                Repeat for k=Pos1 to k=1
                while (k>0 &&  Priority_Bit(URL(Pos)) > Priority_Bit(URL(k))
                        temp=URL(Pos);
                Repeat for m=Pos to m=k
                    URL(m)=URL(m-1);
                URL(k)=temp;
 2.2 else
Set Priority _Bit (URL(i)) ←1;
Insert URL(i) at last position in the list of URLs having same visit  time.
```

Fig 5.8: Algorithm for URL Prioritizer

It may be noted that by adopting the aforementioned scheme, not only the important pages are downloaded/refreshed in prior to others but the chances of downloading the same document again and again are also eliminated.

5.2.5 Page Repository

It contains the entire information regarding the parsed web page. It stores two types of information namely: i) link info related to structural summary of web page within the web graph; ii) term info related to content of the web page. The schema for page repository is shown in Fig 5.9.



PAGE REPOSITORY

Term	Doc_ID	Freq	Position	URL	Next_URL	Page address

Fig 5.9: Schema for Page Repository

The description of various fields of page repository is given in table 5.2

Table 5.2: Description of Fields of Page Repository

Field	Description
Term	The word extracted from page such as engineer, college course etc.
Doc_ID	A unique number or alphanumeric no. is assigned to each page such as D1,D2,and D3.
Frequency	No. of occurrences of term in the page.
Position	It is a numeric value which tells the position of term in the page.eg. College at 53 position indicates that it is the 53rd word of the parsed page Di
URL	The URL of the page being extracted.
Next_URL	The out links of extracted page.
Page address	It specifies the memory location of page on the server site

Once the page repository contains the information about the parsed pages, they are later stored in domain specific database.

5.2.6 URL Scheduler

The URL Scheduler is responsible for fetching the URLs from multiple Prioritized URL Queues maintained by URL prioritizer and assign them to the multiple instances of the Document Downloader for downloading the associated web pages. In order to choose a URL from URL Prioritized queue, the URL Scheduler selects the URL from the list having the highest priority. The URL list with the highest priority means all those URLs that have the lowest value of *VI.* The list is ordered on the basis of number of outlinks.

The first URL in a priority queue is the one which occurs maximum times as an out link in other web pages and thus stored at the first position in the Prioritized URL Queue with respect to same visit interval. URL Scheduler processes all the URLs from the URL list having the highest priority (at first position) to the lowest priority (last position) in a sequential manner. For example, consider the prioritized URL

Queue as depicted in Fig 5.10, where k different priority lists of URLs have been formed by the URL prioritizer.

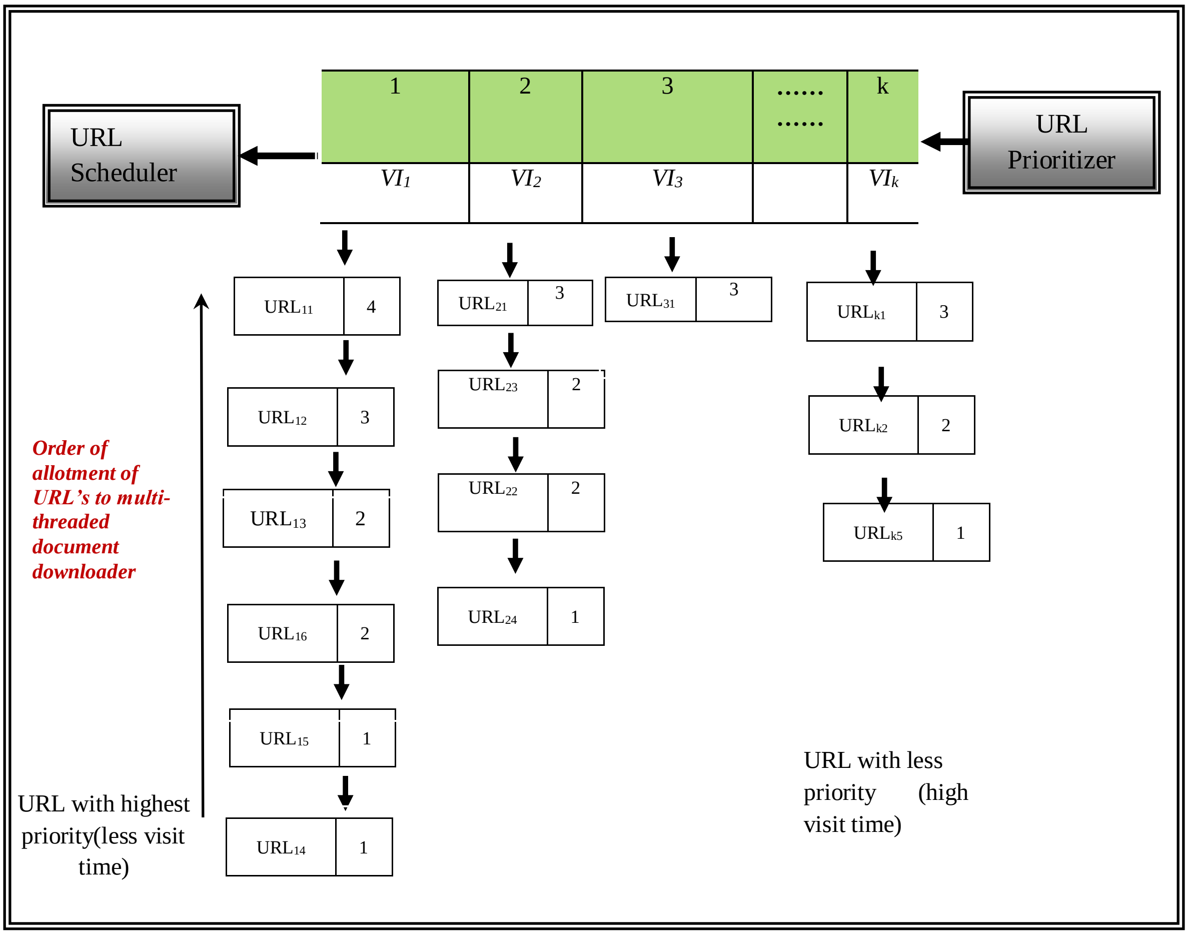


Fig 5.10: An Example: Working of URL Scheduler

These k URL lists are formed by using the different values of *VI* computed by the visit scheduler where each of the new URLs is inserted in their respective URL Pool marked by *VI* in accordance to its priority bit.

The URL Scheduler starts by taking the URL list at the first position headed by the *VI* equal to one as shown in Fig 5.10. It extracts the first URL, denoted by URL_{11} in this list and delegates it to an instance of the Document Downloader for downloading the web page associated with this URL_{11}, fetches the next URL represented by URL_{12} in the same list, delegates it to another instance of the Multi-threaded Document Downloader [96] and proceeds in a same fashion for all the n URLs included in this list.

After allocating all the URLs in the first URL list, the URL Scheduler extracts the URLs from the list occupying the next position i.e. VI=2 in the same manner. Afterwards, it extracts the next URLs from the other lists (those having lesser priority) and so on.

URL Scheduler performs the same task for each URL list at the various k positions in the ascending order of their visit interval denoted as $VI1 < VI2 < VI3 < VI4 <..... < VIk$. The algorithm for URL Scheduler is given in Fig 5.11.

```
URL Scheduler ()
Input: URL Frontier with prioritized URL.
Output: Allotment of URLs to multi- threaded document downloader   .
Method:
While (! empty (all priority queues))
        { Repeat for all position i=1 to  n in  each priority queue
                { while (!empty (URL List(Position i)))
                        Fetch URL from URL List;
                         Pass URL to instance of multi- threaded document downloader;
                           URL=getNextURL(URL List(Position i))}
}}
```

Fig 5.11: Algorithm for URL Scheduler Module

It may be noted that each instance of multi- threaded document downloader independently downloads documents for the URL set assigned by the URL scheduler. Since all instances have given different URL seed set, so minimum overlap of downloaded document is achieved. Thus the architecture requires no coordination overheads among the multi- threaded document downloader rendering it to a highly scalable system.

5.2.7 Multi-threaded Document Downloader

The Multi-threaded Document Downloader is a high performance asynchronous HTTP client capable of downloading several web pages in parallel. It initiates a number of downloader instances equal to the number of URLs received (for downloading) from the URL Scheduler by processing the different prioritized URL queues. The instances download the web pages in parallel from the multiple web servers and pass them to the Visit Scheduler for further processing. The working of the multithreaded document downloader can be understood with the help of the algorithm in Fig 5.12.

Document Downloader ()

Input: URL to be fetched.

Output: Allotment of URLs to multi- threaded document downloader instance .

Method:

```
Do forever
Repeat for each URL set received from URL Scheduler ()
{ Open N number of HTTP connections;
        Repeat for each connection
        Download web page corresponding to allotted URL;}
```

Fig 5.12: Algorithm for Document Downloader

The performance evaluation of proposed arithmetic progression based crawling mechanism is given in next section.

5.3 PERFORMANCE EVALUATION

The proposed crawling mechanism implemented in C# .net with MS SQL Server 2012 at the back end and test run was carried out on sampled seed URL set. In the experiment, the URLs are grouped in three different categories as follows:

- **Less dynamic:** less dynamic web pages are those that do not change over a period of time.
- **Moderate:** Pages that are changed by web site administrator reasonably.
- **Dynamic:** web pages that changes very frequently

To test the idea, a seed set of URLs from five different domains viz education, sports, travelling & tourism, food & beverages and fashion & shopping is chosen (URL set is exponentially grown by adding the extracted hyperlinks up to depth 3 in the hierarchal structure of downloaded pages). Further under each domain, URLs from aforementioned categories i.e. less dynamic, moderate, dynamic are taken. Initially visit interval for URLs belonging to less dynamic category is set to 15 days whereas visit interval for moderate category is set to 1 week, for dynamic category it is set to 12 hours. The priority bit for all the URLs within each category is initially set to 1. After downloading the web pages related to selected seed set by multi – threaded document downloader instances, the visit interval for each downloaded page is updated by the visit scheduler and priority bit is updated by URL prioritizer module. A visit is said to be successful here, if a page is found to be changed in the expected *VI*. On occurrence of any change, the old version of page is replaced by the latest version.

5.3.1 Performance Metric

Precision (P) is defined as a fraction of re-crawled page that encountered change over all the pages re-crawled by the proposed crawler. Mathematically, *Precision* is given by eq^n (5.5).

$$\boldsymbol{P = CP/(CP + NP)} \qquad \textbf{(5.5)}$$

Where:

- *CP* is the number of web pages encounter change i.e. relevant pages re-crawled.
- *NP* is the number of web pages that do not encounter change i.e. irrelevant pages re-crawled.

Recall (R) is defined as a fraction of re-crawled pages that encountered change over all relevant pages .The relevant pages are the pages that are re-crawled and also those pages that are not re- crawled by the proposed crawler but undergone change. Mathematically, *Recall* is given by the eq^n (5.6).

$$\boldsymbol{R = CP/AP} \qquad \textbf{(5.6)}$$

Where:

- *CP* is the number of re-crawled pages encounters change.

- *AP* is the number of all pages undergone change.

F-measure (F) is defined as harmonic mean of precision and recall. Mathematically, *F-measure* is given by eq^{n} (5.7).

$$F = 2PR \; / \; (P+R) \tag{5.7}$$

F-measure evenly weights Precision and Recall.

5.3.2 Procedure

The experiment was conducted on two different systems in computer lab of YMCA University of Science and Technology. The APCM was running on one machine and conventional crawler was running on other machine. The *Precision* and *Recall* of two crawling systems is computed in terms of URL crawled in particular time. The experimental results obtained by two systems for following websites are given below:

Test1 was conducted on website of *Kurukshetra University, Kurukshetra* (www.kuk.ac.i/) for both the crawling schemes i.e. conventional and proposed APCM. The observed results for are summarized in table 5.3.

Table 5.3: Test 1- Re-Crawling Results for KUK Web Site

Run #	**AP (KUK)**	**APCM**		**Conventional**		**Precision (%)**		**Recall (%)**		**F- Measure (%)**	
		CP	**NP**	**CP**	**NP**	**APCM**	**Conventional**	**APCM**	**Conventional**	**APCM**	**Conventional**
1	103	82	14	71	53	85.4	57.25	79.6	68.9	82.39	62.53
2	98	81	15	68	56	84.3	54.83	82.65	69.38	83.46	61.25
3	107	86	15	69	55	85.4	55.64	80.37	64.48	82.80	89.73
4	72	61	12	55	69	83.5	44.35	84.72	76.38	84.10	56.11
Average (%)						**84.6s**	**53.02**	**81.83**	**69.78**	**83.19**	**59.91**

Test 2 was conducted on website of *YMCA University of Science and Technology (http://ymcaust.ac.in/)* for both the crawling schemes i.e. conventional and proposed APCM. The observed results for are summarized in table 5.4.

Table 5.4: Test 2 - Re-Crawling Results for YMCAUST Web Site

Run #	**AP (YMCAUST)**	**APCM**		**Conventional**		**Precision (%)**		**Recall (%)**		**F- Measure (%)**	
		CP	**NP**	**CP**	**NP**	**APCM**	**Conventional**	**APCM**	**Conventional**	**APCM**	**Conventional**
1	24	19	4	15	12	82.6	55.5	79.16	62.5	80.84	58.79
2	19	16	3	13	14	84.2	48.14	84.2	68.42	84.2	56.51
3	22	18	3	15	12	85.71	55.5	81.81	68.18	83.71	61.19
4	20	16	2	14	13	83.3	51.85	80	70	81.61	59.57
Average (%)						**83.92**	**52.74**	**81.29**	**81.29**	**82.89**	**59.01**

Test3 was conducted on website of *University of Delhi (http://www.du.ac.in/du/)* for both the crawling schemes i.e. conventional and proposed APCM. The observed results for are summarized in table 5.5.

Table 5.5: Test 3- Re-Crawling Results for DU Web Site

Run #	**AP (DU)**	**APCM**		**Conventional**		**Precision (%)**		**Recall (%)**		**F- Measure (%)**	
		CP	**NP**	**CP**	**NP**	**APCM**	**Conventional**	**APCM**	**Conventional**	**APCM**	**Conventional**
1	105	80	15	69	51	84.21	57.5	76.19	65.71	79.99	61.33
2	107	86	12	68	52	87.7	56.66	80.37	63.55	83.87	59.90
3	92	79	12	68	52	86.81	56.65	85.86	73.91	86.33	64.13
4	82	71	11	60	60	86.58	50	86.58	73.17	86.58	59.40
Average (%)						**86.3**	**55.20**	**82.25**	**69.08**	**84.19**	**61.19**

The summarized results of Precision, Recall and F-measure for both conventional and proposed system under three tests are given in table 5.6.

Table 5.6: Summarized Results of Re-Crawling for Different Tests

Test#	**Precision (%)**		**Recall (%)**		**F- measure (%)**	
	APCM	**Conventional**	**APCM**	**Conventional**	**APCM**	**Conventional**
Test1	84.65	53.02	81.83	69.78	83.19	59.91
Test2	83.92	52.74	81.29	81.29	82.89	59.01
Test3	86.32	55.20	82.25	69.08	84.19	61.19
Average	**84.96**	**53.65**	**81.79**	**73.38**	**83.42**	**60.03**

The Precision values for both conventional and APCM obtained in three tests are plotted in Fig 5.13.

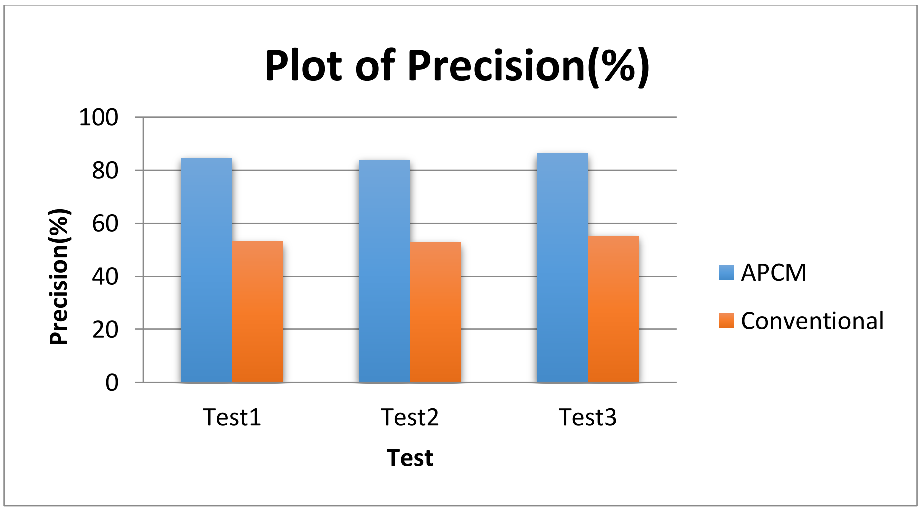


Fig 5.13: Plot of Precision (%) for Proposed APCM and Conventional Crawler

It may be observed from the graph shown in Fig 5.13 that average precision of proposed APCM is 84.96% and gain in Precision over conventional crawler ≃ 31%

The Recall values for both conventional and APCM obtained in three tests are plotted in Fig 5.14.

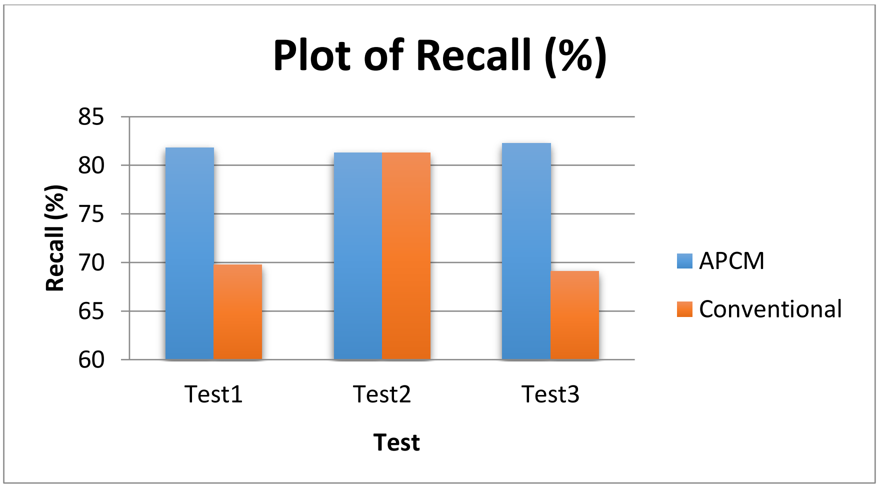


Fig 5.14: Plot of Recall (%) for Proposed APCM and Conventional Crawler

It may be observed from the graph shown in Fig 5.14 that average Recall of proposed APCM is 83.42% and gain in Recall over conventional crawler ≃ 10%

The F-measure values for both conventional and APCM obtained in three tests are plotted in Fig 5.15.

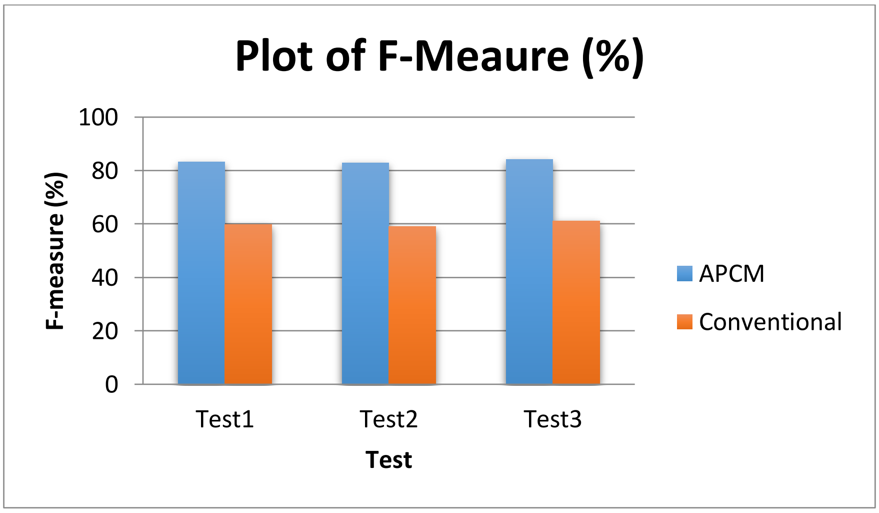


Fig 5.15: Plot of F-measure (%) for Proposed APCM and Conventional Crawler

It may be observed from the graph shown in Fig 5.14 that average F-measure of proposed APCM is 81.79 and gain in over conventional crawler ≃ 24%

The experiment results show that technique is able to maintain significant balance between accumulation of latest information and network load. Besides these advantages the system also eliminated the problem of duplicate document download by incorporating priority bit with each URL to be visited next thereby enriching the search engine repository with quality contents.

The snapshots of implementation are included in appendix E. The next chapter discusses the design of interest based search system using query structuring in detail.

CHAPTER VI

DESIGN OF INTEREST BASED SEARCH SYSTEM USING QUERY STRUCTURING

6.1 INTRODUCTION

A significant number of web users often face problem in retrieving the relevant information from web due to many reasons: web is dynamic, redundant, heterogeneous, and expanding at a staggering rate. Moreover the term ‘relevance’ is subjective and time varying as far as web user is concerned. Different users have different expectations and goals while surfing the web such as informative, navigational, communicational or transactional. They often submit short, vague and instant queries which lead to inclusion of wrong documents in the result set. The informative knowledge hidden behind the web interfaces can be efficiently obtained by optimizing crawling process, building effective indexes, recommending personalized queries to the user and preparing rank list based on user browsing patterns under a unified search system called as ***Interest based Search System using Query Structuring.***. The bi-layer architecture of proposed interest based search system is depicted in Fig 6.1.

All the processes needed by the search engine to maintain its repository of web pages are placed in ***Data Collection Layer*** .The processes concerned with retrieval and fulfillment of user’s specific need are put together in ***Data Presentation layer***. The amalgam of various mining techniques is applied at different functional level to increase the performance of system as compared to general purpose search engine. The detail description of each layer is given in following sections.

6.2 DATA COLLECTION LAYER

Data collection layer is integration of six major components namely: page classifier, page probability calculator, classified database, indexing module crawling subsystem and page repository specifically designed to optimize the back end process carried out

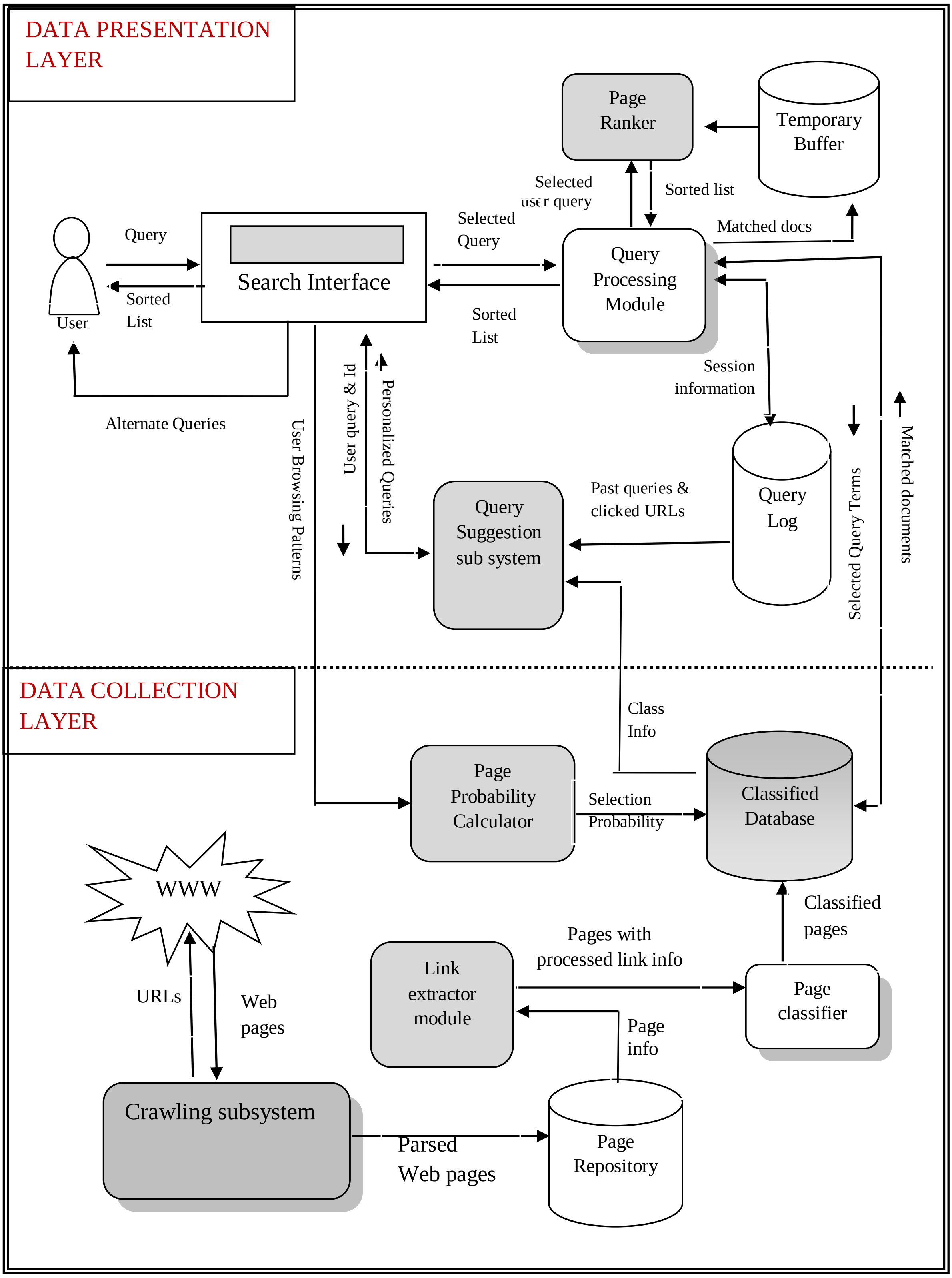


Fig 6.1: Proposed Interest based Search System using Quay Structuring

by conventional search engine. Detail discussion on each functional component of data collection layer is given in following subsections.

6.2.1 Crawling Subsystem

In a general scenario, a crawler updates its collection after a certain period of time in the batch mode. But keeping in view the varying change frequency of one web page from other, here the crawler is made intelligent by assisting it with two new modules namely: visit scheduler and URL Prioritizer module besides the conventional composition of URL Scheduler, downloader instances, URL frontier and page extractor as shown in Fig 6.2.

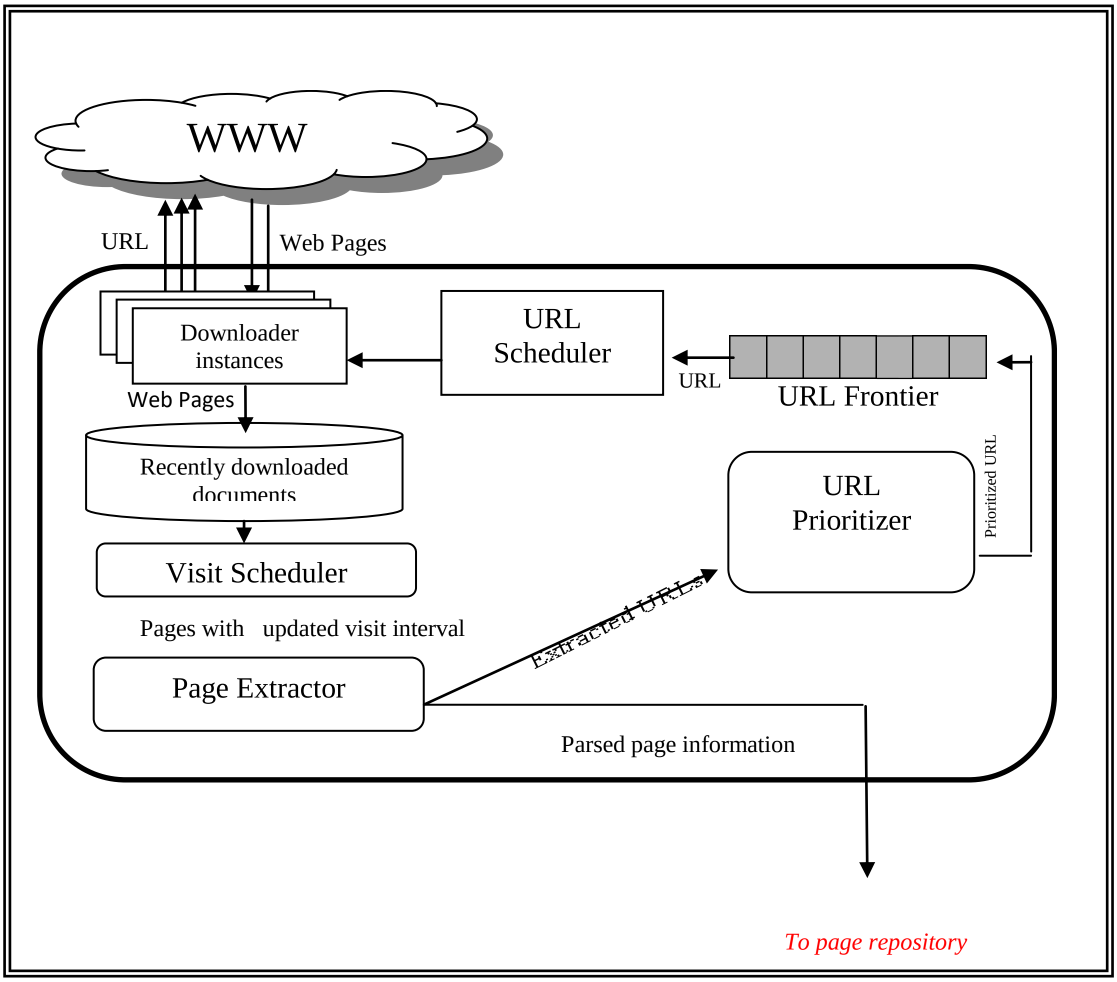


Fig 6.2: Component Diagram of Crawling Subsystem

The process of web data collection starts by gathering the set of seed URLs in URL Frontier. URL Scheduler module schedule theses URLs and allocate them to multithreaded asynchronous HTTP document downloader instances to download the web pages in parallel from various web servers. The downloaded pages are further processed by Visit sSheduler as per the scheme discussed in chapter 5.2.2. The web page sent by the visit scheduler along with updated URL record is parsed by Page Extractor module to get the text and links contained in the page. The extracted text is in turn tokenized to get a list of normalized keywords known as stems. Further, the frequency and position of each stem in the parsed document is determined and stored in page repository .The extracted URLs are passed to URL Prioritizer module as well as stored in Page Repository so as to maintain in/out links information to be later on used in page ranking. After receiving the extracted URL list from the Page Extractor module, URL Prioritizer populate the URL Frontier in accordance with the visit time of URLs , eliminate the duplicate URLs from the queue and prioritize them by means of mining the web graph by using the scheme discussed in chapter 5.2.4.

6.2.2 Page Repository

Page repository aggregates structural and content summary of downloaded page so as to facilitate the indexing process of search system. The structure of its table consists of following fields:

- Term
- Doc_ID
- Frequency
- Position
- URL
- Next_URL

Once the page repository stores the information about the parsed page, the next task is to index the page in classified database with the help of link Extractor and Page Classifier module.

6.2.3 Link Extractor

As the page rank of a page is computed in terms of its back links so it becomes significant to store the information regarding the back links of a page along with other information in the classified database itself. The link extractor module performs this task by fetching the link information of all the pages from page repository, assimilate the in links of the required pages and pass the pages to page classifier module that further store them in classified search database. In order to collect the information regarding the back links of a particular page, it searches the specified page in Next_URL field of all the other pages stored in page repository. An existence of match specifies a hit for a back link and the associated URLs in the matched rows represent the back link itself.

To better understand the process of back link accumulation, let the back link of a page (say C) need to be computed as shown in Fig 6.3. It is easily computed by the searching ‘C’ in Next_URL field of all the pages in page repository. Each occurrence of ‘C’ in Next_URL field specifies its no. of back links and associated URL in URL field represents back links itself. For example there are two back links of page C, named URL A and URL D marked by dotted encircled rows in sample page repository shown in Fig 6.3.

Back link of C

Back link of C

Term	URL	Doc ID	Next_URL	Frequency	Position
Prime	A	D1	B,C,D	3	5,10,23
Prime	B	D2	F,G	3	3,28.56
Number	F	D5	H,I,J	6	12,34,56,87,101,115
System	F	D5	H,I,J	6	13,35,57,88,102,116
Java	C	D3	A,D	5	13, 24,30,48,54
Java	D	D4	B,G,A,C	4	11,20,34,41
Factor	B	D2	F,G	3	4,29,57
Math	B	D2	F,G	3	6,31.59
……	……	…..	……	…..	……

Fig 6.3: Assimilating the Back Links from Page Repository

In the same way, the back links of any web page can be easily identified and this information is further utilized in rank computation.

6.2.4 Page Classifier Module

One of the key goals of proposed interest based search system is to build effective indexes so as to provide personalized query results for each user. To achieve this, Page classifier module is introduced to enhance the capabilities of search engine database. It classifies and stores the downloaded page information under different domain specific classes.

These classes are further used to discover the degree of user interest in different domains so as to fulfill user specific needs. In order to construct various domain specific classes, the Page Classifier has to perform the following:

- Construct the initial set of domain classes populated with some seed keywords set.
- Identify the domain of each and every web page the crawler had come across during incremental crawling process.
- Store the information of page under identified class.
- Progressively enhance the keyword set of each class.

The process of page classification begins by selecting the candidate domains with some seed keywords set. In the present experiment the five domain classes namely: education, travelling & tourism, sports, food and beverages and fashion & shopping have been taken. Initially, seed keywords set of each candidate class is formed by extracting the terms of web pages associated with seed URLs contained in URL frontier of crawling subsystem as shown in Fig.6.4.

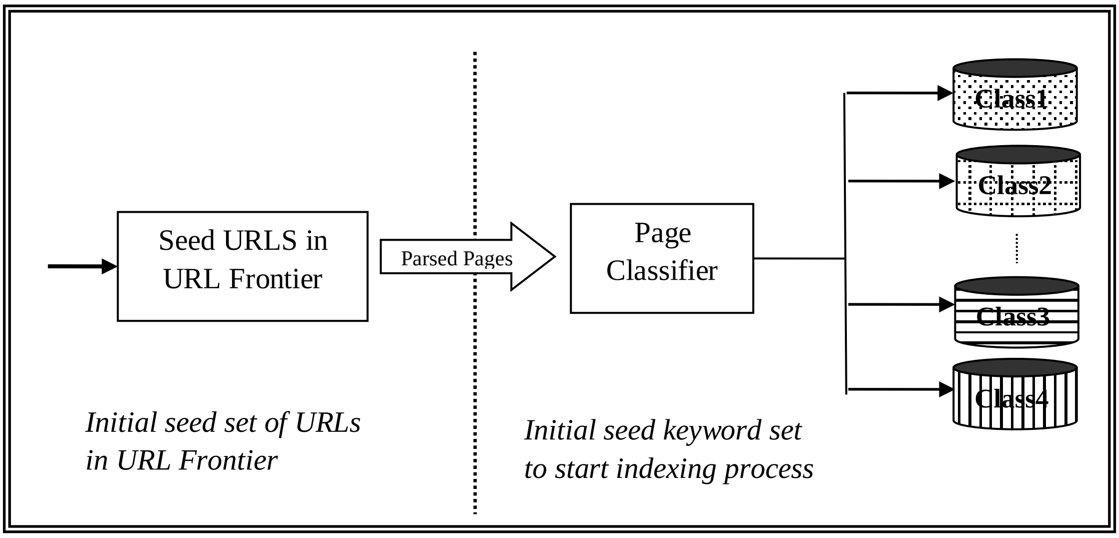

Seed URLS in URL Frontier
Parsed Pages
Page Classifier
Class1
Class2
Class3
Class4
Initial seed set of URLs in URL Frontier
Initial seed keyword set to start indexing process

Fig 6.4: Constructing the Initial Seed Keyword Set for Each Domain

Once the classes have been initialized, the next job of page classifier is to classify each and every web page obtained during the incremental crawling process under specified class. So, it takes the page as input from link extractor module and computes the intersection of page terms with class keyword set.

Depending upon the value of intersection, a page may undergo one of the following actions:

Creation: If the intersection of page terms with class keywords is less than the minimum threshold value of intersection i.e. $I_{val} < I_{min}$, A new class is created with keyword set initialized to page terms.

Insertion & Updation: If the intersection of page terms with class keywords is more than the maximum threshold value of intersection i.e. $I_{val} > I_{max}$, then the page is inserted in the qualified class and page keywords are also used to enhance the keywords set of that class.

Insertion Only: When the intersection of page terms with class keywords lies between the boundary conditions i.e. $I_{val} \in [I_{min}, I_{max}]$, then page is only inserted into the qualified class.

In the experiment, the maximum threshold intersection of two sets is set to 0.8 and minimum threshold intersection is set to 0.2. The flow of aforementioned actions can be understood by the flow graph shown in Fig 6.5.

It is further observed that as initially the no. of keywords in each class are less so it become easy for a page to score high intersection value with class keywords and thus stored under the class. But as more and more pages get inserted in the class, the no. of keywords in the class also increases at a high rate.

So, it becomes difficult for a new page to obtain optimal intersection score. This causes sudden decrease in no. of pages to be inserted in the class.

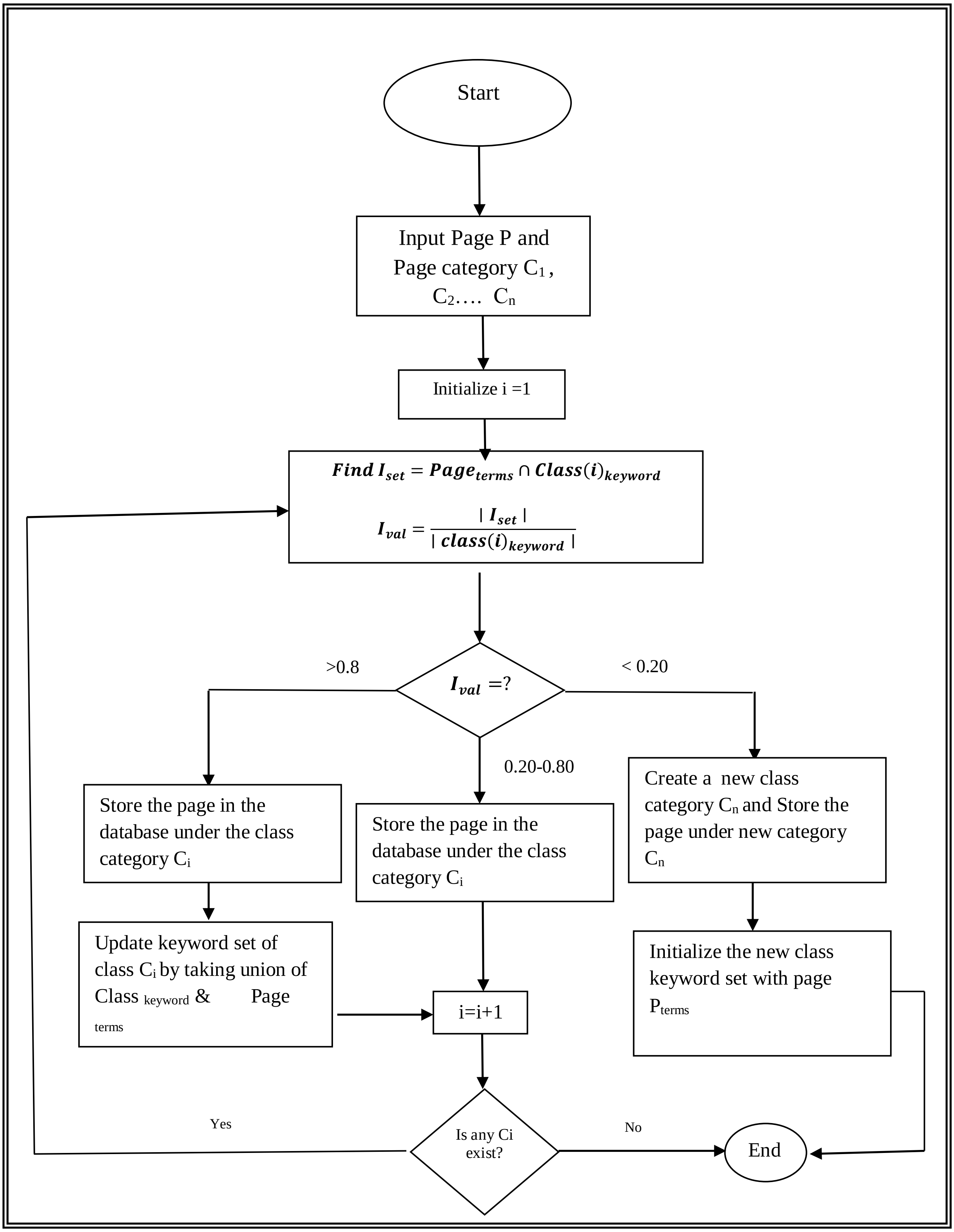

Start
Input Page P and Page category C_1 , C_2…. C_n
Initialize i =1
$Find\ I_{set} = Page_{terms} \cap Class(i)_{keyword}$
$I_{val} = \frac{|\ I_{set}\ |}{|\ class(i)_{keyword}\ |}$
I_{val} =?
>0.8
< 0.20
0.20-0.80
Store the page in the database under the class category C_i
Store the page in the database under the class category C_i
Create a new class category C_n and Store the page under new category C_n
Update keyword set of class C_i by taking union of Class $_{keyword}$ & Page $_{terms}$
i=i+1
Initialize the new class keyword set with page P_{terms}
Yes
Is any Ci exist?
No
End

Fig 65: Flow Chart for Page Classification Process.

Thus the insertion of pages in specific class follows the Gaussian distribution as shown in Fig 6.6.

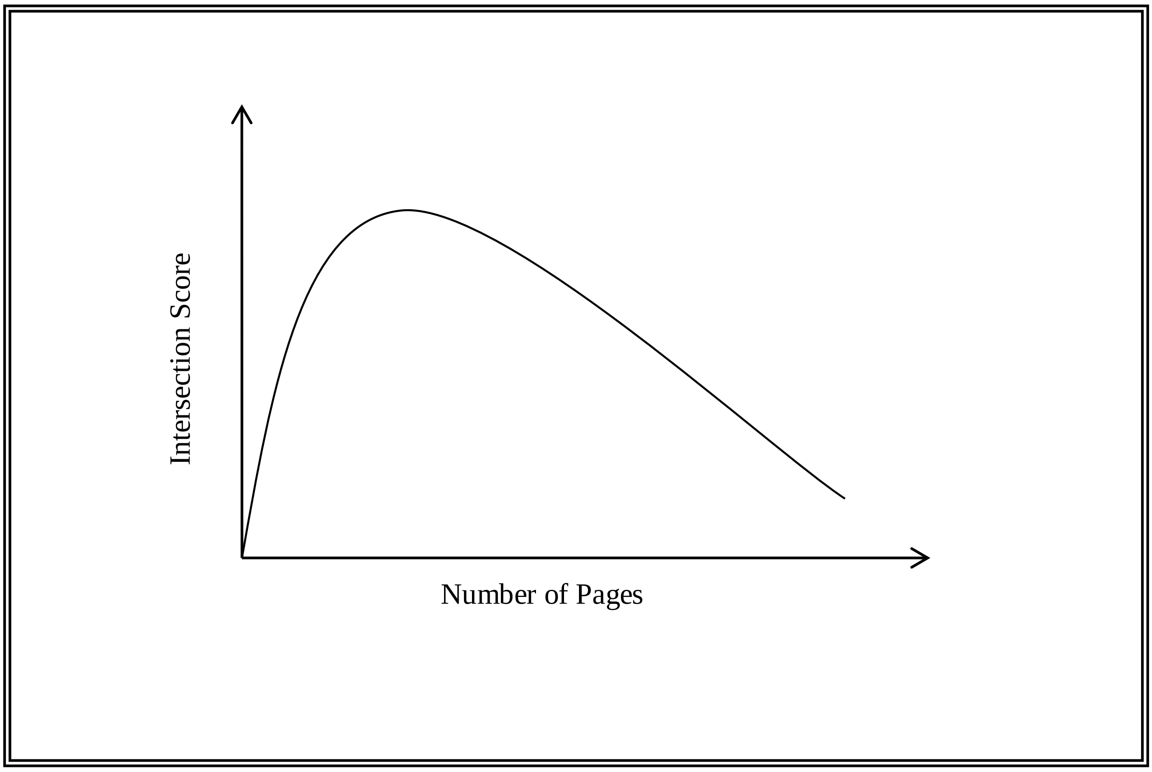


Fig 6.6: Graph Showing Relationship between Pages in the Class and Intersection Score

In light of above discussion, the mechanism for enhancing the class keywords is modified in such a way that a class will contain only those keywords that reflect the whole image of domain category.

To achieve this, an integer parameter named ***hit count*** is attached with each keyword of the class that keeps on increasing every time it occurs in intersection of a particular page with class.

Only the top m keywords having the higher value of ***hit count*** are selected to make class keyword set. The algorithm showing the modified working of page classifier module is depicted in Fig 6.7.

```
Page Classifier ()

Input: Page P with set of terms denoted by $P_{terms}$, classified database with set of keywords defined in each class; Classified database={$class_1$,$class_2$....$class_n$} where each $Class(i)_{keywords}$={$kw_1$,$kw_2$....$kw_n$}, threshold value for number of class keywords.

Output: Store the page P under appropriate class.

Method: Repeat for each class i=1 to n {                // n is no. of classes

    1. Compute intersection of $P_{terms}$ with $Class(i)_{keyword}$ ; $I_{set} \leftarrow P_{terms} \cap Class(i)_{keyword}$

    2. $compute\ I_{val} = \frac{|I_{set}|}{|class(i)_{keyword}|}$

    3. Set threshold on number of class keywords, m ← threshold_value;

    4. Store P under appropriate class

       4.1 If ($I_{val} \geq 0.8$) {

            Store P under class (i);

            For each keyword, $kW \in I_{set}$

            Set hit count (kW) ← hit count (kW) +1;

            Set $class(i)_{keyword} \leftarrow class(i)_{keyword} \cup P_{terms}$   //Update keyword set of class(i)

            Count←|$Class(i)_{keyword}$|          // count no. of keywords in class (i)

             if (count > m)

            {Sort the keyword of class (i) in decreasing order of their hit count;

             Store only top m keywords in class (i) and discard the remaining keywords;}

    4.2 else if ($I_{val} \geq 0.2$ && $I_{val} < 0.8$)

            Store the Page P under class (i);

            4.3 else {Create a new class; $class_{n+1}$ ;

            Set $class(n+1)_{keyword} \leftarrow P_{terms}$}}
```

Fig 6.7: Algorithm for Page Classifier

Finally, the page is stored along with extracted information under the domain specific classified class. The structure of classified database is discussed in section 6.2.6.

6.2.5 Page Probability Calculator

Another important goal of proposed system is to provide the search results as per user's expectations and need. Towards this direction, a new module named as Page Probability Calculator, *PPC* is introduced at the back end of search system whose primarily task is to analyze and discover the hidden trends of user browsing patterns as per the scheme discussed in section 4.3.1.

Based upon the discovered knowledge it assigns a numeric score to each page termed as page probability factor, PPF and store it in classified database as shown in Fig 6.8.

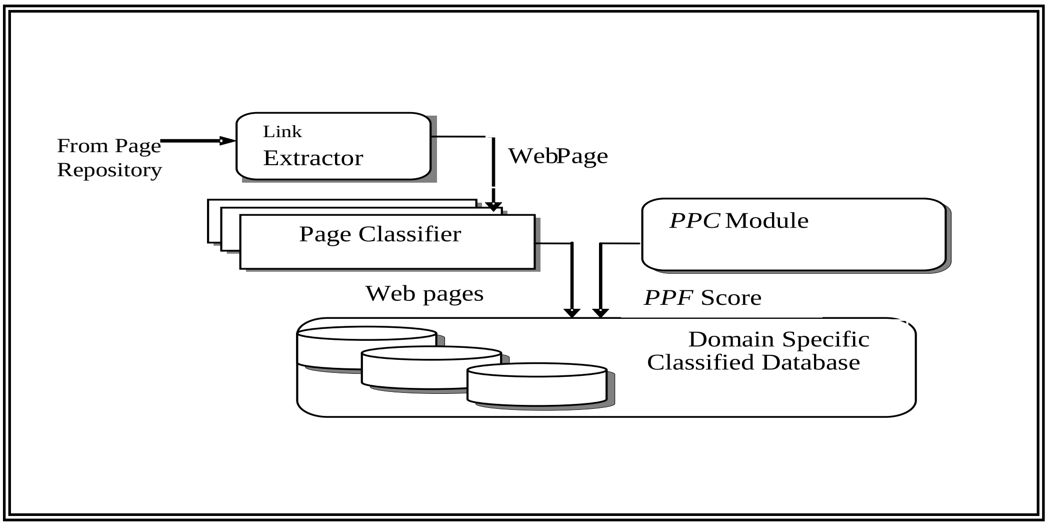


Fig 6.8: Storing the PPF Score with Other Information in Classified Database

6.2.6 Domain specific Classified Database (DSCD)

In the proposed system, the existing data structure of search engine database is modified for the sake of better interpretability, efficiency and personalization. The existing database is classified in various domain specific classes wherein each class contains the information about terms extracted from downloaded documents along with their link information and PPF score.

Here, the term refers to normalized words obtained after stemming process. The broad level schema for each domain specific class is depicted in Fig 6.9.

`

Domain Specific class Schema

Term	URL	Class_ID	Doc_ID	Back links	Forward links	PPF Score	Freq	Position

Fig 6.9: Schema of Data Structure Used for Domain Specific Class

The description of various fields of domain specific classes is summarized in table 6.1.

Table 6.1: Description of Domain Specific Class Schema

Field	Description
Term	The word extracted after stemming process. For example the term obtained after stemming of graphing and graphics is graphic.
Doc_ID	A unique number or alphanumeric no. is assigned to each page such as D1,D 2, D3.
URL	URL of the page being classified and indexed.
Class_ID	Unique identifier for the domain under which the document is stored. This field is assigned by page classifier module.
Frequency	No. of occurrences of each term after stemming process retrieved from page repository.
Position	It is a numeric value which tells the position of term in the page after stemming.
back_links	No. of incoming links to the URL obtained by link extractor module.
forward_lnks	No. of outgoing links from the URL taken from page repository.
PPF score	The numeric value associated with each document assigned by PPC module.

`

Further the information in each class is organized into four portions namely: *Class listing*, *Term vocabulary*, *Document listing* and *Position listing* as shown in Fig 6.10. *Class listing* specifies the class to which the term belongs. The *Term vocabulary* contains the information about the term and number of documents that contains it.

Document listing contains the information about the document identifier, no. of back links & forward links to the document, page probability factor (PPF) and frequency of term in the document.

Finally, *Position listing* contains the information regarding exact occurrence point of term in the document.

In order to better understand the organization of information in a specific class say, **Class C_1,** Let us consider the sample terms and related information given in table.6.2.

Table 6.2: Organization of Information in Classified Database

Term	Doc ID	Back links	Forward links	PPF	Frequency	Position
Java	D3	7	5	4.5	4	4,12,23,56
Java	D8	9	3	7	3	7,45,89
-	-			-	-	-
Factors	D3	7	6	4.5	6	5,13,24,57,79 ,99
Prime	D8	9	7	7	3	8,46,90

For instance, in table 6.2, the term ‘Java’ occurs at four different places: 4, 12, 23 and 56 in document D3 in row_1. The back links of D3 are 7 and forward links are 5 .PPF score is 4.5. Likewise, information about the other terms is also organized. The database containing the fresh information related to different domains are thereof, used by the processes contained in data presentation layer to be discussed in section 6.3.

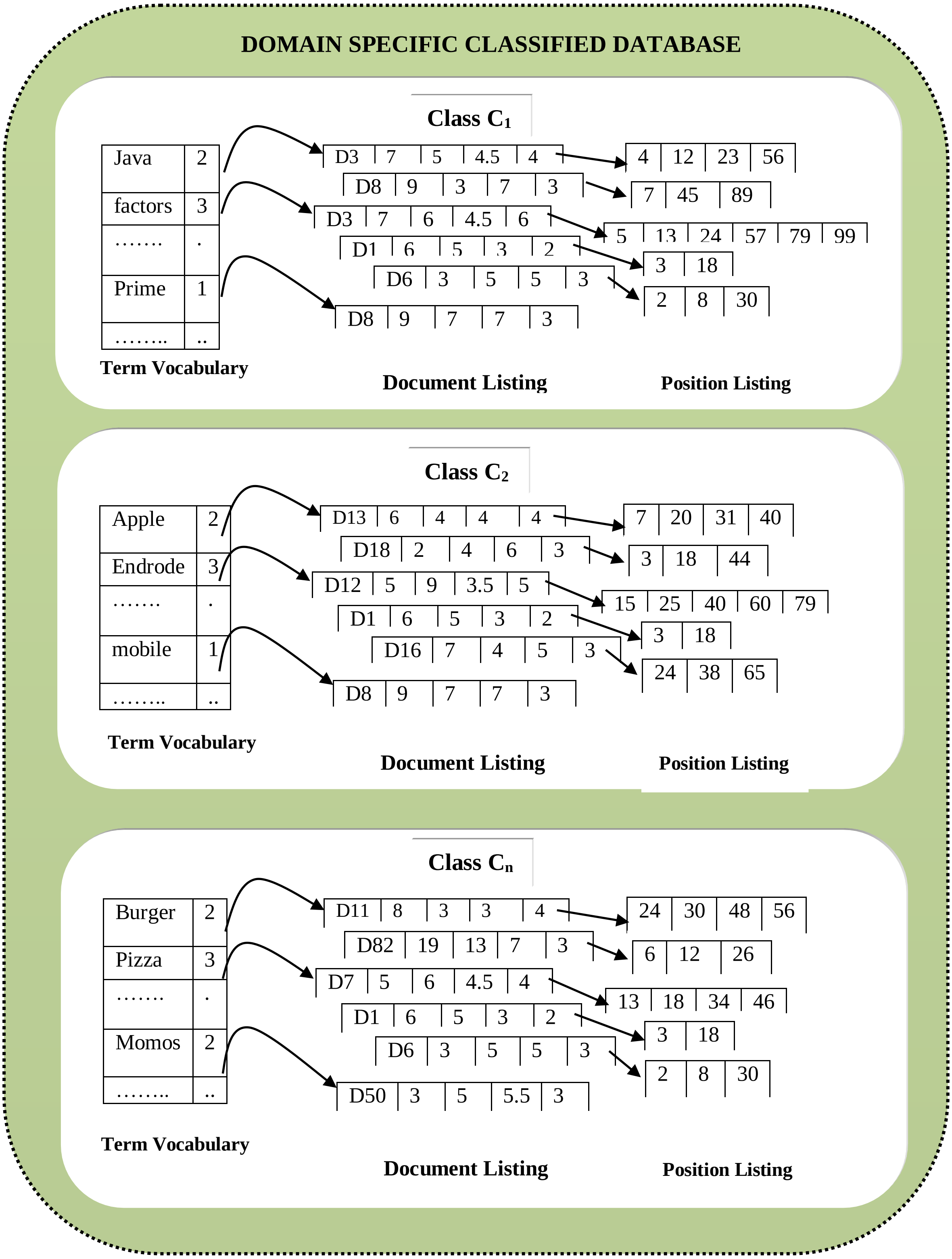


Fig 6.10: Proposed Domain Specific Classified Database

6.2.7 Summary of Data Collection Layer

The techniques proposed in data collection layer at various functional levels are summarized in table 6.3.

Table 6.3: Summary of Proposed Techniques at Data Collection Layer

Techniques → **Parameters↓**	**APCM (crawling mechanism)**	**PR-PPF(Ranking mechanism)**	**DSCD (Indexing mechanism)**
Description	Compute re-crawl interval for each site separately.	Compute selection probability of each page separately.	Indexing downloaded pages in domain specific database.
Application point	Crawling	Ranking	Query recommendation
I/P parameters	Page updation frequency	User browsing patterns	Tokens and related information
Source	Web servers	Session information	Page repositories
Mining Technique	web structure mining	Web usage mining	Web content mining
Advantages	• Maintain latest information in search engine database without incurring extra load on network. • Eliminate repeated downloading of same document. • Fetch important pages prior to others	• Importance of page is determined as per user feedback. • Returns relevant results to the user.	• Helps to compute degree of interest of each user in different domains • Results in personalized search.

The next section describes proposed data presentation layer in detail.

6.3 DATA PRESENTATION LAYER

With the aim to correctly interpret the user query and present relevant results to user in easy accessible way, a couple of mining techniques are applied at various level of data presentation layer. Data Presentation layer consists of four major components: search engine interface, query suggestion sub system, query processing module, and page ranker. A discussion on each component is given in following sub sections.

6.3.1 Search Interface

It is an interface where the user specifies its information need in the form of query. It first creates the account for a novice user or verifies the existing user with the help of special module named as profile generation module as shown in Fig 6.11(step 1). After creation/verification, user can submit its query at search interface (step 2). It then gets a set of personalized alternate queries from query suggestion subsystem and offers them to user (step 3 and 4). After selection of one of the alternate query by the user, it sends a signal *'something to record'* to PPC module and passes the selected query to query processor module (step 5 and 6). At the end of search operation, it receives sorted list of documents from query processor to present back to the user (step 7 and 8).

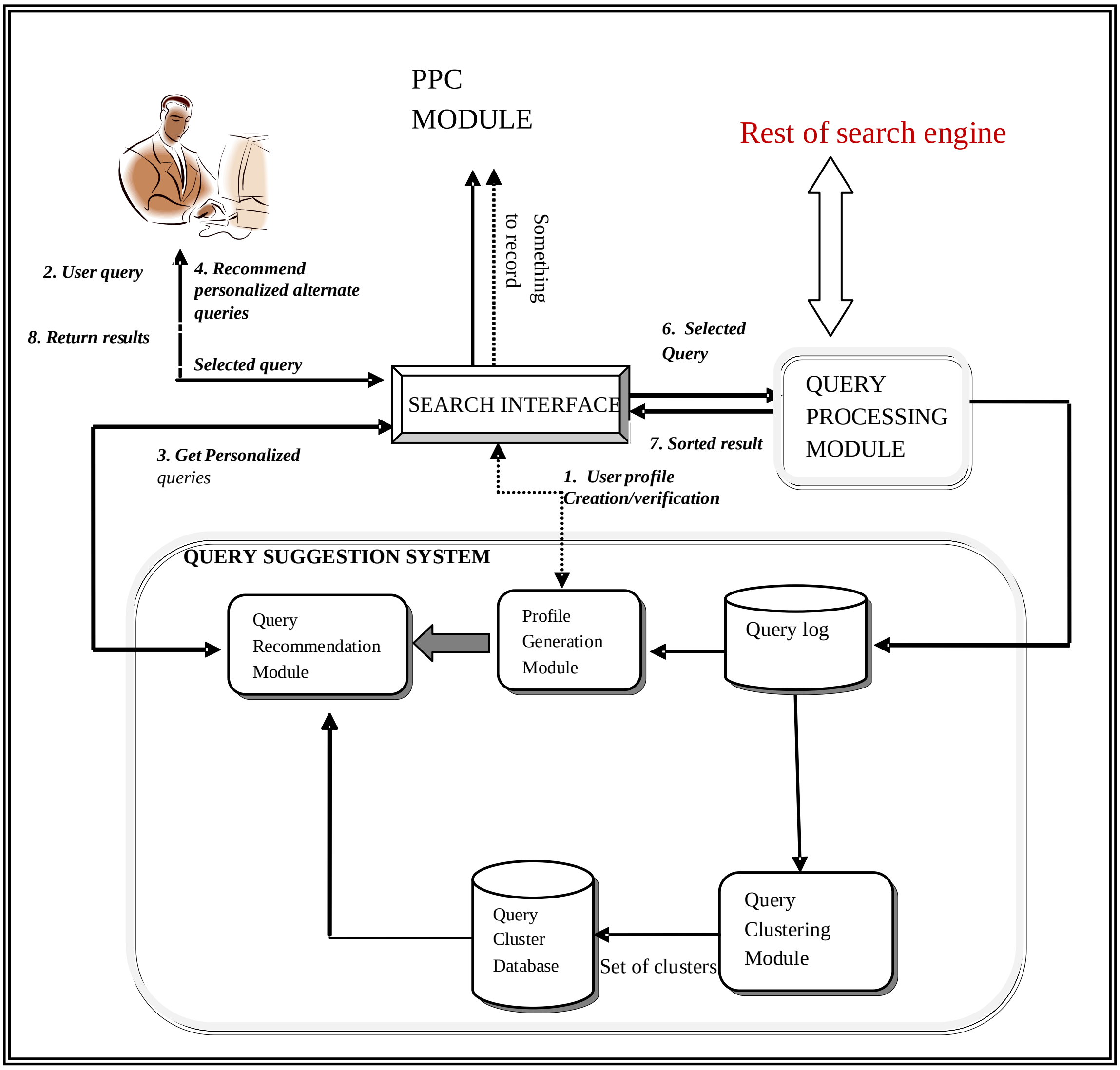


Fig 6.11: Interaction of Search Interface with Various Components

6.3.2 Query Suggestion module

As the keywords of submitted query directly map to selection of document into the result set so it become highly important to filter the user query at early stage of search process.For this purpose, query suggestion sub system is appended in data presentation layer whose primarily task is to generate personalized alternate queries for each user It applies web content mining to get all those combinations of queries which are contextually similar to user need, web structure mining to get all those combination of queries whose clicked documents are similar to issued query and finally obtain the personalized queries

Based on user past interactions with the proposed system as per the scheme discussed in section 3.3 in chapter 3. The algorithm for query suggestion sub system is given in Fig 6.12.

```
Query Suggestion Sub System ()
Input:  user query Q, user id denoted by UID
Output: Set of personalized queries $Q_{personalized}$={q1,q2,q3,q4}
Method:
Do forever
1)  wait( UID, pswd);
2)  call(Profile_Genration(UID);
3)  signal (profile verification information);
4)  wait(Q);
5)  call Query_Recommendation(Q, UID);
6)  return ($Q_{personalized}$) to search interface;
7)  call query clustering ();
```

Fig 6.12: Algorithm for Query Suggestion Sub System

After the query is selected by the user from set of personalized queries , the next task is to execute the same on classified database. The task is performed by Query Processing module discussed in following subsection.

6.3.3 Query Processing Module

The query selected by the user is propagated to query processing module. It is designed to perform four major tasks:

i) Query normalization
ii) Query execution
iii) Building query log
iv) Search result formation

Query normalization: This is most basic and indispensible step in searching the documents related to a query. A query is normalized by removing stop words as these words do not contribute towards retrieval of relevant documents. After removal of stop words, processed query is stemmed using Porter's algorithm [75].

Query Execution: The normalized query is matched with the terms stored in classified database and corresponding page URLs along with related information is retrieved. The fetched information is thus stored in temporary buffer for ranking purpose.

Building query log: In order to assist Query suggestion sub system, query processing module is also delegated the task to record certain attributes related to user interaction with search system and store them in query log. A query log is basically a file consisting of series of search requests wherein each search requests is combination of certain attributes as listed in schema shown in Fig 6.13.

Session ID	Query ID	Query	Clicked URL	Click count	URL class ID

Fig 6.13: Schema for Query Log

Description of various fields of query log is given in table 6.4:

Table 6.4: Query Log Description

Field	Description
Session ID	A numeric number assigned to a series of search requests done by a single user in a period of time.

Query ID	An anonymous identification for a particular query.
Query	The search request submitted by the user in the form of query keywords
Clicked URLs	URL clicked by the user in result list returned in response to its query.
Click count	Number of clicks on selected URL.
URL Class ID	Class identification number to which clicked URL belongs.

Search result formation: After storing the matched documents in temporary buffer, it send the signal *'something to rank'* to page ranker module and get the sorted list to return it back to search engine interface.

6.3.4 Page Ranker

The onus of determining the relevancy of retrieved documents with respect to issued query is delegated to page ranker module. It assigns a numeric score to each page as per the technique discussed in section 4.3.2. The technique works in three ways:

- It applies three important measures to rank a document. They are content similarity, link popularity and selection probability. Combination of these measures helps to move relevant documents upwards i.e. in user look around area.
- Second, content similarity is computed by not only considering the frequency of query terms in the document but also in query itself. It avoids assigning high content score to lengthy documents.
- Third, the technique improves the rank of a page based on user feedback with no extra requirement of creating and storing cumbersome profile for each individual.

6.4 PERFORMANCE EVALUATION

To check the performance of proposed system, a volunteer group of 25 students from YMCA University are asked to use the system for five selected domains i.e. education, travelling & tourism, sports, food and beverage and fashion and shopping class.

The system assisted each user with set of personalized queries. The user selects/refines the query and submits it. Based on user past browsing patterns, a list of URLs corresponding to submitted query is presented back to the user. The feedback about quality of search results in terms of number of satisfactory pages from each user as also collected. On the responses the performance metric *Result Relevancy Score (RRS)* was applied for each user and its average was taken. . The *Result Relevancy Score (RRS*) is computed by eq^n(6.2).

$$Result\ \ Relevance\, Score\ (RRS) in\ \% = \frac{Number\ of\ Relevant\ pages}{Total\ number\ of\ pages} \times 100 \qquad (6.1)$$

The *Result Relevancy Score* for set of queries issued by user 1 (Set 1) is given in table 6.5(a).

Table 6.5 (a): Set 1- Result Relevance Score (RRS) for Set of Queries Submitted by User 1

S.No.	Queries	Query Type	No. of Relevant pages (RP)	Total No. of pages (TP)	RRS=RP/TP*100
Q1	Benefits of green tea	Food	13	19	68.42
Q2	Primier technical university in north india	Education	29	37	78.37
Q3	Best apple phone	Shopping	7	9	77.77
Q4	Rayban	Shopping	10	14	71.42

	sunglasses				
Average RRS (%)					**73.99**

The *Result Relevancy Score* for set of queries issued by user2 (Set 2) is given in table 6.5(b).

Table 6.5 (b): Set 2- Result Relevance Score (RRS) for Set of Queries Submitted by User 2

S.No.	**Queries**	**Query Type**	**No. of Relevant pages (RP)**	**Total No. of pages (TP)**	**RRS=RP/TP*100**
Q1	Singhai currency	Travel	6	8	75
Q2	Singapoe package	Travel	12	16	75
Q3	Best place to live in India	Travel	10	12	83
Q4	Conductors of electricity	Education	9	13	69.23
Average RRS (%)					**75.55**

The *Result Relevancy Score* for set of queries issued by use3 (Set 3) is given in table 6.5(c).

Table 6.5 (c): Set 3- Result Relevance Score (RRS) for Set of Queries Submitted by User 3

S.No.	**Queries**	**Query Type**	**No. of Relevant pages (RP)**	**Total No. of pages (TP)**	**RRS=RP/TP*100**
Q1	Source of vitamin D	Food	9	12	75
Q2	Oops tutorials	Education	19	25	76

Q3	Protein rich food	Food	8	10	80
Q4	7th pay commission	Education	16	19	84.21
Average RRS (%)					**78.80**

The *Result Relevancy Score* for set of queries issued by user4 (Set 4) is given in table 6.5(d).

Table 6.5 (d): Set 4- Result Relevance Score (RRS) for Set of Queries Submitted by User 4

S.No.	**Queries**	**Query Type**	**No. of Relevant pages (RP)**	**Total No. of pages (TP)**	**RRS=RP/TP*100**
Q1	Food of Gujarat	Food	11	17	64.70
Q2	Punjabi dhabba in faridabad	Food	12	14	85.71
Q3	Clean city India eassy	Education	16	19	84.21
Q4	Delhi to Ahmadabad airliners	Travel	8	13	61.53
Average RRS (%)					**74.03**

The *Result Relevancy Score* for set of queries issued by user5 (Set51) is given in table 6.5(e).

Table 6.5 (e): Set 5- Result Relevance Score (RRS) for Set of Queries Submitted by User 5

S.No.	**Queries**	**Query Type**	**No. of Relevant pages (RP)**	**Total No. of pages (TP)**	**RRS=RP/TP*100**
Q1	PV Sindhu awards	sport	13	19	68.42
Q2	Common	sport	28	35	80

	wealth games 2010				
Q3	MRIU	education	8	11	72.72
Q4	PV Sindhu ranking	Digital	12	19	63.16
Average RRS (%)					**71.08**

Plot of average RRS obtained for five sets of queries submitted by different users for for five different domains is depicted in Fig 6.14.

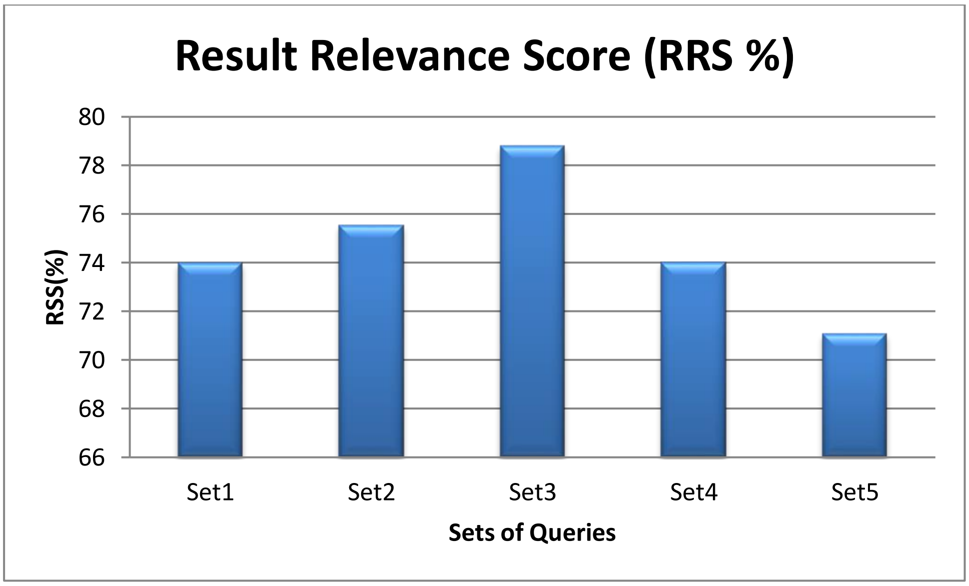


Fig 6.14: Result Relevance Score for various Query Sets

It may be observed from the Fig 6.12 that average *Result Relevancy Score* for five sets is ≃ 75%. So it can be noticed that the proposed search system optimizes the search process beginning from query formation to till ranked list presentation in an efficient way.

Further, the benefits of modified data structures are listed below:

- Placing the lower and upper bound on re-crawl time in URL Frontier helps in maintaining optimization between network load and availability of updated information.
- Priority bit in URL Frontier directs the crawler to download the important

pages first among the URLs having the same visit time. This results in fast accumulation of fresh information.

- Back link and forward link information in classified classes help in evaluating the page rank of a page based on link structure mining.
- Frequency and Position information is used to find the relevancy of page with respect to user query. A page that contains high frequency of query keywords is considered to be highly relevant.
- PPF information signifies the relative importance of page within its alike group of pages based on web usage mining. It is used to precisely cater the user information need thereby improving the quality of web search engine.
- Finally, the organization of various information related to web pages in classified database is highly beneficial in inferring the user interest areas and providing relevant results thereof high user satisfaction level is archived.

The nest chapter concludes the outcome of the work proposed in this thesis. The future research directions are also enumerated in this regard.

CHAPTER VII

CONCLUSION AND FUTURE SCOPE

7.1 CONCLSION

WWW has become tremendous source of information on almost every topic of the world. Everyday millions of pages are created and updated on web. Search engines are used as a vital tool for accessing such volatile information. Many sophisticated algorithms have been used to improve user search experience through search engines. But still there exist many untouched areas of prime interest that can significantly improve the quality of web search. Towards this goal, an extensive survey on web mining in context of web personalization has been done and major challenges towards providing quality information were identified that further become the basis for objectives of work carried out in this dissertation.

The foremost objective of the work is to build search system that can efficiently mechanize query processing, ranking, indexing and crawling tasks to improve overall user search experience. To achieve this, an interest based search system using query structuring has been developed. The contribution made by the present work are listed below:

- **A Unified Framework for Interest Based Search System**

In this work, a unified bi-layer interest based search system has been designed which deals with interest of user both at front and back end level. A couple of web mining techniques specific to query processing and page ranking have been developed under data presentation layer. Techniques related to efficient crawling and storage of web pages are developed under data collection layer. The layered architecture of proposed system not only focuses to improve the performance of concerned processes but also opens the provision of adding new functionalities to the system.

- **Personalized Query Assistance over Keyword based Query Suggestion**

The personalized query suggestion technique has been developed without affecting the keyword based query suggestions employed by existing search systems. The technique relies on web resources such as online lexical dictionary to retrieve contextual meaning of query followed by application of user trends obtained by analysis of query log. The resultant personalized queries help to direct user search in right direction.

- **Relevance Improvement**

The result list returned by the search engine in response to user query must contain the links of user interest. User topic of interest can be obtained by capturing its interaction with search engine during the information seeking period. A novel page ranking technique that assigns rank to a page based on its probability of getting selected by the user has been developed. The page probability is computed by capturing the user browsing patterns in terms of relative clicks, time spent and action performed by user on that particular page which is used to provide user specific document retrieval.

- **Volatile Information Updation**

In order to render latest information to the user, web pages are periodically revisited by the crawler. In the present work, an arithmetic progression based crawling mechanism for computing best data collection cycle for each web page based on its changing behavior is developed. The experimental results show that technique is able to maintain significant balance between accumulation of latest information and network load.

- **Quality of Search Result**

A duplicate URL detection and elimination technique is developed to avoid redundant downloading of the same document. This not only avoids the possible overlap between the URL assignments to parallel crawl instances but also reduces unnecessary load on scare resources such as network bandwidth.

- **Optimization of Crawling Process**

While downloading the documents from web, the crawler must be intelligent enough to crawl the important pages first as compared to others. So URL prioritizer module is developed to identify important URLs in URL frontier on the basis of link structure mining. The technique helps in enriching the quality of search database.

- **Balance Assignment of Downloading Task**

To make the crawling process more efficient, the URLs are evenly distributed among asynchronous multi - threaded downloader by URL Scheduler module.

- **Efficient Data Structure**

To support personalized search environment, existing data structures are modified and novel data structures have been employed. For instance, Query log is modified to identify four levels object associations: association between user queries, association between query keywords, association between accessed page, association between query and accessed page. The structure of URL frontier is also modified to handle changing behavior of web page separately. Besides this novel data structures such as profile database and definition repository are employed to generate personalized queries.

- **Efficient Index Structure for Personalized Document Retrieval**

In this work, the existing data structure of search engine database is modified for the sake of better interpretability, efficiency and personalization. The existing database is classified in various domain specific classes such as education, travelling & tourism, food & beverages, Fashion & shopping and sports. Further, each class contains the structural and usage information about downloaded pages which facilitates in query formation as well as result set construction process.

The proposed interest based search system using query structuring have been implemented in C# .Net and SQL Server 2012 and tested over sampled query log and seed URL set. The results obtained thereof are summarized below:

- **Freshness of downloaded collection**
  - ➢ Average precision = 84.9%
  - ➢ Gain in precision over conventional crawler $\simeq$ 31%
  - ➢ Average Recall = 83.4%
  - ➢ Gain in Recall over conventional crawler $\simeq$ 10%
  - ➢ Average F- measure = 79%
  - ➢ Gain in F-measure over conventional crawler $\simeq$ 24%

- **Relevancy of response list**
    - ➢ Average Result Relevancy (RRS) = 75%

- **Query suggestion accuracy**
    - ➢ Average Query Suggestion Relevancy (SRS)=79%

The result analysis indicates that proposed techniques provide up to date user specific information in quick and an efficient manner without increasing load on network.

7.2 FUTURE SCOPE

In this work, various issues related to personalized search system have been addressed. But there is still a scope of improvement in few areas that are worth exploring for providing friendly, seamless and transparent access to large repositories of search engines. The list of some of these issues ranging from existing system improvement to dealing with web resources that are largely ignored by the current search system is given below:

- **Dealing with Hidden Web**

The proposed search system is designed for general web. However the same can be modified to mine hidden web resources such as non HTML pages, image composition, relational tables etc, so as to improve the relevancy of search results in other area.

- **AI & NLP based Result Retrieval**

More advanced personalization techniques using artificial intelligence and natural language processing concepts may be devised to generate more comprehensive query suggestions for better document retrieval.

- **Estimating User Temporal Need**

The work can be extended further to incorporate the methods that can anticipate user’s temporal needs. For instance, during the time of parliamentary elections, user interest sift to information about political parties or during the summer vacations people start browsing about hill stations or during admission time heavy queries about schools , colleges and universities are issued to search systems. So the mechanism that can be trained to pre fetch the information in anticipation with user’s future need, may be devised for better web search experience.

REFERENCES

[1] D. Beeferman and A. Berger. “Agglomerative clustering of a search engine query log” *In Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, USA,* pp 407 – 416, 2000

[2] http://www.networkworld.com

[3] http://www.internetlivestats.com

[4] http://www.worldwidewebsize.com/

[5] https://www.nngroup.com/articles/100-million-websites/

[6] Dirk Livandowarski, “Web searching, search engine and information retrieval” *Information Services.IOS Press,* 2592005, pp 137-147, 2005

[7] A. Terrence, “ Web search: How web has changed information retrieval” Information Research, April 2003

[8] A. Arasu, J. Cho, “Searching the web” *ACM Transaction of Information Technology,* vol. 1,pp2-43, 2001.

[9] D. Christopher, R. Prabhakar H. Schuitze, “An introduction to information retrieval “ *Online Edition Cambridge University Press,* 2009

[10] Bing Liu, “Web data mining: exploring hyperlinks, contents and usage data” *Springer –Verlag,* 2006.

[11] “Availability of the Windows Desktop Search add-in for Files on Microsoft Networks (Revision: 5.0)" *Microsoft Support. Microsoft Corporation.* 2008, Retrieved 2012.

[12] Protalinski, Emil, "Mastering Windows Search using Advanced Query Syntax", *Ars Technica. Condé Nast Digita,.* Retrieved,2011.

[13] Robert Mohns, "Tiger Review: Examining Spotlight", *Macintouch.com.* Retrieved , 2007.

[14] Hidden Gems, “Boolean Spotlight Queries". Retrieved April 1,2012.

[15] David Hawling, “ Challenges in enterprise search” *In the Proceeding of Fifteenth Australasian Conference in Research and Practices in Information Technology,* vol. 27, 2004.

[16] Yunyao Li, Ziyang Liu, “Enterprise search in the big data era: Recent development and open challenges”, *In the Proceeding of 40th International Conference on Very Large Databases, Hangzhou(China)*, vol.7, issue 13, 2014.

[17] P. Dmitri, P. Serdyukov & S. Chernov, “Enterprise and desktop search”, *In Proceeding of 19th International Conference on World Wide Web*, ISBN: 978-1-60558-799-8, pp 1345-1346, 2010 .

[18] https://www.netmarketshare.com/

[19] http://www.searchenginehistory.com/

[20] M.Shyad, E.Mustafa, “A brief history of web s” *Copyright IBM Canada Ltd.* 2013.

[21] D.Shatma, A. Sharmaa, “Search Engine”, *ICT Influences on Human Development Interaction and Collaboration*, pp 117-131, 2012.

[22] J.K.Khan, “Comparative study of information retrieval model used in search engine”, *In Proceeding of IEEE International Conference on Computational Intelligence and Communication Technology*, 2015

[23] http://www.comscore.com/Insights/Press-Releases/2017/9/comScore-Releases-New-Global-Mobile-Report.

[24] J. Pitokow, S. Himrich, C. Todd, C. Robb, R. Dan, “Personalized search” *Communication of ACM*, vol 45 issue 9, 2002.

[25] A. Henmak, P. Sapiezynski, D. Lazer, A. Mislove “Personalization of web search”, *International World Wide Web Committee (IW3C2) ACM*, ISSN 978-1-4503-2035 , pp 2-13.

[26] J. Hu, J. Zang, H. Li, “Prediction based on user browsing history”, *World Wide Web Consortium*, 2007.

[27] B. Horling, M. Kulick,” Personalized search for everyone”, *Google Official Blog*, 2009

[28] B. yu. G. Cai, “A query aware document ranking method for geographic information retrieval” *GIR*, 2007.

[29] J. Teenvan, Sushan. T. Dumais, Eric Horvitz “ Potential for personalization”, *ACM Transaction Compute- Hum. Interact, Microsoft research*, vol. 17, issue 4, 2010.

[30] Wu. M,Turpin. A,Zorbal. J, “An investigation on a community web search variability “, *In Proceeding of an Australian Computer society*, Pg 117-126, 2008.

[31] Silivan Denny, “Of magic keywords and flavours of personalized search at Google”, 2014

[32] B. Symyth, “Adaptive information access: Personalization and privacy” *International journal of pattern recognition and artificial intelligence,* pp 183-205, 2007.

[33] Tin Wai, Lun Dik, “Deriving concept based user profile from search engine logs” *IEEE truncation on Knowledge and Data Engineering*, vol 22 issue 7, Pg 979-982, 2010.

[34] M. Coyal, B. Symth, “Information recovery and discovery in collaborative web search”, *Advances in Information Retrieval Lecture Notes in Computer Science,* pp 356-367, ISBN:, ISBN 540-71494-1, 2007.

[35] https://www.invensis.net/blog/data-processing/12-data-mining-tools-techniques/

[36] O. Heinonen, H. Ahomen, ”Applying data mining techniques for descriptive phrase extraction in digital document collections” *In Advance in Digital Library,* 2010.

[37] R. Feldman &I, Dagan,”Knowledge discovery in textual data”, In Proceeding of *Ist Internation Conference on knowledge Discovery and Data Mining,* pp 112-117, 2005.

[38] D. Bollsus , M. Pazanni, “A hybrid user model for new story classification” *In Proceeding of 7th Internation Conference on User Modeling*, 2009.

[39] D. Mademic,” Text mining and related intelligent agent”, *In Proceeding of International Conference on Intelligent Systems,* Vol 14, Issue 4 pp 44-54, 2009.

[40] H. Wimmer, L.M. Poweell,” A comparison of data source tool for data science” *In Proceeding of International Conference of Information System Applied Science* Vol 8, Issue 3651, pp 1-9, ISSN 2167-1508, 2015.

[41] B. Zupan, J. Demsar,” Open source tools for web mining” Proceeding of *International Conference on Clinic in Lab rotary Medicine,* Vol 28, Isuue 1, pp 37-54, 2008.

[42] W. Paynter & E. Frank, “Domain specific key phrase extraction”, *In Proceeding In Proceeding of 16th International Conference on Artificial Intelligence,* pp 668-673. 2009.

[43] S. Sederland, ”Learning information extraction rules for semi structured and free data”, *In proceeding of International Conference on Machine Learning,* Vol 34, Issue 3, PP 233-272, 2006.

[44] Hamzaoglu & Kargupta, “Distributed data mining using an agent based architecture”, *In Proceeding of International Conference on Knowledge Discovery and Data Mining,* pp 211-214, 2007.

[45] Yang Willkus, “Information extraction as a core language technology”, *Lecture Notes in Computer Science , Springer* , pp 1-9, 2007.

[46] A. Khan, B. Baharudin & L. H. Lee, “A review of machine learning algorithm for text documents classification” *International Journal of Advances in Information Technology,* Vol 1, Issue1, pp 4-17, 2010.

[47] S. Jusoh & S. Osman, “Ambiguity in text mining”, I*n proceeding of IEEE International Conference on Computer and Communication Engineering,* 2008.

[48] H. Kim, S. Chen, “Associative Naïve –Baues Classifier: Automates linking to Gene ontology to machine learning” *International Journal on Pattern Recognition ACM digital Library,* Vol 42, Issue 9, pp 1777-1785, 2009.

[49] Chaung Haung Lee, Hasin Chang Yang. “Construction of supervised and un supervised learning system for multilingual text categorization “ *International Journal of Expert System with Application,* Vol 36, Issue 2, pp 2400-2410, 2009.

[50] B. Yu, Z. Xu, C. Li. “Latent semantic analysis for text categorization using neural network” *International Journal of Knowledge based System,* Vol 21, Isuue 8 pp 900-904, 2008.

[51] W. Jhang, T. Yoshida, “Text classification based on multi word with support vector machine” *International Journal of Knowledge Based System,* pp 879-886, 2008.

[52] R. Malarvizhi and K. Saraswathi.“Web Content Mining Techniques Tools & Algorithms-A Comprehensive Study", *International Journal of Computer Trends and Technology* , Vol 4, 2013.

[53] B. Singh, Hemant Kumar Singh.“Web data mining research: A survey", *In Proceeding of IEEE International Conference on Computational Intelligence and Computing Research (ICCIC),* 2010.

[54] Johnson, Faustina, and Santosh Kumar Gupta. "Web Content Mining Techniques: A Survey”, *International Journal of Computer Applications,* Vol 23, issue 2, 2012.

[55] Deepti Sharda and Sonal Chawla. "Web Content Mining Techniques: A Study." *International Journal of Innovative Research in Technology & Science,* Vol 4, Issue 2, pp 234-241, 2012.

[56] A. Kumar & P. C. Gupta, "Study & analysis of web content mining Ttols to improve techniques of web data mining." *International Journal of Advanced Research in Computer Engineering & Technology,* Vol 9, 2012.

[57] Kavita, Gulshan Shrivastava and Vikas Kumar. "Web mining: Today and tomorrow.", *In Proceeding of IEEE International Conference on Electronics Computer Technology,* Vol 1, pp 387-403, ISSN 978-1-4244-8679, 2011.

[58] Srividya, M., D. Anandhi and M. I. Ahmed. "Web mining and its categories–a survey", pp 2-4, 2013.

[59] C.Singh, "Improving Focused Crawling With Genetic Algorithms" *International Journal of Computer Applications, Published by Foundation of Computer Science, New York, USA ISBN: 973-93-80873- 64-0,* Vol. 66, Issue 4, pp 40-43, 2013.

[60] Mohsen Jamali et. al "A Method for Focused Crawling Using Combination of Link Structure and Content Similarity", *In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence,* 2006.

[61] Zhaoqiong Gao et. al," Incrementally Updating Concept Context Graph (CCG) for Focused Web Crawling Based on FCA", *In Proceeding of Asia-Pacific Conference on Information Processing,* vol. 2, pp 40-43, 2009.

[62] C. Singh "Domain Specific Collection With Genetic Algorithms" *published in Dronacharya Research Journal,* Vol. 4, Issue 1,ISSN 09753389, 2012.

[63] Milad shokouhi, Pirooz Chubak, Zaynab Raeesy," Enhancing focused crawling with genetic algorithms," *Information Technology: Coding and Computing,* Vol. 2, Issue 4 pp 503 - 508, 2005.

[64] Knut magne risvik and Rolf michelsen, "Search engines and web dynamics" *Journal of Computer Networks,* Vol 39, Issue 3, pp 289- 302, 2002.

[65] Marc Najork and Janet Wiener "Breadth-first search crawling Yields High-Quality Pages" *WWW10, Hong Kong,* 2001.

[66] C. Singh, Ramkala. Article: "Web crawling algorithms" page No. 161-165, ID-raictia-10 - 194.

[67] Alessandro Micarelli and Fabio Gasparetti, "Adaptive focused crawling" *Springer-Verlag Berlin LNCS 4321, Heidelberg* pp.231–262, 2007.

[68] MPS Bhatia, Akshi Kumar Khalid, "A primer on the web information retrieval Paradigm", *Journal of Theoretical and Applied Information Technology,* pp 657-662, 2008.

[69] Gautam Pant, Padmini Srinivasan1, and Filippo Menczer, "Crawling the web" *Web Dynamics: Adapting to Change in Content, Size, Topology and Use, Springer-Verlag, Berlin, Germany*, pp 153-178, 2004.

[70] Anshika Pal, Deepak Singh Tomar, S. C. Shrivastava, "Effective focused crawling based on content and link structure analysis", *International Journal of Computer Science and Information Security,* Vol. 2, Issue 1, 2009.

[71] Qu Cheng et. al. "Efficient Focused crawling strategy using combination of link structure and content similarity" *In Proceedings of IEEE International Symposium on IT in Medicine and Education,* vol.2, pp.797 – 802, 2003.

[72] Najork, M. and Wiener, J. L, "Breadth-first search crawling yields high-quality pages", *In 10th International World Wide Web Conference,* pp 114-118.

[73] A.K.Sharma, J.P. Gupta, " Augmented hypertext documents suitable for Parallel crawler", *In Proceeding of workshop on Information Technology Servicesand Application,* WITSA 2003.

[74] http://www.dmoz.org/

[75] http://people.scs.carleton.ca/~armyunis/projects/KAPI/porter.pdf

[76] Silviu Cueerzan, Ryen W, " Query suggestion based on user landing pages" *In Proceeding of international ACM SIGR Conference of Research and Development in Information Retrieval,* ISBN:9781-1-59593-597-7.

[77] Wenpu Xing and Ali Ghorbani, "Weighted page rank algorithm" *In Proceedings of the Second Annual Conference on Communication Networks and Services Research,* (CNSR'04)0-7695-2096-0, 2004.

[78] Page, S. Brin, R. Motwani, and T. Winograd, "The Pagerank Citation Ranking: Bringing order to the Web". *Technical report,Stanford Digital Libraries* SIDL-WP-1999-0120, 1999.

[79] Gyanendra Kumar, Neelam Duahn, and Sharma A. K., "Page ranking based on number of visits of web pages", *International Conference on Computer & Communication Technology (ICCCT),* 978-1-4577-1385-9 , 2011.

[80] Zdravko Markov and Daniel T. Larose, "Mining the web: Uncovering patterns in web content, structure, and usage data". *Copyright 2007 John Wiley & Sons, Inc*

[81] A. Dixit , A. Sharma " A mathematical model for crawler revisit frequency", *In Proceeding of IEEE 2nd international Advanced computing conference(IACC)* pp 316-319.

[82] Jibran Mustafa, Sharifullah Khan, Khalid Latif, "Ontology based semantic information retrieval," *In Proceeding of 4th International IEEE Conference Intelligent Systems,* pp. 22-14, 2008.

[83] Ivan Marcialis and Emanuela De Vita, "SEARCHY: An agent to Personalize Search results", *In Proceeding 3rd International Conference on Internet and Web Applications and Services,* 2010.

[84] Ming-YenChen,Hui-ChuanChu,Yuh-Mi Chen, "Developing a semantic–enable information retrieval mechanism" *Expert system with Applications,* Vol 37, pp 322-240, 2010.

[85] Veningston .K, Dr. R. Shanmugalakshmi, "Enhancing personalized web search re-ranking algorithm by incorporating user profile", 2009.

[86] Adil.Siddiqui, Sudheer Singh, "URL ordering based performance evaluation of web crawler", *International Journal of Computer and Information Technology,* Vol 4 Issue 1, 2011.

[87] Nicolaas Flip Radinski, "Personalizing web search using long term history", *In Proceeding of WSDM'11 ACM Conference,* ISSN 978-1 4503-0493, 2011.

[88] Jeehym Kim, Quian Gao & Yong In Cho, "A comtext aware based dynamic user preference profile construction model" *International Joural of Advanced Engineering and Global Technolog* , Vol 1 Issue 4, ISSN 2309-4893, 2011.

[89] Hao Wu, Guoliang Li (2014), "GNIX: Genrealised inverted index for keyword search", Vol 18 Issue 1 ISSN 1007-0214 10, 2012.

[90] http://assets.cengage.com/pdf/smp_4444_DC02_Fin.pdf

[91] Jensen. B. J, Spink. A, "An analysis of web searching by european althecom user", *In Proceeding of Information Management Conference,* Vol 41, Issue 2, pp 361-381, 2005.

[92] A. Arasu J Cho and Molina. "Searching the web", *ACM Transactions on Internet Technology,* Vol 1. , pp 2-43, 2001.

[93] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, "Query expansion by mining user logs," *IEEE Transaction on Knowledge Data Engineering.*, Vol 15, Issue 4, pp. 829-839, 2003.

[94] J.Cho and H. Garcia-Molina, "Parallel crawlers", *In Proceedings of 11th International Conference on World Wide Web- WWW,* Vol 2, 2002.

[95] J. Cho and H. Garcia-Molina, "The evolution of the web and implications for an incremental crawler", *In Proceeding of the 26th International Conference on Very Large Database,* 2000.

[96] S. Gupta, K. Bhatia, "CrawlPart: Creating crawl partitions in parallel crawler" *In Proceeding of IEEE Conference on Computational and Business Intelligence,* 2014.

[97] N. Singhal, A. Dixit, R.P Agarwal, A.K Sharma, "Using migrating agents in designing web search engines and property analysis of available platforms". *International Journal of Advancement in Technology (IJOAT)*, Vol. 3, Issue 4, pp.254-269, ISSN 0976-4860, 2012.

[98] S. Raghavan and H. Gareia Molina, "Crawling the hidden web", *VLDB conference*, 2001.

[99] O. Papapetrou, S. Papastavrou and G. Samaras, "UCYMICRA: Distributed indexing of the web using migrating crawlers." *Advances in Databases and Information System Lecture Notes in Computer Science*, pp 133-147, 2003.

[100] S Lawerence, "Searching the World Wide Web", vol 280 Issue 5360, pp-98-100, 1998.

[101] B.E Brewington and G Cybenko, "How dynamic is the Web?.", *Computer Networks*, vol 33 Issue 6 pp 257-276, 2000

[102] B. Grossan. "Search Engines: What they are, how they are, how they work, and practical suggestion for getting the most out of them.", February 1997.

[103] A. Dixit, N Singhal, "Need of Search Engines and Role of a Web Crawler", *National Conference on Recent Trends in Computer and Information Technologies Century (RTCIT- 2009) Panipat, Haryana, India*, April 2009.

[104] M. Hersovici, M. Jacovi , Y.S Maarek , D.Pelleg, M. Shtaliama and S. Ur " The shark search algorithm", *In Proceeding of 3rd International Conference on Computer Networks* , Vol 30, Issue 7, pp 317-326, 1998.

[105] J.M Kleinberg, "Hubs, authorities and communities," ACM Computing Surveys, Vol 31 Issue 4, 1998.

[106] J.C Miller, G Rae, F. Schaefer, L.A Ward, T. Lofaro and A. Farahat "Modifications of kleinbergs HITS algorithm using matrix exponentiation and web log records", *In Proceeding of the 24th Annual International ACM SIGR Conference on Resarch and Development in Information Retrieval- SIGIR,* Vol 01, 2001.

[107] S. Chakrabati, M. V. Berg and B. Dorn, "Focused crawling : a new approach to topic-specific web resource discovery.", *Computer Networks,* Vol. 31, Issue 11, pp. 1623-1640, 1999.

[108] K. Khan, U. Rahman, “DBSCAN: Past, present and future”, *In the Proceeding of IEEE Conference on Application of Digital Information anf Web Technology*, 2014.

[109] A Guerrico , F Ragni and C. Martines, “ A dynamic URL assignment method for parallel web crawler”, *In Proceeding of IEEE International Conference on Computational Intelligence for Measurement System and Application*, 2010.

[110] L. Leitlo, P. Calado and M. Herschel, “Efficient and Effective Duplicate Detection in Hierarchical Data”, *IEEE Transaction on knowledge and Data Engineering* , vol 25, Issue 5, pp 101-10281,2013.

[111] H, L. Yu, L Bingwa and Y. Fang, “Similarity Computation of Web Pages of Focused Crawler”, *In Proceeding of International forum on Information technology and Applications,* 2010.

[112] C. Y Kang, “DOM based web pages to determine the structure of the similarity algorithm”, *In proceeding of 3^rd^ International Symposium on Inteligent Information Technology Application*, 2009.

[113] Nierman and H. V Jagdish. “ Evaluating Structural Similarity in XML documents”, *In Proceeding of 5th International Workshop on the Web and Database(Web DB2002) Madison Wisconsin, USA*, June 2002.

[114] E Bertino G Gueerini and Mesiti, “A matching algorithm for measuring the structural similarity between an XML document and a DTD and its application”, *In Proceeding of International Conference in Information System,* Vol 29 Issue 1. pp 23-46, 2004.

[115] M morita and Y. Shinoda, “Information filtering based on user behavior analysis and best match text retrieval.” *Springer Publication*, Vol 94, pp 272-281, 2004.

[116] M. Claypool , P. Le M Wased and D. Brown, “ Implicit interest indicators”, *In Proceedings of the 15th International Conference on World Wide Web WWW 06*, 2006.

[117] J. Goeks and J. Shavlik, “Learning users interest by unobtrusively observing their normal behavior”, *In Proceeding of the 5th International Conference on Intelligent User Interface- IUI00,* 2000.

[118] X. Ying, “The research on user mmodeling for internet personalized services”, *National University Of Defenvce Technology*, 2003.

[119] Z Jingling , X . Wang And Y. Zhou, “Study and implementations of user behaviour analysis,” *In Proceeding of 12th IEEE International Conference on Advance Communication Technology (ICACT),* Vol 1, 2010.

[120] Y Qinhong Hao and X Neng, “The research on user interest model based on quantization browsing behavior”, *International Conference on Computer Science & Education (ICCSE)*, 2012.

[121] A.K Sharma J.P Gupta, D.P Agarwal, “PARCHYD: An architecture of a parallel crawler based on augmented hypertext documents”, *Ph.D Thesis, HIT & M, Gwalior*, 2003

[122] X.Wang, M.Yang,S.Li, “Incremental clustering of search history in personalized search”, *International Journal of Computational Systems* , pp 2285-2292, 2013.

[123] Q. Mei, D. Zhou, and K. Church, “Query Suggestion Using Hitting Time,” CIKM ’08: Proc. 17th ACM Conf. Information and Knowledge Management, pp. 469-477, 2008.

[124] A. Sharma, N. Duhan, “Web search result optimization by mining search engine query log”, *International Conference on Methods and Models in Computer Science* pp 40-45, 2010

[125] Chandel, G. S., Patidar, K., & Mali, M. S, “A result evolution approach for web using mining using fuzzy c-mean clustering algorithm” *International Journal of Computer Science and Network Secrity (IJCSNS),* Vol 16, Issue 1, pp 135-145, 2012.

[126] Yuan Hong, Jaideep, Vaidya and Haibing Lu Rutgers University “Search engine query clustering using top - K search results”, *In Proceeding of International Conferences on Web Intelligence and Intelligent Agent Technology,* 2011.

[127] A. Ladekar, P. Pawar, D. Raikar, and J. Chaudhari, “ Web log based analysis of user’s browsing behaviour”, *International Journal of Computer Applications,* vol 115, Issue 11, pp, 5-8, 2015.

[128] D. Anupama and S. D. Gowda, “Clustering of web user sessions to maintain occurrence of sequence in navigation pattern,” *In Proceeding of International conference of Computer Science*, Vol. 58, pp. 558-564, 2015.

[129] K. Filipowski, “Comparison of Scheduling Algorithms for Domain Specific Web Crawler,” *2014 European Network Intelligence Conference*, 2014.

[130] W. R. Bhaginath, S. Shingade, and M. Shirole, “Virtualized dynamic URL assignment web crawling model”, *In Proceeding of International Conference on Advances in Engineering & Technology Research (ICAETR – 2014),* Feb-2014.

[131] G. H. Agre and N. V. Mahajan, “Keyword focused web crawler,” *In Proceeding of 2nd International Conference on Electronics and Communication Systems (ICECS),* 2015.

[132] M. Kumar and R. Bhatia, “Design of a mobile Web crawler for hidden Web,” *In Proceeding of 3rd International Conference on Recent Advances in Information Technology(RAIT),* 2016.

[133] P. Devi, A. Gupta, A.Dixit , “Comparative study of HITS and PageRank link based ranking algorithms”, *International Journal of Advanced Research in Computer and Communication Engineering* , Vol 3,Issue 5, ISSN 2278-1021, Feb-2014.

[134] B. Mehta and M. Narvekar, “DOM tree based approach or Web content extraction”, *In Proceeding of International conference on Communication Information & Computing Technology (ICCICT),* 2015.

[135] A. Dixit and A. K. Sharma, “Self adjusting refresh time based architecture for incremental web crawler”, *International Journal of Computer Science and Network Security (IJCSNS) Korea,* Vol 8 Issue 12 pp 349-354, ISSN:1738-7906, 2008.

[136] A. Surya and D. K. Sharma, “An approach for web page ordering using user session”, *In Proceeding of IEEE Conference on Information And Communication Technologies*,2013.

[137] S. Niwattankul &J. Singthongchai, “Using Jaccard coefficient for keyword similarity” *In Proceeding of International Multi Conference of Engineers and Computer Scientist,* Vol 1, ISSN 978-988-19251-8-3 (2013).

[138] Z-K Wei and J-p Du “Research on several key issues about Search Engines” *In Proceeding of International Conference on Machine Learning and Cybermetics,* 2008.

[139] V. Thanda & V. Jaglan, “Comparison of jaccard, dice,cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm”, *International Journal of Innovation in Engineering and Technology,* Vol 2, Issue 4 pp 202-205, ISSN 2319-1058, 2013.

[140] S. Mizuno & T. Yamaguchi, “Overlap coefficient for accessing the similarity of pharmacokinetic data between ethnically different population” *SAGE Journal,* Vol 2, Issue 2, pp 174-181, 2005.

[141] S. Brim, L. Page, “The page rank citation ranking: Bringing orders to the web”, *Technical Report Sandford Dig Libraries SIDLWP,* 1999-0120.

[142] https://www.intmath.com/applications-integration/5-centroid-area.php

[143] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna,“The query-flow graph: Model and applications,” *In Proceeding of 17thACM Conference on Information and Knowledge Management (CIKM),* 2008

[144] J.Wen, J.Nie, and H.Zhang. “Query clustering using user logs”. *ACM Transactions on Information Systems*, Vol 1, pp 59– 81, 2002.

[145] J.Wen, J.Nie, and H.Zhang. “Clustering user queries of a search engine”, *In Proceedings of the 10th International World Wide Web Conference, Hong-Kong, China*, pp 162–168, 2001.

APPENDIX-A

A fragment of Query Log containing adequate amount of queries belonging to five different domains issued by different users are given in Table A 1.1

Table A 2.1: Query Log for Analysis

Host	User _Query	Date_Time	Request	User_Agent	Clicked URL (Base Address)	Domain Class
172.20.10.10	Samsung mobile phone	11/20/2016 23:06	*GET/gadgets.ndtv.com/mobiles/samsung-phones*	Google Chrome	gadgets.ndtv.com	Shopping
172.20.10.10	Motorola mobile phone	11/20/2016 23:06	*GET/gadgets.ndtv.com/mobiles/motorola-phones*	Google Chrome	gadgets.ndtv.com	Shopping
172.20.10.10	Iphone Price List	11/20/2016 23:06	*GET/http://www.mysmartprice.com/mobile/pricelist/apple-mobile-price-list-in-india.html*	Google Chrome	www.mysmartprice.com	Shopping
172.20.10.10	Oppo Price List	11/20/2016 23:07	*GET/http://www.mysmartprice.com/mobile/pricelist/oppo-mobile-price-list-in-india.html*	Google Chrome	www.mysmartprice.com	Shopping
172.20.10.10	Apple vs Samsung	11/20/2016 23:11	*GET/https://en.wikipedia.org/wiki/Apple_Inc._v._Samsung_Electronics_Co.*	Google Chrome	en.wikipedia.org	Shopping
172.20.10.10	Oppo vs Samsung	11/20/2016 23:13	*GET/https://gadgets.ndtv.com/samsung-galaxy-j7-max-4216-vs-oppo-f3-4141-vs-vivo-v5s-4131*	Google Chrome	gadgets.ndtv.com	Shopping
172.20.10.10	Samsung j7 camera	11/20/2016 23:15	*GET/https://www.gsmarena.com/samsung_galaxy_j7_2016-review-1632p8.php*	Google Chrome	www.gsmarena.com	Shopping
172.20.10.10	iphone 6 camera	11/20/2016 23:16	*GET/http://www.trustedreviews.com/reviews/iphone-6-camera-page-5*	Google Chrome	www.trustedreviews.com	Shopping
172.20.10.10	Android os features	11/20/2016 23:31	*GET/https://en.wikipedia.org/wiki/List_of_features_in_Android*	Google Chrome	en.wikipedia.org	Shopping
172.20.10.10	iphone os features	11/20/2016 23:31	*GET/https://www.theguardian.com/technology/2016/sep/19/ios-11*	Google Chrome	www.theguardian.com	Shopping
172.20.10.10	Anroid vs Ios	11/20/2016 23:49	*GET/https://www.diffen.com/difference/Android_vs_iOS*	Google Chrome	www.diffen.com	Shopping
172.20.10.10	Ios vs Windows	11/20/2016 23:50	*GET/https://www.trustedreviews.com/opinion/which-mobile-operating-system-is-best-2928049*	Google Chrome	www.trustedreviews.com	Shopping
172.20.10.10	Anroid vs Windows	11/20/2016 23:50	*GET/https://www.trustedreviews.com/opinion/which-mobile-operating-system-is-best-2928049*	Google Chrome	www.trustedreviews.com	Shopping
172.20.10.10	Anroid vs Bada	11/20/2016 23:50	*GET/https://androidqueries.com/comparison-between-android-bada-operating-system-1481.html*	Google Chrome	www.androidqueries.com	Shopping
172.20.10.10	Mobile phone amazon	11/21/2016 0:01	*GET/https://www.amazon.in/Mi-4-Redmi-Gold-32GB*	Google Chrome	www.amazon.in	Shopping
172.20.1	redmi4 gold	11/21/2016	*GET/https://www.amazon.in/Redmi	Google	www.amazo	Shopp

0.10	on amazon	0:02	-4-Gold-32*	Chrome	n.in	ing
172.20.10.10	Amazon Phone sale	11/21/2016 0:02	*GET/https://www.amazon.in/mobile-phones/b?ie=UTF8&amp;node=1389401031*	Google Chrome	www.amazon.in	Shopping
172.20.10.10	Flipkart Phone sale	11/21/2016 0:02	*GET/https://www.flipkart.com/mobile-phones-store*	Google Chrome	www.flipkart.com	Shopping
172.20.10.10	Iphone 6s price on amazon	11/21/2016 0:02	*GET/https://www.amazon.in/Apple-iPhone-Space-Grey-32GB/dp/B01LX3A7CC*	Google Chrome	www.amazon.in	Shopping
172.20.10.10	Iphone 6s price on paytm	11/21/2016 0:02	*GET/https://paytm.com/offer/iPhone6s/*	Google Chrome	paytm.com	Shopping
172.20.10.10	Dell laptop	11/21/2016 14:42	*GET/http://www.dell.com/in/business/p/laptops*	Google Chrome	/www.dell.com	Shopping
172.20.10.10	Hp laptop	11/21/2016 14:42	*GET/https://www.hpshopping.in/laptops-tablets.html*	Google Chrome	www.hpshopping.in	Shopping
172.20.10.10	Dell price	11/21/2016 14:42	*GET/https://www.google.co.in/search?q=dell+price&amp;oq=dell+price+&amp;aqs=chrome..69i57j0l5.3510j0j7&amp;sourceid=chrome&amp;ie=UTF-8*	Google Chrome	https://www.google.co.in/	Shopping
172.20.10.10	Acer Price	11/21/2016 14:43	*GEThttps://www.smartprix.com/laptops/acer-brand*	Google Chrome	https://www.smartprix.com	Shopping
172.20.10.10	Laptop	11/21/2016 14:43	*GET/https://www.flipkart.com/laptops/pr*	Google Chrome	www.flipkart.com	Shopping
172.20.10.10	acer laptop on flipkart	11/21/2016 14:43	*GET/https://www.flipkart.com/laptops/acer~brand/pr?sid=6bo,b5g*	Google Chrome	www.flipkart.com	Shopping
172.20.10.10	dell laptop configuration	11/21/2016 14:43	*GET/http://www.dell.com/in/p/laptops*	Google Chrome	www.dell.com	Shopping
172.20.10.10	Mac configuration	11/21/2016 14:43	*GET/https://www.apple.com/in/macbook-pro/specs/*	Google Chrome	www.apple.com	Shopping
172.20.10.10	dell vs mac	11/21/2016 14:44	*GET/https://www.pcworld.com/article/3179677/computers/dell-xps-15-vs-macbook-pro-15-fight.html*	Google Chrome	www.pcworld.com	Shopping
172.20.10.10	dell vs hp	11/21/2016 14:44	*GET/https://www.quora.com/Which-laptop-I-should-buy-Dell-Lenovo-or-HP*	Google Chrome	www.quora.com	Shopping
172.20.10.10	hp vs acer	11/21/2016 14:44	*GET/http://www.tomsguide.com/answers/id-2945609/choice-acer.html*	Google Chrome	www.tomsguide.com	Shopping
172.20.10.10	Acer vs lenovo	11/21/2016 14:44	*GET/https://www.techpowerup.com/forums/threads/acer-or-lenovo-laptops-both-have-same-specifications.221299/*	Google Chrome	www.techpowerup.com	Shopping
172.20.10.10	latest processor in laptop	11/21/2016 14:52	*GET/https://www.laptopmag.com/articles/cpu-comparison*	Google Chrome	www.laptopmag.com	Shopping
172.20.10.10	latest processor in mobile	11/21/2016 14:52	*GET/http://www.samsung.com/semiconductor/minisite/Exynos/?gclid*	Google Chrome	www.samsung.com	Shopping
172.20.10.10	dell i3 6th gen	11/21/2016 14:52	*GET/https://www.flipkart.com/dell-inspiron-core-i3-6th-gen-4-gb-1-tb-hdd-windows-10-home-5559-	Google Chrome	www.flipkart.com	Shopping

			laptop/p/itmenugrzrxmzvws*			
172.20.10.10	Ultrathin	11/21/2016 14:59	*GET/https://www.amazon.in/s/?ie=UTF8&amp;keywords=ultra+thin*	Google Chrome	www.amazon.in	Shopping
172.20.10.10	Ultrabook	11/21/2016 15:00	*GET/http://www.techradar.com/news/mobile-computing/laptops/best-ultrabook-18-top-thin-and-lights-1054355*	Google Chrome	www.techradar.com	Shopping
172.20.10.10	dell i3 7th gen	11/21/2016 14:59	*GET/https://www.flipkart.com/dell-inspiron-5000-core-i3-7th-gen-4-gb-1-tb-hdd-windows-10-home-5378-2-1-laptop/p/itmestksraepm6xc*	Google Chrome	www.flipkart.com	Shopping
172.20.10.10	current best laptop in india	11/21/2016 15:00	*GET/https://www.digit.in/top-products/top-10-laptops-5.html*	Google Chrome	www.digit.in	Shopping
172.20.10.10	current best laptop in America	11/21/2016 15:00	*GET/https://www.stuff.tv/top-10/laptops*	Google Chrome	www.stuff.tv	Shopping
172.20.10.10	Digital camera	11/21/2016 15:23	*GET/https://www.bestbuy.com/site/cameras-camcorders/digital-cameras/*	Google Chrome	www.bestbuy.com	Shopping
172.20.10.10	SLR camera	11/21/2016 15:23	*GET/https://en.wikipedia.org/wiki/Singlelens_reflex_camera*	Google Chrome	en.wikipedia.org	Shopping
172.20.10.10	DSLR camera	11/21/2016 15:23	*GET/http://www.fujifilmusa.com/products/digital_cameras/index.html*	Google Chrome	www.fujifilmusa.com	Shopping
172.20.10.10	DSLR lens	11/21/2016 15:23	*GET/https://www.target.com/s/1%20dslr%20lens?ref=tgt_adv_XS000000&amp;AFID=google&amp;fndsrc=tgtao&amp;CPNG*	Google Chrome	www.target.com	Shopping
172.20.10.10	SLR lens price	11/21/2016 15:24	*GET/https://www.walmart.com/browse/camera-accessories/all-camera-lenses/3944_133277_132913_1079107?adid*	Google Chrome	www.walmart.com	Shopping
172.20.10.10	Canon Digital Camera	11/21/2016 15:29	*GET/https://www.officedepot.com/a/browse/canon/N=5+509515&amp;cbxRefine*	Google Chrome	www.officedepot.com	Shopping
172.20.10.10	Nikon Digital Camera	11/21/2016 15:30	*GET/https://www.nikonusa.com/en/index.page?cid=img_en_us:SEM:EC:Rise:Ongoing:Google:P*	Google Chrome	www.nikonusa.com	Shopping
172.20.10.10	Average Digital camera weight	11/21/2016 15:33	*GET/http://www.red.com/?utm_source=google&amp;utm_medium=adwords&amp;utm_campaign=RED_Brand_AdWords&amp;gclid*	Google Chrome	www.red.com	Shopping
172.20.10.10	Camera weight calculator	11/21/2016 15:35	*GET/http://cameraweight.com/*	Google Chrome	cameraweight.com	Shopping
172.20.10.10	Nikon handicam	11/21/2016 15:40	*GET/https://www.nikonusa.com/en/index.page?cid=img_en_us:SEM:EC:Rise:Ongoing:Google:P*	Google Chrome	www.nikonusa.com	Shopping
172.20.10.10	Sony handicam	11/21/2016 15:41	*GET/https://www.bestbuy.com/site/searchpage.jsp?_dyncharset=UTF-8&amp;ks=960&amp;sc=Global&amp;list=y&amp;usc*	Google Chrome	www.bestbuy.com	Shopping
172.20.10.10	Best resolution camera	11/21/2016 15:44	*GET/http://www.backgroundexposure.com/blog/2007/02/film-vs-digital/?gclid*	Google Chrome	www.backgroundexposure.com	Shopping
172.20.1	Best resolution	11/21/2016	*GET/http://www.red.com/products/	Google	www.red.co	Shopp

0.10	camera phone	15:45	weapon?utm_source=google&amp;utm_medium=cpc&amp;utm_campaign*	Chrome	m	ing
172.20.10.10	Digital camera on amazon	11/21/2016 15:57	*GET/https://www.amazon.com/gp/product/B01C4UY0JK?tag=googhydr-20&amp;hvadid*	Google Chrome	www.amazon.com	Shopping
172.20.10.10	Digital camera on flipkart	11/21/2016 15:58	*GET/https://www.flipkart.com/cameras*	Google Chrome	www.flipkart.com	Shopping
172.20.10.10	camera dslr	11/21/2016 16:02	*GET/https://slrhut.co.uk/search/?q=slr+cameras&amp;gclid*	Google Chrome	slrhut.co.uk	Shopping
172.20.10.10	mobile camera vs dslr	11/21/2016 16:03	*GET/https://www.marchnetworks.com/products/cameras/mobile-cameras/?utm_source*	Google Chrome	www.marchnetworks.com	Shopping
172.20.10.10	mobile lens camera	11/21/2016 16:05	*GET/http://www.bandpro.com/blog/raptor/?gclid*	Google Chrome	www.bandpro.com	Shopping
172.20.10.10	apple camera	11/21/2016 16:06	*GET/https://www.dvwarehouse.com/Used-Macs-c-253_53.html?gclid*	Google Chrome	www.dvwarehouse.com	Shopping
172.20.10.10	kodak camera	11/21/2016 16:07	*GET/https://www.officedepot.com/a/browse/digital-cameras/N*	Google Chrome	www.officedepot.com	Shopping
172.20.10.10	Digtial Watch	11/22/2016 1:06	*GET/https://www.amazon.in/Watches-Digital/s?ie=UTF8&amp;page=1&amp;rh=n%3A1350387031%2Cp_n_feature_seven_browse-bin%3A1480901031*	Google Chrome	www.amazon.in	Shopping
172.20.10.10	Apple Watch	11/22/2016 1:06	*GET/https://www.apple.com/in/watch/*	Google Chrome	www.apple.com	Shopping
172.20.10.10	Apple Watch	11/22/2016 1:06	*GET/https://www.amazon.in/s/?ie=UTF8&amp;keywords=watches+iphone*	Google Chrome	/www.amazon.in	Shopping
172.20.10.10	Apple Watch	11/22/2016 1:06	*GET/https://www.flipkart.com/watches/pr*	Google Chrome	www.flipkart.com	Shopping
172.20.10.10	Apple Watch vs Samsung Gear	11/22/2016 1:06	*GET/https://www.gadgetsnow.com/compare-smartwatch/Apple-Watch-Series-2-vs-Samsung-Gear-S3-Classic*	Google Chrome	www.officedepot.com	Shopping
172.20.10.10	Fitness Watch	11/22/2016 1:06	*GET/https://www.menshealth.com/fitness/best-fitness-watches-track-workouts*	Google Chrome	www.menshealth.com	Shopping
172.20.10.10	Apple Watch Water Resistent	11/22/2016 1:07	*GET/https://support.apple.com/en-in/HT205000*	Google Chrome	support.apple.com	Shopping
172.20.10.10	Mi band	11/22/2016 1:07	*GET/http://www.mi.com/in/miband/*	Google Chrome	www.mi.com	Shopping
172.20.10.10	Samsung band	11/22/2016 1:07	*GET/http://www.samsung.com/global/galaxy/gear-fit2/*	Google Chrome	www.samsung.com	Shopping
172.20.10.10	Digital band	11/22/2016 1:07	*GET/https://www.google.co.in/search?q=digital+abnd&amp;oq=digital+abnd&amp;aqs=chrome..69i57j0l5.5521j0j7&amp;sourceid=chrome&amp;ie=UTF-8*	Google Chrome	https://www.google.co.in/	Shopping
172.20.10.10	Smart tv	11/22/2016 1:17	*GET/https://www.amazon.in/smart-tvs/b?ie=UTF8&amp;node=7198570	Google Chrome	www.amazon.in	Shopping

			031*			
172.20.10.10	Samsung tv	11/22/2016 1:18	*GET/http://www.samsung.com/in/tvs/?cid=in_ppc_google_ce_tv-aos-rest-led-tv-brand_samsung-tv_20160907*	Google Chrome	www.samsung.com	Shopping
172.20.10.10	Sony tv	11/22/2016 1:18	*GET/https://www.amazon.in/s/ref=nb_sb_noss?url=node%3D5903486031&amp;field-keywords=&amp;tag=googinkenshoo-21&amp;ascsubtag=c6313b01-9a5c-400c-a3c9-d131d9cae836*	Google Chrome	/www.amazon.in	Shopping
172.20.10.10	CRT tv	11/22/2016 1:18	*GEThttps://www.amazon.in/s/ref=nb_sb_noss?url=node%3D1389396031&amp;field-keywords=&amp;tag=googinkenshoo-21&amp;ascsubtag=c6313b01-9a5c-400c-a3c9-d131d9cae836*	Google Chrome	www.amazon.com	Shopping
172.20.10.10	Led tv	11/22/2016 1:18	*GET/https://www.amazon.in/TVs/b/ref=nav_shopall_sbc_tvelec_television?ie=UTF8&amp;node=1389396031&amp;tag=googinkenshoo-21&amp;ascsubtag*	Google Chrome	www.amazon.com	Shopping
172.20.10.10	lcd	11/22/2016 1:18	*GET/https://www.amazon.in/s/?ie=UTF8&amp;keywords=television+lcd+tv&amp;tag=googinhydr1-21&amp;index=aps&amp;hvadid=213894697094*	Google Chrome	www.amazon.com	Shopping
172.20.10.10	4k tv	11/22/2016 1:18	*GET/https://www.sony.co.in/electronics/bravia-4k-hdr-tv?cid=sem-am-1652*	Google Chrome	www.sony.co.in	Shopping
172.20.10.10	Curved tv	11/22/2016 1:18	*GET/https://www.gozefo.com/ncr/category/TV?ref=ad&amp;cbanner=aplcondition&amp;cname%3DTV+Category+-+Condition+-+APL*	Google Chrome	www.gozefo.com	Shopping
172.20.10.10	TV Price	11/22/2016 1:19	*GET/https://www.amazon.in/s/ref=nb_sb_noss?url=node%3D1389396031&amp;field-keywords=&amp;tag=googinkenshoo-21&amp;ascsubtag=c6313b01-9a5c-400c-a3c9-d131d9cae836*	Google Chrome	www.amazon.in	Shopping
172.20.10.10	TV	11/22/2016 1:20	*GET/https://www.google.co.in/search?q=digital+abnd&amp;oq=digital+abnd&amp;aqs=chrome..69i57j0l5.5521j0j7&amp;sourceid=chrome&amp;ie=UTF-8*	Google Chrome	https://www.google.co.in/	Shopping
172.20.10.10	Tv at amazon	11/22/2016 1:20	*GET/https://www.google.co.in/search?q=digital+abnd&amp;oq=digital+abnd&amp;aqs=chrome..69i57j0l5.5521j0j7&amp;sourceid=chrome&amp;ie=UTF-8*	Google Chrome	https://www.google.co.in/	Shopping
192.168.43.167	Baby food	11/22/2016	*GET/https://www.amazon.in/Baby-Food/b?ie=UTF8&node=1953449031*	Mozilla Firefox	https://www.amazon.in/	Food
192.168.43.167	Nestle Baby food	11/22/2016 1:21	*GET/https://www.amazon.com/Baby-Foods-Nestle-Feeding/s?ie=UTF8&page=1&rh=n%3A16323111%2Cp_4%3ANestle*	Mozilla Firefox	https://www.amazon.in/	Food
192.168.43.167	burger pizza	11/22/2016 1:21	*GET/https://www.dominos.co.in/menu/burger-pizza*	Mozilla Firefox	https://dominos.co.in/	Food

192.168.43.167	dominos pizza	11/22/2016 1:22	*GET/https://pizzaonline.dominos.co.in/menu/D/Faridabad/Sector%206/66142*	Mozilla Firefox	https://pizzaonline.dominos.co.in/	Food
192.168.43.167	Pizza hut pizza	11/22/2016 1:20	*GET/https://online.pizzahut.co.in/product/pizza*	Mozilla Firefox	https://online.pizzahut.co.in/	Food
192.168.43.167	burger	11/22/2016 1:20	*GET/http://www.bk.com/menu/burgers*	Mozilla Firefox	http://www.bk.com/	Food
192.168.43.167	veg momos	11/23/2016 1:20	*GET/http://food.ndtv.com/recipe-vegetable-momos-99111*	Mozilla Firefox	http://food.ndtv.com/	Food
192.168.43.167	chicken momos	11/24/2016 1:20	*GET/http://www.bawarchi.com/recipe/chicken-momo-oesvbhiaibbdi.html*	Mozilla Firefox	http://www.bawarchi.com/	Food
192.168.43.167	fried momos	11/25/2016 1:20	*GET/http://nishamadhulika.com/1452-veg-fried-momo-recipe.html*	Mozilla Firefox	http://nishamadhulika.com/	Food
192.168.43.167	fried chicken momos	11/26/2016 1:220:13 AM	*GET/http://spicyworld.in/chicken-momo.html*	Mozilla Firefox	http://spicyworld.in/	Food
192.168.43.167	chicken recipes	11/27/2016 1:20	*GET/https://recipes.timesofindia.com/recipes/masala-chicken/rs54673639.cms*	Mozilla Firefox	https://recipes.timesofindia.com/	Food
192.168.43.167	chicken biryani	11/28/2016 2:20	*GET/https://indianhealthyrecipes.com/chicken-biryani-in-pressure-cooker/*	Mozilla Firefox	https://indianhealthyrecipes.com/	Food
192.168.43.167	chicken soup	11/29/2016 1:20	*GET/http://allrecipes.com/recipe/8814/homemade-chicken-soup/*	Mozilla Firefox	http://allrecipes.com/	Food
192.168.43.167	chicken noodle soup	11/30/2016 4:23	*GET/http://allrecipes.com/recipe/26460/quick-and-easy-chicken-noodle-soup/*	Mozilla Firefox	http://allrecipes.com/	Food
192.168.43.167	chicken kari	12/01/2016 03:34	*GET/http://allrecipes.com/recipe/212721/indian-chicken-curry-murgh-kari/*	Mozilla Firefox	http://allrecipes.com/	Food
192.168.43.167	chicken tandoori	12/02/2016 04:20	*GET/https://www.youtube.com/watch?v=-CKvt1KNU74*	Mozilla Firefox	https://www.youtube.com/	Food
192.168.43.167	chicken kebabs	12/03/2016 01:45	*GET/https://indianhealthyrecipes.com/chicken-kebab-recipe-chicken-kabab/*	Mozilla Firefox	https://indianhealthyrecipes.com/	Food
192.168.43.167	Cake	12/04/2016 03:45	*GET/https://www.bbcgoodfood.com/recipes/category/cakes-baking*	Mozilla Firefox	https://www.bbcgoodfood.com/	Food
192.168.43.167	chocolate cake	12/05/2016 03:56	*GET/https://www.bbc.co.uk/food/recipes/easy_chocolate_cake_31070*	Mozilla Firefox	https://www.bbc.co.uk/	Food
192.168.43.167	fruit cake	12/06/2016 04:20	*GET/http://www.bbc.co.uk/food/fruit_cake*	Mozilla Firefox	http://www.bbc.co.uk/	Food
192.168.43.167	wedding cake	12/07/2016 07:50	*GET/http://bakeshop.carlosbakery.com/wedding-cakes/*	Mozilla Firefox	http://bakeshop.carlosbakery.com/	Food
192.168.43.167	birthday cake	12/08/2016 01:20	*GET/https://www.fnp.com/cakes/birthday*	Mozilla Firefox	https://www.fnp.com/	Food
192.168.43.167	Cookies	12/9/2016 3:60:13 AM	*GET/http://allrecipes.com/recipes/362/desserts/cookies/*	Mozilla Firefox	http://allrecipes.com/	Food

192.168.43.167	chocolate cookies	12/10/2016 07:50	*GET/http://www.geniuskitchen.com/recipe/chewy-chocolate-cookies-5049*	Mozilla Firefox	http://www.geniuskitchen.com/	Food
192.168.43.167	breakfast food	12/11/2016 02:40	*GET/https://greatist.com/health/healthy-fast-breakfast-recipes*	Mozilla Firefox	https://greatist.com/	Food
192.168.43.167	Sandwich	12/12/2016 4:60:13 AM	*GET/http://www.seriouseats.com/sandwiches*	Mozilla Firefox	http://www.seriouseats.com/	Food
192.168.43.167	grilled sandwich	12/13/2016 9:40	*GET/https://www.youtube.com/watch?v=1ow9_3pDydc*	Mozilla Firefox	https://www.youtube.com/	Food
192.168.43.167	grilled cheese sandwich	12/14/2016 13:50:13 AM	*GET/http://allrecipes.com/recipe/23891/grilled-cheese-sandwich/*	Mozilla Firefox	http://allrecipes.com/	Food
192.168.43.167	corn cheese sandwich	12/15/2016 6:04	*GET/https://indianhealthyrecipes.com/corn-cheese-sandwich-recipe/*	Mozilla Firefox	https://indianhealthyrecipes.com/	Food
192.168.43.167	corn cheese soup	12/16/2016 8:32	*GET/http://www.geniuskitchen.com/recipe/corn-cheese-soup-114208*	Mozilla Firefox	http://www.geniuskitchen.com/	Food
192.168.43.167	corn salad	12/17/2016 1:54	*GET/https://www.tasteofhome.com/recipes/summer-corn-salad*	Mozilla Firefox	https://www.tasteofhome.com/	Food
192.168.43.167	Corn cheese samosa	12/18/2016 3:45	*GET/https://www.youtube.com/watch?v=kEJf6DcDjwI*	Mozilla Firefox	https://www.youtube.com/	Food
192.168.43.167	samosa	12/19/2016 1:20	*GET/https://www.youtube.com/watch?v=eawFnRbrdnc*	Mozilla Firefox	https://www.youtube.com/	Food
192.168.43.167	samosa chaat	12/20/2016 3:46	*GET/https://www.youtube.com/watch?v=Fuaf2BuzGAs*	Mozilla Firefox	https://www.youtube.com/	Food
192.168.43.167	pizza samosa	12/21/2016 3:47	*GET/https://www.tastemade.co.uk/videos/pizza-samosas*	Mozilla Firefox	https://www.tastemade.co.uk/	Food
192.168.43.167	Pasta	12/22/2016 3:48	*GET/http://www.taste.com.au/recipes/collections/pasta-recipes*	Mozilla Firefox	http://www.taste.com.au/	Food
192.168.43.167	Pasta salad	12/23/2016 3:49	*GET/http://allrecipes.com/recipes/215/salad/pasta-salad/*	Mozilla Firefox	http://allrecipes.com/	Food
192.168.43.167	red sauce pasta	12/24/2016 3:50	*GET/http://food.ndtv.com/recipe-pasta-with-tomato-sauce-100507*	Mozilla Firefox	http://food.ndtv.com/	Food
192.168.43.167	white sauce pasta	12/25/2016 3:51	*GET/http://foodviva.com/snacks-recipes/white-sauce-pasta/*	Mozilla Firefox	http://foodviva.com/	Food
192.168.43.167	cheese pasta	12/26/2016 3:52	*GET/https://www.pillsbury.com/recipes/four-cheese-pasta/839aba2c-1560-4de3-a5c9-1dbd88072d85*	Mozilla Firefox	https://www.pillsbury.com/	Food
192.168.43.167	cold coffee	12/27/2016 3:53	*GET/https://recipes.timesofindia.com/beverage/non-alcoholic/cold-coffee/rs53842591.cms*	Mozilla Firefox	https://recipes.timesofindia.com/	Food
192.168.43.167	Iced Tea	12/28/2016 3:54	*GET/https://www.lipton.com/us/en/cult-of-tea/how-to-make-iced-tea.html*	Mozilla Firefox	https://www.lipton.com/	Food
192.168.43.167	milkshake	12/29/2016 3:55	*GET/http://www.delish.com/cooking/g1504/milkshake-recipes/*	Mozilla Firefox	http://www.delish.com/	Food
192.168.43.167	mango shake	12/30/2016 3:56	*GET/http://www.vegrecipesofindia.com/mango-milkshake-recipe/*	Mozilla Firefox	http://www.vegrecipesofin	Food

					dia.com/	
192.168.43.167	mango juice	12/31/2016 3:57	*GET/http://www.spiceupthecurry.com/mango-juice-recipe/*	Mozilla Firefox	http://www.spiceupthecurry.com/	Food
192.168.43.167	chocolate milkshake	01/01/2017 03:58	*GET/http://www.vegrecipesofindia.com/chocolate-milkshake-recipe/*	Mozilla Firefox	http://www.vegrecipesofindia.com/	Food
170.10.20.30	SSC EXAM	11/21/2016 14:15	*get/en.wikipedia.org/wiki/SSC_Combined_Graduate_Level_Examination*	Google Chrome	en.wikipedia.org	Education
170.10.20.30	SSC EXAM	11/21/2016 14:15	*GET/testbook.com/blog/ssc-exam-dates/*	Google Chrome	testbook.com	Education
170.10.20.30	SSC EXAM	11/21/2016 14:15	*GET/https://www.successcds.net/... Exam/staff-selection-commission-combined-graduate-le…	Google Chrome	www.successcds.net	Education
170.10.20.30	engeineering colleges	11/21/2016 14:15	*GET/https://www.shiksha.com/b-tech/ranking/top-engineering-colleges-in.../44-2-0-0-0	Google Chrome	www.shiksha.com	Education
170.10.20.30	engeineering colleges	11/21/2016 14:15	*GET/https://www.niche.com/colleges/search/best-colleges-for-engineering/	Google Chrome	www.niche.com	Education
170.10.20.30	engeineering colleges	11/21/2016 14:15	*GET/https://www.collegechoice.net/rankings/best-engineering-degrees/	Google Chrome	www.collegechoice.net	Education
170.10.20.30	ymca university	11/21/2016 14:15	*GET/ymcaust.ac.in/	Google Chrome	ymcaust.ac.in	Education
170.10.20.30	ymca university	11/21/2016 14:15	*GET/ymcaust.ac.in/.../552-ymca-university-ranked-amongst-top-150-engineering-institutio...	Google Chrome	ymcaust.ac.in	Education
170.10.20.30	ymca university	11/21/2016 14:15	*GET/ymcaust.ac.in/index.php/useful-resources/recruitments	Google Chrome	ymcaust.ac.in	Education
170.10.20.30	engeineering colleges in faridabad	11/21/2016 14:15	*GET/https://www.mapsofindia.com/education/engineering-colleges/faridabad.html	Google Chrome	www.mapsofindia.com	Education
170.10.20.30	engeineering colleges in faridabad	11/21/2016 14:15	*GET/www.dec.edu.in/	Google Chrome	www.dec.edu.in	Education
170.10.20.30	engeineering colleges in faridabad	11/21/2016 14:15	*GET/ymcaust.ac.in/	Google Chrome	ymcaust.ac.in	Education
170.10.20.30	computer books	11/21/2016 14:15	*GET/https://www.amazon.com/Computers-Technology-Books/b?ie=UTF8&node=5	Google Chrome	www.amazon.com	Education
170.10.20.30	computer books	11/21/2016 14:15	*GET/https://www.barnesandnoble.com/b/books/computers/_/N-29Z8q8Zug4	Google Chrome	www.barnesandnoble.com	Education
170.10.20.30	computer books	11/21/2016 14:15	*GET/bookboon.com/en/it-programming-ebooks	Google Chrome	bookboon.com	Education
170.10.20.30	primary education	11/21/2016 14:15	*GET/https://en.wikipedia.org/wiki/Primary_education	Google Chrome	en.wikipedia.org	Education
170.10.20.30	primary education	11/21/2016 14:15	*GET/www.moec.gov.cy/dde/en/	Google Chrome	www.moec.gov.cy	Education

170.10.20.30	primary education	11/21/2016 14:15	*GET/www.gov.sz/index.php?option=com_content&view=article&id=295...408	Google Chrome	www.gov.sz	Education
170.10.20.30	education facility in india	11/21/2016 14:15	*GET/https://en.wikipedia.org/wiki/Education_in_India	Google Chrome	en.wikipedia.org	Education
170.10.20.30	education facility in india	11/21/2016 14:15	*GET/https://en.wikipedia.org/wiki/Education_in_Delhi	Google Chrome	en.wikipedia.org	Education
170.10.20.30	education facility in india	11/21/2016 14:16	*GET/https://www.classbase.com/countries/India/Education-System	Google Chrome	www.classbase.com	Education
170.10.20.30	iit collegas	11/21/2016 14:16	*GET/https://www.mapsofindia.com/education/iit-colleges-in-india.html	Google Chrome	en.wikipedia.org	Education
170.10.20.30	iit collegas	11/21/2016 14:16	*GET/https://en.wikipedia.org/wiki/Indian_Institutes_of_Technology	Google Chrome	en.wikipedia.org	Education
170.10.20.30	iit collegas	11/21/2016 14:16	*GET/https://engineering.careers360.com › Home › Tags	Google Chrome	engineering.careers360.com	Education
170.10.20.30	NET Exam pattern	11/21/2016 14:16	*GET/https://www.successcds.net/Entrance-Exam/ugc-net-exam-pattern.html	Google Chrome	www.successcds.net	Education
170.10.20.30	NET Exam pattern	11/21/2016 14:16	*GET/https://testbook.com/blog/net-exam-pattern/	Google Chrome	testbook.com	Education
170.10.20.30	NET Exam pattern	11/21/2016 14:16	*GET/https://scoop.eduncle.com/ugc-net-exam-pattern	Google Chrome	scoop.eduncle.com	Education
170.10.20.30	latest science research	11/21/2016 14:16	*GET/https://www.sciencedaily.com/	Google Chrome	www.sciencedaily.com	Education
170.10.20.30	latest science research	11/21/2016 14:16	*GET/https://www.sciencedaily.com/news/	Google Chrome	www.sciencedaily.com	Education
170.10.20.30	latest science research	11/21/2016 14:16	*GET/https://www.livescience.com/news	Google Chrome	www.livescience.com	Education
170.10.20.30	latest science research	11/21/2016 14:16	*GET/https://www.sciencedaily.com/news/	Google Chrome	www.sciencedaily.com	Education
170.10.20.30	nobel price in physics	11/21/2016 14:16	*GET/https://www.nobelprize.org/nobel_prizes/physics/laureates/	Google Chrome	www.nobelprize.org	Education
170.10.20.30	nobel price in physics	11/21/2016 14:16	*GET/https://www.nobelprize.org/nobel_prizes/physics/laureates/2016/	Google Chrome	www.nobelprize.org	Education
170.10.20.30	nobel price in physics	11/21/2016 14:16	*GET/https://www.nobelprize.org/nobel_prizes/physics/	Google Chrome	www.nobelprize.org	Education
170.10.20.30	schools in rajasthan	11/21/2016 14:16	*GET/https://en.wikipedia.org/wiki/List_of_schools_in_Rajasthan	Google Chrome	en.wikipedia.org	Education
170.10.20.30	schools in rajasthan	11/21/2016 14:16	*GET/https://targetstudy.com/school/schools-in-rajasthan.html	Google Chrome	targetstudy.com	Education
170.10.20.30	schools in rajasthan	11/21/2016 14:16	*GET/www.icbse.com/schools/state/rajasthan	Google Chrome	www.icbse.com	Education
170.10.20.30	smart schools	11/21/2016 14:16	*GET/https://www.smart-schools.com/	Google Chrome	www.smart-schools.com	Education
170.10.20.30	smart schools	11/21/2016 14:16	*GET/https://www.fedena.com/blog/2015/10/6-impacts-of-smart-schools-	Google Chrome	www.fedena.com	Education

			on-education.html			
170.10.20.30	data warehouse	11/21/2016 14:16	*GET/https://docs.oracle.com/cd/B10501_01/server.920/a96520/concept.htm	Google Chrome	docs.oracle.com	Education
170.10.20.30	data warehouse	11/21/2016 14:16	*GET/https://en.wikipedia.org/wiki/Data_warehouse	Google Chrome	en.wikipedia.org	Education
170.10.20.30	data warehouse	11/21/2016 14:16	*GET/searchsqlserver.techtarget.com › BI and Data Warehousing › Database	Google Chrome	searchsqlserver.techtarget.com	Education
170.10.20.30	data mining	11/21/2016 14:16	*GET/https://en.wikipedia.org/wiki/Data_mining	Google Chrome	en.wikipedia.org	Education
170.10.20.30	data mining	11/21/2016 14:16	*GET/searchsqlserver.techtarget.com › BI and Data Warehousing › Software applications	Google Chrome	searchsqlserver.techtarget.com	Education
170.10.20.30	data mining	11/21/2016 14:16	*GET/www.thearling.com/text/dmwhite/dmwhite.htm	Google Chrome	www.thearling.com	Education
170.10.20.30	smart study	11/21/2016 14:16	*GET/https://www.youtube.com/channel/UCSxd8i0Imrz3qwh0jyR1CcQ	Google Chrome	www.youtube.com	Education
170.10.20.30	smart study	11/21/2016 14:16	*GET/https://www.youtube.com/channel/UCOG5FqyZ84IzGgAbU31oOGw	Google Chrome	www.youtube.com	Education
170.10.20.30	smart study	11/21/2016 14:16	*GET/https://www.facebook.com/studysmart00/	Google Chrome	www.facebook.com	Education
170.10.20.30	highest educated state india	11/21/2016 14:16	*GET/https://en.wikipedia.org/wiki/Indian_states_ranking_by_literacy_rate	Google Chrome	en.wikipedia.org	Education
170.10.20.30	highest educated state india	11/21/2016 14:16	*GET/top10wala.in › 2015	Google Chrome	top10wala.in	Education
170.10.20.30	highest educated state india	11/21/2016 14:17	*GET/besttimepass.com/top-10-most-educated-states-in-india/	Google Chrome	besttimepass.com	Education
170.10.20.30	highest literacy rate india	11/21/2016 14:17	*GET/https://en.wikipedia.org/wiki/Literacy_in_India	Google Chrome	en.wikipedia.org	Education
170.10.20.30	highest literacy rate india	11/21/2016 14:17	*GET/https://en.wikipedia.org/wiki/Indian_states_ranking_by_literacy_rate	Google Chrome	en.wikipedia.org	Education
170.10.20.30	highest literacy rate india	11/21/2016 14:17	*GET/https://en.wikipedia.org/wiki/Indian_states_ranking_by_literacy_rate	Google Chrome	en.wikipedia.org	Education
170.10.20.30	Literacy rate women	11/21/2016 14:17	*GET/https://www.mapsofindia.com › Answers › India	Google Chrome	www.mapsofindia.com	Education
170.10.20.30	Literacy rate women	11/21/2016 14:17	*GET/trendyfeeds.com/literate-states-of-india/	Google Chrome	trendyfeeds.com	Education
170.10.20.30	scholarship for general category student	11/21/2016 14:17	*GET/mhrd.gov.in/scholarships-education-loan-0	Google Chrome	mhrd.gov.in	Education
170.10.20.30	scholarship for general category	11/21/2016 14:17	*GET/https://www.quora.com/What-are-some-good-scholarship-options-for-a-general-categ...	Google Chrome	www.quora.com	Education

	student					
170.10.20.30	SC/ST scholarship	11/21/2016 14:17	*GET/https://scholarships.gov.in/	Google Chrome	scholarships.gov.in	Education
170.10.20.30	SC/ST scholarship	11/21/2016 14:17	*GET/https://scholarships.gov.in/	Google Chrome	scholarships.gov.in	Education
170.10.20.30	SC/ST scholarship	11/21/2016 14:17	*GET/www.mpsc.mp.nic.in/scholarships/	Google Chrome	www.mpsc.mp.nic.in	Education
170.10.20.30	foreign education	11/21/2016 14:17	*GET/indiatoday.intoday.in/education/story/foreign-education/1/770560.html	Google Chrome	indiatoday.intoday.in	Education
170.10.20.30	education abroad	11/21/2016 14:17	*GET/pradhanmantri-yogana.in/beti-bachao-beti-padhao-scheme-in-hindi/	Google Chrome	pradhanmantri-yogana.in	Education
170.10.20.30	beti bchao beti padhao	11/21/2016 14:17	*GET/www.deepawali.co.in/beti-bachao-beti-padhao-yojana-in-hindi.html	Google Chrome	www.deepawali.co.in	Education
170.10.20.30	beti bchao beti padhao	11/21/2016 14:17	*GET/pradhanmantri-yogana.in/beti-bachao-beti-padhao-scheme-in-hindi/	Google Chrome	pradhanmantri-yogana.in	Education
170.10.20.30	research paper CK Nagapal	11/21/2016 14:17	*GET/www.bvicam.ac.in/indiacom/.../Brief%20Profile%20C%20K%20Nagpal.pdf	Google Chrome	www.bvicam.ac.in	Education
170.10.20.30	research paper CK Nagapal	11/21/2016 14:17	*GET/ymcaust.ac.in/f_detail.php?id=11	Google Chrome	ymcaust.ac.in	Education
170.10.20.30	manjeet singh research	11/21/2016 14:17	*GET/https://www.researchgate.net/profile/Manjeet_Singh25	Google Chrome	www.researchgate.net	Education
170.10.20.30	manjeet singh research	11/21/2016 14:17	*GET/https://www.researchgate.net/profile/Manjeet_Singh35	Google Chrome	www.researchgate.net	Education
170.10.20.30	research paper manjeet singh tomar ymca	11/21/2016 14:17	*GET/ymcaust.ac.in/f_detail.php?id=42	Google Chrome	ymcaust.ac.in	Education
170.10.20.30	research paper manjeet singh tomar ymca	11/21/2016 14:17	*GET/ymcaust.ac.in/faculty/f_biodata/42cv_dr_manjeet_singh.pdf	Google Chrome	ymcaust.ac.in	Education
170.10.20.30	research paper manjeet singh tomar ymca	11/21/2016 14:17	*GET/ymcaust.ac.in/faculty/f_biodata/45parultomar.pdf	Google Chrome	ymcaust.ac.in	Education
170.10.20.30	WHY Smriti irani	11/21/2016 14:17	*GET/https://en.wikipedia.org/wiki/Smriti_Irani	Google Chrome	en.wikipedia.org	Education
170.10.20.30	WHY Smriti irani	11/21/2016 14:17	*GET/https://www.scoopwhoop.com/Smriti-Iranis-Demotion-Is-A-Stern-Message-To-Every...	Google Chrome	www.scoopwhoop.com	Education
170.10.20.30	WHY Smriti irani	11/21/2016 14:17	*GET/www.elections.in/political-leaders/smriti-irani.html	Google Chrome	www.elections.in	Education
170.10.20.30	nalanda university	11/21/2016 14:17	*GET/https://www.nalandauniv.edu.in/	Google Chrome	www.nalandauniv.edu.in	Education
170.10.20.30	nalanda university	11/21/2016 14:17	*GET/https://www.nalandauniv.edu.in/	Google Chrome	www.nalandauniv.edu.in	Education
170.10.20.30	nalanda university	11/21/2016 14:17	*GET/https://en.wikipedia.org/wiki/Nalanda_University	Google Chrome	en.wikipedia.org	Education
170.10.20.30	first university	11/21/2016 14:17	*GET/www.guinnessworldrecords.com/world-records/oldest-university	Google Chrome	www.guinnessworldrecor	Education

					ds.com	
170.10.20.30	first university india	11/21/2016 14:17	*GET/https://en.wikipedia.org/wiki/University_of_Calcutta	Google Chrome	en.wikipedia.org	Education
170.10.20.30	first parer book	11/21/2016 14:18	*GET/https://en.wikipedia.org/wiki/History_of_books	Google Chrome	en.wikipedia.org	Education
170.10.20.30	biggest library	11/21/2016 14:18	*GET/https://en.wikipedia.org/wiki/List_of_the_largest_libraries_in_the_United_States	Google Chrome	en.wikipedia.org	Education
170.10.20.30	biggest library	11/21/2016 14:18	*GET/https://en.wikipedia.org/wiki/List_of_largest_libraries	Google Chrome	en.wikipedia.org	Education
170.10.20.30	library japan	11/21/2016 14:18	*GET/https://en.wikipedia.org/wiki/Imperial_Library_(Japan)	Google Chrome	en.wikipedia.org	Education
170.10.20.30	central library india	11/21/2016 14:18	*GET/www.nationallibrary.gov.in/	Google Chrome	www.nationallibrary.gov.in	Education
170.10.20.30	compiler design books	11/21/2016 14:18	*GET/https://www.amazon.com/Compiler-Design-Languages-Tools-Books/b?ie=UTF8...	Google Chrome	www.amazon.com	Education
170.10.20.30	compiler design books	11/21/2016 14:18	*GET/https://www.amazon.com/Best-Sellers-Books-Compiler-Design/zgbs/books/3970	Google Chrome	www.amazon.com	Education
170.10.20.30	sachin tendulkar books	11/21/2016 14:18	*GET/https://www.amazon.in/Books-Sachin-Tendulkar/s?...27%3ASachin%20Tendulkar	Google Chrome	www.amazon.in	Education
170.10.20.30	sachin tendulkar books	11/21/2016 14:18	*GET/https://www.amazon.in/Sachin-Tendulkar-Playing-Way.../dp/1473605202	Google Chrome	www.amazon.in	Education
170.10.20.30	artificial intelligence boooks	11/21/2016 14:18	*GET/https://www.amazon.com/Artificial-Intelligence/b?ie=UTF8&node=491300	Google Chrome	www.amazon.com	Education
170.10.20.30	artificial intelligence boooks	11/21/2016 14:18	*GET/bigdata-madesimple.com/20-free-books-to-get-started-with-artificial-intelligence/	Google Chrome	bigdata-madesimple.com	Education
170.10.20.30	yashwant kathkar books	11/21/2016 14:18	*GET/https://en.wikipedia.org/wiki/Yashavant_Kanetkar	Google Chrome	en.wikipedia.org	Education
170.10.20.30	chetan bhagat books	11/21/2016 14:18	*GET/https://en.wikipedia.org/wiki/Chetan_Bhagat	Google Chrome	en.wikipedia.org	Education
170.10.20.30	chetan bhagat books	11/21/2016 14:18	*GET/https://en.wikipedia.org/wiki/One_Indian_Girl	Google Chrome	en.wikipedia.org	Education
170.10.20.30	java books	11/21/2016 14:18	*GET/https://www.amazon.com/Best-Sellers-Books-Java-Programming/zgbs/books/3608	Google Chrome	www.amazon.com	Education
170.10.20.30	java projects	11/21/2016 14:18	*GET/https://www.udemy.com/learn-java-by-building-projects/	Google Chrome	www.udemy.com	Education
170.10.20.30	java books	11/21/2016 14:18	*GET/https://www.amazon.com/Java-Programming-Computers-Internet-Books/b?ie...	Google Chrome	www.amazon.com	Education
170.10.20.30	mca aims and objectives	11/21/2016 14:18	*GET/www.dsktraining.co.uk/aims-and-objectives-of-the-mental-capacity-act-deprivation-o...	Google Chrome	www.dsktraining.co.uk	Education

170.10.20.30	mca aims and objectives	11/21/2016 14:19	*GET/www.mca.gov.md/en/aim_and_objectives_Tr.html	Google Chrome	www.mca.gov.md	Education
192.168.1.10	Test Match	11/22/2016 14:19	https://en.wikipedia.org/wiki/Test_cricket	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	One day Match	11/23/2016 14:19	https://www.theguardian.com/sport/live/2016/sep/24/india-v-australia-third-one-day-international-live	Google Chrome	https://www.theguardian.com	Sports
192.168.1.10	T 20 Match	11/24/2016 14:19	https://sports.ndtv.com/cricket/schedules-fixtures	Google Chrome	https://sports.ndtv.com/	Sports
192.168.1.10	Cricket Ball	11/25/2016 14:19	https://en.wikipedia.org/wiki/Cricket_ball	Google Chrome	https://en.wikipedia.org	Sports
192.168.1.10	Volley Ball	11/26/2016 14:19	https://en.wikipedia.org/wiki/Volleyball	Google Chrome	https://en.wikipedia.org	Sports
192.168.1.10	Foot ball	11/27/2016 14:19	https://en.wikipedia.org/wiki/Football	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	Hand Ball	11/28/2016 14:19	https://en.wikipedia.org/wiki/Handball	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	Basket Ball	11/29/2016 14:19	https://en.wikipedia.org/wiki/Basketball	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	India at Common wealth Games	11/30/2016 14:19	https://en.wikipedia.org/wiki/India_at_the_Commonwealth_Games	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	India at Olympic Games	12/01/2016 14:20	https://en.wikipedia.org/wiki/India_at_the_Olympics	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	India at Asian Games	12/02/2016 14:20	https://en.wikipedia.org/wiki/India_at_the_Asian_Games	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	Fifa World Cup	12/03/2016 14:20	http://www.fifa.com/worldcup/index.html	Google Chrome	http://www.fifa.com	Sports
192.168.1.10	Indoor Games	12/04/2016 14:22	https://www.familyfuntwincities.com/fun-indoor-games-for-kids-of-all-ages-categorized/	Google Chrome	https://www.familyfuntwincities.com/	Sports
192.168.1.10	Outdoor Games	12/05/2016 14:22	https://in.pinterest.com/explore/outdoor-games/	Google Chrome	https://in.pinterest.com/	Sports
192.168.1.10	Pro Kabaddi	11/20/2016 23:06	https://www.prokabaddi.com/	Google Chrome	https://www.prokabaddi.com/	Sports
192.168.1.10	Pro Kabaddi Rules	11/20/2016 23:06	https://www.prokabaddi.com/prokabaddi-rules	Google Chrome	https://www.prokabaddi.com/	Sports
192.168.1.10	Online Sports Games	11/20/2016 23:06	http://www.agame.com/games/sports	Google Chrome	http://www.agame.com	Sports
192.168.1.10	Online Games	11/20/2016 23:07	https://www.miniclip.com/games/en/	Google Chrome	https://www.miniclip.com	Sports
192.168.1.10	Cricket Equipments	11/20/2016 23:11	https://www.sportsdirect.com/cricket	Google Chrome	https://www.sportsdirect.com/	Sports
192.168.1.10	Foot ball Equipments	11/20/2016 23:13	https://www.dickssportinggoods.com/products/football-equipment-gear.jsp	Google Chrome	https://www.dickssportinggoods.com/	Sports

192.168.1.10	Gym Equipments	11/20/2016 23:15	https://www.amazon.in/Exercise-Fitness/b?ie=UTF8&node=3403635031	Google Chrome	https://www.amazon.in/	Sports
192.168.1.10	Indian Cricket Team	11/20/2016 23:16	www.bcci.tv/team-india	Google Chrome	www.bcci.tv/	Sports
192.168.1.10	Indian Football Team	11/20/2016 23:31	https://twitter.com/IndianFootball	Google Chrome	https://twitter.com/	Sports
192.168.1.10	Indian Hockey Team	11/20/2016 23:31	www.thehindu.com › Sport › Hockey	Google Chrome	www.thehindu.com	Sports
192.168.1.10	Indian Kabaddi Team	11/20/2016 23:49	https://www.sportskeeda.com/team/india-kabaddi-team	Google Chrome	https://www.sportskeeda.com/	Sports
192.168.1.10	Indian Table Tennis Team	11/20/2016 23:50	https://en.wikipedia.org/wiki/Category:Indian_table_tennis_players	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	Indian Chess Team	11/20/2016 23:50	https://www.chessbase.in/news/icf-wins-inter-railway-2016/	Google Chrome	https://www.chessbase.in/	Sports
192.168.1.10	Indian Badminton Team	11/20/2016 23:50	https://en.wikipedia.org/wiki/India_national_badminton_team	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	Indian Basketball Team	11/21/2016 0:01	https://en.wikipedia.org/wiki/India_national_basketball_team	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	Indian Kho Kho Team	11/21/2016 0:02	https://en.wikipedia.org/wiki/Kho_kho	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	Indian Swimming Team	11/21/2016 0:02	https://en.wikipedia.org/wiki/List_of_Indian_records_in_swimming	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	Indian Squash Team	11/21/2016 0:02	https://en.wikipedia.org/wiki/India_men%27s_national_squash_team	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	Commonwealth Games	11/21/2016 0:02	https://en.wikipedia.org/wiki/Commonwealth_Games	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	Olympic Games	11/21/2016 0:02	https://www.olympic.org/olympic-games	Google Chrome	https://www.olympic.org/	Sports
192.168.1.10	Asian Games	11/21/2016 14:42	https://en.wikipedia.org/wiki/India_at_the_Asian_Games	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	South Asian Games	11/21/2016 14:42	https://en.wikipedia.org/wiki/India_at_the_2016_South_Asian_Games	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	Cricket World Cup	11/21/2016 14:42	https://www.icc-cricket.com/cricket-world-cup	Google Chrome	https://www.icc-cricket.com/	Sports
192.168.1.10	Blind Cricket World Cup	11/21/2016 14:43	https://www.sportskeeda.com/cricket/india-win-the-cricket-world-cup-for-the-blind-2...	Google Chrome	https://www.sportskeeda.com/	Sports
192.168.1.10	Hockey World Cup	11/21/2016 14:43	https://en.wikipedia.org/wiki/Hockey_World_Cup	Google Chrome	https://en.wikipedia.org	Sports
192.168.1.10	Kabaddi World Cup	11/21/2016 14:43	https://en.wikipedia.org/wiki/Kabaddi_World_Cup	Google Chrome	https://en.wikipedia.org	Sports
192.168.1.10	Kho Kho World Cup	11/21/2016 14:43	https://en.wikipedia.org/wiki/Kho-Kho_at_the_2016_South_Asian_Games	Google Chrome	https://en.wikipedia.org/	Sports

192.168.1.10	Basketball World Cup	11/21/2016 14:43	https://en.wikipedia.org/wiki/2017_FIBA_Women%27s_Basketball_World_Cup	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	Badminton World Cup	11/21/2016 14:44	https://timesofindia.indiatimes.com/.../badminton/world-championships...world.../602...	Google Chrome	https://timesofindia.indiatimes.com/	Sports
192.168.1.10	Swimming World Cup	11/21/2016 14:44	https://en.wikipedia.org/wiki/2016_FINA_Swimming_World_Cup	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	Chess World Cup	11/21/2016 14:44	https://en.wikipedia.org/wiki/Chess_World_Cup_2016	Google Chrome	https://en.wikipedia.org	Sports
192.168.1.10	Squash World Cup	11/21/2016 14:44	https://en.wikipedia.org/wiki/WSF_World_Team_Squash_Championships	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	Table Tennis World Cup	11/21/2016 14:52	https://en.wikipedia.org/wiki/Table_Tennis_World_Cup	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	indian winner of world junior badminton	11/21/2016 14:52	https://en.wikipedia.org/wiki/2008_BWF_World_Junior_Championships	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	ICC Women World Cup	11/21/2016 14:52	www.cricbuzz.com/cricket-series/2571/icc-womens-world-cup-2016/matches	Google Chrome	www.cricbuzz.com/cricket-series/	Sports
192.168.1.10	Women Hockey World Cup	11/21/2016 14:59	https://en.wikipedia.org/wiki/Women%27s_Hockey_World_Cup	Google Chrome	https://en.wikipedia.org/	Sports
192.168.1.10	Women Football World Cup	11/21/2016 15:00	www.fifa.com/womens-football/index.html	Google Chrome	www.fifa.com/	Sports
172.20.10.10	E-Books	11/23/2016 1:46	*GET/https://en.wikipedia.org/wiki/E-book*	Google Chrome	en.wikipedia.org	Education
172.20.10.10	E-Books	11/23/2016 1:46	*GET/https://www.ebooks.com/*	Google Chrome	www.ebooks.com/	Education
172.20.10.10	E-Books	11/23/2016 1:47	*GET/https://www.goodreads.com/ebooks*	Google Chrome	www.goofreads.com	Education
172.20.10.10	E-Books	11/23/2016 1:47	*GET/https://www.bookrix.com/books.html*	Google Chrome	www.bookrix.com	Education
172.20.10.10	E-Books	11/23/2016 1:48	*GET/http://bookboon.com/*	Google Chrome	http://bookboon.com/	Education
172.20.10.10	20 Best site to Download E-Book	11/23/2016 1:48	*GET/https://www.lifewire.com/download-free-books-3482754*	Google Chrome	www.lifewire.com	Education
172.20.10.10	Hotels	11/23/2016 1:48	*GET/https://in.hotels.com/?pos=HCOM_IN&amp;locale=en_IN&amp;rffrid=sem.hcom.IN.google.003.00.03.s.kwrd=c.184545664487.48215317826.981516661.1t1.kwds*	Google Chrome	n.hotels.com	Travel
172.20.10.10	Hotels	11/23/2016 1:48	*GET/https://www.booking.com/index.en.html*	Google Chrome	/www.booking.com	Travel
172.20.10.10	Hotels in manali	11/23/2016 1:48	*GET/https://www.tripadvisor.in/SmartDeals-g297618-Manali_Manali_Tehsil_Kullu_District_Himachal_Pradesh-Hotel-Deals.html*	Google Chrome	www.tripadvisor.in	Travel

172.20.10.10	Hotels in nainital	11/23/2016 1:48	*https://www.tripadvisor.in/SmartDeals-g660548-Nainital_Nainital_District_Uttarakhand-Hotel-Deals.html*	Google Chrome	www.tripadvisor.in	Travel
172.20.10.10	hill-station near delhi	11/23/2016 1:48	*GET/https://www.thrillophilia.com/hill-stations-near-delhi*	Google Chrome	www.thrillophilia.com	Travel
172.20.10.10	ladakh trip cost	11/23/2016 1:48	*GET/https://www.makemytrip.com/holidays-india/ladakh-tour-packages.html*	Google Chrome	www.makemytrip.com	Travel
172.20.10.10	package tour leh ladakh	11/23/2016 1:48	*GET/http://ladakh-packages.ladakh-tour.in/?gclid=Cj0KCQiA3dTQBRDnARIsAGKSflmSsuoAadPPvH4_x6ix43irRa4Jy2_O7Bj5kyy1EgkmbhTdBWD8dy4aAre0EALw_wcB*	Google Chrome	ladakh-packages.ladakh-tour.in	Travel
172.20.10.10	package tour to kerala	11/23/2016 1:49	*GET/http://www.zenithholidays.com/winter-getaway-kerala.aspx?gclid=Cj0KCQiA3dTQBRDnARIsAGKSflmNwJuipgJha8er_RdKjrSYG9aYC*	Google Chrome	www.zenithholidays.com	Travel
172.20.10.10	package tour to goa	11/23/2016 1:49	*GET/https://www.makemytrip.com/holidays-india/goa-travel-packages.html*	Google Chrome	www.makemytrip.com	Travel
172.20.10.10	Cheap package holidays	11/23/2016 1:49	*GET/https://www.holidaypirates.com/holidaypackages*	Google Chrome	www.holidaypirates.com	Travel
172.20.10.10	is it safe to travel turkey now	11/27/2016 7:33	*GET/http://www.telegraph.co.uk/travel/advice/is-turkey-safe-for-tourists/*	Google Chrome	www.telegraph.co.uk	Travel
172.20.10.10	is it safe to travel australia	11/27/2016 7:35	*GET/https://www.gov.uk/foreign-travel-advice/australia*	Google Chrome	www.gov.uk	Travel
172.20.10.10	how to become a travel agent	11/27/2016 7:37	*GET/http://www.hungrybags.com/travshoppe/index.php?x=tab&amp;gclid*	Google Chrome	www.hungrybags.com	Travel
172.20.10.10	how to become travel agent in india	11/27/2016 7:38	*GET/https://www.indiafilings.com/learn/starting-a-travel-agency-business-in-india/*	Google Chrome	www.indiafilings.com	Travel
172.20.10.10	how to travel cheap	11/27/2016 7:41	*GET/https://www.makemytrip.com/flights?cmp*	Google Chrome	www.makemytrip.com	Travel
172.20.10.10	how to travel cuba	11/27/2016 7:41	*GET/https://expertvagabond.com/travel-to-cuba-for-americans//*	Google Chrome	www.expertvagabond.com	Travel
172.20.10.10	how long to travel to mars	11/27/2016 7:44	*GET/https://www.mars-one.com/faq/mission-to-mars/how-long-does-it-take-to-travel-to-mars*	Google Chrome	www.mars-one.com	Travel
172.20.10.10	how long to travel o moon	11/27/2016 7:46	*GET/https://www.space.com/18145-how-far-is-the-moon.html*	Google Chrome	www.space.com	Travel
172.20.10.10	how to start travel agency	11/27/2016 7:49	*GET/http://www.travelbookingagent.in/*	Google Chrome	www.travelbookingagent.in	Travel
172.20.10.10	how to start travel blog	11/27/2016 7:50	*GET/https://artoftravel.tips/?gclid*	Google Chrome	www.artoftravel.tips	Travel
172.20.10.10	where to travel in december	11/27/2016 7:55	*GET/https://www.huffingtonpost.com/viator/where-to-go-in-december-2_b_4387664.html*	Google Chrome	www.huffingtonpost.com	Travel
172.20.1	where to travel	11/27/2016	*GET/http://www.cntraveller.com/recommended/itineraries/top-10-	Google	www.cntrave	Travel

0.10	in november	7:55	holiday-destinations/page/top-10-holiday-destinations-november*	Chrome	ller.com	
172.20.10.10	where to travel in august in india	11/27/2016 7:58	*GET/https://www.hellotravel.com/stories/best-places-to-visit-in-august-in-india?gclid*	Google Chrome	www.hellotravel.com	Travel
172.20.10.10	where to travel in july in india	11/27/2016 7:59	*GET/https://www.holidify.com/blog/places-to-visit-in-july-in-india/*	Google Chrome	www.holidify.com	Travel
172.20.10.10	where to travel in winter	11/27/2016 8:02	*GET/http://www.travelchannel.com/interests/winter/photos/editors-picks-hot-winter-vacations*	Google Chrome	www.travelchannel.com	Travel
172.20.10.10	where to travel summer	11/27/2016 8:02	*GET/https://www.cntraveler.com/galleries/2015-06-29/10-best-places-to-visit-this-summer*	Google Chrome	www.cntraveler.com	Travel
172.20.10.10	where you can travel with a passport	11/27/2016 8:07	*GET/http://indiatoday.intoday.in/education/story/indian-passport/1/469258.html*	Google Chrome	www.indiatoday.intoday.in	Travel
172.20.10.10	where you can travel without a passport	11/27/2016 8:07	*GET/http://www.travelchannel.com/destinations/us/photos/no-passport-required*	Google Chrome	www.travelchannel.com	Travel
172.20.10.10	distance between india and usa	11/27/2016 8:10	*GET/https://www.distancefromto.net/distance-from-india-to-united-states*	Google Chrome	www.distancefromto.net	Travel
172.20.10.10	distance between india and australia	11/27/2016 8:10	*GET/https://www.distancefromto.net/distance-from-australia-to-india*	Google Chrome	www.distancefromto.net	Travel
172.20.10.10	flight from bengluru to delhi	11/27/2016 8:14	*GET/http://flights.makemytrip.com/makemytrip/search/O/O/E/1/0/0/S/V0/BLR_DEL_15-01-2018?lang*	Google Chrome	www.flights.makemytrip.com	Travel
172.20.10.10	flight from india to usa	11/27/2016 8:15	*GET/http://www.thaiairways.com/en_IN/index.page?gclid*	Google Chrome	www.thaiairways.com	Travel
172.20.10.10	travel apps	11/27/2016 8:18	*GET/https://www.danhostel.dk/en/content/best-free-travel-apps-denmark?gclid*	Google Chrome	www.danhostel.dk	Travel
172.20.10.10	travel agency	11/27/2016 8:19	*GET/http://www.detoursindia.com/*	Google Chrome	www.detoursindia.com	Travel
172.20.10.10	only booking flight app	11/27/2016 8:22	*GET/https://www.makemytrip.com/flights?cmp*	Google Chrome	www.makemytrip.com	Travel
172.20.10.10	only booking hotel app	11/27/2016 8:23	*GET/https://www.trivago.in/?iSemThemeId*	Google Chrome	www.trivago.in	Travel
172.20.10.10	places to travel in india	11/27/2016 8:25	*GET/https://www.inspirock.com/india-trip-planner?gclid*	Google Chrome	www.inspirock.com	Travel
172.20.10.10	cheap place to travel	11/27/2016 8:26	*GET/https://www.makemytrip.com/flights?cmphttps://www.makemytrip.com/flights?cmp*	Google Chrome	www.makemytrip.com	Travel
172.20.10.10	tour travel agency in india	11/27/2016 8:29	*GET/https://travefy.com/pro?km_marketing=google&amp;km_google*	Google Chrome	www.travefy.com	Travel
172.20.10.10	top two travel agency in india	11/27/2016 8:30	*GET/https://travefy.com/pro?km_marketing*	Google Chrome	www.travefy.com	Travel
172.20.10.10	maximum visited place in	11/27/2016 8:33	*GET/http://www.india.com/travel/articles/top-10-most-visited-cities-in-	Google Chrome	www.india.com	Travel

	world		the-world-in-2016/?gclid*			
172.20.10.10	maximum visited place in india	11/27/2016 8:34	*GET/http://www.india.com/travel/articles/top-10-most-famous-tourist-places-in-india/?gclid*	Google Chrome	www.india.com	Travel
172.20.10.10	famous cities in world	11/27/2016 8:37	*GET/https://transferwise.com/gb/blog/10-visited-cities*	Google Chrome	www.transferwise.com	Travel
172.20.10.10	famous cities in india	11/27/2016 8:38	*GET/http://www.indiafamousfor.com/cities-in-india.html*	Google Chrome	www.indiafamousfor.com	Travel
172.20.10.10	travel by train	11/27/2016 8:41	*GET/https://www.b-europe.com/EN/Legal/About-SNCB-Europe?gclid*	Google Chrome	www.b-europe.com	Travel
172.20.10.10	travel by bus	11/27/2016 8:41	*GET/https://www.redbus.in/bus-tickets/?gclid*	Google Chrome	www.redbus.in	Travel
172.20.10.10	travel by car	11/27/2016 8:44	*GET/http://www.booktaxiinamritsar.com/*	Google Chrome	www.booktaxiinamritsar.com	Travel
172.20.10.10	travel by flight	11/27/2016 8:44	*GET/http://www.shermanstravel.com/travel_search/flights?refer*	Google Chrome	www.shermanstravel.com	Travel
172.20.10.10	self dive car for travelling	11/27/2016 8:47	*GET/https://www.mylescars.com/?gclid*	Google Chrome	www.mylescars.com	Travel
172.20.10.10	rent car for travelling	11/27/2016 8:47	*GET/https://volercars.com/delhi?gclid*	Google Chrome	www.volercars.com	Travel
172.20.10.10	best restaurant for indian food in usa for visiters	11/27/2016 8:51	*GET/http://darbarny.com/?gclid*	Google Chrome	www.darbarny.com	Travel
172.20.10.10	best restaurant for chinees food in india for visiters	11/27/2016 8:52	*GET/https://www.tripadvisor.in/Restaurants-g304551-New_Delhi_National_Capital_Territory_of_Delhi.html*	Google Chrome	www.tripadvisor.in	Travel
172.20.10.10	emergencies security for visiters in usa	11/27/2016 8:54	*GET/https://www.visitorscoverage.com/?gclid*	Google Chrome	www.visitorscoverage.com	Travel
172.20.10.10	emergencies security for visiters in india	11/27/2016 8:57	*GET/http://www.dailymail.co.uk/travel/travel_news/article-2897592/India-introduces-emergency-helplines-travel-advice-visitors-tourism*	Google Chrome	www.dailymail.co.uk	Travel
172.20.10.10	cost for travelling to goa	11/27/2016 9:00	*GET/https://www.tripadvisor.in/SmartDeals-g297604-Goa-Hotel-Deals.html*	Google Chrome	www.tripadvisor.in	Travel
172.20.10.10	cost for travelling to shimla	11/27/2016 9:00	*GET/https://www.inspirock.com/india/shimla-trip-planner?gclid*	Google Chrome	www.inspirock.com	Travel
172.20.10.10	type of clothes in shimla for visters	11/27/2016 9:03	*GET/http://www.shimlaindia.net/travel-tips/what-to-wear.html*	Google Chrome	www.shimlaindia.net	Travel
172.20.10.10	type of clothes in goa for visters	11/27/2016 9:03	*GET/https://www.skyscanner.co.in/news/things-you-must-pack-your-goa-trip*	Google Chrome	www.skyscanner.co.in	Travel
172.20.10.10	haunted places in india	11/27/2016 9:06	*GET/https://www.speakingtree.in/allslides/13-most-haunted-places-of-india?gclid*	Google Chrome	www.speakingtree.in	Travel

172.20.10.10	haunted place in asia	11/27/2016 9:07	*GET/https://www.hellotravel.com/stories/5-most-haunted-places-on-earth-hair-raising-experience*	Google Chrome	www.hellotravel.com	Travel
172.20.10.10	visiting time for tajmahal	11/27/2016 9:11	*GET/https://www.tajmahal.gov.in/*	Google Chrome	www.tajmahal.gov.in	Travel
172.20.10.10	visiting time for redfort	11/27/2016 9:11	*GET/https://www.ixigo.com/red-fort-new-delhi-india-opening-visiting-timing-hours-closed-days-ne-1174212*	Google Chrome	www.ixigo.com	Travel
172.20.10.10	best time to visit bhangarh	11/27/2016 9:14	*GET/https://www.inspirock.com/india/alwar/bhangarh-fort-a494327439?gclid*	Google Chrome	www.inspirock.com	Travel
172.20.10.10	best time to visit manali	11/27/2016 9:15	*GET/http://www.india.com/travel/articles/best-time-to-visit-manali-for-honeymoon/?gclid*	Google Chrome	www.india.com	Travel
172.20.10.10	most beautiful places in india	11/27/2016 9:17	*GET/http://viralstories.in/33-naturally-beautiful-places-india-absolutely-must-visit-die/*	Google Chrome	www.viralstories.in	Travel
172.20.10.10	most beautiful places in shimla	11/27/2016 9:18	*GET/https://www.inspirock.com/india/shimla-trip-planner?gclid*	Google Chrome	www.inspirock.com	Travel
172.20.10.10	most rainy place in world for visiters	11/27/2016 9:28	*GET/https://www.hellotravel.com/stories/10-rainiest-places*	Google Chrome	www.hellotravel.com	Travel
172.20.10.10	most rainy place in india for visiters	11/27/2016 9:29	*GET/https://www.tripoto.com/trip/highest-rainfall-places-in-india*	Google Chrome	www.tripoto.com	Travel
172.20.10.10	travelling jobs in india	11/27/2016 9:31	*GET/https://www.shine.com/job-search/travel-tourism-jobs?akamai_redirect=1&amp;aka_ind*	Google Chrome	www.shine.com	Travel
172.20.10.10	travelling insurance india	11/27/2016 9:31	*GET/https://www.tataaig.com/?gclid*	Google Chrome	www.tataaig.com	Travel
172.20.10.10	family holiday package in india	11/27/2016 9:37	*GET/http://www.manahotels.in/?utm=GoogleAdwords&amp;gclid*	Google Chrome	www.manahotels.in	Travel
172.20.10.10	family holiday package in asia	11/27/2016 9:39	*GET/http://www.backyardtravel.com/tours/?keyword*	Google Chrome	www.backyardtravel.com	Travel
172.20.10.10	christmas holidays package	11/27/2016 9:42	*GET/http://www.getsholidays.in/lp/Australia_packs.html?x*	Google Chrome	www.getsholidays.in	Travel
172.20.10.10	diwali holiday packages	11/27/2016 9:43	*GET/https://holidayz.makemytrip.com/holidays/india?cmp*	Google Chrome	www.holidayz.makemytrip.com	Travel
172.20.10.10	london city tour package	11/27/2016 9:45	*GET/https://www.virginexperiencedays.co.uk/explore-london-with-hop-on-hop-off-sightseeing-bus-tour-and-river-cruise-for-a-family-of-four?gclid*	Google Chrome	www.virginexperiencedays.co.uk	Travel
172.20.10.10	india to dubai tour package	11/27/2016 9:46	*GET/https://www.gofro.com/mkt/destination/dubai-DXB?v*	Google Chrome	www.gofro.com	Travel
172.20.10.10	south india travel package	11/27/2016 9:48	*GET/https://www.travelogyindia.com/south-india/?gclid*	Google Chrome	www.travelogyindia.com	Travel

172.20.10.10	north india travel packages	11/27/2016 9:48	*GET/http://www.rubyholiday.com/landing-page.html*	Google Chrome	www.rubyholiday.com	Travel
172.20.10.10	india vs australia live score	11/27/2016 10:15	*GET/http://www.cricbuzz.com/cricket-match/live-scores*	Google Chrome	www.cricbuzz.com	Sports
172.20.10.10	england vs australia live score	11/27/2016 10:15	*GET/http://www.espn.in/watch/?cmp=Nov_17|Live_Scores/_Eng_vs_Aus|Exact*	Google Chrome	www.espn.in	Sports
172.20.10.10	ashes details	11/27/2016 10:16	*GET/http://www.cricbuzz.com/cricket-series/2538/the-ashes-2016-18?gclid=Cj0KCQiA6enQBRDUARIsAGs1YQjo1IcPobuXcelvr8WVBllfv0WEuwdDC*	Google Chrome	www.cricbuzz.com	Sports
172.20.10.10	paytm cup 2016	11/27/2016 10:18	*GET/http://www.hindustantimes.com/cricket/cricket-calendar-at-a-glance-featuring-ind-sa-aus-ban-sl-pak-nz-wi-afg/story-HDDRr2p3ZbUMQ6Glc4kFTN.html*	Google Chrome	/www.hindustantimes.com	Sports
172.20.10.10	highest chase by india in t20	11/27/2016 10:24	*GET/http://www.howstat.com/cricket/Statistics/Matches/MatchRunChasesHighest_T20.asp*	Google Chrome	www.howstat.com	Sports
172.20.10.10	highest chase by india in odi	11/27/2016 10:25	*GET/https://cricket.yahoo.com/photos/highest-successful-chases-by-india-in-odis-slideshow/*	Google Chrome	cricket.yahoo.com	Sports
172.20.10.10	india gold medals in olympics	11/27/2016 10:27	*GET/http://www.firstpost.com/sports/seeing-indian-hockey-team-win-an-olympic-gold-medal-is-the-last-desire-in-my-life-balbir-singh-sr-2933054.html*	Google Chrome	/www.firstpost.com	Sports
172.20.10.10	china gold medals in olympics	11/27/2016 10:27	*GET/https://en.wikipedia.org/wiki/China_at_the_Olympics*	Google Chrome	en.wikipedia.org	Sports
172.20.10.10	best football player in 2016	11/27/2016 10:37	*GET/http://www.businessinsider.com/the-best-footballers-in-the-world-ronaldo-messi-neymar-2016-8?IR=T*	Google Chrome	www.businessinsider.com	Sports
172.20.10.10	best football player in 2016	11/27/2016 10:37	*GEThttps://www.theguardian.com/football/ng-interactive/2016/dec/20/the-100-best-footballers-in-the-world-2016-interactive*	Google Chrome	www.theguardian.com	Sports
172.20.10.10	games in commonwealth	11/27/2016 10:40	*GEThttps://www.thecgf.com/sports/sports_index.asp*	Google Chrome	www.thecgf.com	Sports
172.20.10.10	games in Olympics	11/27/2016 10:40	*GET/http://www.topendsports.com/events/summer/sports/*	Google Chrome	www.topendsports.com	Sports
172.20.10.10	countries participated in olympics	11/27/2016 10:42	*GET/https://en.wikipedia.org/wiki/List_of_participating_nations_at_the_Summer_Olympic_Games*	Google Chrome	/en.wikipedia.org	Sports
172.20.10.10	country participated in commonwealth games	11/27/2016 10:42	*GET/https://en.wikipedia.org/wiki/Commonwealth_Games*	Google Chrome	en.wikipedia.org	Sports
172.20.10.10	ronaldo total goals	11/27/2016 10:47	*GET/http://messivsronaldo.net/all-time-stats/*	Google Chrome	messivsronaldo.net	Sports

172.20.10.10	messi total goals	11/27/2016 10:47	*GET/http://messivsronaldo.net/all-time-stats/*	Google Chrome	messivsronaldo.net	Sports
172.20.10.10	history of sports	11/27/2016 10:49	*GET/http://www.topendsports.com/resources/history.htm*	Google Chrome	www.topendsports.com	Sports
172.20.10.10	indian sports history	11/27/2016 10:49	*GET/https://en.wikipedia.org/wiki/Sport_in_India*	Google Chrome	en.wikipedia.org	Sports
172.20.10.10	sport in china history	11/27/2016 10:52	*GET/https://en.wikipedia.org/wiki/Sport_in_China*	Google Chrome	en.wikipedia.org	Sports
172.20.10.10	sports in india history	11/27/2016 10:52	*GET/https://en.wikipedia.org/wiki/Sport_in_India*	Google Chrome	en.wikipedia.org	Sports
172.20.10.10	sports	11/27/2016 10:54	*GET/http://www.rediff.com/sports*	Google Chrome	www.rediff.com	Sports
172.20.10.10	cricket	11/27/2016 10:54	*GET/http://www.cricbuzz.com/*	Google Chrome	www.cricbuzz.com	Sports
172.20.10.10	cricket rules	11/27/2016 10:56	*GET/http://cricket-rules.com/*	Google Chrome	cricket-rules.com	Sports
172.20.10.10	football rules	11/27/2016 10:56	*GET/http://www.rulesofsport.com/sports/football.html*	Google Chrome	www.rulesofsport.com	Sports
172.20.10.10	hockey rules	11/27/2016 10:59	*GET/https://www.winnetkahockey.com/index.php/basic-rules-of-hockey*	Google Chrome	www.winnetkahockey.com	Sports
172.20.10.10	vollyball rules	11/27/2016 11:00	*GET/https://www.theartofcoachingvolleyball.com/basic-volleyball-rules-and-terminology/*	Google Chrome	theartofcoachingvolleyball.com	Sports
172.20.10.10	kabadi	11/27/2016 11:03	*GET/https://en.wikipedia.org/wiki/Kabaddi*	Google Chrome	/en.wikipedia.org	Sports
172.20.10.10	last running tea in kabaddi	11/27/2016 11:03	*GET/https://en.wikipedia.org/wiki/Indian_Kabaddi_Team_(men%27s_division)*	Google Chrome	en.wikipedia.org	Sports
172.20.10.10	Golf Rules	11/27/2016 11:05	*GET/http://golf-info-guide.com/golf-rules/?gclid=Cj0KCQiA6enQBRDUARIsAGs1YQjAEodRyEa3KT-U1gM91DojJAd8Ri4oIUZNELseVu7qLMNLRgqqSr0aAoJ0EALw_wcB*	Google Chrome	golf-info-guide.com	Sports
172.20.10.10	Baseball Rules	11/27/2016 11:06	*GET/https://en.wikipedia.org/wiki/Baseball_rules*	Google Chrome	/en.wikipedia.org	Sports
172.20.10.10	india rank in football	11/27/2016 11:09	*GET/http://www.firstpost.com/sports/india-jump-11-places-to-152-in-fifa-rankings-after-back-to-back-wins-over-laos-2894092.html*	Google Chrome	/www.firstpost.com	Sports
172.20.10.10	india rank in cricket	11/27/2016 11:09	*GET/http://www.cricbuzz.com/cricket-stats/icc-rankings?gclid=Cj0KCQiA6enQBRDUARIsAGs1YQjlWw6JWn0-NC_aHZe-yp2wRPpJSTxC1e8RzttJgFVhWxNf8x9Ba20aAvyeEAL*	Google Chrome	www.cricbuzz.com	Sports
172.20.10.10	Boxer in haryana	11/27/2016 11:12	*GET/https://en.wikipedia.org/wiki/Vijender_Singh*	Google Chrome	en.wikipedia.org	Sports
172.20.10.10	Boxer in india	11/27/2016 11:12	*GET/https://en.wikipedia.org/wiki/Boxing_in_India*	Google Chrome	en.wikipedia.org	Sports

172.20.10.10	no of player in kabaddi	11/27/2016 11:15	*GET/http://www.facts-about-india.com/number-of-players-in-sports.php*	Google Chrome	www.facts-about-india.com	Sports
172.20.10.10	no of player in basketball	11/27/2016 11:15	*GET/https://simple.wikipedia.org/wiki/Basketball*	Google Chrome	simple.wikipedia.org	Sports
172.20.10.10	red card in football means	11/27/2016 11:17	*GET/https://en.wikipedia.org/wiki/Fouls_and_misconduct_(association_football)*	Google Chrome	/en.wikipedia.org	Sports
172.20.10.10	red card in hockey means	11/27/2016 11:17	*GET/http://news.bbc.co.uk/sport2/hi/other_sports/hockey/4188396.stm*	Google Chrome	news.bbc.co.uk	Sports
172.20.10.10	no 1 badmintion player in world female	11/27/2016 11:29	*GET/https://en.wikipedia.org/wiki/BWF_World_Ranking*	Google Chrome	en.wikipedia.org	Sports
172.20.10.10	no 1 badmintion player in world male	11/27/2016 11:44	*GET/https://en.wikipedia.org/wiki/BWF_World_Ranking*	Google Chrome	en.wikipedia.org	Sports
172.20.10.10	no 1 badmintion player in india female	11/27/2016 11:44	*GET/https://en.wikipedia.org/wiki/Saina_Nehwal*	Google Chrome	en.wikipedia.org	Sports
172.20.10.10	no 1 badmintion player in india male	11/27/2016 11:44	*GET/https://en.wikipedia.org/wiki/Srikanth_Kidambi*	Google Chrome	en.wikipedia.org	Sports
172.20.10.10	no 1 tennis player in world female	11/27/2016 11:46	*GEThttps://en.wikipedia.org/wiki/List_of_WTA_number_1_ranked_tennis_players*	Google Chrome	en.wikipedia.org	Sports
172.20.10.10	no 1 tennis player in world male	11/27/2016 11:46	*GET/https://en.wikipedia.org/wiki/List_of_ATP_number_1_ranked_singles_tennis_players*	Google Chrome	en.wikipedia.org	Sports
172.20.10.10	no 1 tennis player in india female	11/27/2016 11:47	*GET/https://www.ranker.com/list/famous-tennis-players-from-india/reference*	Google Chrome	www.ranker.com	Sports
172.20.10.10	no 1 tennis player in india male	11/27/2016 11:47	*GET/https://www.ranker.com/list/famous-tennis-players-from-india/reference*	Google Chrome	www.ranker.com	Sports
172.20.10.10	what is ski sports	11/27/2016 11:47	*GET/https://en.wikipedia.org/wiki/Skiing*	Google Chrome	en.wikipedia.org	Sports
172.20.10.10	sports	11/27/2016 11:47	*GEThttps://en.wikipedia.org/wiki/Sport*	Google Chrome	en.wikipedia.org	Sports
172.20.10.10	largest capacity stadium in india	11/27/2016 11:47	*GET/https://en.wikipedia.org/wiki/List_of_cricket_grounds_by_capacity*	Google Chrome	en.wikipedia.org	Sports
172.20.10.10	largest capacity stadium in australia	11/27/2016 11:47	*GET/https://en.wikipedia.org/wiki/List_of_cricket_grounds_by_capacity*	Google Chrome	en.wikipedia.org	Sports
192.168.43.167	orange juice	01/02/2017 03:59	*GET/https://www.bigbasket.com/pc/beverages/fruit-drinks-juices/orange-apple-juices/*	Mozilla Firefox	https://www.bigbasket.com/	Food
192.168.43.167	Real juice	01/03/2017 04:00	*GET/http://www.realfruitpower.com/*	Mozilla Firefox	http://www.realfruitpower	Food

					.com/	
192.168.43.167	Tropicana juice	01/04/2017 04:01	*GET/http://www.tropicana.com/*	Mozilla Firefox	http://www.tropicana.com/	Food
192.168.43.167	Tropicana juice	01/05/2017 04:02	*GET/https://www.amazon.in/Tropicana-Orange-100-Juice-1000ml/dp/B00QPS8KW6*	Mozilla Firefox	https://www.amazon.in/	Food
192.168.43.167	fruit juice	01/06/2017 04:03	*GET/https://www.amazon.in/Fruit-Juice/b?ie=UTF8&node=4859554031*	Mozilla Firefox	https://www.amazon.in/	Food
192.168.43.167	paperboat juice	01/07/2017 04:04	*GET/http://www.paperboatdrinks.com/drinks*	Mozilla Firefox	http://www.paperboatdrinks.com/	Food
192.168.43.167	paperboat juice	01/08/2017 05:10	*GET/https://www.amazon.in/Paper-Boat-Aamras-Juice-250ml/dp/B00RLHKJCO*	Mozilla Firefox	https://www.amazon.in/	Food
192.168.43.167	cold drinks	01/09/2017 05:12	*GET/https://www.bigbasket.com/pc/beverages/soft-drinks/*	Mozilla Firefox	https://www.bigbasket.com/	Food
192.168.43.167	soft drinks	01/10/2017 05:13	*GET/https://www.bigbasket.com/pc/beverages/soft-drinks/*	Mozilla Firefox	https://www.bigbasket.com/	Food
192.168.43.167	red bull	01/11/2017 05:20	*GET/http://energydrink-in.redbull.com/red-bull-energy-drink*	Mozilla Firefox	http://energydrink-in.redbull.com/	Food
192.168.43.167	amul milk	01/12/2017 05:23	*GET/http://www.amul.com/products/amul-cow-milk-info.php*	Mozilla Firefox	http://www.amul.com/	Food
192.168.43.167	amul cheese	1/13/2017 5:24	*GET/http://www.amul.com/products/cheese.php*	Mozilla Firefox	http://www.amul.com/	Food
192.168.43.167	amul cheese	1/14/2017 5:25	*GET/https://www.bigbasket.com/pb/amul/cheese/*	Mozilla Firefox	https://www.bigbasket.com/	Food
192.168.43.167	amul butter	1/15/2017 5:40	*GET/http://www.amul.com/products/amul-tablebutter-info.php*	Mozilla Firefox	http://www.amul.com/	Food
192.168.43.167	amul butter	1/16/2017 1:20	*GET/https://dir.indiamart.com/impcat/amul-butter.html*	Mozilla Firefox	https://dir.indiamart.com/	Food
192.168.43.167	amul ice cream	1/17/2017 1:20	*GET/http://www.amul.com/products/icecream.php*	Mozilla Firefox	http://www.amul.com/	Food
192.168.43.167	amul ice cream	1/18/2017 1:20	*GET/https://www.bigbasket.com/pb/amul/ice-creams/*	Mozilla Firefox	https://www.bigbasket.com/	Food
192.168.43.167	amul cream	1/19/2017 1:20	*GET/http://www.amul.com/products/freshcream.php*	Mozilla Firefox	http://www.amul.com/	Food
192.168.43.167	amul cream	1/20/2017 1:20	*GET/https://dir.indiamart.com/impcat/amul-cream.html*	Mozilla Firefox	https://dir.indiamart.com/	Food
192.168.43.167	Mojito	1/21/2017 5:40	*GET/https://www.thespruce.com/mojito-cocktail-recipe-759319*	Mozilla Firefox	https://www.thespruce.com/	Food
192.168.43.167	brownie sundae	1/22/2017 8:30	*GET/http://www.foodnetwork.com/recipes/ina-garten/brownie-sundaes-recipe-1941095*	Mozilla Firefox	http://www.foodnetwork.com/	Food

192.168.43.167	brownie cake	1/23/2017 1:20	*GET/https://www.bbcgoodfood.com/recipes/2882/chocolate-brownie-cake-*	Mozilla Firefox	https://www.bbcgoodfood.com/	Food
192.168.43.167	brownie fudge	1/24/2017 1:20	*GET/http://allrecipes.com/recipe/9566/fudge-brownies-i/*	Mozilla Firefox	http://allrecipes.com/	Food
192.168.43.167	brownie trifle	1/25/2017 1:20	*GET/http://www.geniuskitchen.com/recipe/brownie-trifle-16028*	Mozilla Firefox	http://www.geniuskitchen.com/	Food
192.168.43.167	chocolava	1/26/2017 8:35	*GET/http://www.vegrecipesofindia.com/eggless-choco-lava-cake-recipe/*	Mozilla Firefox	http://www.vegrecipesofindia.com/	Food
192.168.43.167	eggless chocolava	1/27/2017 8:35	*GET/http://www.vegrecipesofindia.com/eggless-choco-lava-cake-recipe/*	Mozilla Firefox	http://www.vegrecipesofindia.com/	Food
192.168.43.167	dominos chocolava	1/28/2017 8:35	*GET/https://www.dominos.co.in/menu/side-orders/lava-cake*	Mozilla Firefox	https://www.dominos.co.in/	Food
192.168.43.167	pickles	1/29/2017 8:35	*GET/https://www.placeoforigin.in/staples/pickles-sauces*	Mozilla Firefox	https://www.placeoforigin.in/	Food
192.168.43.167	mango pickle	1/30/2017 9:48	*GET/https://indianhealthyrecipes.com/mango-pickle-recipe/*	Mozilla Firefox	https://indianhealthyrecipes.com/	Food
192.168.43.167	lemon pickle	1/31/2017 9:48	*GET/http://www.vegrecipesofindia.com/easy-lemon-pickle-recipe/*	Mozilla Firefox	http://www.vegrecipesofindia.com/	Food
192.168.43.167	south indian food	02/01/2017 09:48	*GET/http://food.ndtv.com/lists/10-best-south-indian-recipes-736459*	Mozilla Firefox	http://food.ndtv.com/	Food
192.168.43.167	gujrati food	02/02/2017 09:48	*GET/http://food.ndtv.com/lists/10-best-gujarati-recipes-695837*	Mozilla Firefox	http://food.ndtv.com/	Food
192.168.43.167	Popcorn	02/03/2017 09:48	*GET/https://www.bigbasket.com/pc/branded-foods/veg-snacks/popcorn/*	Mozilla Firefox	https://www.bigbasket.com/	Food
192.168.43.167	cheese popcorn	02/04/2017 11:32	*GET/https://www.thespruce.com/real-cheddar-cheese-popcorn-recipe-2098775*	Mozilla Firefox	https://www.thespruce.com/	Food
192.168.43.167	chocolate popcorn	02/05/2017 11:32	*GET/http://allrecipes.com/recipe/160595/chocolate-popcorn/*	Mozilla Firefox	http://allrecipes.com/	Food
192.168.43.167	fast food	02/06/2017 11:32	*GET/https://www.fastfoodmenuprices.com/11-fast-food-restaurants-open-thanksgiving-day/*	Mozilla Firefox	https://www.fastfoodmenuprices.com/	Food
192.168.43.167	street food	02/07/2017 11:32	*GET/https://www.roughguides.com/gallery/the-best-street-food-around-the-world/*	Mozilla Firefox	https://www.roughguides.com/	Food
192.168.43.167	Sugar candy	02/08/2017 11:32	*GET/https://www.indiamart.com/proddetail/sugar-coated-jelly-soft-candy-10748817855.html*	Mozilla Firefox	https://www.indiamart.com/	Food
192.168.43.167	Diet food	02/09/2017 11:32	*GET/https://www.healthline.com/nutrition/20-most-weight-loss-friendly-foods*	Mozilla Firefox	https://www.healthline.com/	Food
192.168.43.167	french fries	02/10/2017 02:18	*GET/http://allrecipes.com/recipe/219634/chef-johns-french-fries/*	Mozilla Firefox	http://allrecipes.com/	Food

192.168.43.167	chinese food	02/11/2017 02:18	*GET/https://www.zomato.com/ncr/restaurants/chinese*	Mozilla Firefox	https://www.zomato.com/	Food
192.168.43.167	Energy drinks	02/12/2017 02:18	*GET/https://www.caffeineinformer.com/top-10-energy-drink-dangers*	Mozilla Firefox	https://www.caffeineinformer.com/	Food
192.168.43.167	sports drink	2/13/2017 2:18	*GET/http://www.bestproducts.com/fitness/health/g734/electrolyte-sports-drinks/*	Mozilla Firefox	http://www.bestproducts.com/	Food
192.168.43.167	fruit jam	2/14/2017 2:18	*GET/http://food.ndtv.com/recipe-mixed-fruit-jam-218684*	Mozilla Firefox	http://food.ndtv.com/	Food
192.168.43.167	apple jam	2/15/2017 2:18	*GET/http://nishamadhulika.com/en/427-apple-jam-recipe.html*	Mozilla Firefox	http://nishamadhulika.com/	Food
192.168.43.167	dry fruits	2/16/2017 2:18	*GET/https://www.placeoforigin.in/snacks/dry-fruits*	Mozilla Firefox	https://www.placeoforigin.in/	Food
192.168.43.167	sweets	2/17/2017 2:18	*GET/https://www.placeoforigin.in/sweets-confectionery/sweets*	Mozilla Firefox	https://www.placeoforigin.in/	Food
192.168.43.167	Snacks	2/18/2017 7:14	*GET/http://food.ndtv.com/recipes/snacks-recipes*	Mozilla Firefox	http://food.ndtv.com/	Food
192.168.43.167	biscuits	2/19/2017 7:14	*GET/http://www.pauladeen.com/biscuits*	Mozilla Firefox	http://www.pauladeen.com/	Food
192.168.43.167	cream biscuit	2/20/2017 7:14	*GET/http://www.foodnetwork.com/recipes/paula-deen/cream-biscuits-recipe-1941899*	Mozilla Firefox	http://www.foodnetwork.com/	Food
192.168.43.167	chocolate biscuit	2/21/2017 7:14	*GET/http://allrecipes.co.uk/recipe/29760/easy-chocolate-biscuits.aspx*	Mozilla Firefox	http://allrecipes.co.uk/	Food
192.168.43.167	green tea	2/22/2017 7:14	*GET/https://www.medicalnewstoday.com/articles/269538.php*	Mozilla Firefox	https://www.medicalnewstoday.com/	Food
192.168.43.167	herbal tea	2/23/2017 7:14	*GET/http://www.besthealthmag.ca/best-eats/nutrition/7-herbal-teas-that-will-make-you-healthy/*	Mozilla Firefox	http://www.besthealthmag.ca/	Food
192.168.43.167	fresh fruits	2/24/2017 7:14	*GET/https://www.bigbasket.com/pc/fruits-vegetables/fresh-fruits/*	Mozilla Firefox	https://www.bigbasket.com/	Food
192.168.43.167	Summer Vacation	2/25/2017 7:14	*GET/https://www.indianholiday.com/summer-destination.html*	Mozilla Firefox	https://www.indianholiday.com/	Travel
192.168.43.167	Winter Vacation	2/26/2017 7:14	*GET/https://www.tripstodiscover.com/18-amazing-and-affordable-winter-vacation-destinations*	Mozilla Firefox	https://www.tripstodiscover.com/	Travel
192.168.43.167	Cultural Vacation	2/27/2017 4:10	*GET/http://www.culturalvacations.com*	Mozilla Firefox	http://www.culturalvacations.com/	Travel
192.168.43.167	Honeymoon Destination	2/28/2017 4:10	*GET/https://www.theknot.com/content/best-honeymoon-destinations *	Mozilla Firefox	https://www.theknot.com/	Travel
192.168.43.167	Adventure Destination	03/01/2017 04:10	*GET/http://www.adventuredestinations.com.au*	Mozilla Firefox	http://www.adventuredestinations.com.au	Travel

192.168.43.167	Spiritual Destination	03/02/2017 04:10	*GET/https://www.tourmyindia.com/blog/top-25-religious-tourism-places-in-india*	Mozilla Firefox	www.tourmyindia.com/	Travel
192.168.43.167	Family Destination	03/03/2017 04:10	*GET/https://www.fodors.com/trip-ideas/family*	Mozilla Firefox	https://www.fodors.com/	Travel
192.168.43.167	Historical Destination	03/04/2017 04:10	*GET/http://traveltriangle.com/blog/famous-historical-places-in-india/*	Mozilla Firefox	http://traveltriangle.com/	Travel
192.168.43.167	Shopping Destination	03/05/2017 04:10	*GET/http://www.tourisme-metz.com/en/a-shopping-destination.html#.WhOcT1WWbIU/*	Mozilla Firefox	http://www.tourisme-metz.com/	Travel
192.168.43.167	Hill Area	03/06/2017 06:36	*GET/https://www.tourmyindia.com/blog/top-35-hill-stations-in-india/*	Mozilla Firefox	https://www.tourmyindia.com/	Travel
192.168.43.167	Coastal Area	03/07/2017 06:36	*GET/https://www.makemytrip.com/blog/top-10-beach-holidays-india*	Mozilla Firefox	https://www.makemytrip.com/	Travel
192.168.43.167	Food Available	03/08/2017 06:36	*GET/http://www.travelandleisure.com/slideshows/the-foodies-travel-bucket-list *	Mozilla Firefox	http://www.travelandleisure.com/	Travel
192.168.43.167	Vehicle Available	03/09/2017 06:36	*GET/https://www.savaari.com/*	Mozilla Firefox	https://www.savaari.com/	Travel
192.168.43.167	Five Star Hotel	03/10/2017 06:36	*GET/http://www.luxurylink.com/5star/hotel-deals/caribbean*	Mozilla Firefox	http://www.luxurylink.com/	Travel
192.168.43.167	Four Star Hotel	03/11/2017 06:36	*GET/https://www.tripadvisor.in/Hotels-g150807-zfc4-Cancun_Yucatan_Peninsula-Hotels.html*	Mozilla Firefox	https://www.tripadvisor.in	Travel
192.168.43.167	Travel offers	03/12/2017 06:36	*GET/https://www.tripadvisor.in/Hotels-g150807-zfc4-Cancun_Yucatan_Peninsula-Hotels.html*	Mozilla Firefox	https://www.tripadvisor.in/	Travel
192.168.43.167	Travel Coupons	3/13/2017 6:36	*GET/http://www.grabon.in/travel-coupons/*	Mozilla Firefox	http://www.grabon.in/	Travel
192.168.43.167	HOLIDAY PACKAGE	3/14/2017 4:52	*GET/http://www.thomascook.in/tcportal/india-holidays*	Mozilla Firefox	http://www.thomascook.in/	Travel
192.168.43.167	Bus Booking	3/15/2017 4:52	*GET/https://www.makemytrip.com/bus-tickets/laxmi-holidays-booking.html*	Mozilla Firefox	https://www.makemytrip.com/	Travel
192.168.43.167	Flight Booking	3/16/2017 4:52	*GET/http://www.ezeego1.co.in/*	Mozilla Firefox	http://www.ezeego1.co.in/	Travel
192.168.43.167	International Flight	3/17/2017 4:52	*GET/https://www.yatra.com/international-flights/*	Mozilla Firefox	https://www.yatra.com/	Travel
192.168.43.167	Domestic Flight	3/18/2017 4:52	*GET/https://www.yatra.com/domestic-flights/*	Mozilla Firefox	https://www.yatra.com/	Travel
192.168.43.167	Manali Trip	3/19/2017 4:52	*GET/http://www.himachaltourpackagesonline.com/*	Mozilla Firefox	http://www.himachaltourpackagesonline.com/	Travel
192.168.43.167	Shimla Trip	3/20/2017 4:52	*GET/https://www.yatra.com/india-tour-packages/holidays-in-shimla*	Mozilla Firefox	https://www.yatra.com/	Travel

192.168.43.167	kashmir Trip	3/21/2017 4:52	*GET/http://www.srinagartourpackages.net.in/*	Mozilla Firefox	http://www.srinagartourpackages.net.in/	Travel
192.168.43.167	kerala Trip	3/22/2017 9:24	*GET/http://inventiveholidays.com/Holiday/kerala-tour-packages/*	Mozilla Firefox	http://inventiveholidays.com/	Travel
192.168.43.167	US Trip	3/23/2017 9:24	*GET/https://www.yatra.com/international-tour-packages/holidays-in-usa*	Mozilla Firefox	https://www.yatra.com/	Travel
192.168.43.167	Chennai Trip	3/24/2017 9:24	*GET/https://www.yatra.com/*	Mozilla Firefox	https://www.yatra.com/	Travel
192.168.43.167	Australia Trip	3/25/2017 9:24	*GET/https://www.yatra.com/international-tour-packages/holidays-in-australia*	Mozilla Firefox	https://www.yatra.com/	Travel
192.168.43.167	Singapore Trip	3/26/2017 9:24	*GET/https://www.yatra.com/international-tour-packages/holidays-in-singapore*	Mozilla Firefox	https://www.yatra.com/	Travel
192.168.43.167	Macau Trip	3/27/2017 9:24	*GET/https://www.tripadvisor.in/Tourism-g664891-Macau-Vacations.html*	Mozilla Firefox	https://www.tripadvisor.in/	Travel
192.168.43.167	Sea Diving	3/28/2017 9:24	*GET/http://www.india.com/travel/articles/5-best-scuba-diving-destinations-in-india/*	Mozilla Firefox	http://www.india.com/	Travel
192.168.43.167	Trekking	3/29/2017 9:24	*GET/https://www.tourmyindia.com/treks/*	Mozilla Firefox	https://www.tourmyindia.com/	Travel
192.168.43.167	Camping	3/30/2017 9:24	*GET/https://www.indianholiday.com/tour-packages-india/camping.html*	Mozilla Firefox	https://www.indianholiday.com/	Travel
192.168.43.167	Gujarat Trip	3/31/2017 2:06	*GET/https://www.yatra.com/india-tour-packages/holidays-in-gujarat*	Mozilla Firefox	https://www.yatra.com/	Travel

The table A 1.2 shows the list of terms and their frequency in documents related to URLs taken in table 4.9 under section 4,3,2,1

Table A 1.2: List of Terms Obtained by Page Extractor for URLs given in Table 4.9

Terms	Query	URL1	URL2	URL3
homogeneous	0	1	0	0
transparent	0	1	0	0
medium	0	1	0	2
Light	1	4	19	4
travel	0	2	7	
straight	0	3	4	2
line	0	3	2	2
path	0	1	1	0
know	0	1	1	0
rectilinear	1	2	6	1
propgate	1	2	5	1
demostrate	0	1	1	0
follow	0	1	0	0
experiment	0	1	0	1
take	0	1	0	0
three	0	3	0	1
B	0	1	0	0
cardboard	0	5	0	3

make	0	1	2	0
pinole	0	4	0	0
center	0	1	0	1
place	0	1	0	0
burn	0	1	0	0
candle	0	2	0	0
one	0	2	0	1
side	0	1	0	1
arrange	0	1	0	0
way	0	1	0	0
flame	0	3	0	0
visible	0	1	0	0
C	0	2	0	0
slightly	0	1	0	0
displace	0	1	0	0
try	0	1	0	0
see	0	1	0	3
clear	0	1	0	0
example	0	1	2	0
define	0	0	3	0
lesson	0	0	3	0
5	0	0	1	0
instruct	0	0	1	0
Mathew	0	0	1	0
Berstresser	0	0	1	0
2484	0	0	1	0
view	0	0	1	0
source	0	0	1	0
emit	0	0	1	0
direct	0	0	1	0
ray	0	0	3	0
learn	0	0	1	0
specific	0	0	1	0
compare	0	0	1	0
motion	0	0	3	0
object	0	0	2	0
remember	0	0	1	0
be	0	0	2	0
school	0	0	1	0
teach	0	0	1	0
projrct	0	0	4	0
something	0	0	2	0
screen	0	0	1	0
kid	0	0	1	0
love	0	0	1	0
experience	0	0	1	0
shadow	0	0	4	0
animal	0	0	2	0
hand	0	0	1	0
contort	0	0	1	0
giraffe	0	0	1	0
cat	0	0	1	0
actual	0	0	4	0
tell	0	0	2	0
mean	0	0	1	0
Unique	0	0	1	0
phenomenon	0	0	1	0
Investigate	0	0	1	0
Further	0	0	1	0
Understand	0	0	1	0
Think	0	0	2	0
football	0	0	3	0
Throw	0	0	1	0
Archer	0	0	4	0
Air	0	0	2	0
Lunch	0	0	2	0
Arrow	0	0	4	0
Target	0	0	2	0
50	0	0	1	0
Meter	0	0	1	0
Away	0	0	1	0

Arrive	0	0	1	0
Aim	0	0	2	0
Bull	0	0	4	0
Eye	0	0	3	0
Strike	0	0	2	0
Below	0	0	1	0
Pull	0	0	1	0
Gravity	0	0	1	0
Down	0	0	1	0
Entire	0	0	1	0
Length	0	0	1	0
Same	0	0	3	0
Happen	0	0	1	0
Mass	0	0	1	0
Completely	0	0	1	0
Differ	0	0	2	0
Laser	0	0	1	0
Beam	0	0	1	0
Wavelength	0	0	1	0
Align	0	0	1	1
Point	0	0	1	0
Distance	0	0	1	0
Hit	0	0	2	1
Like	0	0	2	0
Dual	0	0	2	0
Nature	0	0	2	0
Say	0	0	1	0
Property	0	0	1	0
Particle	0	0	5	0
wave	0	0	5	6
Average	0	0	1	0
Discuss	0	0	2	0
Water	0	0	3	0
Sound	0	0	1	0
Lake	0	0	1	0
Fill	0	0	1	0
Little	0	0	1	0
Island	0	0	1	0
Pier	0	0	1	0
Boat	0	0	1	0
Stop	0	0	1	0
Instant	0	0	1	0
Barrier	0	0	2	0
Bend	0	0	2	1
Wall	0	0	1	1
Hear	0	0	1	1
People	0	0	1	1
Talk	0	0	1	1
Room	0	0	2	1
Shine	0	0	1	1
Hold	0	0	1	1
Flashlight	0	0	1	1
Probable	0	0	1	1
Corner	0	0	1	1
Camera	0	0	1	1
Film	0	0	1	1
Image	0	0	1	1
Produce	0	0	1	1
Trillion	0	0	1	1
Imprint	0	0	1	1
Tiny	0	0	1	1
Recognize	0	0	1	1
Proof	0	0	1	1
Color	0	0	1	1
Block	0	0	1	1
Create	0	0	1	1
Account	0	0	1	1
Start	0	0	1	1
Course	0	0	1	1
Explore	0	0	1	1
Library	0	0	1	1

7000	0	0	1	1
Page	0	0	0	1
Issue	0	0	0	1
Electromagnetic	0	0	0	1
Front	0	0	0	1
Create	0	0	0	1
Pond	0	0	0	1
Rock	0	0	0	1
Individual	0	0	0	1
Move	0	0	0	1
Sense	0	0	0	1
Scatter	0	0	0	1
Inhomogeneous	0	0	0	1
Case	0	0	0	1
N	0	0	0	1
1	0	0	0	1
Refract	0	0	0	1
Index	0	0	0	1
Natter	0	0	0	1
Square	0	0	0	1
Small	0	0	0	1
Hole	0	0	0	1
Behind	0	0	0	1
Set	0	0	0	1
Look	0	0	0	1
Bit	0	0	0	1
Long	0	0	0	1
Able	0	0	0	1
Explain	0	0	0	1
Help	0	0	0	1
Human	0	0	0	1
Apply	0	0	0	1
Number	0	0	0	1
Real	0	0	0	1
Life	0	0	0	1

APPENDIX -B

Query Suggestion based on Semantic Ssimilarity had been implemented using JDK 1.8, Eclipse Indigo, Apache Tomcat 7.0 with oracle at the at the back end. To support definition _repository English lexical dictionary WordNet 3.1 is used to derive the various semantic meaning of term .WS4Jdemo tool is used to find the number of nodes between two terms based on the position in is-a hierarchy of WordNet.

The user interface of QSSS is depicted in Fig B1.1 where the user can submit its information need in the form of query.

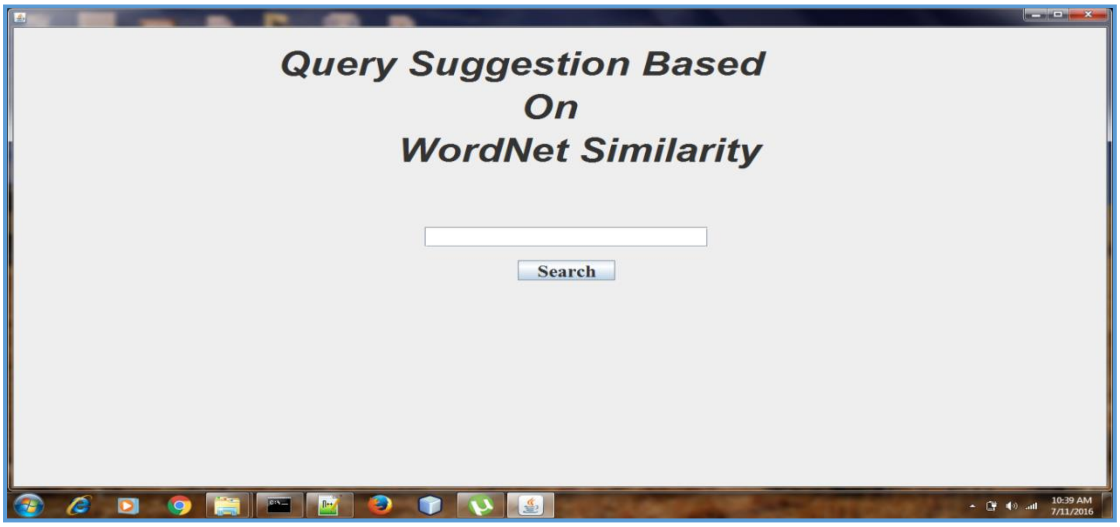
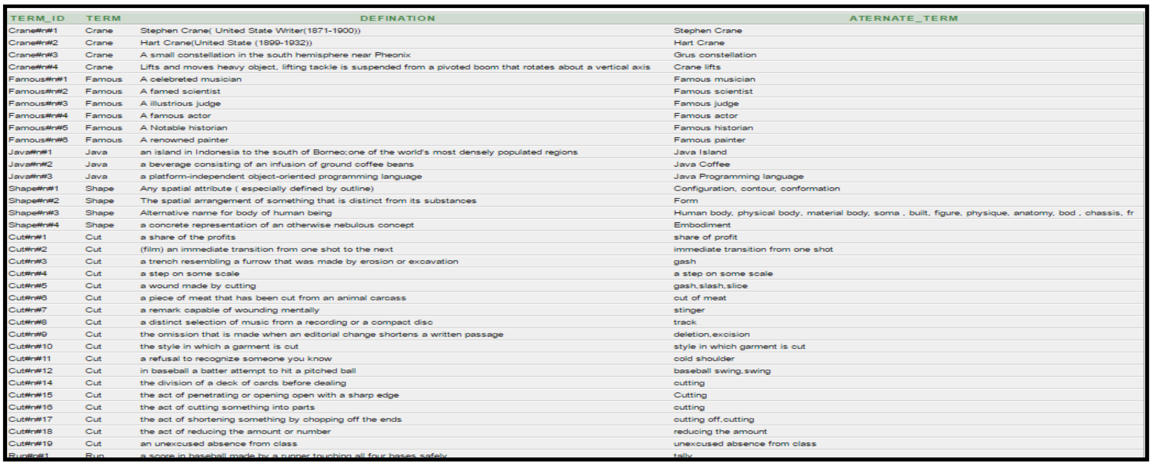


Fig B 1.1: Home Page of QSSS (WordNet)

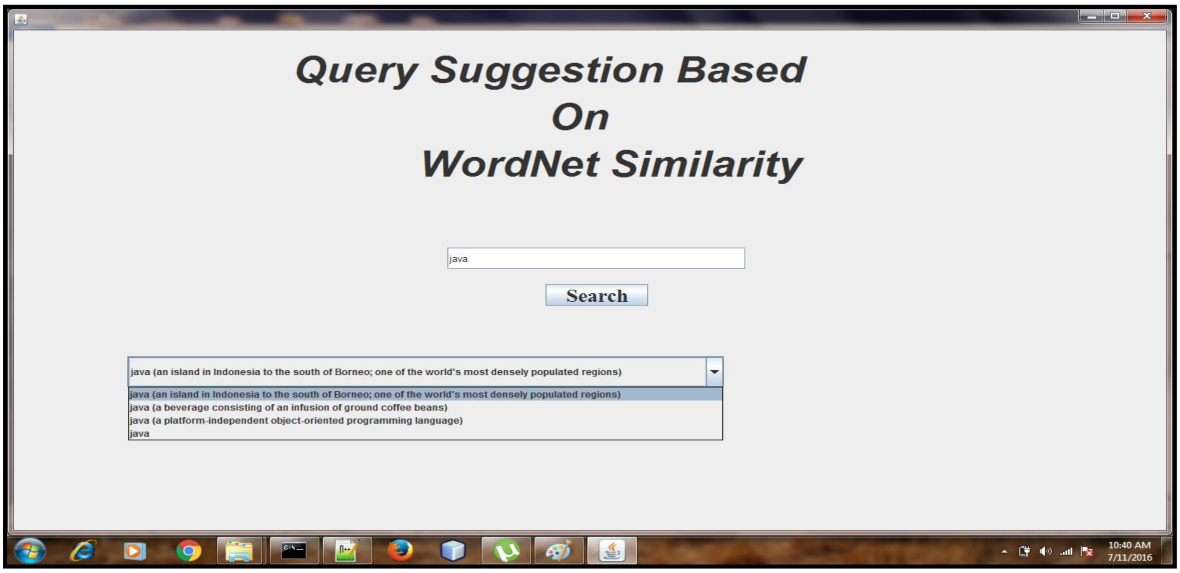
The snapshot of definition _repository maintained using oracle is shown in Fig B1.2.



TERM_ID	TERM	DEFINATION	ATERNATE_TERM
Crane#n#1	Crane	Stephen Crane(United State Writer(1871-1900))	Stephen Crane
Crane#n#2	Crane	Hart Crane(United State (1899-1932))	Hart Crane
Crane#n#3	Crane	A small constellation in the south hemisphere near Pheonix	Grus constellation
Crane#n#4	Crane	Lifts and moves heavy object, lifting tackle is suspended from a pivoted boom that rotates about a vertical axis	Crane lifts
Famous#n#1	Famous	A celebreted musician	Famous musician
Famous#n#2	Famous	A famed scientist	Famous scientist
Famous#n#3	Famous	A illustrious judge	Famous judge
Famous#n#4	Famous	A famous actor	Famous actor
Famous#n#5	Famous	A Notable historian	Famous historian
Famous#n#6	Famous	A renowned painter	Famous painter
Java#n#1	Java	an island in Indonesia to the south of Borneo;one of the world's most densely populated regions	Java Island
Java#n#2	Java	a beverage consisting of an infusion of ground coffee beans	Java Coffee
Java#n#3	Java	a platform-independent object-oriented programming language	Java Programming language
Shape#n#1	Shape	Any spatial attribute (especially defined by outline)	Configuration, contour, conformation
Shape#n#2	Shape	The spatial arrangement of something that is distinct from its substances	Form
Shape#n#3	Shape	Alternative name for body of human being	Human body, physical body, material body, soma , built, figure, physique, anatomy, bod , chassis, fr
Shape#n#4	Shape	a concrete representation of an otherwise nebulous concept	Embodiment
Cut#n#1	Cut	a share of the profits	share of profit
Cut#n#2	Cut	(film) an immediate transition from one shot to the next	immediate transition from one shot
Cut#n#3	Cut	a trench resembling a furrow that was made by erosion or excavation	gash
Cut#n#4	Cut	a step on some scale	a step on some scale
Cut#n#5	Cut	a wound made by cutting	gash,slash,slice
Cut#n#6	Cut	a piece of meat that has been cut from an animal carcass	cut of meat
Cut#n#7	Cut	a remark capable of wounding mentally	stinger
Cut#n#8	Cut	a distinct selection of music from a recording or a compact disc	track
Cut#n#9	Cut	the omission that is made when an editorial change shortens a written passage	deletion,excision
Cut#n#10	Cut	the style in which a garment is cut	style in which garment is cut
Cut#n#11	Cut	a refusal to recognize someone you know	cold shoulder
Cut#n#12	Cut	in baseball a batter attempt to hit a pitched ball	baseball swing,swing
Cut#n#14	Cut	the division of a deck of cards before dealing	cutting
Cut#n#15	Cut	the act of penetrating or opening open with a sharp edge	Cutting
Cut#n#16	Cut	the act of cutting something into parts	cutting
Cut#n#17	Cut	the act of shortening something by chopping off the ends	cutting off,cutting
Cut#n#18	Cut	the act of reducing the amount or number	reducing the amount
Cut#n#19	Cut	an unexcused absence from class	unexcused absence from class
Run#n#1	Run	a score in baseball made by a runner touching all four bases safely	tally

Fig B 1.2: Definition_Repository

The various alternate queries for the query ***'Java'*** obtained when the query is submitted on the QSSS interface is shown in Fig A 1.3. It can be observed that QSSS offers different context of user query along with the main query such as Java Island. Java coffee and Java programming language. It also displays the related semantic definition of each alternate term included in parenthesis in Fig B 1.3.



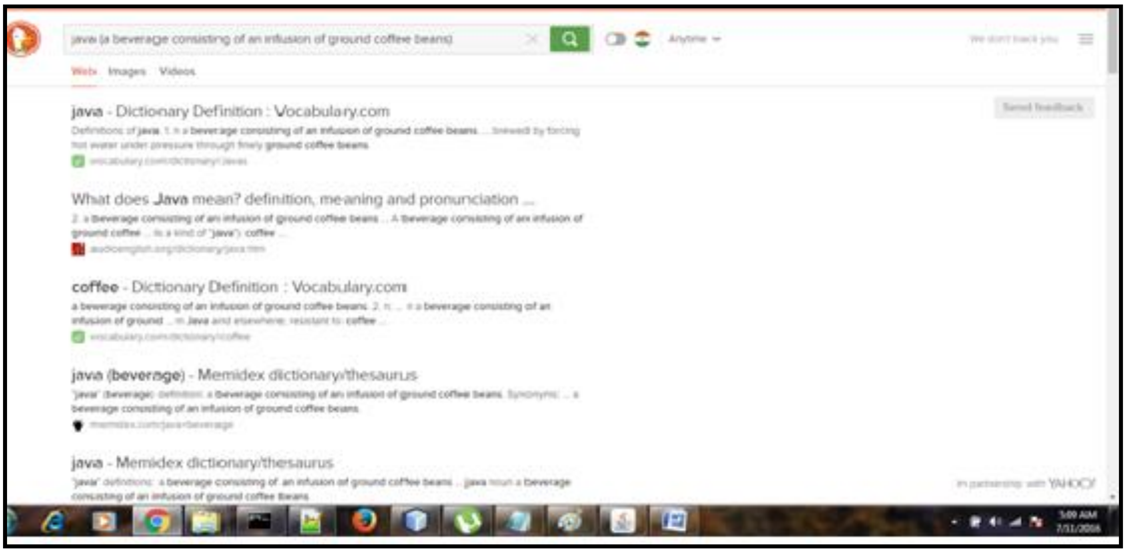
Query Suggestion Based On WordNet Similarity

java

java (an island in Indonesia to the south of Borneo; one of the world's most densely populated regions)
java (an island in Indonesia to the south of Borneo; one of the world's most densely populated regions)
java (a beverage consisting of an infusion of ground coffee beans)
java (a platform-independent object-oriented programming language)
java

Fig B 1.3: Query Assistance Provided by QSSS (WordNet)

The result set obtained after selecting the query 'Java Coffee' and 'Jana programs language by the user at QSSS is shown in Fig B1.4 and Fig. B 1.5.



java (a beverage consisting of an infusion of ground coffee beans)

java - Dictionary Definition : Vocabulary.com
Definitions of java. 1. n a beverage consisting of an infusion of ground coffee beans ... brewed by forcing hot water under pressure through finely ground coffee beans
vocabulary.com/dictionary/java

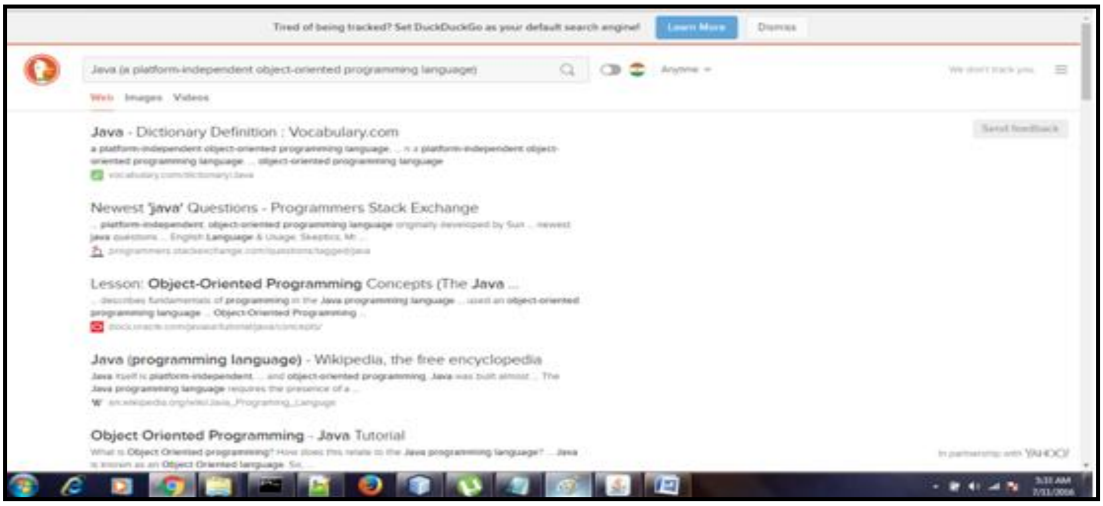
What does Java mean? definition, meaning and pronunciation ...
2. a beverage consisting of an infusion of ground coffee beans ... A beverage consisting of an infusion of ground coffee ... is a kind of 'java') coffee ...
audioenglish.org/dictionary/java.htm

coffee - Dictionary Definition : Vocabulary.com
a beverage consisting of an infusion of ground coffee beans. 2. n. ... n a beverage consisting of an infusion of ground ... in Java and elsewhere; resistant to coffee ...
vocabulary.com/dictionary/coffee

java (beverage) - Memidex dictionary/thesaurus
'java' (beverage) definition: a beverage consisting of an infusion of ground coffee beans. Synonyms: ... a beverage consisting of an infusion of ground coffee beans
memidex.com/java+beverage

java - Memidex dictionary/thesaurus
'java' definitions: a beverage consisting of an infusion of ground coffee beans ... java noun a beverage consisting of an infusion of ground coffee beans

Fig B 1.4: Result Screen for the selected option 'Java coffee'



Tired of being tracked? Set DuckDuckGo as your default search engine!
Learn More
Dismiss
Anytime
We don't track you
Web
Images
Videos
Send feedback
In partnership with YAHOO!
5:31 AM
7/31/2016

Java (a platform-independent object-oriented programming language)

Java - Dictionary Definition : Vocabulary.com
a platform-independent object-oriented programming language. ... n a platform-independent object-oriented programming language ... object-oriented programming language
vocabulary.com/dictionary/Java

Newest 'java' Questions - Programmers Stack Exchange
... platform-independent, object-oriented programming language originally developed by Sun ... newest java questions ... English Language & Usage, Skeptics, Mi ...
programmers.stackexchange.com/questions/tagged/java

Lesson: Object-Oriented Programming Concepts (The Java ...
... describes fundamentals of programming in the Java programming language ... used an object-oriented programming language ... Object-Oriented Programming ...
docs.oracle.com/javase/tutorial/java/concepts/

Java (programming language) - Wikipedia, the free encyclopedia
Java itself is platform-independent, ... and object-oriented programming. Java was built almost ... The Java programming language requires the presence of a ...
en.wikipedia.org/wiki/Java_Programming_Language

Object Oriented Programming - Java Tutorial
What is Object Oriented programming? How does this relate to the Java programming language? ... Java is known as an Object Oriented language. So, ...

Fig B 1.5: Result Screen for the selected option 'Java programming language'

APPENDIX-C

Another query suggestion technique called Query suggestion based on user browsing history (QSUB) discussed in section 3.3 is implemented. At the front end HTML is used to implement user interface while C#.NET is used to store data in MS SQL Server 2012 at back end. QSUB algorithm has been tested on the query log given in Appendix A. The user interface of QSUB is depicted in Fig C 1.1.

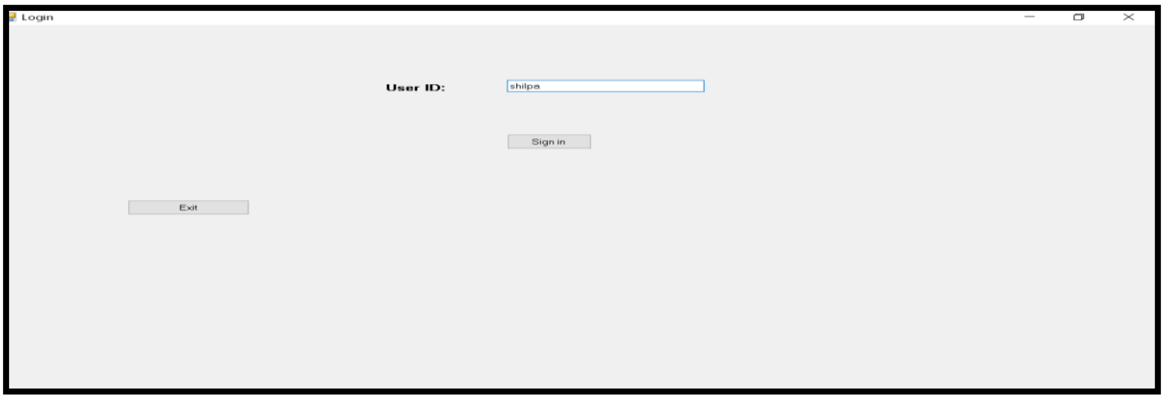
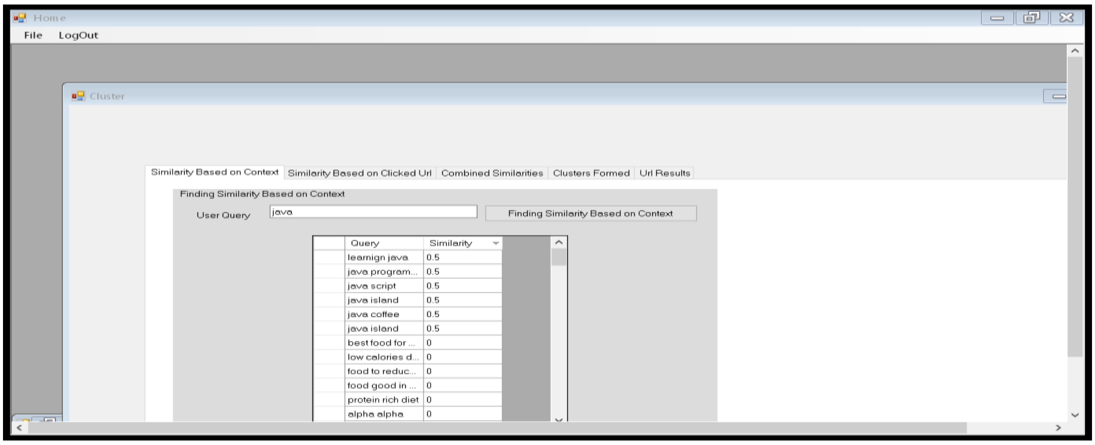


Fig C1.1: User interface of QSUB

In order to search information through QSUB, the user is first asked to login into the system. The step is required to create each individual profile for personalized query suggestions. The next step is to submit the information need in the form of query as shown in Fig C 1.2.

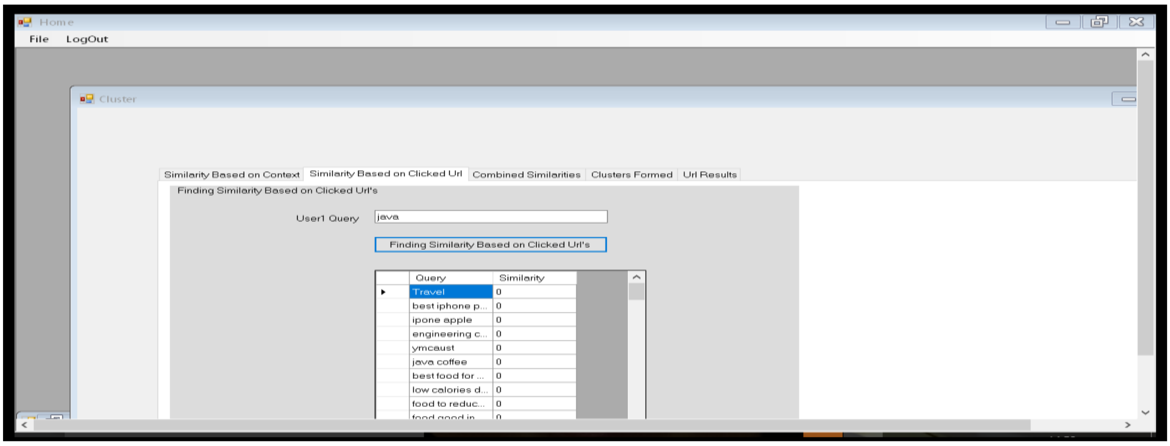

Home
File
LogOut
Cluster
Similarity Based on Context
Similarity Based on Clicked Url
Combined Similarities
Clusters Formed
Url Results
Finding Similarity Based on Context
Finding Similarity Based on Context

User Query: java

Query	Similarity
learnign java	0.5
java program...	0.5
java script	0.5
java island	0.5
java coffee	0.5
java island	0.5
best food for ...	0
low calories d...	0
food to reduc...	0
food good in ...	0
protein rich diet	0
alpha alpha	0

Fig C 1.2: Query Similarity based on Context

On clicking the button labeled ***Finding the similarity based on Context*** aside of user query test box, the application displays all the query suggestion relevant to user query along with their contextual similarity score. The next window displays all the alternate queries relevant to user query on the basis of common clicked URLS as depicted in Fig C1.3.


Home
File
LogOut
Cluster
Similarity Based on Context
Similarity Based on Clicked Url
Combined Similarities
Clusters Formed
Url Results
Finding Similarity Based on Clicked Url's

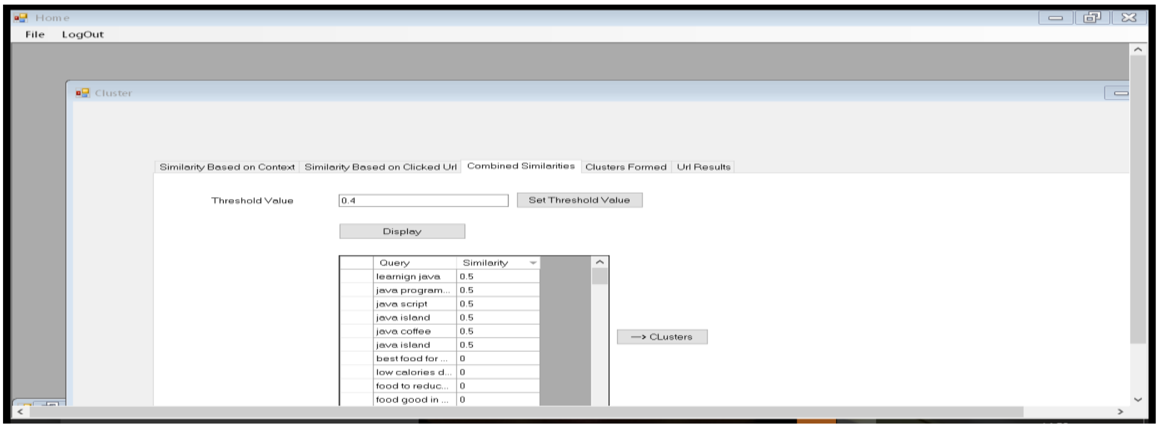
User1 Query: java

Finding Similarity Based on Clicked Url's

Query	Similarity
Travel	0
best iphone p...	0
ipone apple	0
engineering c...	0
ymcaust	0
java coffee	0
best food for ...	0
low calories d...	0
food to reduc...	0
food good in	0

Fig C 1.3: Query Similarity based on Clicked URLs

In order to clearly show the functioning of QSUB, the threshold value is put open to administrator/user. On receiving the value of threshold, the combined similarity based on context and clicked URL is displayed and clusters of queries are formed as depicted in Fig C 1.4.

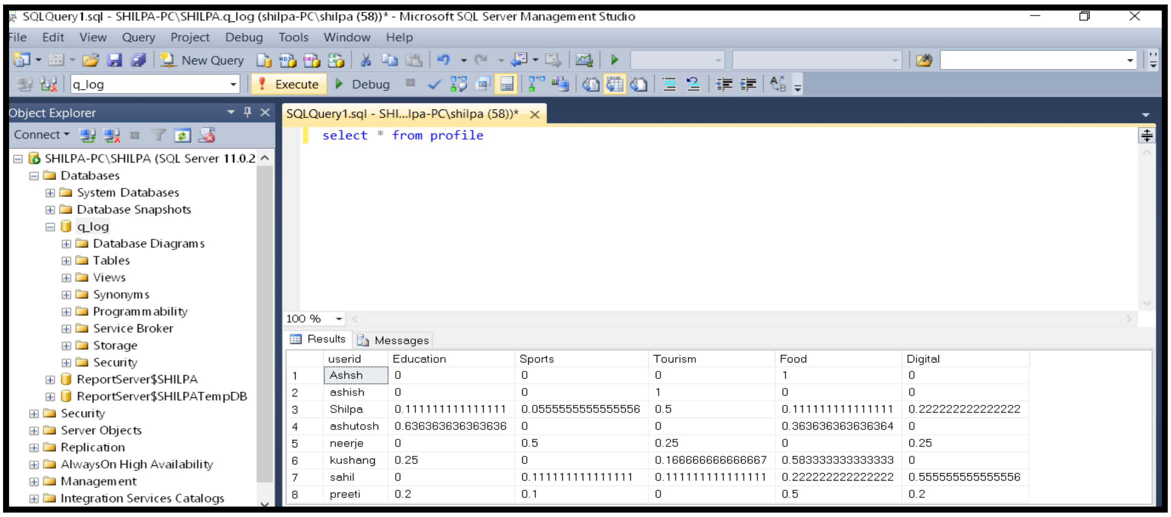


Threshold Value: 0.4

Query	Similarity
learnign java	0.5
java program...	0.5
java script	0.5
java island	0.5
java coffee	0.5
java island	0.5
best food for ...	0
low calories d...	0
food to reduc...	0
food good in ...	0

Fig C 1.4: Combined Similarity

On clicking the button labeled Display as shown in Fig C 1.4, the application shows the personalized queries to login user on the basis of its interest area stored in Profile database as shown in Fig C 1.5.

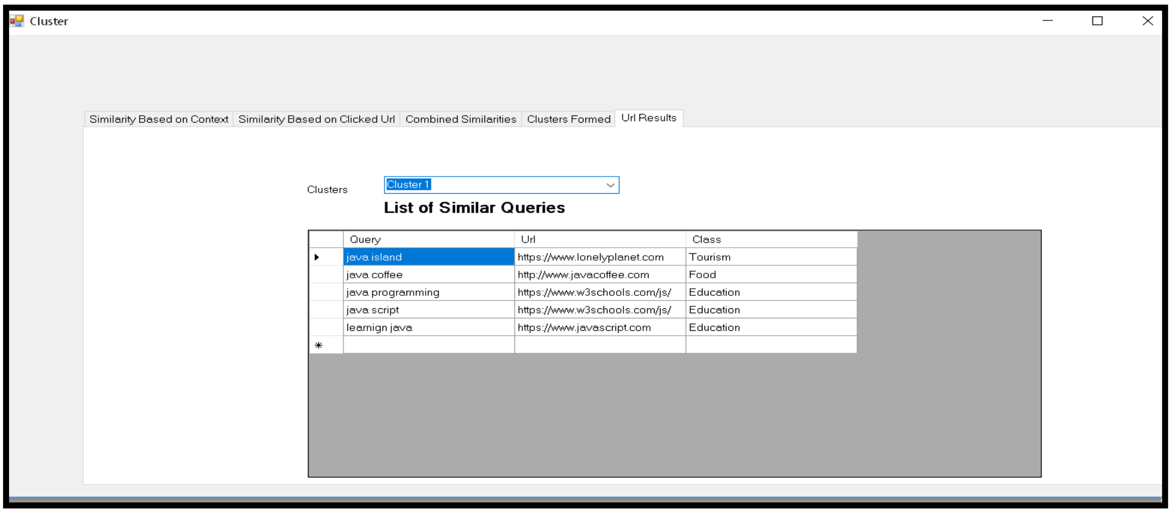

SQLQuery1.sql - SHILPA-PC\SHILPA.q_log (shilpa-PC\shilpa (58))* - Microsoft SQL Server Management Studio
File Edit View Query Project Debug Tools Window Help
New Query
q_log
Execute
Debug
Object Explorer
Connect
SHILPA-PC\SHILPA (SQL Server 11.0.2
Databases
System Databases
Database Snapshots
q_log
Database Diagrams
Tables
Views
Synonyms
Programmability
Service Broker
Storage
Security
ReportServer$SHILPA
ReportServer$SHILPATempDB
Security
Server Objects
Replication
AlwaysOn High Availability
Management
Integration Services Catalogs
SQLQuery1.sql - SHI...lpa-PC\shilpa (58))*
100 %
Results
Messages

```
select * from profile
```

	userid	Education	Sports	Tourism	Food	Digital
1	Ashsh	0	0	0	1	0
2	ashish	0	0	1	0	0
3	Shilpa	0.111111111111111	0.0555555555555556	0.5	0.111111111111111	0.222222222222222
4	ashutosh	0.636363636363636	0	0	0.363636363636364	0
5	neerje	0	0.5	0.25	0	0.25
6	kushang	0.25	0	0.166666666666667	0.583333333333333	0
7	sahil	0	0.111111111111111	0.111111111111111	0.222222222222222	0.555555555555556
8	preeti	0.2	0.1	0	0.5	0.2

Fig C 1.5: Snapshot of Profile Database

The set of personalized query suggestion is then presented back to user as shown in Fig C1.6.



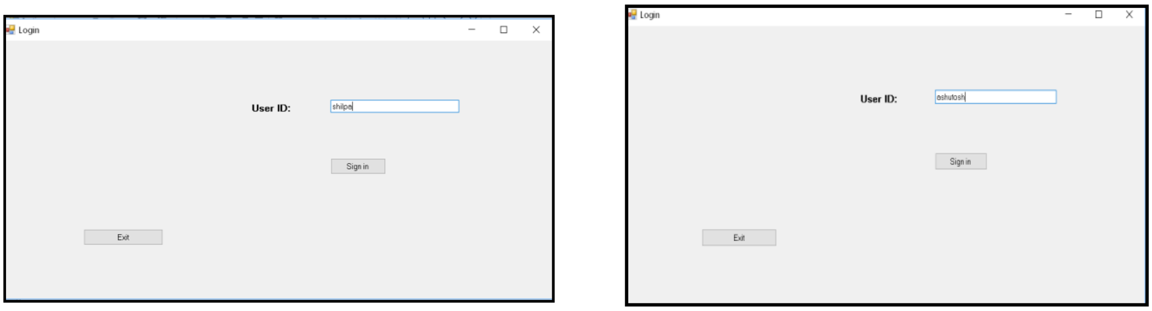
Clusters Cluster 1

List of Similar Queries

Query	Url	Class
java island	https://www.lonelyplanet.com	Tourism
java coffee	http://www.javacoffee.com	Food
java programming	https://www.w3schools.com/js/	Education
java script	https://www.w3schools.com/js/	Education
learnign java	https://www.javascript.com	Education

Fig C1.6: Personalized Query Suggestions based on UI

To distinguish between QSUB and conventional query suggestions system, the test is conducted for various users submitting the same query. For instance, the Fig C 1.7 shows the two login screen related to two different users.



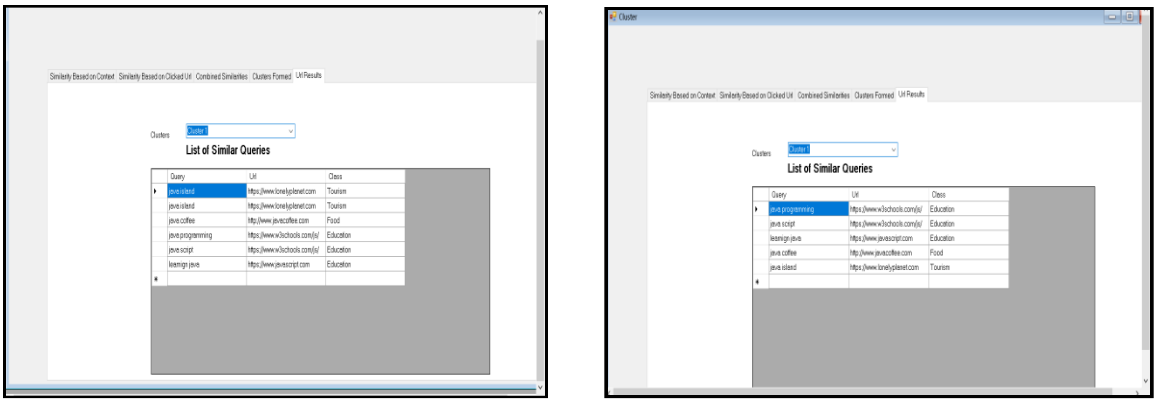
User ID: shilpa

User ID: ashutosh

Fig C 1.7: Two Users Having Different Interest Factors

It can be noted from row with serial number 3 and 4 in Fig C 1.6 that user named Shilpa and Ashutosh possesses different degree of interests indicated by corresponding numeric score in different domains. For instance, the degree of interest in education domain possessed by shilpa is 0.11 whereas other has 0.63.

Application works same for both the users till combined similarity i.e. steps shown in Fig C 1.1 to C 1.5 but in the last filtration step the query suggestion are shown on the basis of their interest factors retrieved from profile database as depicted in Fig C 1.8.


Similarity Based on Context
Similarity Based on Clicked Url
Combined Similarities
Clusters Formed
Url Results
Cluster
Similarity Based on Context
Similarity Based on Clicked Url
Combined Similarities
Clusters Formed
Url Results

Clusters: Cluster 1

List of Similar Queries

Query	Url	Class
java island	https://www.lonelyplanet.com	Tourism
java island	https://www.lonelyplanet.com	Tourism
java coffee	http://www.javacoffee.com	Food
java programming	https://www.w3schools.com/js/	Education
java script	https://www.w3schools.com/js/	Education
learnign java	https://www.javascript.com	Education

Clusters: Cluster 1

List of Similar Queries

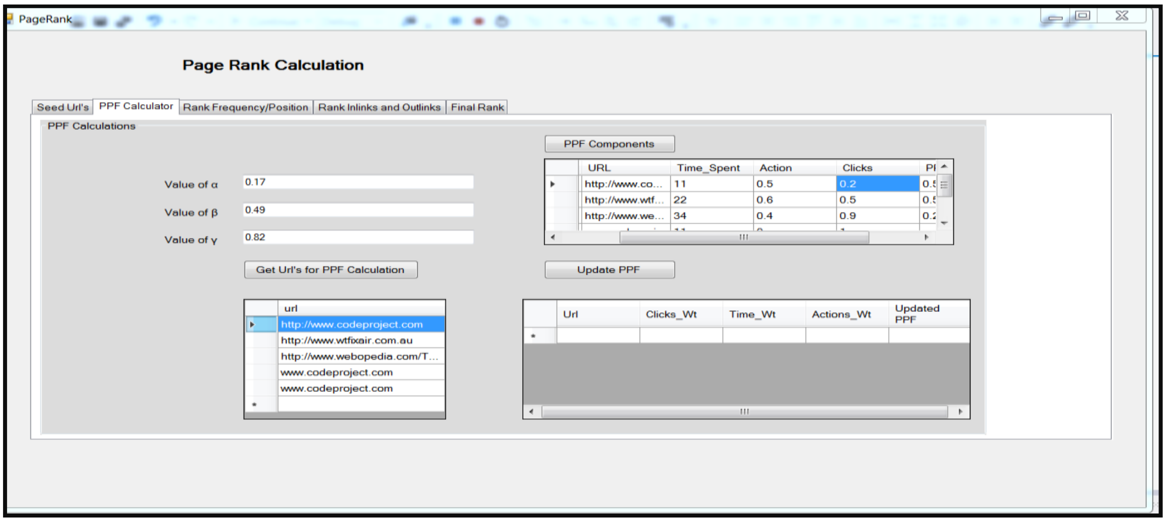
Query	Url	Class
java programming	https://www.w3schools.com/js/	Education
java script	https://www.w3schools.com/js/	Education
learnign java	https://www.javascript.com	Education
java coffee	http://www.javacoffee.com	Food
java island	https://www.lonelyplanet.com	Tourism

Fig C 1.8 (A) Query Suggestions Shown to toUser User “Shilpa”

(B) Query Suggestions Shown “Ashutosh”

APPENDIX- D

The proposed ranking technique based on user browsing patterns is implemented in C# .net with MS SQL Server 2012 at the back end and test run was carried out on sampled seed URL set given in appendix A. The user interface for the same is depicted in Fig D1.1.



Page Rank Calculation

PPF Calculations

Value of α 0.17

Value of β 0.49

Value of γ 0.82

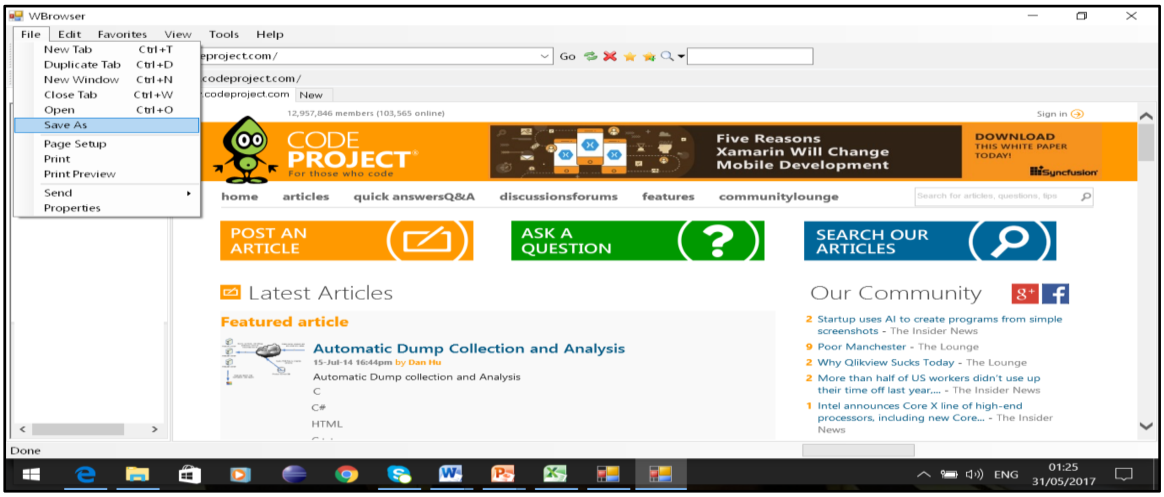
URL	Time_Spent	Action	Clicks	PI
http://www.co...	11	0.5	0.2	0.5
http://www.wtf...	22	0.6	0.5	0.5
http://www.we...	34	0.4	0.9	0.2

url
http://www.codeproject.com
http://www.wtfixair.com.au
http://www.webopedia.com/T...
www.codeproject.com
www.codeproject.com

Url	Clicks_Wt	Time_Wt	Actions_Wt	Updated PPF

Fig D1.1: User Interface for PR-PPF

The snapshots related to user actions such as save, print and bookmark is shown in Fig D 1.2, Fig D 1.3 and D1.4 respectively



eproject.com/

codeproject.com/

codeproject.com

12,957,846 members (103,565 online)

Five Reasons Xamarin Will Change Mobile Development

DOWNLOAD THIS WHITE PAPER TODAY!

Syncfusion

Latest Articles

Featured article

Automatic Dump Collection and Analysis

15-Jul-14 16:44pm by Dan Hu

Automatic Dump collection and Analysis

C

C#

HTML

Our Community

2 Startup uses AI to create programs from simple screenshots - The Insider News

9 Poor Manchester - The Lounge

2 Why Qlikview Sucks Today - The Lounge

2 More than half of US workers didn't use up their time off last year.... - The Insider News

1 Intel announces Core X line of high-end processors, including new Core... - The Insider News

Fig D1.2: User Action-Save

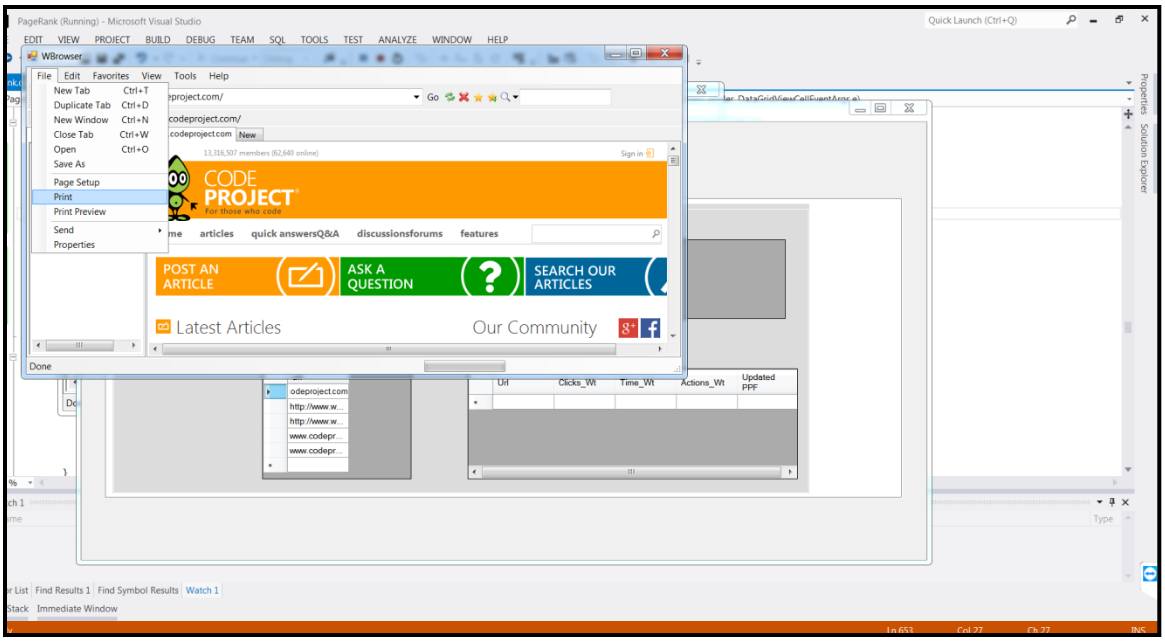
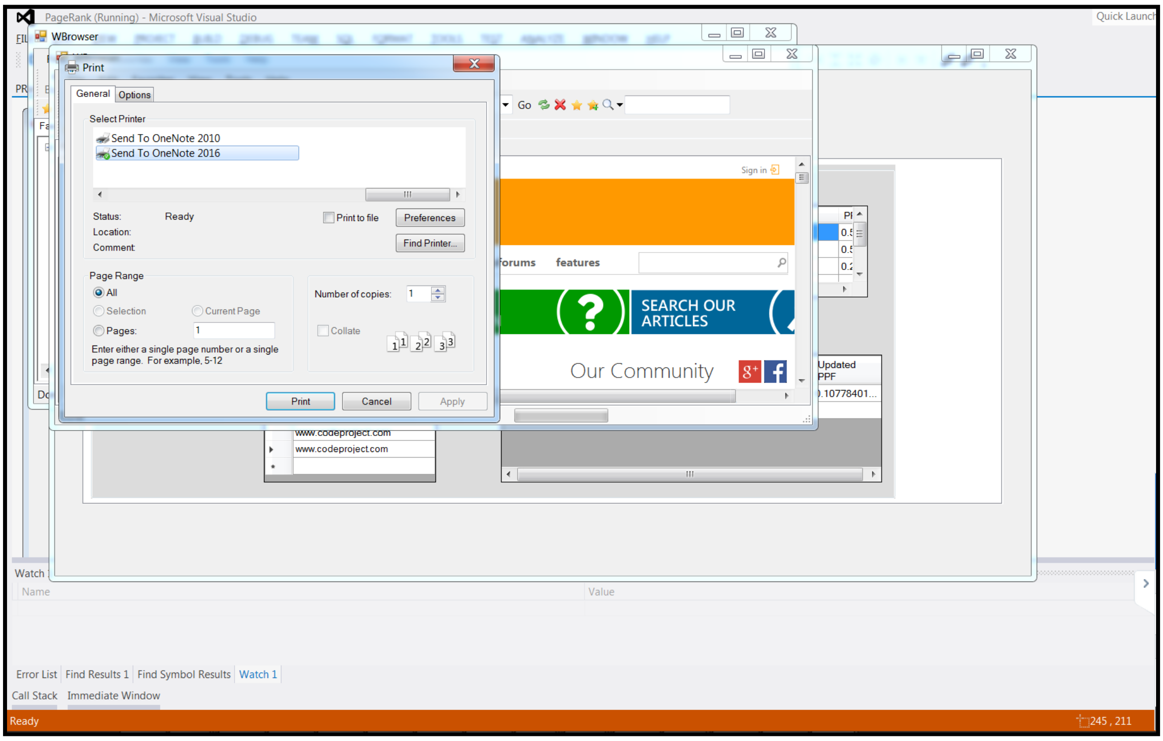


Fig D1.3 (a): User Action- Print



Select Printer
Send To OneNote 2010
Send To OneNote 2016
Status: Ready
Location:
Comment:
□ Print to file
Page Range
● All
○ Selection
○ Current Page
○ Pages: 1
Enter either a single page number or a single page range. For example, 5-12
Number of copies: 1
□ Collate

Fig D1.3 (b): User Action- Print

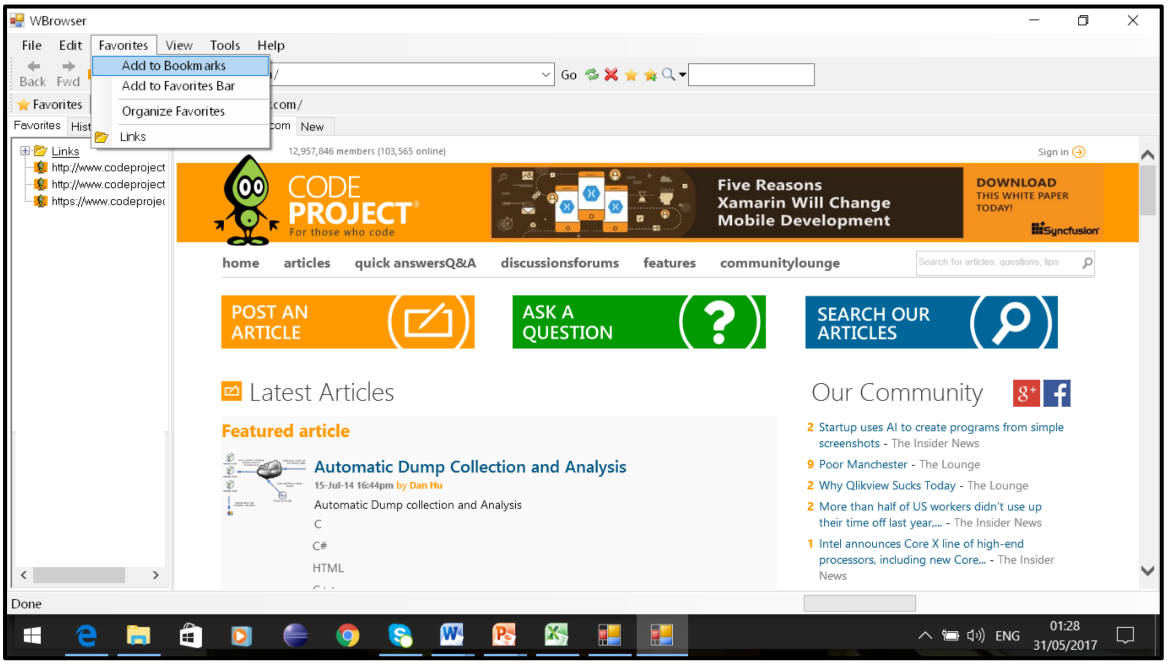


Fig D1.4: User Action- Bookmark

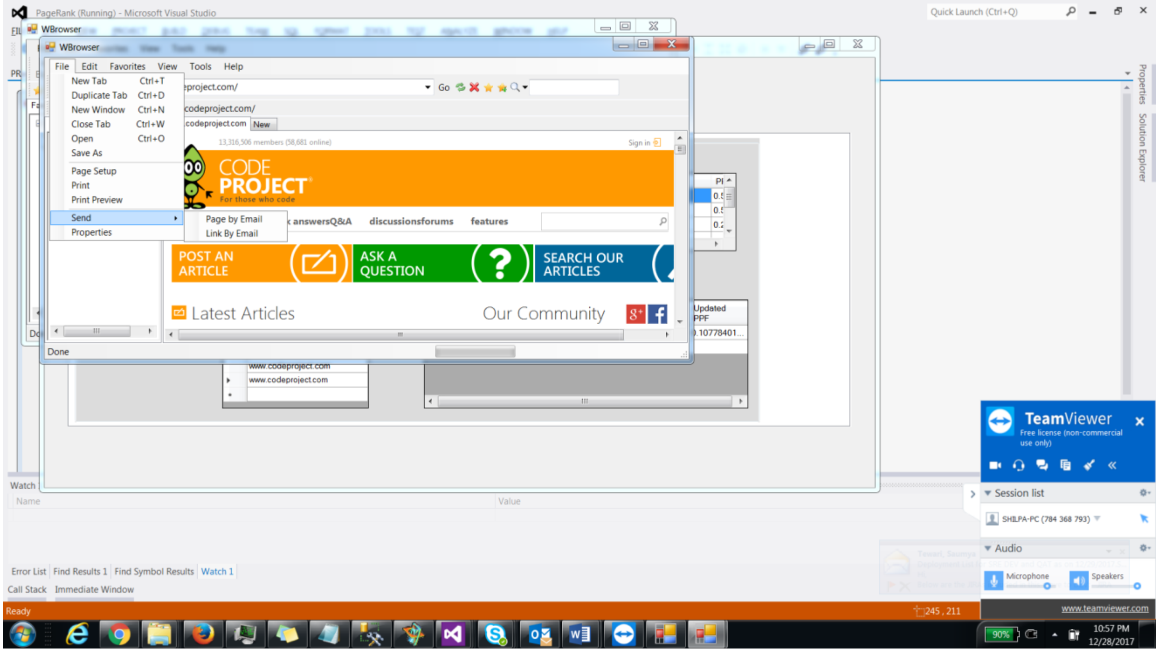
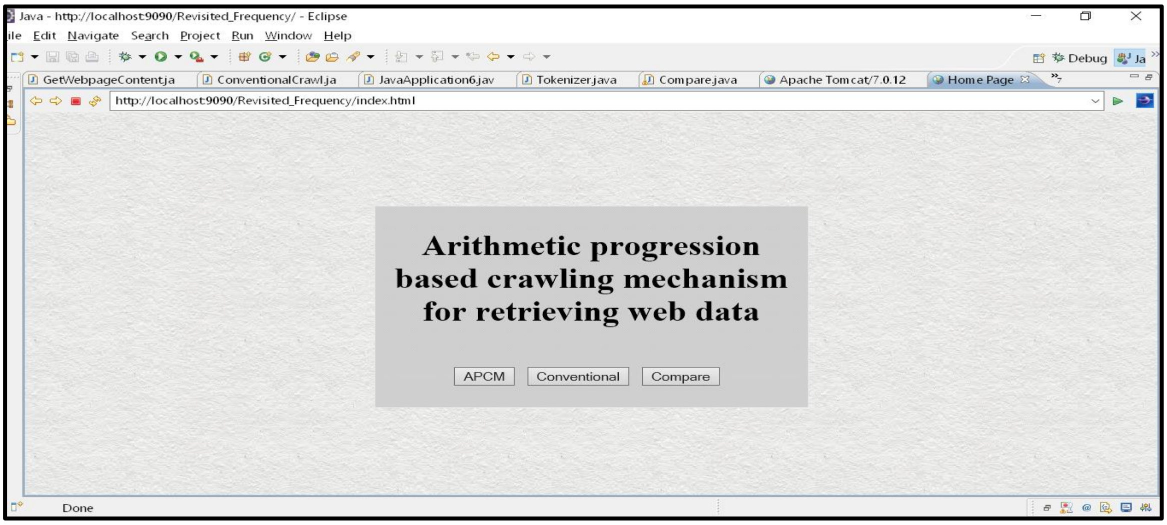


Fig D1.5: User Action- Send

APPENDIX –E

The proposed crawling mechanism implemented in C# .net with MS SQL Server 2012 at the back end. The user interface of application is shown in Fig E 1.1



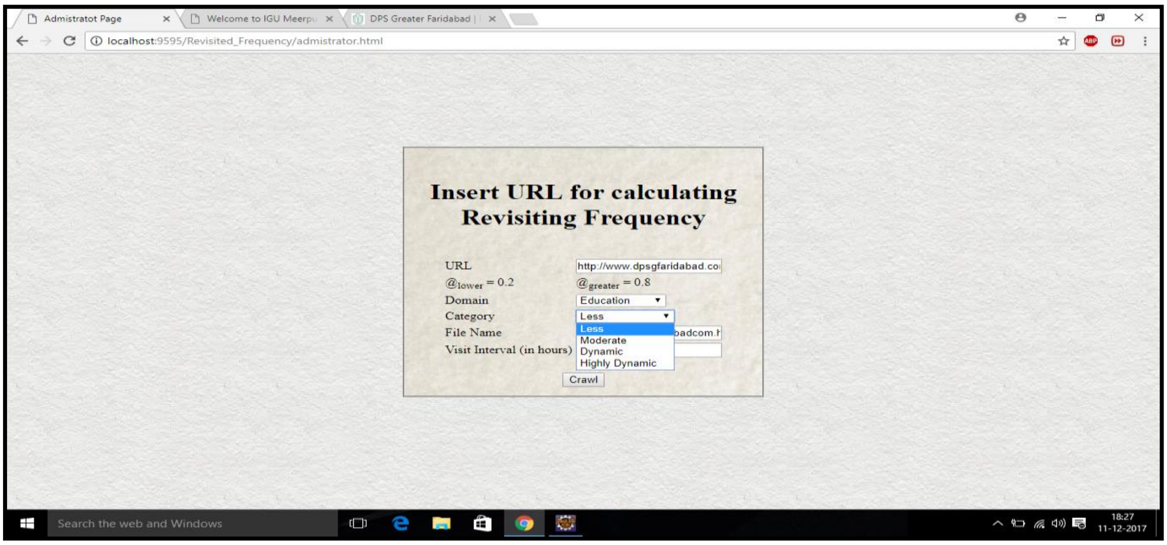
http://localhost:9090/Revisited_Frequency/index.html

Arithmetic progression based crawling mechanism for retrieving web data

Fig E 1.1: User interface for APCM

The Screen contains button for both APCM and Conventional crawling mechanism. On clicking APCM, the administrator is asked to enter the URL to be visited .

The screen shit for different options pertaining to domain categories are depicted in Fig E1.2, E 1.3 and E1.4.



localhost:9595/Revisited_Frequency/admistrator.html

Insert URL for calculating Revisiting Frequency

URL http://www.dpsgfaridabad.co

$@_{lower} = 0.2$ $@_{greater} = 0.8$

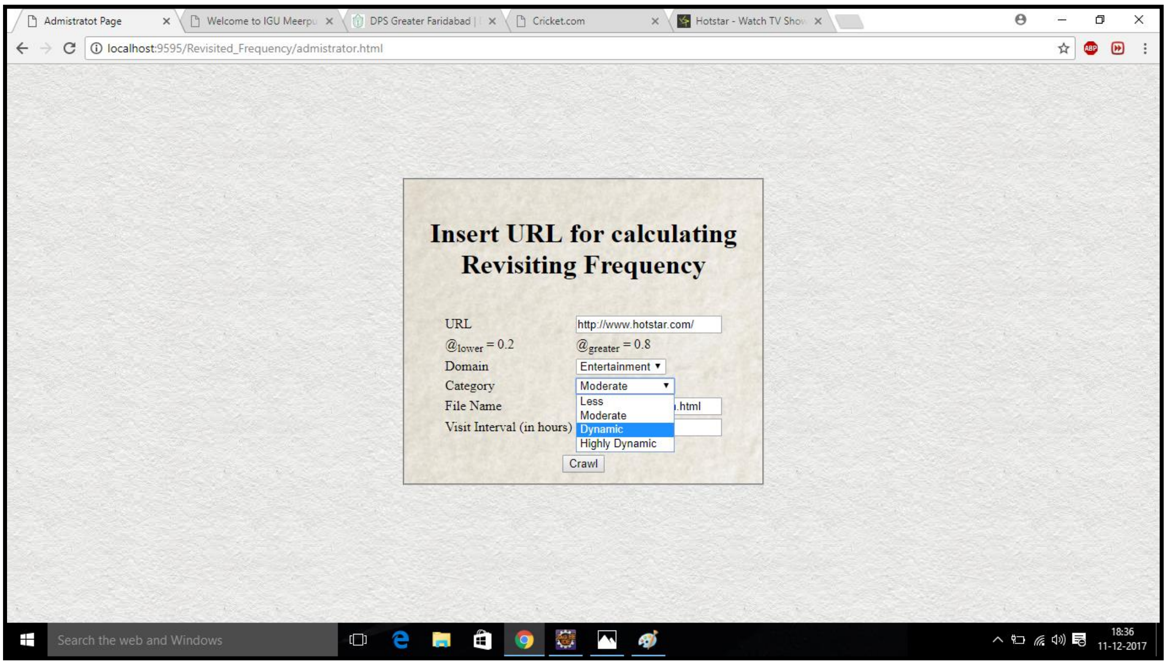
Domain Education

Category Less

File Name badcom.h

Visit Interval (in hours)

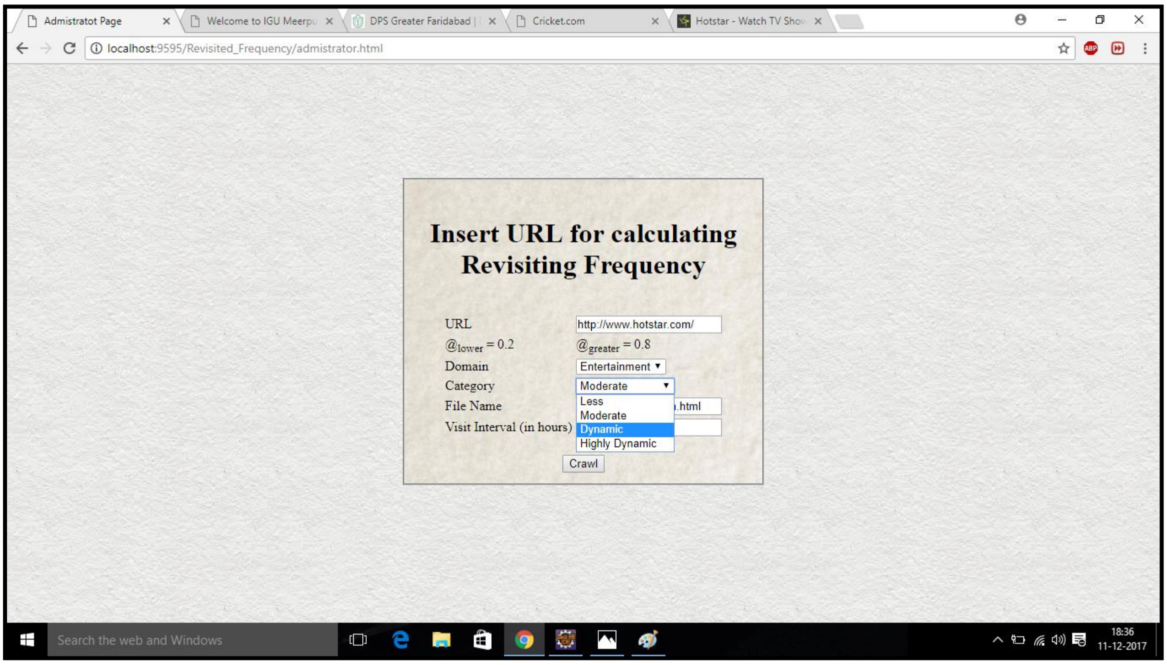
Fig E1.2: URL Crawled Under Domain Category –Less



Insert URL for calculating Revisiting Frequency

URL http://www.hotstar.com/
$@_{lower}$ = 0.2 $@_{greater}$ = 0.8
Domain Entertainment
Category Moderate
Less
Moderate
Dynamic
Highly Dynamic
File Name .html
Visit Interval (in hours)
Crawl

Fig E1.3: URL Crawled Under Domain Category – Moderate

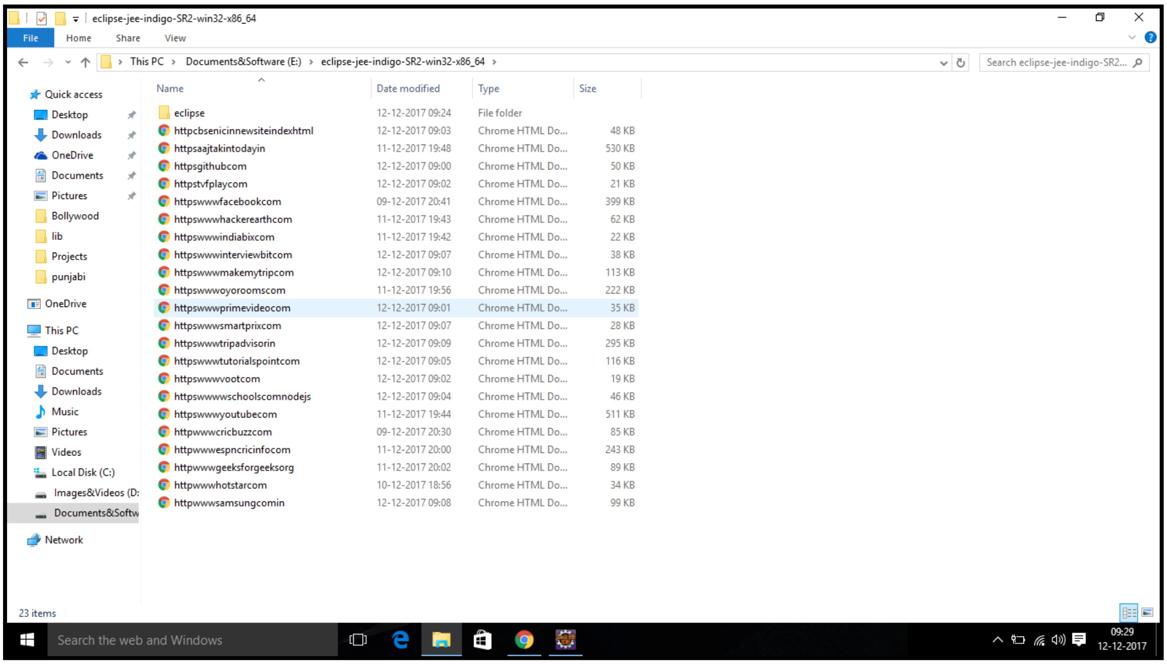


Insert URL for calculating Revisiting Frequency

URL http://www.hotstar.com/
$@_{lower}$ = 0.2 $@_{greater}$ = 0.8
Domain Entertainment
Category Moderate
Less
Moderate
Dynamic
Highly Dynamic
File Name .html
Visit Interval (in hours)
Crawl

Fig E1.4: URL Crawled Under Domain Category – Dynamic

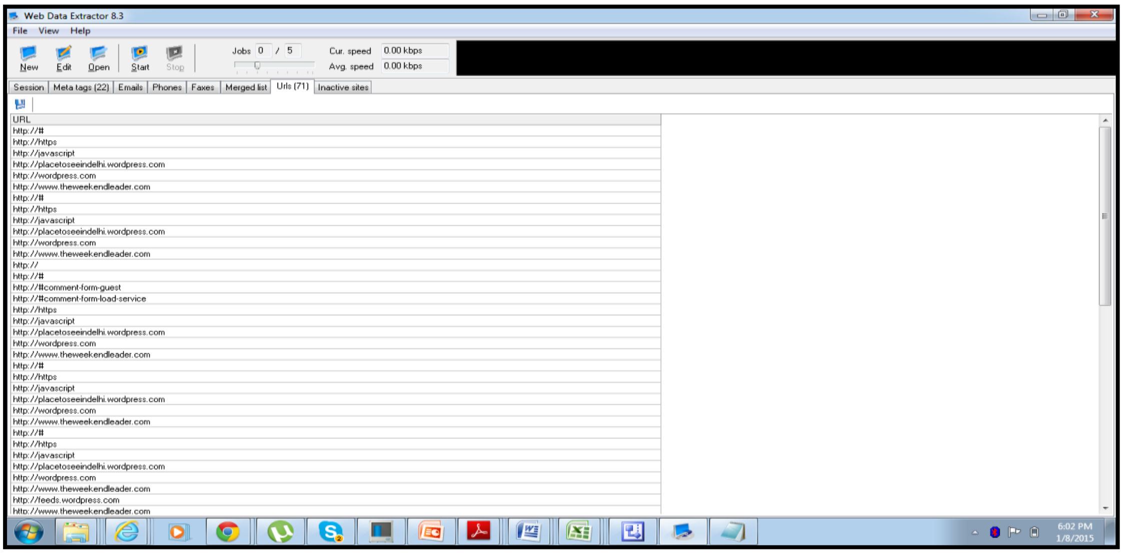
The documents corresponding to URLs downloaded by document downloader are shown in Fig E 1.5.



Name	Date modified	Type	Size
eclipse	12-12-2017 09:24	File folder	
httpcbsenicinnewsiteindexhtml	12-12-2017 09:03	Chrome HTML Do...	48 KB
httpsaajtakintodayin	11-12-2017 19:48	Chrome HTML Do...	530 KB
httpsgithubcom	12-12-2017 09:00	Chrome HTML Do...	50 KB
httpstvfplaycom	12-12-2017 09:02	Chrome HTML Do...	21 KB
httpswwwfacebookcom	09-12-2017 20:41	Chrome HTML Do...	399 KB
httpswwwhackerearthcom	11-12-2017 19:43	Chrome HTML Do...	62 KB
httpswwwindiabixcom	11-12-2017 19:42	Chrome HTML Do...	22 KB
httpswwwinterviewbitcom	12-12-2017 09:07	Chrome HTML Do...	38 KB
httpswwwmakemytripcom	12-12-2017 09:10	Chrome HTML Do...	113 KB
httpswwwoyoroomscom	11-12-2017 19:56	Chrome HTML Do...	222 KB
httpswwwprimevideocom	12-12-2017 09:01	Chrome HTML Do...	35 KB
httpswwwsmartprixcom	12-12-2017 09:07	Chrome HTML Do...	28 KB
httpswwwtripadvisorin	12-12-2017 09:09	Chrome HTML Do...	295 KB
httpswwwtutorialspointcom	12-12-2017 09:05	Chrome HTML Do...	116 KB
httpswwwvootcom	12-12-2017 09:02	Chrome HTML Do...	19 KB
httpswwwwschoolscomnodejs	12-12-2017 09:04	Chrome HTML Do...	46 KB
httpswwwyoutubecom	11-12-2017 19:44	Chrome HTML Do...	511 KB
httpwwwcricbuzzcom	09-12-2017 20:30	Chrome HTML Do...	85 KB
httpwwwespncricinfocom	11-12-2017 20:00	Chrome HTML Do...	243 KB
httpwwwgeeksforgeeksorg	11-12-2017 20:02	Chrome HTML Do...	89 KB
httpwwwhotstarcom	10-12-2017 18:56	Chrome HTML Do...	34 KB
httpwwwsamsungcomin	12-12-2017 09:08	Chrome HTML Do...	99 KB

Fig E 1.5: Documents Downloaded by Document Downloader

The downloaded documents are parsed to get hyperlinks for incremental crawling process as shown in Fig E 1.6.



Jobs 0 / 5
Cur. speed 0.00 kbps
Avg. speed 0.00 kbps

```
URL
http://#
http://https
http://javascript
http://placetoseeindelhi.wordpress.com
http://wordpress.com
http://www.theweekendleader.com
http://#
http://https
http://javascript
http://placetoseeindelhi.wordpress.com
http://wordpress.com
http://www.theweekendleader.com
http://
http://#
http://#comment-form-guest
http://#comment-form-load-service
http://https
http://javascript
http://placetoseeindelhi.wordpress.com
http://wordpress.com
http://www.theweekendleader.com
http://#
http://https
http://javascript
http://placetoseeindelhi.wordpress.com
http://wordpress.com
http://www.theweekendleader.com
http://#
http://https
http://javascript
http://placetoseeindelhi.wordpress.com
http://wordpress.com
http://www.theweekendleader.com
http://feeds.wordpress.com
http://www.theweekendleader.com
```

Fig E 1.6: A snapshot of the List of Hyperlinks extracted from Downloaded Page

BRIEF PROFILE OF RESEARCH SCHOLAR

Shilpa Sethi received her Master in Computer Application from Kurukshetra University, Kurukshetra in the year 2005 and M. Tech. in Computer Engineering from Maharishi Dyanand University, Rohtak in the year 2009. She is working as Assistant Professor in the department of Information Technology & Computer Application at YMCA University of Science & Technology, Faridabad India. She is having 12 years of teaching experience. Her area of research includes Internet Technologies, Web Mining and Information Retrieval System. She has authored papers in various national and international journals.

LIST of PUBLICATIONS

List of Published Papers: International Journals

Sr. No	Title of Paper along with Volume, Issue No, Year of Publicatiio	Publisher	Impact Factor	Referred// Non-reffered	Whether You Paid Money for the Publication	Remarks
1	"An Automatic User Interest Mining Technique for Retrieving Quality Data" International Journal of Business Analytics. Volume 4 • Issue 2 • April-June 2017, pp 62-79, ISSN: 2334-4547	IGI Global		Referred	No	Scopus, Google Directories, Google Scholar, INSPEC
2	"Query Recommendation based on User Browsing History" International Journal of Database Theory and Application (IJDTA) Vol.9 • No.6 • (June 2016), pp.131-144, ISSN: 2005-4270.	SERSC		Referred	No	Scopus indexed
3	"A Crawling Mechanism to Maintain Freshness of Downloaded Collection based on User Perspective and Page Updation Frequency" International Journal of Network communications and Emerging Technologies (JNCET) Vol-5 •special-issue-2 December 2015.pp 42-46, ISSN: 2395-5317.	EverScience	4.33	Referred	No	UGC Approved Journal
4	"A Comparative Study of Link based Page Ranking Algorithm" International Journal of Advanced Technology in Engineering and Science, Vol 03 • Issue 01 • May 2015. pp 180-187, ISSN (online): 2348 – 7550	AR Research	2.87	Referred	No	UGC Approved Journal, Google Scholar. EBSCO. DOAJ, ISSUU, CiteSeer, Cabell .DOC. book browse, IndexCoperncus
5	"An Adaptive Web Search System Based on Web Usage Mining" International Journal of Computer Engineering and Applications (IJCEA) Vol-X • Issue- I • Jan 2016, pp 09-18, ISSN: 2321-3469	AR Publication	4.382	Referred	Yes	UGC Approved Journal

6	“An Arithmetic Progression based Crawling Mechanism for Retrieving Quality Data” Communicated to International Journal of Information Technology and Web Engineering	IGI Global		Referred	No	SCI Index, Scopus, ACM Digital Library, DBLP, Google Scholar

List of Published Papers International Conferences

Sr. No	Title of Paper along with Volume, Issue No, Year of Publicatiio	Publisher	Impact Factor	Referred// Non-reffered	Whether You Paid Money for the Publication	Remarks
1	An Efficient Personalized Query Suggestion Technique for Providing Relevant Results”, IEEE 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016, Organized by BVICAM New Delhi Mar 16-18 2016, ISBN 978-9-3805-4421-2 pp 3404 - 3408.	IEEE			Yes	Scopus indexed
2	A Novel Page Ranking Mechanism Based on User` Browsing Patterns”, 50th Golden Jubilee International Annual Convention of Computer Society of India (CSI-2015) Theme Digital Life, Organised by BVICAM New Delhi, 2nd to 5th December 2015. Conference Proceedings	Springer			Yes	CSI Sponsored
3	Design of personalised search system based on user interest and query structuring”, IEEE 2nd International Conference on Computing for Sustainable Global Development (INDIACom), 2015, Organised by BVICAM New Delhi. Mar11-13 2015, pp 1346- 1351. Print ISBN: 978-9-3805-4415-1	IEEE			Yes	Scopus indexed
4	Personalization: An exploration into the world of searching”, Presented in	YMCAUST			Yes	

	International Conference Paradigms Shift in Management and Technology organized by Department of Management Studies					
5	Design of an Automatic Ontology Construction Mechanism using Semantic Analysis of the Documents" In the proceedings of 4th IEEE International Conference on Computational Intelligence and Communication Networks (CICN-2012) Nov 03-05, 2012 Organized by MIR Labs Gwalior at GLA University Mathura, India. pp 611- 616 ISBN: 978-1-4673-2981-1.	IEEE			Yes	Scopus indexed