Optimized Search Engine Recommandation System

Project ID: 20161709

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering

by

Suraj Khot
Pruthvi Mandaliya
Swapnil Mohite
Rakesh Varma

Roll No: 31, 38, 42, 65

Supervisor:

Prof. Sunantha Krishnan

UNIVERSITY OF MUMBAI

2016-2017

Optimized Search Engine Recommandation System

Project ID: 20161709

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering

by

Suraj Khot
Pruthvi Mandaliya
Swapnil Mohite
Rakesh Varma

Roll No: 31, 38, 42, 65

Supervisor:

Prof. Sunantha Krishnan

Department of Information Technology

Don Bosco Institute of Technology

2016-2017

DON BOSCO INSTITUTE OF TECHNOLOGY
Vidyavihar Station Road, Mumbai - 400070

Department of Information Technology

CERTIFICATE

This is to certify that the project entitled **'Optimized Search Engine Recommandation System'** is a bonafide work of

Suraj Khot	**31**
Pruthvi Mandaliya	**38**
Swapnil Mohite	**42**
Rakesh Varma	**65**

submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Undergraduate** in **Bachelor of Information Technology**

Date:

Prof. Sunantha Krishnan
(GUIDE NAME)

Prof. Janhavi Baikerikar
(HOD, IT Department)

Dr. Prasanna Nambiar
(Principal)

DON BOSCO INSTITUTE OF TECHNOLOGY

Vidyavihar Station Road, Mumbai - 400070

Department of Information Technology

Project Report Approval for B.E.

This project report entitled **'Search Engine Recommendation System'** by **Suraj Khot, Pruthvi Mandaliya, Swapnil Mohite, Rakesh Varma** is approved for the degree of **Bachelor of Engineering in Information Technology**

(Examiner's Name and Signature)

1. ______________________________

2. ______________________________

(Supervisor's Name and Signature)

1. ______________________________

(Chairman)

1. ______________________________

Date:

Place:

DON BOSCO INSTITUTE OF TECHNOLOGY
Vidyavihar Station Road, Mumbai - 400070

Department of Information Technology

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

____________________-

Suraj Khot : 31
Pruthvi Mandaliya : 38
Swapnil Mohite : 42
Rakesh Varma : 65

Date:

Abstract

This system is designed to provide optimal search result to the user.Our idea is to build search engine which considers certain factors to provide user search result. We introduce an algorithm to index the web pages using two concept based namely web page hyperlink analysis using page ranking and weightage to terms in page for classification.Our system also provide recommandation to the user based on his area of interest and past history of the user.This system basically build to provide search result based on query entered by user in a search box.This system satisfy user requirement by providing optimal search result.

Contents

1 Introduction **2**

1.1 Problem Statement 2

1.2 Scope of the Project 2

1.3 Current Scenario 2

1.4 Need for the Proposed System 3

1.5 Summary of the Results / Task completed 3

2 Review of Literature **4**

2.1 Summary of the investigation in the published papers 4

2.2 Comparison between the algorithms 4

3 Analysis and Design **5**

3.1 Methodology / Procedure adopted 5

3.2 Analysis 6

3.3 Proposed System 6

3.3.1 Advantages of the proposed system over existing systems 6

3.3.2 Development Hardware / Software requirements 7

3.3.3 Design Details 7

4 Implementation **11**

4.1 Implementation Plan 11

4.1.1 Algorithms 11

4.1.2 Workbreak down structure 13

4.1.3 Gantt chart 13

4.2 Coding Standard 14

4.2.1 White Space 14

4.2.2 Blank Space 14

4.2.3 Method 15

4.2.4 Type 15

4.2.5 Variable 15
4.2.6 Class 15
4.2.7 Interface 16
4.2.8 Package 16
4.3 Testing 17
4.3.1 Test Cases 17
4.3.2 Results of Testing and System Performance 17

5 Results and Discussion **18**

6 Conclusion **20**

List of Figures

3.1 Extreme Programming 5
3.2 Activity diagram for Search Engine Recommandation System . 8
3.3 Data Flow diagram for Search Engine Recommandation System 8
3.4 Use Case diagram for Search Engine Recommandation System . 9

4.1 workbreak down structure 13
4.2 Timeline Chart of Search Engine Recommandation System . . . 13

5.1 Home page 18
5.2 Results page 19

Chapter 1

Introduction

1.1 Problem Statement

Our system provides an optimal search result to the user using page ranking, classification of the pages, and indexing algorithm for crawler to retrieve relevant web pages.We are using classification algorithm KNN and page ranking method Markov chain on web textual content related IT/Computer subjects.

1.2 Scope of the Project

We introduce an algorithm to index the web pages using two concept based namely web page hyperlink analysis using page ranking and weightage to terms in page for classification. Since ranking and quality of web page are both important factors which increase accuracy of search result.

1.3 Current Scenario

FEATURES	GOOGLE	YAHOO	BING	NEXTER
S/W LICENCE	PROPERITORY	PROPERITORY	PROPERITORY	Free
PAGES INDEXED	YES	YES	YES	YES
HTTP TRACKING COOKIE	NO	YES	NO	NO
PERSONALIZED RESULTS	YES	NO	NO	NO
IP TRACKING	YES	YES	YES	NO
INFORMATION SHARING	YES	YES	YES	NO

1.4 Need for the Proposed System

This system is basically design to provide optimized search result to the user which will satisfy user requirement.We are using pagerank and classification algorithm to get optimal search result according to user query.This system is developed to provide search result which has higher accuracy by considering factors which necessary in calculating page rank of the web pages.Our system also provide context based search and handle link spam to provide more accurate search result.

1.5 Summary of the Results / Task completed

System configuration and setup is completedand the literature survey of various search engine systems is completed and system requirements has been recognized.We are Starting the implementation of proposed algorithm which are require for system.

Chapter 2

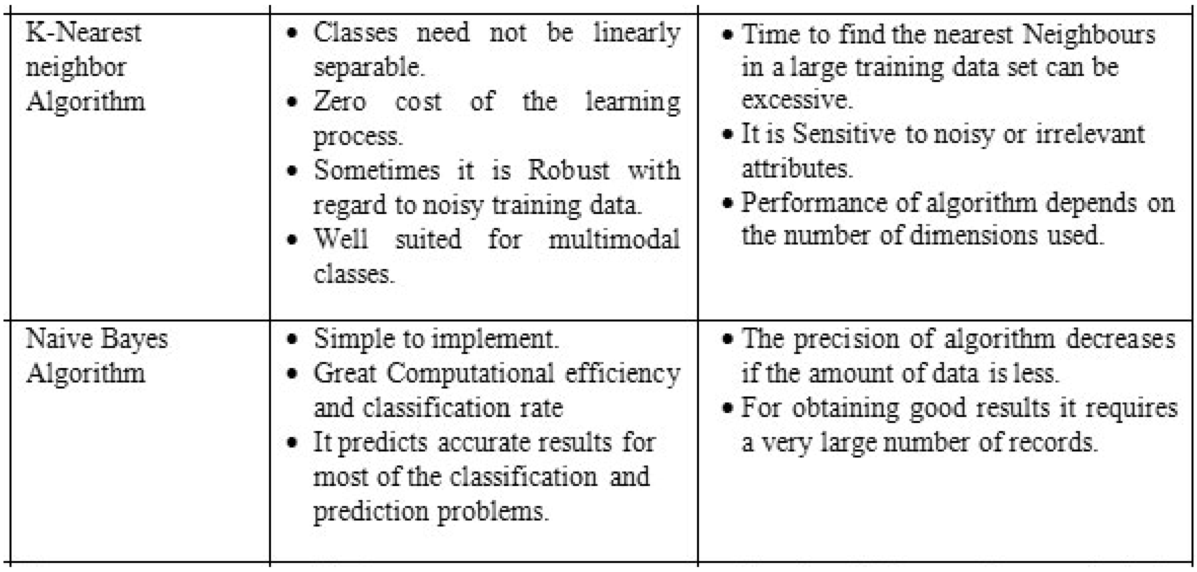
Review of Literature

2.1 Summary of the investigation in the published papers

IEEE Papers and Algorithm which are studied:

- ”Topic-Sensitive PageRank”- A Context-Sensitive Ranking Algorithm for Web Search[7]
- ”IMPLEMENTATION OF WEB CRAWLER”- Knutt-Morri-Pratt, Boyer-Moore, Finite Automata algorithm[1]
- ”A Kind of Algorithm For Page Ranking Based on Classified Tree In Search Engine” - Page ranking algorithm[2]

2.2 Comparison between the algorithms



K-Nearest neighbor Algorithm	• Classes need not be linearly separable. • Zero cost of the learning process. • Sometimes it is Robust with regard to noisy training data. • Well suited for multimodal classes.	• Time to find the nearest Neighbours in a large training data set can be excessive. • It is Sensitive to noisy or irrelevant attributes. • Performance of algorithm depends on the number of dimensions used.
Naive Bayes Algorithm	• Simple to implement. • Great Computational efficiency and classification rate • It predicts accurate results for most of the classification and prediction problems.	• The precision of algorithm decreases if the amount of data is less. • For obtaining good results it requires a very large number of records.

Chapter 3

Analysis and Design

3.1 Methodology / Procedure adopted

We choose Extreme Programming (agile development) methodology for our project, Because this model Provides following Features:

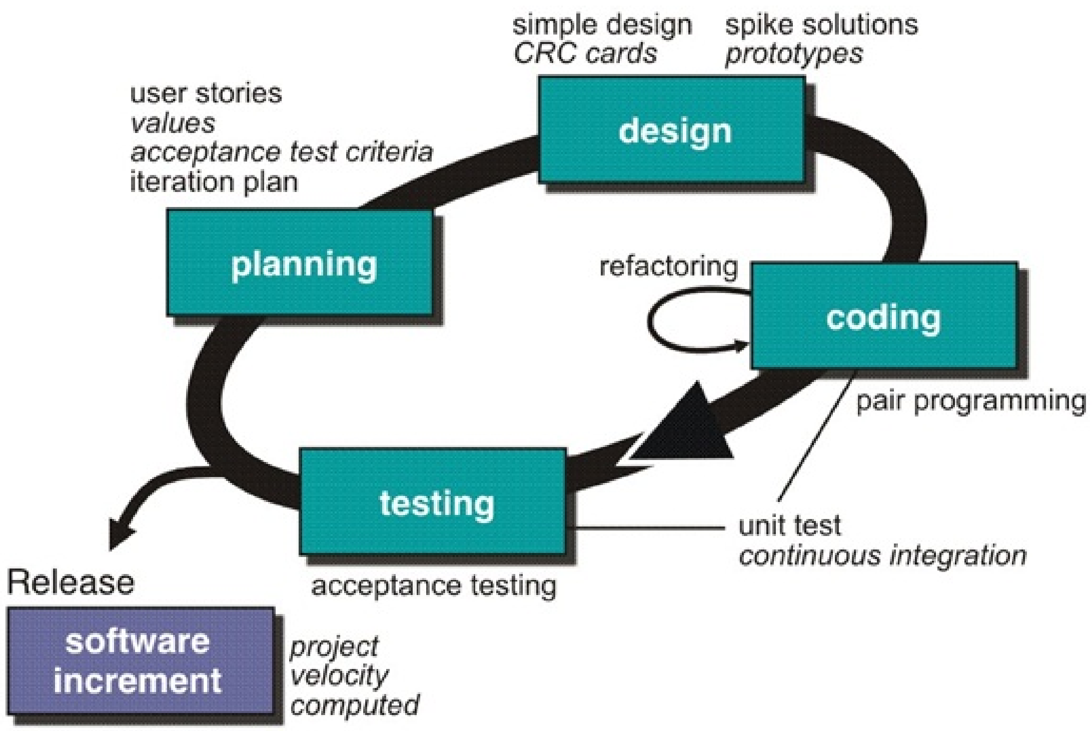


Figure 3.1: Extreme Programming

- Decrease the time required to avail some system features.
- Face to face communication and continuous inputs from customer representative leaves no space for guesswork.

- The end result is the high quality software in least possible time duration and satisfied customer.

We will conduct the weekly meetings with our project guide and do the project work/activities which will be assigned by guide and we will submit the weekly report to guide. We will maintain activity sheets from the beginning of the project work to track the activities which are planned, activities completed and carry forwarded.

3.2 Analysis

The analysis of our system contains planning and requirement analysis of our system.The requirements of our system is identified and the features which are needed to implemented in our system is specified.The analysis of our system has information about our system from user point of view.The functionality of our system is specified.The analysis of various system is also done to identify requirements of our system.

3.3 Proposed System

This system provides search result to user based on the page ranking and classification algorithm.The user enter the query in the web page then web crawler gets seed url from url database and crawl to that page and check the page content and analyse the link present on that page and crawl to one page using link analysis algorithm.This will continue until all pages are found.The pagerank of the downloaded pages are calculated and display in the search result according to their page rank value.

3.3.1 Advantages of the proposed system over existing systems

1. This system can handle big data.
2. Provide context based search result to user.
3. No Pricing.
4. Uses two main algorithm to get optimal search result.
5. Provide efficient search result.

3.3.2 Development Hardware / Software requirements

Hardware requirements:

- Two commodity machines

Software requirements:

- Front End : Java
- Backend : Hadoop System

Software tools:

- Eclipse

3.3.3 Design Details

Different UML diagrams as per the project requirement (For e.g. Use Case Diagram,Data Flow Diagram(DFD),Activity Diagram)

This use case diagram represents how the user will interact with the system.It consists of actor and use case in which actors are the user which are involve in the system and use case shows the functionality of the system.

Activity Diagram:

1. The user sends query to search engine system.
2. The page rank of the pages related to user query is calculated.
3. The page rank of the web pages are indexed so that it can display on the search result.
4. The search result is provided back to the user based on the query entered by the user.

Data Flow Diagram:

1. The user sends query to search engine system and which in turn goes to the hadoop system.
2. The hadoop system sends the data query to the database then it fetches the appropriate web pages.

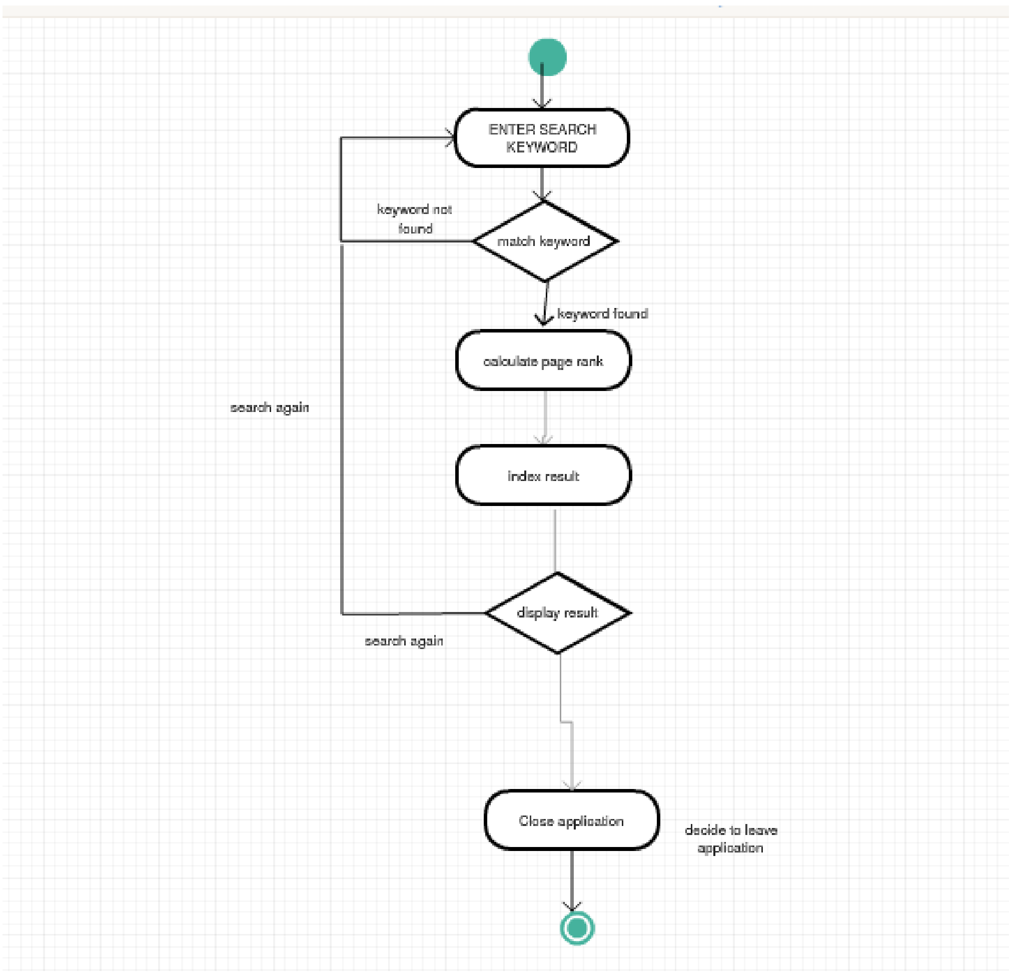


Figure 3.2: Activity diagram for Search Engine Recommandation System

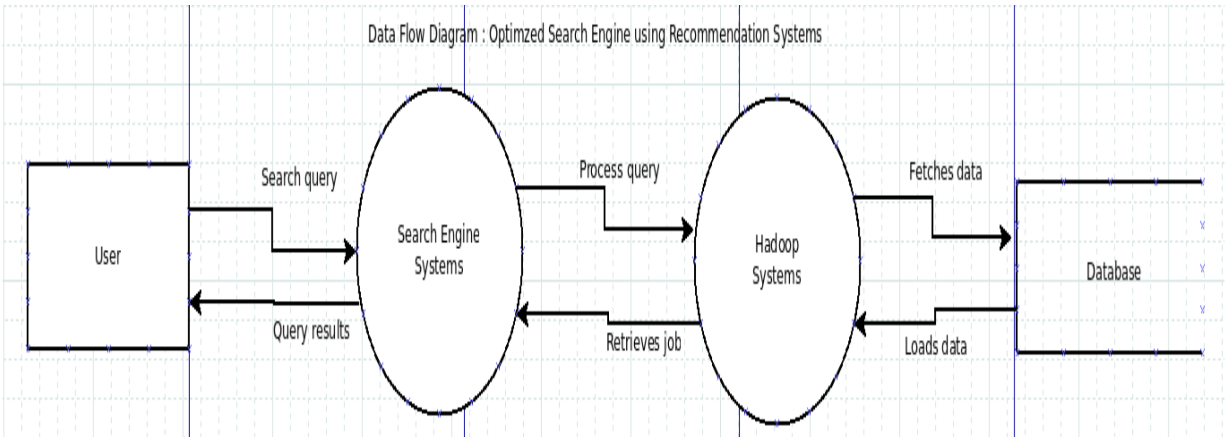


Figure 3.3: Data Flow diagram for Search Engine Recommandation System

3. The page rank of the web pages are calculated then search are provided to user based on the page rank value.

4. The search result is provided back to the user based on the query entered by the user.

Use Case Diagram:

Usecase Diagram for Optimized Search Engine Recommendation System.

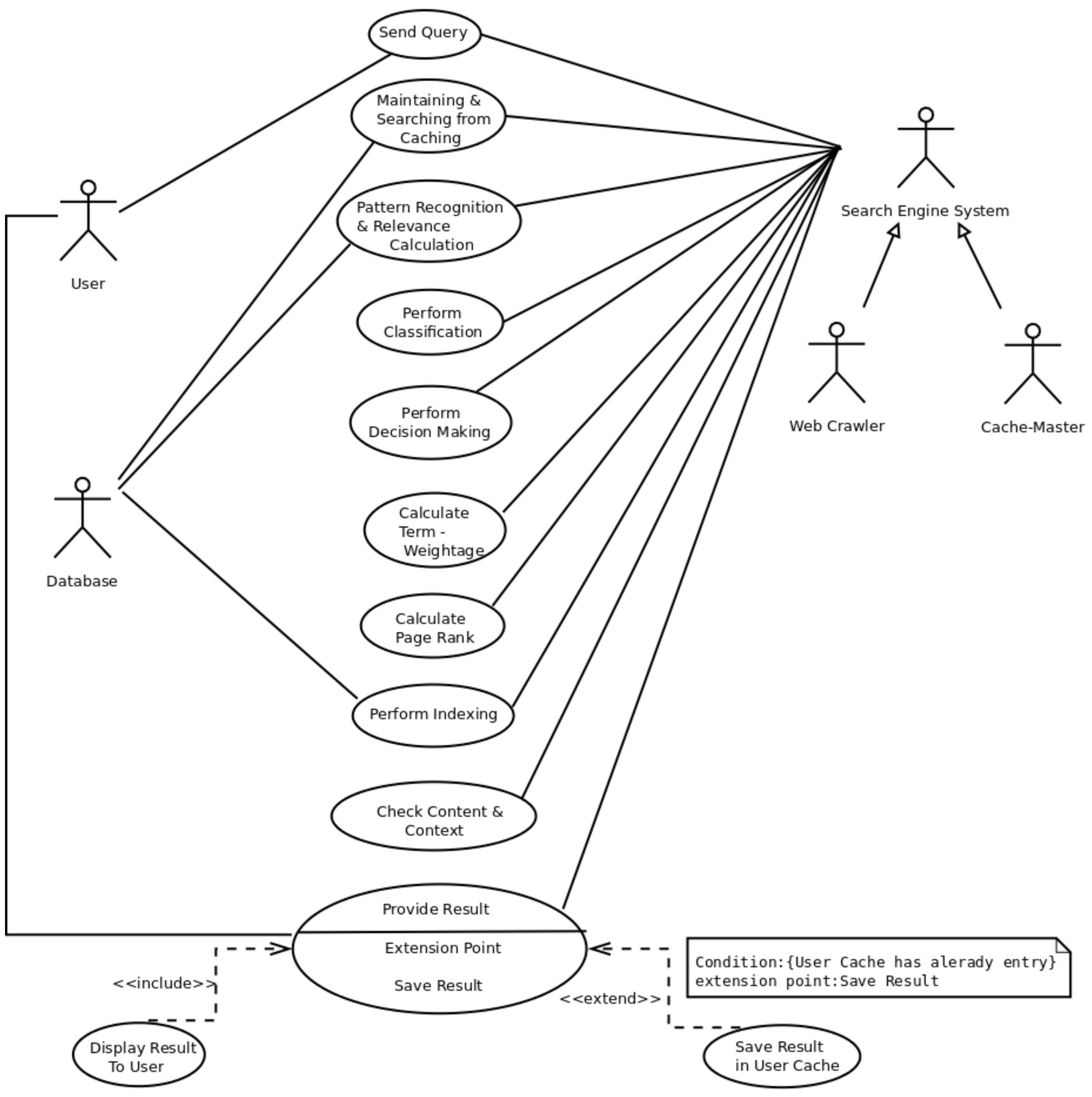


Figure : Use Case Diagram for Optimized Search Engine Recommendation System.

Figure 3.4: Use Case diagram for Search Engine Recommandation System

1. The user sends query to search engine and which in turn goes to the hadoop system

2. The hadoop system classify the data to the different records the based on user query the system will search only related web pages stored in the system.

3. The page rank of the web pages are calculated then search are provided to user based on the page rank value.
4. The search result contains the web pages based on different page rank factors and fulfill the user requirements.
5. The context based search result is also provided to the user.

Chapter 4

Implementation

4.1 Implementation Plan

4.1.1 Algorithms

1.Page rank

1. PageRank is a "vote", by all the other pages on the Web, about how important a page is. A link to a page counts as a vote of support.
2. We assume page A has pages T1 Tn which point to it (i.e., are citations). The parameter d is a damping factor which can be set between 0 and 1. We usually set d to 0.85. There are more details about d in the next section. Also C(A) is defined as the number of links going out of page A. The PageRank of a page A is given as follows:
3. PR(A) = (1-d) + d (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn))
4. Note that the PageRanks form a probability distribution over web pages, so the sum of all web pages' PageRanks will be one.
5. PageRank or PR(A) can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix of the web.

2.Term Frequency - Inverse Document Frequency

- **Term Frequency**: Which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization:

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).

- **Inverse Document Frequency**: Which measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as ”is”, ”of”, and ”that”, may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following: IDF(t) = log e(Total number of documents / Number of documents with term t in it).

3.BM25 Algorithm

- BM25 is a bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in each document, regardless of the inter-relationship between the query terms within a document (e.g., their relative proximity). It is not a single function, but actually a whole family of scoring functions, with slightly different components and parameters. One of the most prominent instantiations of the function is as follows.

$$f(q,d) = \sum_{i=1}^{N} xiyi = \sum_{w\varepsilon a\cap d} \frac{a(w,q)\,((k+1)\,c(w,d))}{c(w,d)+k} log \frac{m+1}{df(w)}$$

4. K Nearest Neighbors

- K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions).

- KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970’s as a non-parametric technique.

- Algorithm-A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor.

4.1.2 Workbreak down structure

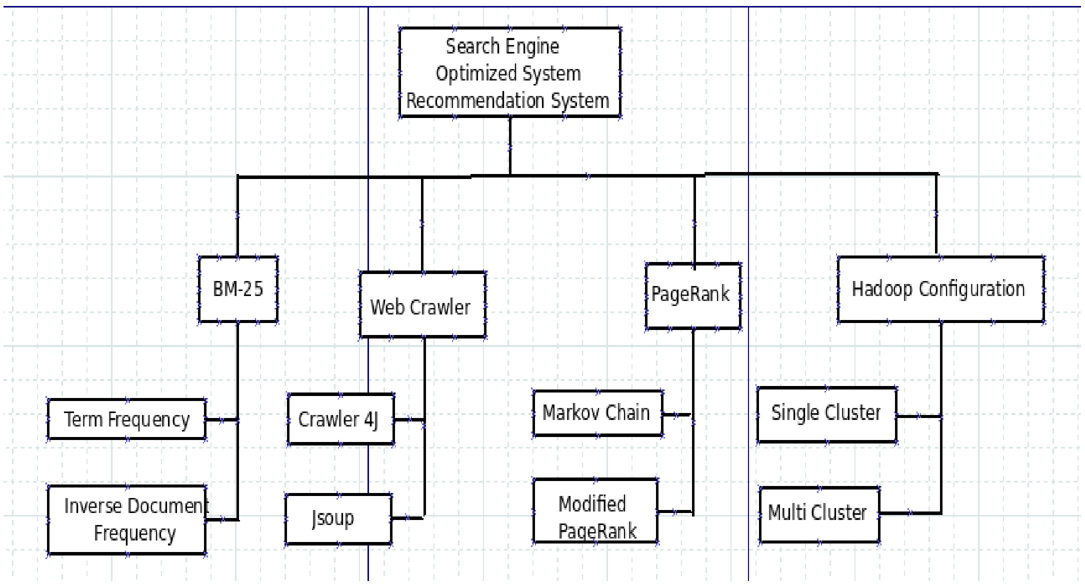
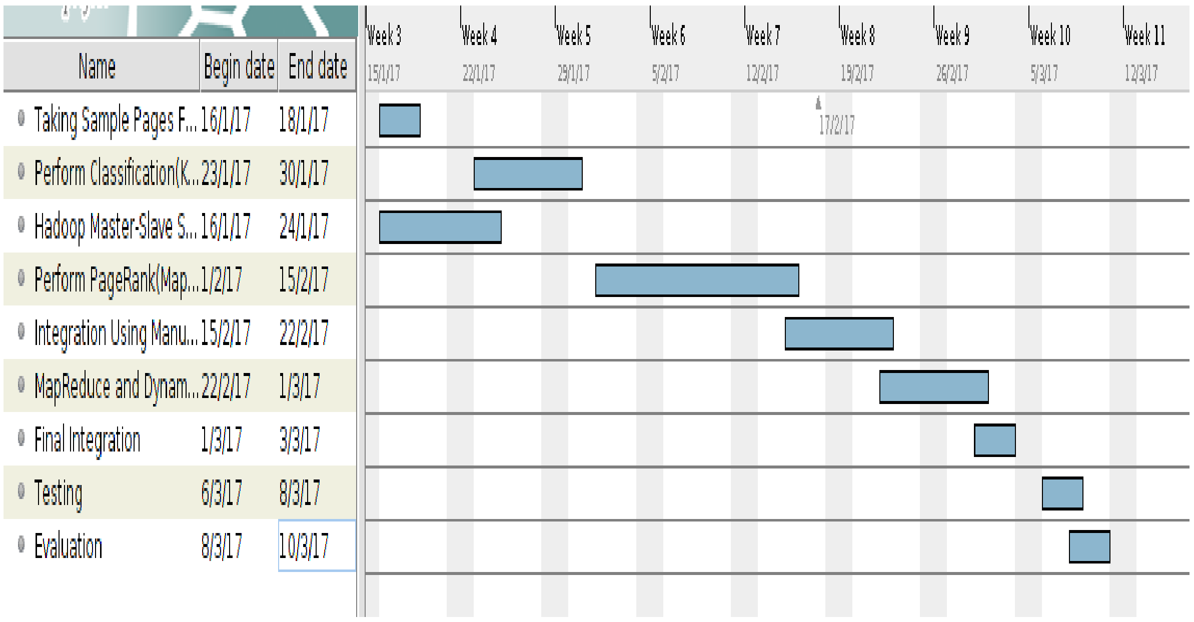


Figure 4.1: workbreak down structure

4.1.3 Gantt chart



Name	Begin date	End date
Taking Sample Pages F...	16/1/17	18/1/17
Perform Classification(K...	23/1/17	30/1/17
Hadoop Master-Slave S...	16/1/17	24/1/17
Perform PageRank(Map...	1/2/17	15/2/17
Integration Using Manu...	15/2/17	22/2/17
MapReduce and Dynam...	22/2/17	1/3/17
Final Integration	1/3/17	3/3/17
Testing	6/3/17	8/3/17
Evaluation	8/3/17	10/3/17

Figure 4.2: Timeline Chart of Search Engine Recommandation System

The above timeline chart shows how the system is going to be developed in different phases within time period.It contains analysis,coding,testing phases which are necessary in any software project.This chart also give description of the tasks that are need to be completed within certain time period.

The installation of the master-slave setup,page rank algorithm,map-reduce job,and integration of the entire system is shown on the time domain.The different tasks that are dependent on each other and time period of the tasks are also represented.The master slave configuration is needed to implementing system.After that the classification and page rank is implemented and then tested on the sample data in the February.The testing of each modules and integration testing will conducted in march to check the working of the entire system.

4.2 Coding Standard

4.2.1 White Space

Blank Lines Blank lines improve readability by setting of sections of code that are logically related. Two blank lines should be used in the following circumstances:

- Between sections of a source file
- Between class and interface definitions One blank line should always be used in the following circumstances:
- Between methods.
- Between the local variables in a method and its first statement.
- Before a block or single line comment.
- Between logical sections inside a method to improve readability.
- Before and after comments.

4.2.2 Blank Space

Blank spaces should be used in the following circumstances

- A keyword followed by a parenthesis should be separated by a space. Example: while (true)
- A blank space should appear after commas in argument lists.
- All binary operators except a period '.' should be separated from their operands by. spaces. Blank spaces should never separate unary operators such as unary minus, increment ("++"), and decrement ("–") from their operands
- The expressions in a for statement should be separated by blank spaces.

4.2.3 Method

Method modifiers should be given in the following order: ¡access¿ static abstract synchronized ¡unusual¿ final native The ¡access¿ modifier (if present) must be the first modifier.

- public static double square(double a); // NOT: static public double square(double a);
- ¡access¿ is one of public, protected or private while ¡unusual¿ includes volatile and transient. The most important lesson here is to keep the access modifier as the first modifier. Of the possible modifiers, this is by far the most important, and it must stand out in the method declaration. For the other modifiers, the order is less important, but it make sense to have a fixed convention.

4.2.4 Type

- Type conversions must always be done explicitly. Never rely on implicit type conversion.
- floatValue = (int) intValue; // NOT: floatValue = intValue; By this, the programmer indicates that he is aware of the different types involved and that the mix is intentional.

4.2.5 Variable

- Variables should be initialized where they are declared and they should be declared in the smallest scope possible.
- Variables must never have dual meaning.
- Class variables should never be declared public.
- Arrays should be declared with their brackets next to the type.
- Variables should be kept alive for as short a time as possible.

4.2.6 Class

- **Naming Classes**
  Class names should be simple full English descriptor nouns, in mixed case

starting with the first letter capitalized and the first letter of each internal word also capitalized. Whole words should be used instead of acronyms and abbreviations unless the abbreviation is more widely used than the long form, such as URL or HTML.

- **Class Visibility**
  Package or default visibility may be used for classes internal to a component.The reason why the class is public should be documented. Each class should have an appropriate constructor.

4.2.7 Interface

- The Java convention is to name interfaces using mixed case with the first letter of each word capitalized like classes.The preferred convention for the name of an interface is to use a descriptive adjective, such as Runnable or Cloneable. Interfaces should be documented specifying the purpose of the interface and how it should and shouldn't be used. Method declarations in interfaces should explicitly declare the methods as public for clarity.

4.2.8 Package

- **Documenting a Package**
  There should be one or more external documents in html format with the package name that describe the purpose of the packages documenting the rationale for the package, the list of classes and interfaces in the package with a brief description of each so that other developers know what at the package contains

- **Naming Packages**
  Unless required otherwise, a package name should include the organization's domain name, with the top-level domain type in lower case ASCII letters i.e. com.¡Name of company¿ followed by project name and sub project name as specified in ISO Standard 3166, 1981.Subsequent components of the package name vary according to requirements.Package names should preferably be singular.

4.3 Testing

We use manual testing. Manual Testing is a process carried out to find the defects. In this method the tester plays an important role as end user and verify all features of the application to ensure that the behavior of the application and helps to find the bugs in the application under test.

4.3.1 Test Cases

Test case ID	Test Cases
1	Check whether the blank space is trimmed on starting of the first word in the text box
2	On no entry done in text box, check the engine does not display any result
3	Check whether result is displayed according to page rank
4	Check the search response time
5	Check if no keyword is entered display alert box
6	Display top five result

4.3.2 Results of Testing and System Performance

Results of Testing and Integration Testing

Test case ID	Test Cases	Status	Improvisions
1	Check the search response time	Successfull	Response time reduced upto 2 minutes from 4 minutes
2	Check whether the blank space is trimmed on starting of Check whether the blank space is trimmed on starting of the first word in the text box	Successfull	
3	Check whether result is displayed according to page rank	Successfull	
4	Check if no keyword is entered display alert box	Successfull	
5	Display top five result	Successfull	Improvised results relevance by including BM25 algorithm

Table 4.1: test result

Chapter 5

Results and Discussion

.

During the development of project research was conducted based on similar systems,it was found that using hadoop framework is good for storing and processing of large data. During the development of the project it was kept in mind that system will be developed in user friendly and produced result was optimal. The user will be enter the query and system provide the result of top five website according to our algorithm that match user query.

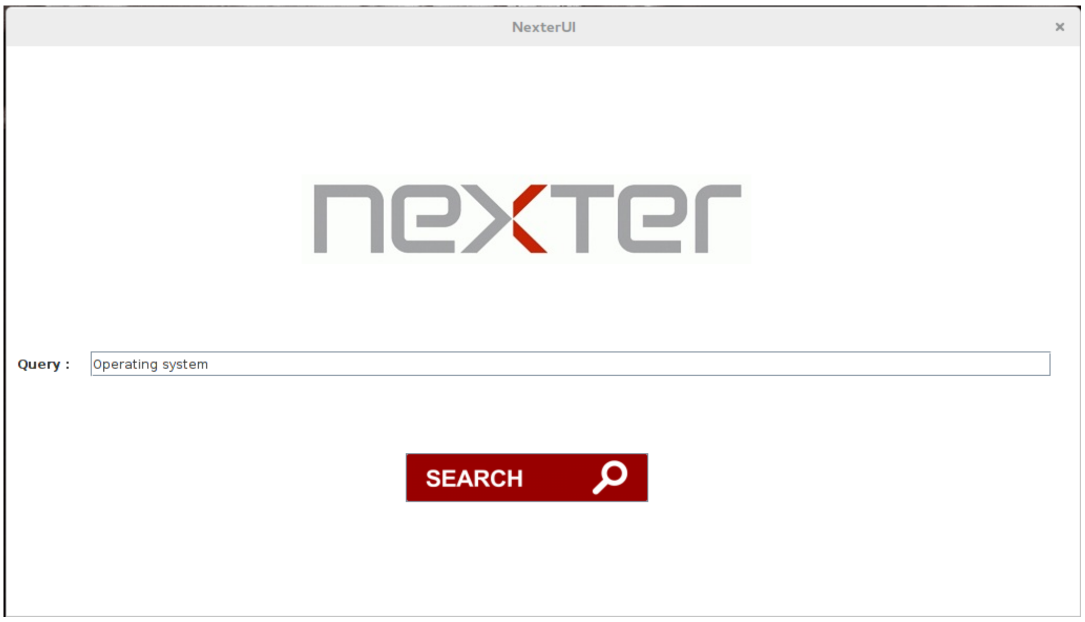
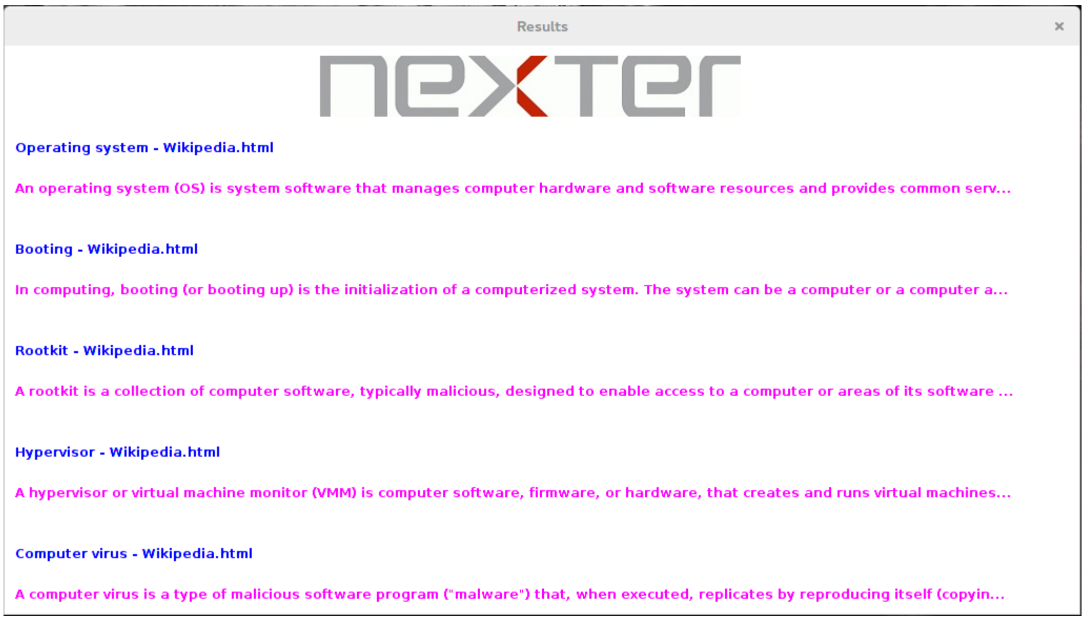


Figure 5.1: Home page



Operating system - Wikipedia.html

An operating system (OS) is system software that manages computer hardware and software resources and provides common serv...

Booting - Wikipedia.html

In computing, booting (or booting up) is the initialization of a computerized system. The system can be a computer or a computer a...

Rootkit - Wikipedia.html

A rootkit is a collection of computer software, typically malicious, designed to enable access to a computer or areas of its software ...

Hypervisor - Wikipedia.html

A hypervisor or virtual machine monitor (VMM) is computer software, firmware, or hardware, that creates and runs virtual machines...

Computer virus - Wikipedia.html

A computer virus is a type of malicious software program ("malware") that, when executed, replicates by reproducing itself (copyin...

Figure 5.2: Results page

Chapter 6

Conclusion

This search engine system consists of page rank algorithm,classification and indexing of the web pages.It uses KNN classification algorithm to classify the web pages and then page rank of the pages related to the user query is calculated so that it will provide search result according to page rank value of the web pages.We are implementing the page rank algorithm in a such way that it will provide optimal search result to the user.The context based search result can also be provided to enhance the accuracy of search result.This system is made to provide search result to the user which will satisfy the requirement of the user.

References

[1] Pooja gupta Mrs. Kalpana Johari,*"IMPLEMENTATION OF WEB*. CRAWLER," in Second International Conference on Emerging Trends in Engineering and Technology, ICETET-09.

[2] TIAN Chong, "A Kind of Algorithm For Page Ranking Based on Classified Tree In Search Engine," International Conference on Computer Application and System Modeling(ICCASM 2010).

[3] Debashis Hati,Biswajit Sahoo,Amritesh Kumar, "Adaptive Focused Crawling Based on Link Analysis," 2nd International Conference on Education Technology and Computer(ICETC), 2010.

[4] Fan Chung, "A Brief Survey of PageRank Algorithms," IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, 2014.

[5] Wang Hui-chang,Ruan Shu-hua,Tang Qijie, "The Implementation of a Web Crawler URL Filter Algorithm Based on Caching," Second International Workshop on Computer Science and Engineering, 2009.

[6] Chia-Chen Yen,Jih-Shih Hsu, "Pagerank Algorithm Improvement by Page Relevance Measurement," FUZZ-IEEE 2009.

[7] Radha Shankarmani,M.Vijayalakshmi, "Big data Analytics"

Acknowledgements

I would like to express my special gratitude and thanks to our guide for giving me such attention and time. My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

Suraj Khot : 31
Pruthvi Mandaliya : 38
Swapnil Mohite : 42
Rakesh Varma : 65

Date: