

**Visvesvaraya Technological University,
Jnana Sangama, Belgaum - 590014**



A Project Report on

“CLUSTERING OF DOCUMENTS BASED ON TEXT CONTENT”

Submitted in partial fulfilment of the requirements for the award of degree of

Computer Science & Engineering

Submitted by:

1PI13CS100

NIKET RAJ

1PI13CS128

S BHASKAR

1PI13CS155

SHREYASH S PATIL

Under the guidance of
Prof V R Badri Prasad
Associate Professor

Jan – May 2017



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
PES INSTITUTE OF TECHNOLOGY,
(AN AUTONOMOUS INSTITUTE UNDER VTU, BELGAUM AND UGC, NEW DELHI)
100FT RING ROAD, BSK 3RD STAGE, BENGALURU - 560085



PES INSTITUTE OF TECHNOLOGY

(An Autonomous Institute under VTU, Belgaum)

100 Feet Ring Road, BSK- III Stage, Bangalore – 560 085

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

Certified that the eighth semester project work titled “**Clustering of Documents Based on Text Content**” is a bonafide work carried out by

1PI13CS100

NIKET RAJ

1PI13CS128

S BHASKAR

1PI13CS155

SHREYASH S PATIL

in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belgaum during the academic semester January 2017 – May 2017. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Bachelor of Engineering.

Signature of the Guide

Prof. V R BADRI PRASAD

Signature of the HOD

Prof. Nitin V. Pujari

Signature of the Principal

Dr. K S Sridhar

External Viva

Name of Examiners

Signature with Date

ACKNOWLEDGEMENT

This project has resulted from suggestions, guidance and encouragement of numerous individuals. We deeply express our heartfelt gratitude to all those who have helped us in its completion.

To begin with, we are grateful to our Guide, **Prof. V R Badri Prasad**, Professor for his valuable guidance, novel ideas and excellent support during the course of our study and project work. He has always been a source of constant help and encouragement. We would also want to thank him for his continuous inputs and ideas for this project.

We would like to thank **Prof. Nitin V. Pujari**, Head of the Department CSE for his constant support and encouragement.

We are grateful to our Principal, **Dr. K S Sridhar** who provided us adequate time and sufficient facilities to carry out the project. We express our gratitude for having given us an opportunity to learn and to experience the essence of teamwork in the due course of the project.

Last but not the least, we would also like to thank our parents and other family members for their continuous support to us, without which, we would not have been able to achieve whatever we have. We are also thankful to our friends who directly or indirectly helped us to complete our project successfully.

Abstract

Text mining is used to retrieve data and information from the documents. In day to day the number of unstructured data is increasing, and it is really hard to find related data or information from the documents. Text mining which is also called as text data mining help to extract relevant information from the documents.

Feature extraction is used in data mining for extracting the main terms from the documents. The terms can be removed by using NLP (NLP post tagger) tool. This terms are used by the TCFS algorithms to produce the corpus. This corpus are used to classify the documents in particular predefined domain.

The text mining can be used to categories the set of documents in their respective domain. For example a set of new documents which contains all types of data such as sports or national and international .We can use text mining to produce respective news as per our need. In the same way our project uses text mining to classify a set of mixed research papers into predefined research papers domains like networking, data mining etc.

Table of Contents

1. Introduction.....	1
1.1. Problem Statement.....	2
1.2. Generic Proposed Solution.....	2
1.3.Acknowledgment.....	3
2. Literature Survey.....	4
3. System Requirements Specification.....	6
3.1. High – Level Block Diagram.....	6
3.2. Environment Used in the Project	6
3.2.1. Hardware Interface Requirement.....	7
3.2.2. Software Requirements.....	6
3.3.Functional Requirements.....	7
3.4.Non – Functional Requirements.....	8
3.5.Constraints and Dependencies	9
3.6.Assumption	10
3.7.Use Case Diagram for the requirements	11
3.8.Requirement Traceability Matrix.....	12
4. Schedule.....	15
5. High Level Design.....	16
5.1. Architectural Diagram.....	16
5.2. Sequence Diagram.....	17
5.3.User Interface Design.....	19
6. Detail Design.....	22
6.1. Modules.....	22
6.2.Updated RTM.....	24
7. Implementation.....	25
7.1. Pseudo code/algorithms	25
7.2.Codebase structure	27
7.3. Coding Guidelines Used	28
7.3.1. Why Have Coding Guidelines	28
7.3.2. Java Coding Guidelines Followed	28

7.4. Sample Code.....	29
7.5. Unit Test Cases.....	31
7.6. Metrics for Unit Test Cases.....	32
7.7. Updated RTM.....	33
8. Testing.....	34
8.1. System Test Specifications.....	35
8.2. Test Environment Used.....	36
8.3. Test Procedure.....	36
8.4. Example Test Result.....	36
8.5. Test Metrics.....	37
8.6. Updated RTM	38
9. Results and Discussion.....	39
10. Retrospective.....	40
11. Bibliography.....	41

1. Introduction

The term Data Mining for the most part alludes to a procedure by which precise and already obscure data can be removed from expansive volumes of information in a shape that can be comprehended, followed up on, and utilized for enhancing choice procedures. Data Mining is regularly connected with the more extensive procedure of KDD. By similarity, this framework characterizes Textual Data Mining as the way toward securing substantial, possibly valuable and understandable content from the text provided.

Text data contains large amount of text which is raw and unstructured. This text data can be anything book, articles, news or research too. As the internet is growing the number of pdfs, new etc. are increasing in large amount with large number of structured and unstructured data. The process of mining this data is text mining. As increase of research in all fields have increased the number of research paper publication. For example every research paper contains few structured fields tables and title etc. and the remaining part of the paper is unstructured like abstract and content. Without knowing what the pdf is about it is hard to extract useful information from the research paper. Text mining is used to extract the features from the unstructured data and analysis the data.

Now a days during research survey the junior scientist or the PhD students are given a huge number of research papers in thousands of number which might be relevant or might not be relevant to what they are looking for. If the research person is looking for image processing papers from the huge set of papers which include other research papers such as Data mining or NLP, the person has to check all the thousands of research papers to find the image processing paper which is relevant to him. This is a waste of time and effort. If there was a system which could cluster or classify these documents in the domain and present it to the user, the time and efforts of the user would be saved and will spend more time on the research not sorting them.

The existing system uses keyword search to find the relevant documents from the huge set of thousands of research papers. This system is not efficient because if a NLP documents contains some reference to Image processing paper, the output of the system will a NLP paper not image processing paper which will make the system's efficacy less and poor. The challenges there are

to read all the documents and give out the required set of documents which user has asked without using the search and sort technique which is a challenge to the programmer to use text mining and extra feature and use NLP over it so produce an efficient output.

1.1 Problem Statement

Given a mixed set of documents of different computer science domains mainly networking , cloud computing ,mobile computing data mining ,classify or cluster these documents such that each domains related papers are together .The result is the given document belongs to which domain.

1.2 Generic Proposed Solution

The solution goes in two way. First we need to train the system with the train data. And test the results with the testing data.

The train data is given to the system in batch wise according to the domains and system is trained for that domain. First stage training, would preprocess the input data, we remove unwanted data .we will make a frequency matrix having terms which are extracted using NLP tool, count of the term and the name document in which it is present.

TCFS algorithm is used later to extract the corpus from the data set and store them in particular Domains database.

Testing data is given to our system to test the system. We will perform the same procedure, first unwanted data will be removed. Frequency matrix and document matrix is done. After this feature extraction is performed and TCFS algorithm is applied. We get the terms and these terms are matched with the corpus of all domains database (corpus database) and score is generated which help us to classify the document, where it belongs i.e. which domain it belongs.

1.3 Acknowledgement



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

PES INSTITUTE OF TECHNOLOGY

(An Autonomous Institute under VTU, Belgaum)

100 Feet Ring Road, BSK- III Stage, Bangalore – 560 085

Project ID : 099

Project Title : Clustering Of Documents Based On Text Content

Project Team :	1PI13CS100	NIKET RAJ
	1PI13CS128	S BHASKAR
	1PI13CS155	SHREYASH S PATIL

This project report was submitted for review on **May 6, 2017**. I acknowledge that the project team has implemented all recommended changes in the project report.

Guide signature with date:

Guide Name: Prof. V R Badri Prasad

2. Literature Survey

A lot of research and work is going on in the field of text mining. Much progress has happened in the field of text mining in the recent years. Many proposed techniques are looked and analyzed to have a clear idea and to proceed further for implementation. Most percent of the information in the world is currently stored in unstructured textual format. Even though if we use some techniques of Natural Language Processing, that will not accomplish complete text analysis. Because of this, text mining is emerging which helps in extracting the information effectively. It is very difficult to manually organize and extract useful relevant information from huge datasets. A common problem is that classifying the documents into user's requirement. As the document size or number of documents increases, the computation involved also increases. We looked into some feature extraction algorithms to reduce the complexity.

One of the main challenges in text mining is high dimensionality. Text-mining algorithms should also deal with word ambiguities such as synonyms, acronyms, pronouns, noisy data, spelling mistakes, abbreviations and improperly structured text.

According to the paper there are many techniques for the text mining. Author of this paper explained about some selected text mining methods i.e. Latent Semantic Analysis, Hierarchical Latent Dirichlet Allocation, Probabilistic Latent Semantic Analysis, Latent Dirichlet Allocation, Principal Component Analysis and Support Vector Machines with some examples.

In the paper author proposed the methodology for extracting the helpful information from the Medline papers. Because it is very difficult problem to extract and get the useful data from the currently available search engines and other tools. He proposed some special technique and specific keywords it rank those medicines which are frequently used.

In the paper some new methodologies for text mining are proposed by the author but they have some major drawbacks also i.e. The ABC principle, and NLP based method it is for text mining but NLP based systems are more computationally intensive than co-occurrence based methods. And another pitfall of some NLP based IE methods is that when a huge number of relations needs to be detected, these are limited by the availability and quality of the training data and do not scale well.

From the study it was observed that Text mining algorithms can be classified in two Broad categories: Supervised learning and unsupervised learning.

Supervised learning is a technique in which the algorithm uses predictor and target attribute value pairs to observe the predictor and the target value relation. The training datasets consist of pairs of predictor and target values. Each predictor value is tagged with a target value. If the algorithm can predict a categorical value for a target attribute, it is called a classification function. Class can be an example of a categorical variable. Ahmad and Dey (2007) presents a clustering algorithm based on k-mean paradigm .K-means is one of popularly known clustering algorithm which as its name suggests is used to partition n unlabeled observations in the dataset into k clusters in which each observation belongs to the cluster with the nearest mean .Unsupervised learning is a technique in which the algorithm uses only the predictor attribute values. There is no target attribute value and the learning task is to gain some understanding of relevant structural patterns in the data

Text Clustering Issues to be considered

Selection of Features: the process of determining the quality terms that have the positive impact on the clustering process.

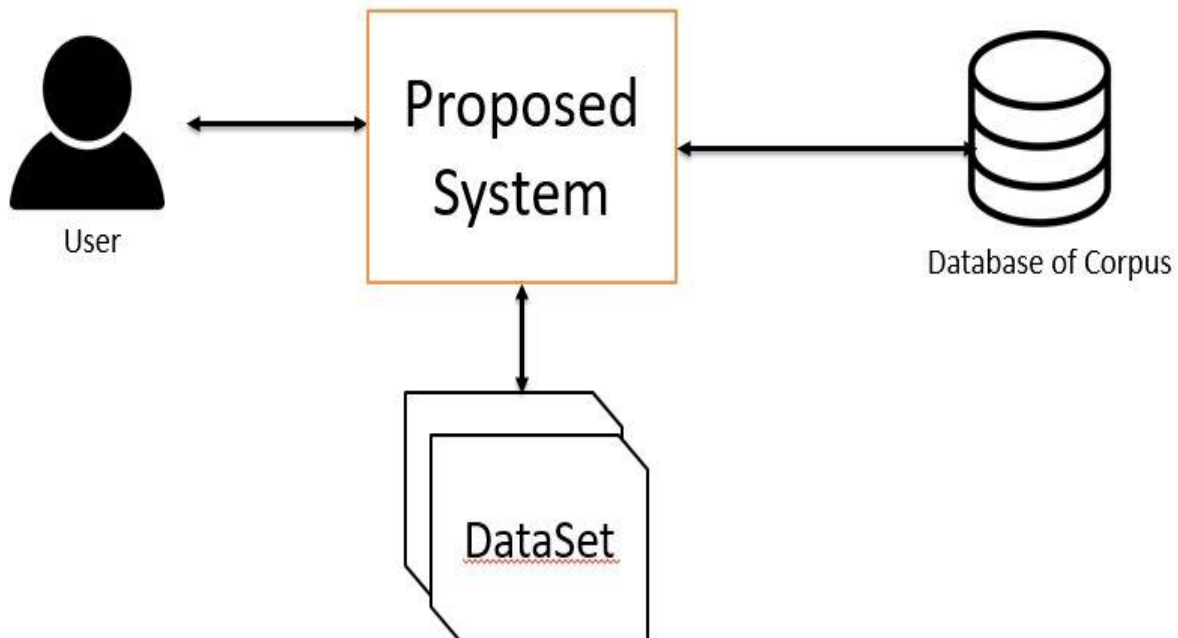
Dimensionality of Feature Space Process: The features thus selected are of high dimension. The reducing this high dimensionality is the main idea of the cluster algorithms.

Clustering Process: calculating similarity measure which denotes the similarity of content between two vectors of two documents.

Clustering Algorithm: There are many different clustering algorithms available, which are K-means, Fuzzy Clustering, Self-Organizing Maps, Expectation Minimization algorithm and so on. Selecting the right algorithm that best suits an application is a tough task.

3. Software Specification Requirements

3.1 High-Level Block Diagram



3.2 Environment Used in the Project

3.2.1 Hardware Interface Requirements

Some of the hardware requirements are as follows:

Processor : Any Intel based processor

RAM : Minimum of 1GB of RAM

Disc Space : At least 500 MB free disk space for storing project files

3.2.2 Software Requirements

Some of the software requirements are as follows:

Operating System : Windows /8/8.1/10

Languages : Java, HTML/CSS, JavaScript

Software and IDEs : Eclipse

Web Service : Tomcat

Database used : MySQL

Java Version : JDK1.6 or higher

3.3 Functional Requirements

The major functions that are provided by our system are,

FR1. Accepting input from the user

FR2. Parsing and preprocessing the document

FR3. Forming document matrix

FR4. Applying algorithm

FR5. Cluster formation

FR6. User Interface

- The system aims at providing an efficient interface for choosing between different functionalities of the project
- The web-based interface will be simple and efficient and allow users to actively interact with the system.

3.4 Non-Functional Requirements

NFR1. Performance

Performance is generally measured in terms of the time in which the system responds for the given transaction per user. The performance is the bottleneck criteria given that the accident dataset is enormous. Extra care must be taken that the processing time must not be too long.

NFR2. Scalability

Scalability is the capacity to support the expected quality of the system even if the load on the system increases. In our system, there is scope to increase the data load. We must make sure that adding more datasets to our system does not have any Regressive Effects on it.

NFR3. Reliability

Ensuring the existence of integrity and consistency of application and transactions is Reliability. Our system can be called reliable if the results of our analysis match to that of the real world. This near-real world results can be obtained if our analysis is carried out on large number of datasets rather than a very small data.

NFR4. Availability

Ensuring that a service or resource can always be accessed by a user irrespective of the situations is called Availability. Our system depends on the user input to carry out the analysis. The user input may be unexpected. This should not lead to the breakdown of our system. We must set up an environment to ensure that unexpected parameters are dealt the right way.

NFR5. Extensibility

Extensibility is the addition of extra functionality or modification of an existing functionality without causing much of an effect to the existing system. Since our system is divided into different modules, any change in one of them is least likely to affect the other. Even if it does so, we must make sure that there is consistency maintained.

NFR6. Maintainability

Following standard coding guidelines to ensure readability, easy to understand and maintainability. Maintaining the system involves correcting the flaws in the existing functionality without impacting any other components of the system. This requirement is also satisfied by the modularity implemented in our project.

NFR7. Security

Security is one of the most important requirement of any system. It involves protection to the system as a whole and also to its components such as Data. Since the datasets are collected from various sources, we need to ensure that they are kept secure. The security of our system is not compromised.

3.5 Constraints and Dependencies

Constraints:

C1. Time constraint:

Finishing the different sprints within given deadline associated.

C2. Dataset Constraint:

The algorithms proposed are not generic and cannot be applied to all the dataset. Dataset which we are providing are of specific domain with in all boundary.

C3.Database Constraint:

Limited corpus words of a particular domain for feature extraction.

Dependencies:

D1. Apache PDFbox library to read and process the pdf's.

D2. NLP library for extracting nouns.

D3. Bootstrap for designing the web interface.

3.6 Assumption

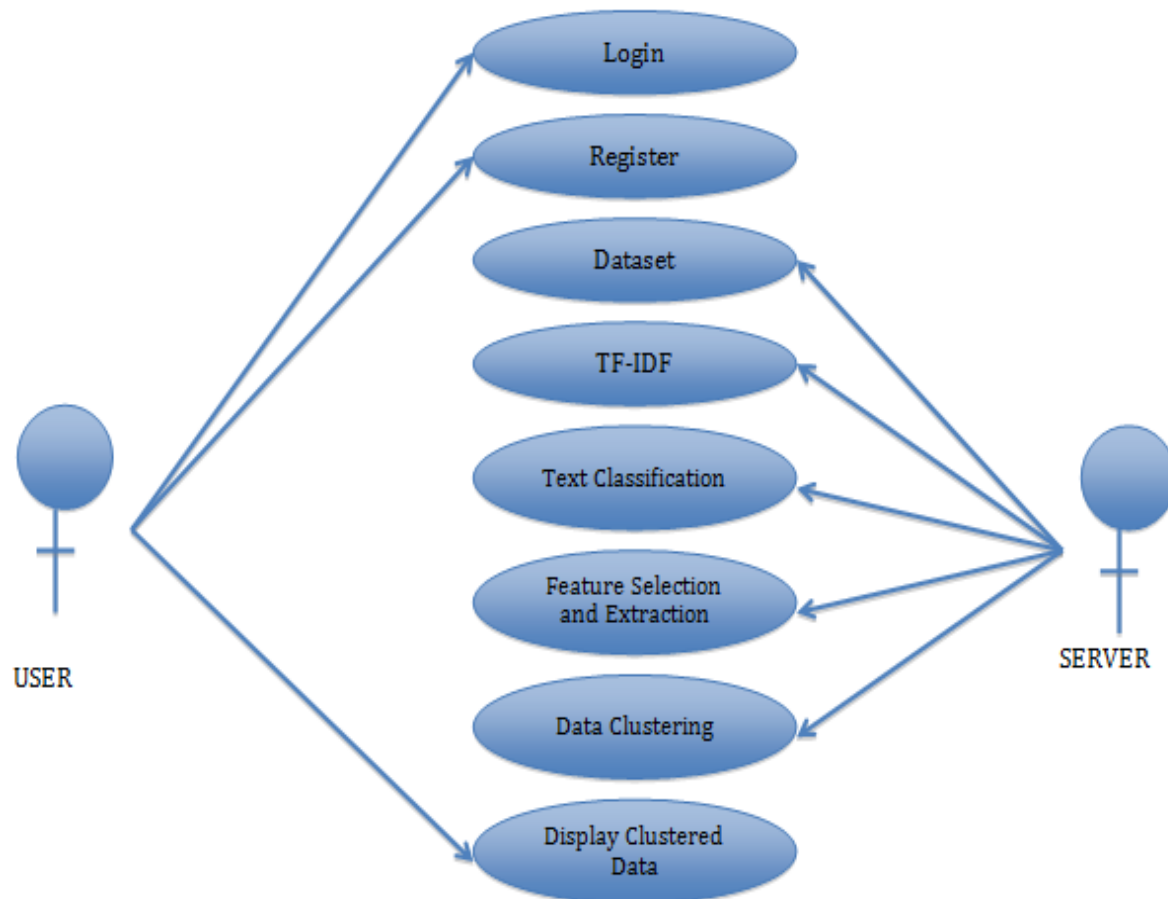
A1. Research papers are in pdf format.

A2. Since Text mining is a complex task which require expertise in computer vision and Text mining, it is not possible to start all the algorithms from scratch so we use some predefined algorithms which are very optimistic and reliable.

A3. We rely on these API's, or Algorithm so that we enforce agility in our project and also enhance reusability of the algorithms rather than doing from scratch.

A4. We assume that dataset provided for texting will be related to the trained dataset of the given specific domain.

3.7 Use Case Diagram for the requirements



3.8 Requirement Traceability Matrix

Req Id	Test Scenario Id	Test Case Id
F1	TS1	TC1
		TC2
F2	TS2	TC3
		TC4
F3	TS3	TC5
		TC6
F4	TS4	TC7
		TC8
F5	TS5	TC9
		TC10
F6	TS6	TC11
		TC12

Description of RTM:

F1: User

F2: Data cleansing

F3: forming document matrix

F4: preprocessing

F5: Algorithm

F6: Formatting the output

TS1: Front End page

TC1: UI for Clients and able to access all the features of application

TC2: Dropdown to select the domain.

TS2: process pdf

TC3: able to extract data from the pdf document.

TC4: check whether data is cleaned.

TS3: further preprocess.

TC5: removed stopwords.

TC6: removed redundant data.

TS4: feature extraction

TC7: check the formed document matrix.

TC8: database updating.

TS5: Implementation

TC9: clustering and association mining.

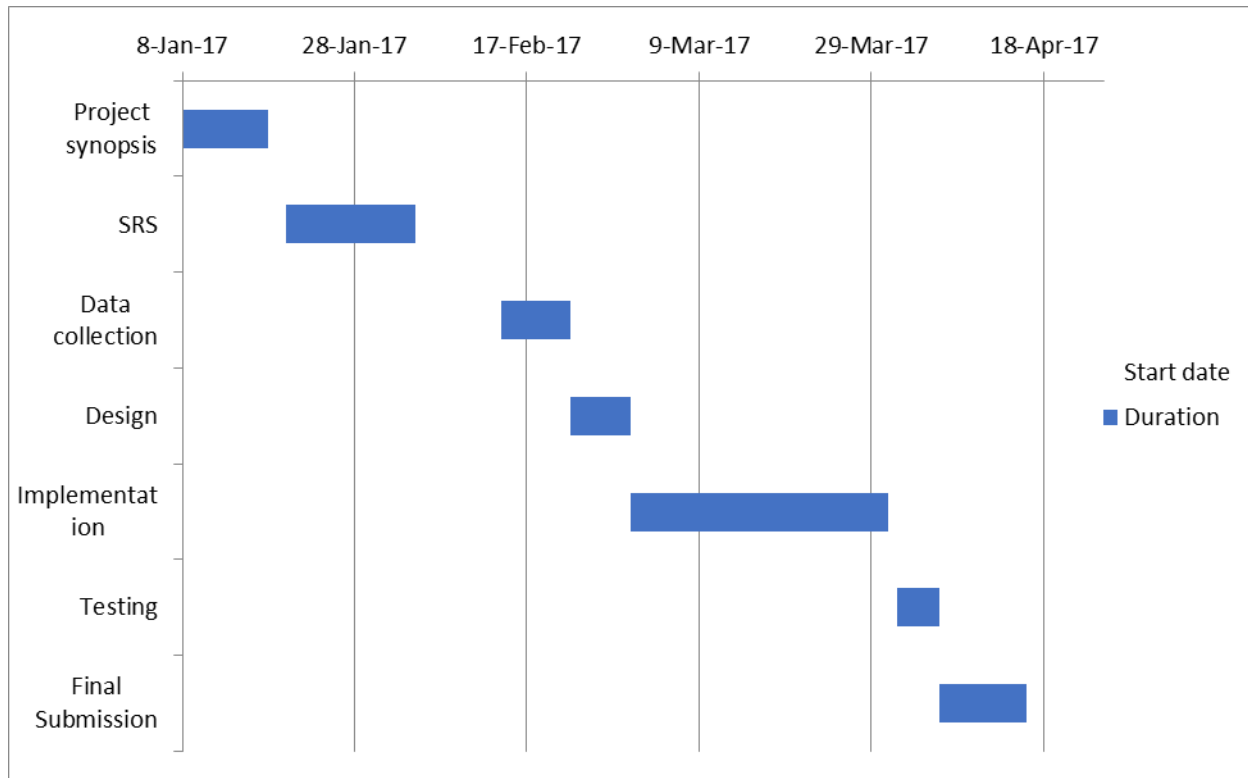
TC10: corpus updated to database

TS6: output correctness

TC11: Testing whether the interpreted results are fine.

TC12: To check whether the output is formatted and shown to user

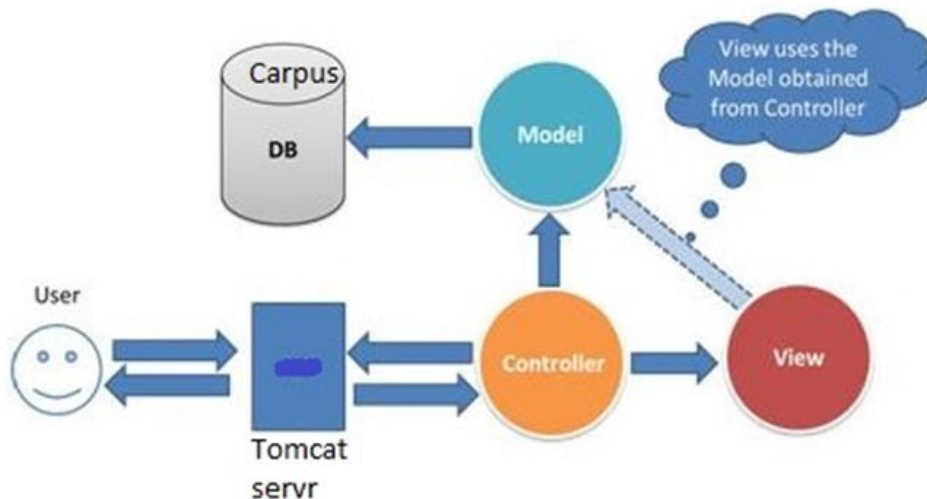
4. Schedule



5. System Design or High Level Design

5.1 Architectural Diagram

MVC Architecture



Model

This is one of the component in MVC architectural model .the work of this component is to coordinate between the other two components view and controller respectively .The make task of this component is to send data through and for.

View

This is one of the component in MVC architectural model .the work of this component is to Display the detail .It is the main UI component of the system .The main task of this system is to take the user inputs through user interaction and send the data to the model to interpret it.

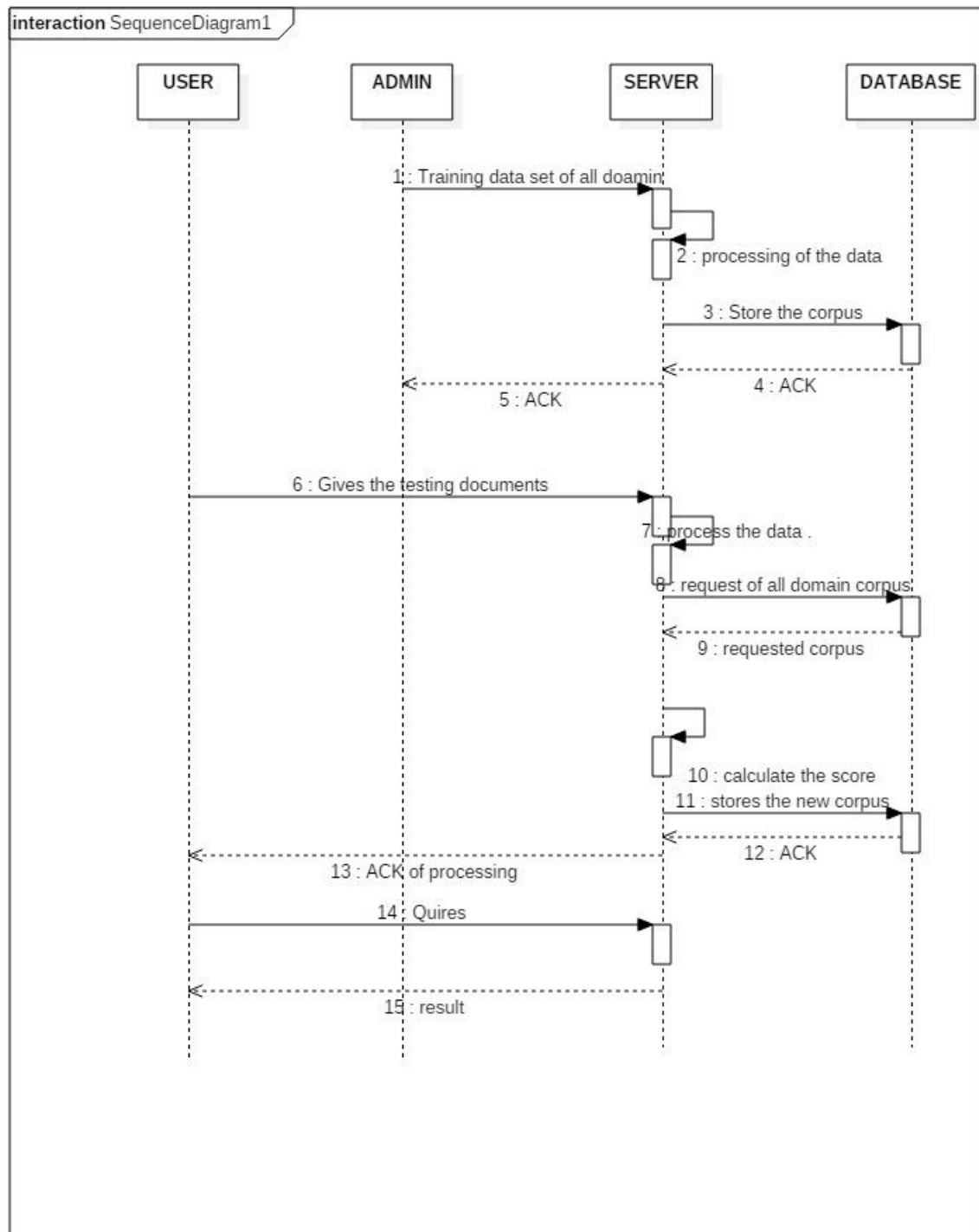
Controller

This is one of the component in MVC architectural model .the work of this component is to make logic decision for the user input and produce related command to update model component. It coordinate between the other two components view and controller respectively.

Tomcat Server

It is an open-source Java Servlet Container. Tomcat is also used to develop JSP project.it is HTTP localhost server.

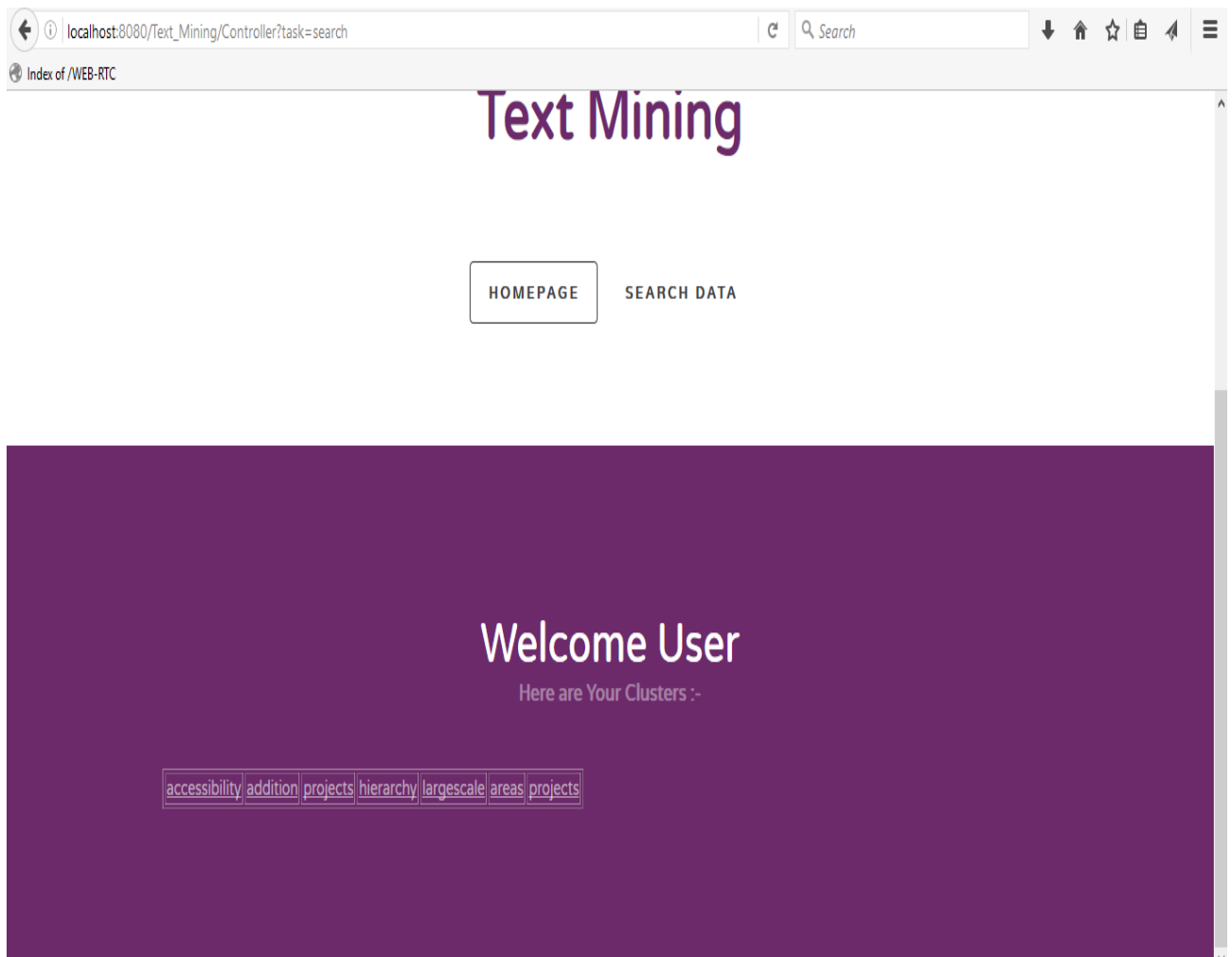
5.2 Sequence Diagrams



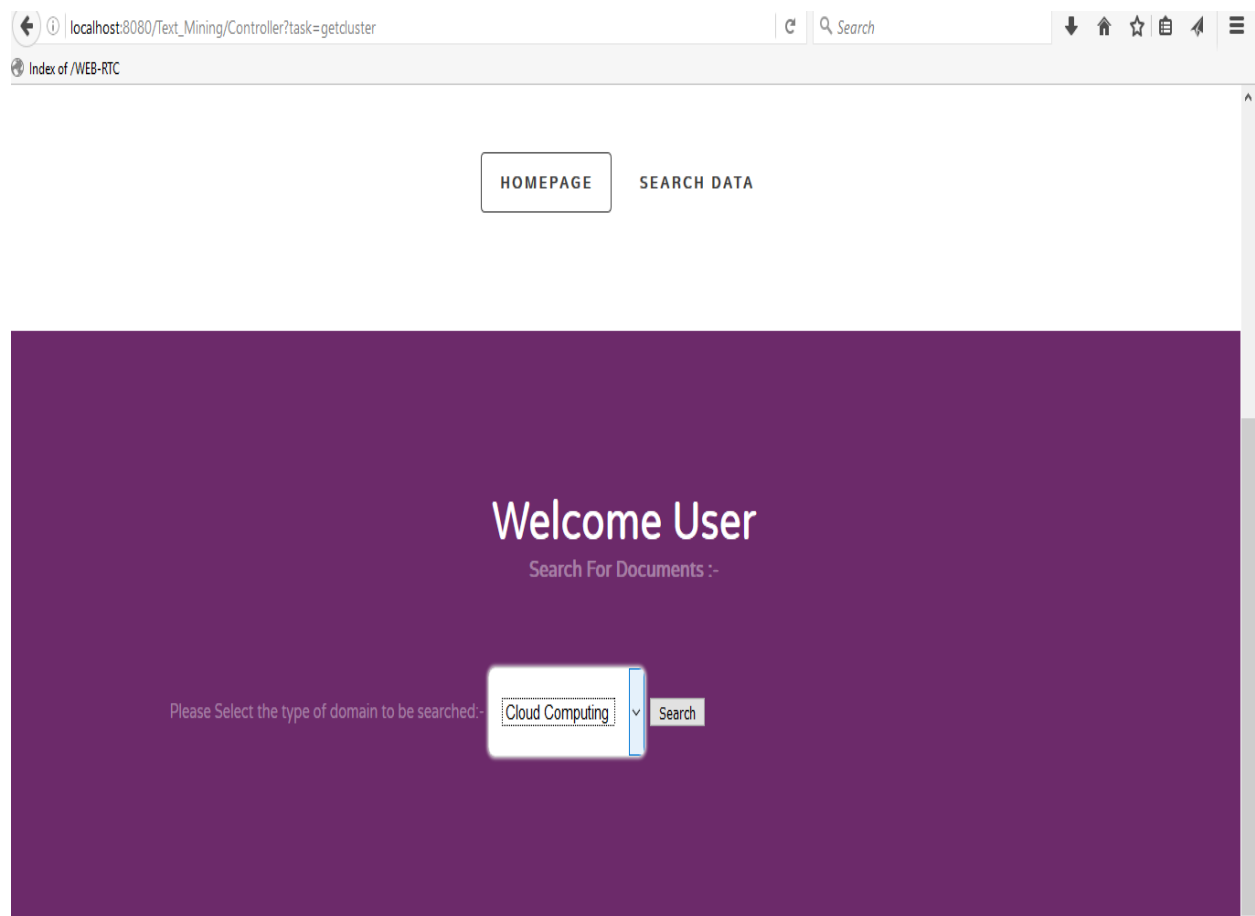
Basically in the sequence diagram there are four life line namely User, Admin, server and database. The details of each message is given below.

1. This message is from admin to server .admins provides the training dataset to the server which contains all domains data set papers.
2. In this message the data is pre-processed by the server and the terms and corpus are extracted.
3. This message is from server to database to store the corpus of each domain that has been extracted.
4. Acknowledgement is send by the database to the server.
5. Server sends acknowledgement to admin saying it has completed the training.
6. This message is from user to server which contains the test data set.
7. The pre-processing is done and terms and corpus are extracted from the dataset
8. The server ask the database to get each domains corpus.
9. The database send all the corpus of all domain
10. The server calculate the scored and finds out the papers are in which domains.
11. The new extracted corpus from test dataset ae then getting updated to the database.
12. Acknowledgement is send by the database to the server
13. Server sends acknowledgement to admin saying it has completed the testing.
14. The user quires is send to the server .It contains which domain the user wants to search the paper.
15. The result are displayed on the screen to the user .which contains a list of papers of the requested domains.

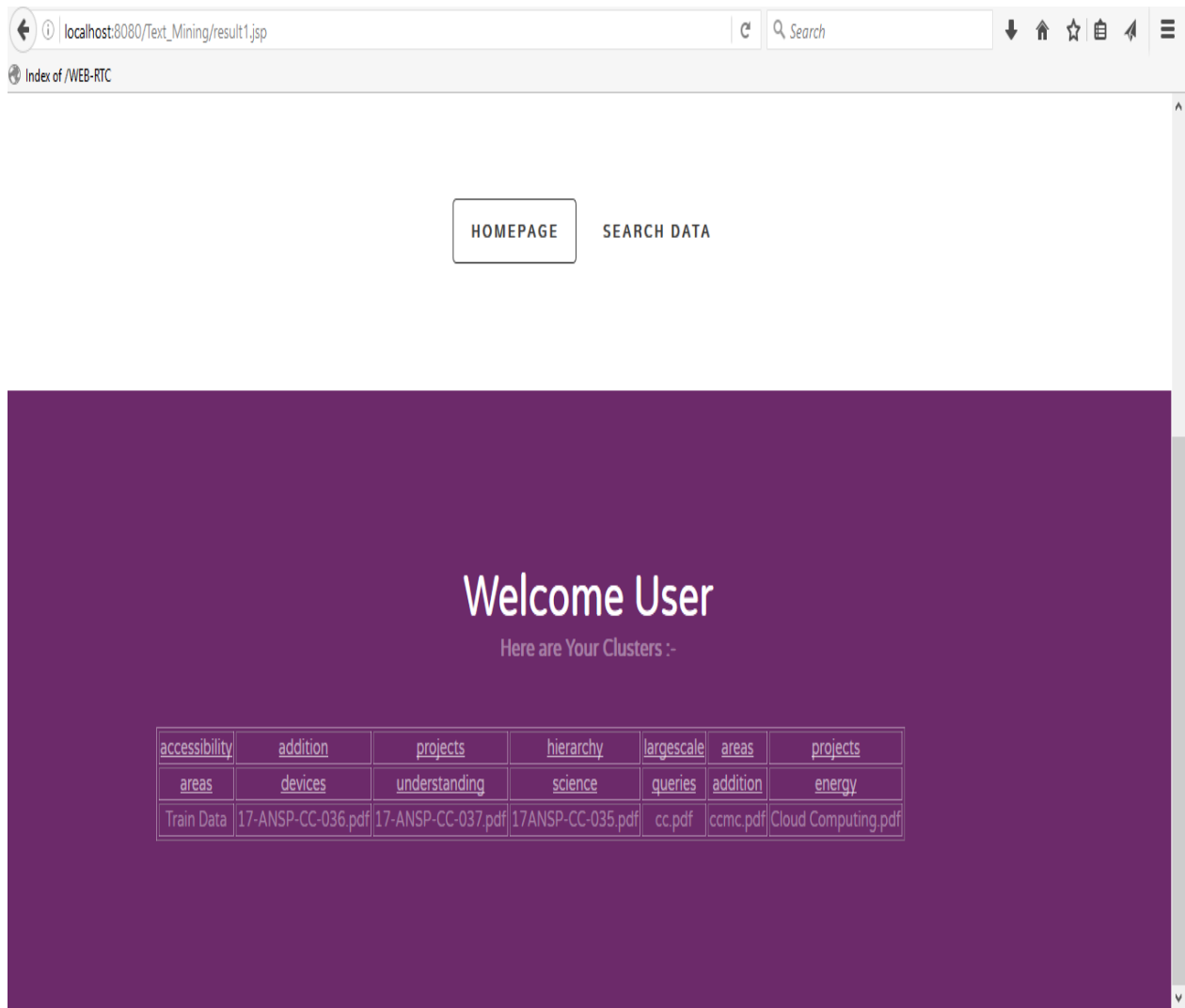
5.3 UI design



These is the first page that will appear on the screen when the user opens over application. The page consist of Home page and data Search Tap .The data search tap is shown below.



This is a screen shoot of the search data tab. When the user has placed data in the test folder, the user has to choice which domain related paper he want to search from the set. A drop down will appear in the search tab which include a list of predefined domain names where the users has to select one.



localhost:8080/Text_Mining/result1.jsp

Index of /WEB-RTC

HOMEPAGE SEARCH DATA

Welcome User

Here are Your Clusters :-

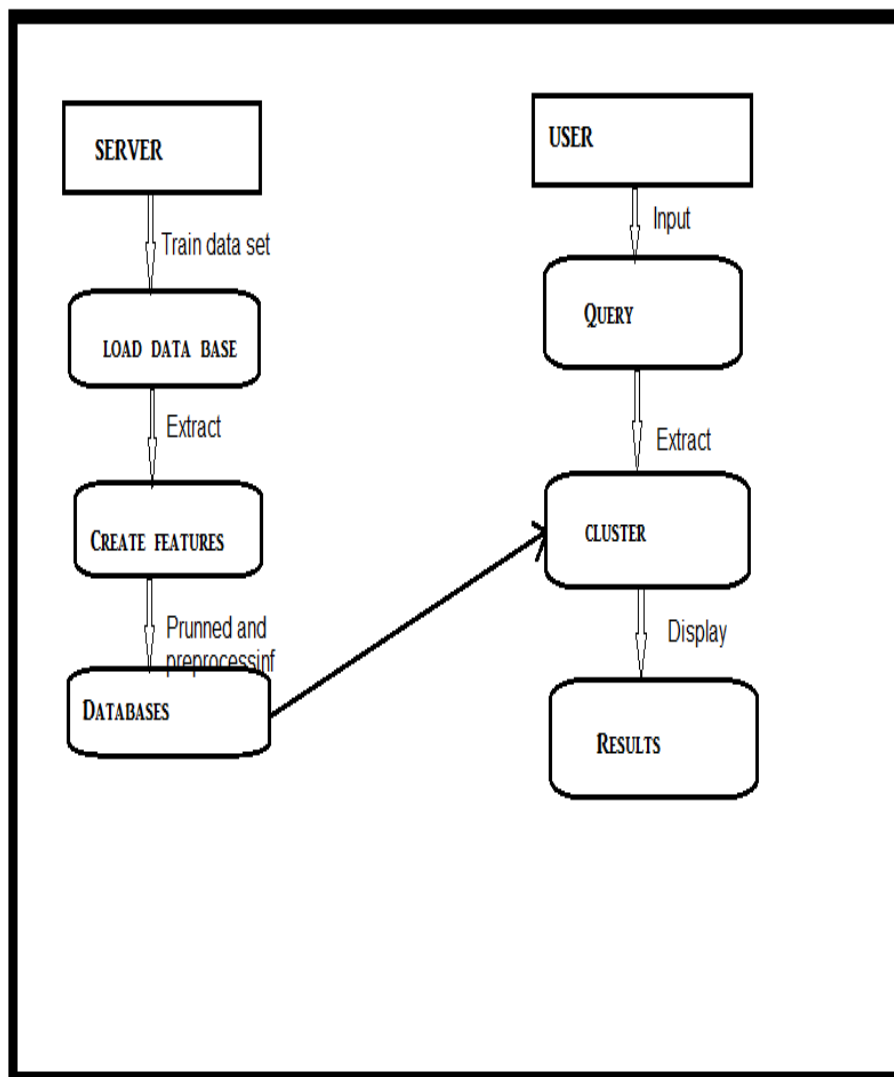
accessibility	addition	projects	hierarchy	largescale	areas	projects
areas	devices	understanding	science	queries	addition	energy
Train Data	17-ANSP-CC-036.pdf	17-ANSP-CC-037.pdf	17ANSP-CC-035.pdf	cc.pdf	ccmc.pdf	Cloud Computing.pdf

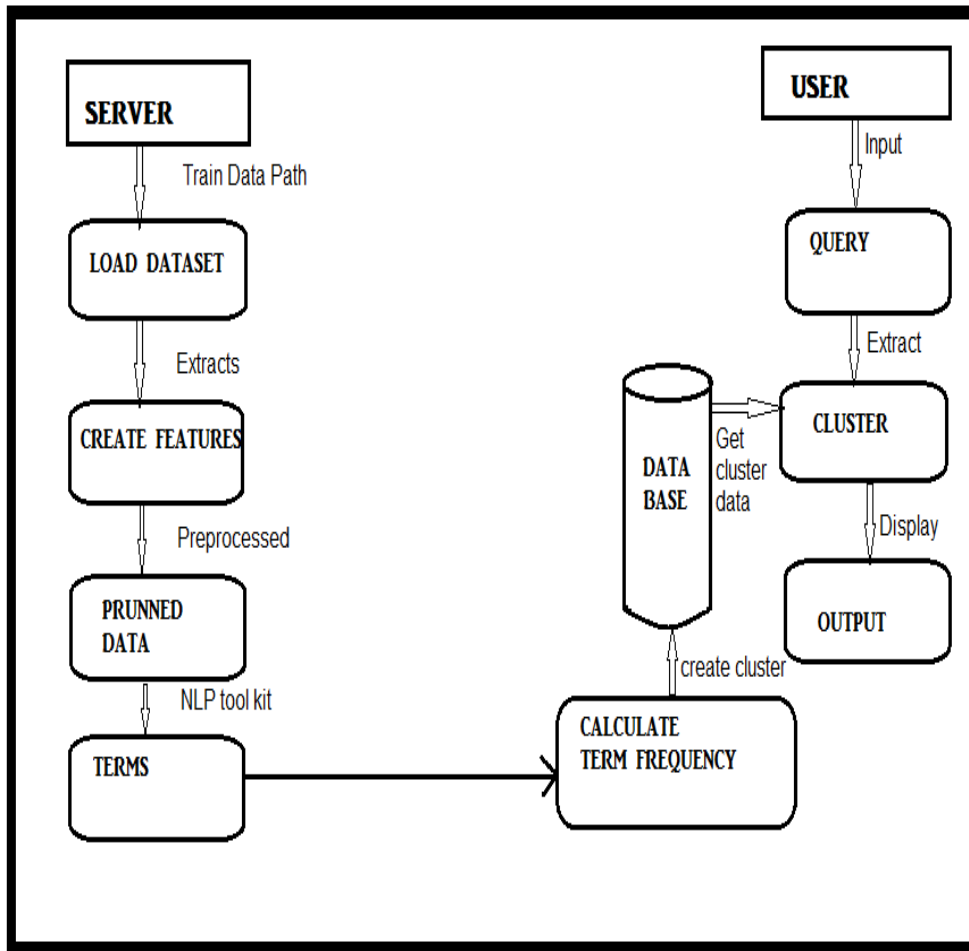
This is the main result page where the user will find the result i.e. the list of name papers of the user requested domain from the drop down.

6. Detailed Design

6.1 Modules

There are multiple modules as stated in the report. This section will iterate over the modules and other stages of the project. It will also elaborate the requirements on each module, how the module achieves its objective and what other modules it interacts with to get the objective completed.





The data set module contains the training data set in which all the domains research papers are there. These data set is the used to extract features from it and send it to pruned data module. In punned data module the data is pre-processed which include removing of stop words and noise from the data set and passed to terms module .in terms module were NLP tool kit is used to removed terms from the set and calculate the terms frequency.

After calculating the term frequency cluster are created and corpus are extracted .the corpus are updated in the corpus data base in each of the particular domain.

When the test data set are is provide it follows the same process and corpus are

6.2 Updated RTM

Requirement ID	Requirement	Design
R1	Data cleansing	D1
R2	Preprocessing	D2
R3	Document matrix	D3
R4	Algorithm	-
R5	Format Output	-

7. Implementation

7.1 Pseudo code/algorithms

We have mainly used customized TCFS algorithm in our project for clustering and association rule mining respectively.

Clustering is a very important step in text mining. It is a process of grouping data items together such that data items in one group are similar to each other in one way or another. Clustering is not any algorithm as such but is the final goal that can be achieved through various algorithms. One of the most famous and most used algorithm for clustering is K-mean algorithm. This algorithm randomly chooses different elements as the centroids of the clusters, then the iterative step of calculating distance of every particular data element from the centroid of the cluster is calculated and hence grouped. The centroid is calculated at every iteration, in this way, the elements of similar nature are grouped. We thought of implementing this particular algorithm. But then we realized that this algorithm was suitable for datasets having numerical attributes. Ours was a dataset which contained categorical values. The distance factor between any two categorical values was meaningless. Hence, we had to switch to another lesser known algorithm called TCFS Algorithm.

Another important part of our project was to find out the interesting correlations among clusters. We chose Association rule mining for performing this task. Association rule mining is a machine learning methodology to find out the interesting relations between the different variables.

During our Literature survey we found that in all the existing text clustering algorithms documents are represented by using the vector space model. Each document is considered as a vector in the term-space and is represented by the following term frequency (TF) vector:

$$\text{doc_term_freq} = [\text{term_freq}_1, \text{term_freq}_2, \dots, \text{term_freq}_h] \dots\dots\dots(1)$$

Where term_freq_i is the frequency of the i th term in the document, and h is the dimension of the text db, which is the aggregate no. of unique terms. Typically, there are few preprocessing steps, including the stop words expulsion and the stemming, on the documents. A generally utilized refinement to this model is to weight each term based on its inverse document frequency (IDF) in the corpus db. To account for the diff docs of different sizes, the length of each document vector is normalized to a unit length. Standardized vector space model is weighted by TF-IDF is utilized to represent documents during the clustering.

For the issue of clustering text docs, there are distinctive model functions available. The most generally utilized is the cosine function. The cosine function measures the similitude between two docs as the correlation between the document vectors representing them.

For two documents doc_i and doc_j, the similarity between them can be calculated as

$$\text{Cosine}(\text{doc_i}, \text{doc_j}) = \text{doc_i} * \text{doc_j} / \| \text{doc_i} \| \| \text{doc_j} \| \quad \dots\dots (2)$$

Where X represents the vector dot product and |doc_i| denotes the length of vector doc_i. The cosine value is 1 when two docs are identical and 0 if there is nothing in common between these two documents. The larger this cosine value means these two docs share more terms which implies they are more similar.

7.2 Codebase structure

TextMiningProject

src

(Default package)

Controller.java

com.algorithm

ExtractData.java

LoadDataset.java

StopWords.java

TermDocumentMatrix.java

com.database

DBConnection.java

DBFormat.java

DBQuery.java

com.implementation

Connector.java

Constants.java

PropertyImpl.java

com.nlp

TaggerDemo.java

Constants.java

PropertyImpl.java

UI

index.jsp

search.jsp

result.jsp

result1.jsp

Libraries

Mysql-connector-java-5.1.6-bin.jar

Pdfbox-app-2.0.4.jar

Stanford-postagger-3.7.0.jar

7.3 Coding Guidelines Used

7.3.1 Why Have Coding Guidelines

Coding Guidelines are very important to a developer for so many reasons:

1. 70-80% of the overall cost of a system goes to maintenance.
2. Hardly any system is maintained for its entire life by its original developer/author.
3. Coding Guidelines increases the readability of the software, allowing other developers to understand the code more rapidly and completely

7.3.2 Java Coding Guidelines Followed

1. Mention the Package name in the very beginning.
2. Make all the imports at the start of the file. Imports are grouped into following order:
 - a. Standard library imports
 - b. Third-party imports
3. Single quotes followed for all String literals because they are easier to read to type and also avoiding escape characters for double quotes
4. Meaningful names for each function and variables.
5. All possible exceptions are handled with finally block.
6. Coding lines are not terminated with semicolons and neither used semicolons to put two commands on the same line.
7. Maximum line length is 70 characters.
8. Code blocks are indented with 4 spaces.
9. Two blank lines between top-level definitions, one blank line between method definitions
10. All the functions, modules and classes are commented in-lined.
11. Explicitly closing files and images opened.
12. Naming convention followed :-

module_name, package_name, ClassName, method_name, ExceptionName, function_name,
GLOBAL_CONSTANT_NAME global_variable_name, instance_variable_name,
function_parameter_name, local_variable_name
13. Followed standard typographic rules for the use of spaces around punctuation.

7.4 Sample Code

```

Java EE - Text Mining/src/com/algorithm/LoadDataset.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
ExtractData.java LoadDataset.java StopWords.java Term_document_Matrix.java

package com.algorithm;
import java.io.File;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.Iterator;
import java.util.LinkedHashSet;
import com.database.DBQuery;
import com.impl.Constants;
import com.nlp.Impl.TaggerDemo;

public class LoadDataset extends DBQuery{
    private ExtractData extract = null;
    private TaggerDemo posTagger = null;

    public LoadDataset() throws SQLException{
        extract = ExtractData.getInstance();
        posTagger = new TaggerDemo();
        trainData();
        start();
    }

    public static void main(String[] args) throws SQLException {
        new LoadDataset();
    }
  
```

```

Java EE - Text Mining/src/com/algorithm/LoadDataset.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
ExtractData.java LoadDataset.java StopWords.java Term_document_Matrix.java

    public static void main(String[] args) throws SQLException {
        new LoadDataset();
    }

    private void trainData() throws SQLException{
        String dataset_path = System.getProperty(Constants.system_dir);
        dataset_path = dataset_path+File.separator+"Training Dataset";
        String taggerPath = Constants.taggerPath;
        ArrayList<String> document = new ArrayList<String>();
        HashSet<String> terms = new HashSet<String>();

        File[] files = new File(dataset_path).listFiles();
        for(File f:files){
            if(f.isDirectory()){
                String type = f.getName();
                System.out.println("Name = "+type);
                File[] F = f.listFiles();
                for(File f1:F){
                    String data = extract.getData(f1,type);
                    data = StopWords.sorting(data);
                    data = StopWords.stopwords_new(data);
                    ArrayList<String> nouns = LoadDataset.removeRedundantData(posTagger.extractNouns(taggerPath, data));
                    String N = convertArrayListToString(nouns);
                    document.add(N+"-"+type);
                    String[] n = N.split(" ");
                }
            }
        }
    }
  
```

```

Java EE - Text Mining/src/com/algorithm/LoadDataset.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access Java EE Java

ExtractData.java LoadDataset.java StopWords.java Term_document_Matrix.java

49 String[] n = N.split(" ");
50 for(String n1:n){
51     terms.add(n1);
52 }
53 }
54 HashSet<String> closedTerms = new HashSet<>();
55 Iterator<String> itr = terms.iterator();
56 while(itr.hasNext()){
57     String term = itr.next();
58     if(!term.equalsIgnoreCase("abstract") || !term.equalsIgnoreCase("introduction")){
59         double d = Term_document_Matrix.tf(document, term);
60         if(d==1 && term.length()>=5){
61             closedTerms.add(term);
62         }
63     }
64 }
65 if(type.equalsIgnoreCase("cc")){
66     closedTerms.add("cloud");
67 }
68 else if(type.equalsIgnoreCase("nw")){
69     closedTerms.add("network");
70 }
71 else if(type.equalsIgnoreCase("mc")){
72     closedTerms.add("mobile");
73 }
74 }
  
```

```

Java EE - Text Mining/src/com/algorithm/LoadDataset.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access Java EE Java

ExtractData.java LoadDataset.java StopWords.java Term_document_Matrix.java

212 private ArrayList<String> getData(String type) throws SQLException{
213     ArrayList<String> out = new ArrayList<String>();
214     ResultSet rs = DB_SELECT(null, null, "corpus", "without condition");
215     while(rs.next()){
216         String d = rs.getString(type);
217         out.add(d);
218     }
219     return out;
220 }
221 }
222 public static ArrayList<String> removeRedundantData(ArrayList<String> data)
223 {
224     LinkedHashSet<String> set = new LinkedHashSet<String>(data);
225     return new ArrayList<String>(set);
226 }
227 }
228 public static String convertArrayListToString(ArrayList<String> data)
229 {
230     String out = "";
231     Iterator<String> itr = data.iterator();
232     while(itr.hasNext()){
233         out+=itr.next()+" ";
234     }
235     out = out.substring(0, out.lastIndexOf(" "));
236     return out;
  
```

7.5 Unit Test Cases

A unit testing is a level of testing where smallest part of individual unit/component is tested to determine if they are fit for use.

The test case fields can be:

- Test Case ID
- Test Case Purpose
- Procedure / steps to be performed
- Expected Result
- Actual Result
- Remarks

Test Cases for Login Page:

- Test if user is able to login successfully.
- Test if unregistered users is not able to login to the site.
- Test with valid username and empty password such that login must get failed.

Test Cases for Selecting the Domain:

- Provide drop down for all the available domains.
- Output should come after selecting a domain.

Test Cases for Results:

- Check whether it is showing the corresponding documents of selected domain.
- Test whether every document is getting displayed for that particular domain,
- Test nothing is working before uploading the dataset on to the database.

7.6 Metrics for Unit test Cases

S.No.	Unit Test Metrics	Data Retrieved during test case development and execution
1	No. of Requirements	3
2	No. of Test cases written for all requirements	12
3	Total No. of Test cases executed	12
4	No. of test cases Failed	0
5	No. of test cases unexecuted	-
6	Total no. of defects identified	1
7	No. of test cases passed	12
8	Critical defects found	0
9	Medium defects found	2
10	Low defects found	3

7.7 Updated RTM

Requirement ID	Requirement	Design	Code
R1	Data cleansing	D1	Is_data_clean
R2	Preprocessing	D2	Is_preprocessed
R3	Document matrix	D3	doc_matrix
R4	Algorithm	-	Is_cluster
R5	Format Output	-	Is_format

8. Testing

It does not matter, how good of an algorithm one has used, or how many hours have been spent on coding the proposed algorithm, in the field of prediction, if the results are not good enough, then everything is a waste. Saying that our project did give us good results at some place and mixed ones at others.

One of the main reasons for this anomaly was that single paper can have terms which belongs to two different domain. The accuracy ratio can be calculated total number of documents passed for clustering which belongs to a particular domain, and the number of documents actually got clustered in that particular domain. Hence the accuracy will fall into a big range, for us it was from 70 % to as high as 80 %.

When we paid a closer attention to these lower accuracies, we found that some of the corpus got stored on the database are not relevant at all and some belongs to two or more than two domain also.

Stating this, now let's have a detailed analysis of how testing was done.

8.1 System Test Specifications

The system testing specification used for the project is derived from the software requirements specification and the functional requirements specification of the software system. The system testing specification includes test cases to maintain the overall acceptability/viability of the software for the end user apart from testing the most basic features and functionality of the software. It includes performance testing, checks for basic security practices, usability test scenarios and other general test cases.

Some of the test cases that were executed:

General test cases:

1. Application crash messages, database errors and other implementation specific error details must not be displayed in the production version of the software. All such instances of error must be redirected to a particular error page.
2. Corpus must be formatted and stored properly.
3. Timeout values wherever used must be configured adequately keeping the system states and expected behavior for the user in mind.
4. Validation and error messages must be shown at correct locations in the user interface.
5. All error messages must follow a standard style procedure.

GUI Test scenarios:

1. Formatting parameters such as font sizes, style, font color, background color for various elements on the user interface etc. must have values in adherence to the ones agreed upon in software requirements specifications and functional requirements specifications.
2. Check all the window with proper background.
3. Textboxes that are only readable are made to be prevented from writing.
4. As the page loads, default values of radio buttons, placeholder values in input fields etc. must be made.
5. Make sure the user interface loads within an expectable time frame.

Security:

1. Sanitize input fields for SQL injection attacks.

8.2 Test Environment Used

As mentioned above, most of the test cases was the actual output of the project. We have used Windows Operating System for out test environment.

Some of the hardware requirements are as follows:

- Processor : Any Intel-based processor
- Ram : Minimum of 2 GB of RAM
- Disk Space : 4-6 GB for a typical installation 1 GB for datasets

8.3 Test Procedure

On initial basis, the testing was done manually, i.e., looking at the passed dataset of different domains and checking whether it got clustered in its respective cluster matching them with the text content, then using a calculator to get the % accuracy of the concerned plants.

Due to limited knowledge of NLP and Semantic ontology, we took only a handful of documents in our list, rather than taking n number of documents

8.4 Example Test Result

Test Case ID: TC5

Test Priority: High

Module Name: Stop Words Removal

Test Name: Verifying whether all the stop words got removed

Test Description: Testing in different ways to make sure that all the stop words which were initially present should get removed.

Precondition: Pdf Content is passed as it is from the respective documents.

Post condition: All the stop words should get removed in the output.

Dependencies: The only dependency is it looks for the stored stop words in the db to actually remove the it.

8.4 Test Metrics

S.No.	Unit Test Metrics	Data Retrieved during test case development and execution
1	No. of Requirements	4
2	No. of Test cases written for all requirements	8
3	Total No. of Test cases executed	8
4	No. of test cases Failed	0
5	No. of test cases unexecuted	-
6	Total no. of defects identified	2
7	No. of test cases passed	8
8	Critical defects found	1
9	Medium defects found	1
10	Low defects found	6

8.5 Updated RTM

Requirement ID	Requirement	Design	Code	Test Cases
R1	Data cleansing	D1	Is_data_clean	T1
R2	Preprocessing	D2	Is_preprocessed	T2
R3	Document matrix	D3	doc_matrix	T3
R4	Algorithm	-	Is_cluster	T4
R5	Format Output	-	Is_format	T5.

Description:

T1: Is data cleaned for further stage.

T2: Is preprocessing done.

T3: Ensure that document matrix generated.

Verify whether data is updated in the database.

T4: Is cluster done in right manner.

Ensure the rules have given correctly.

T5: Verify whether output is formatted and displaying in the UI.

9. Results and Discussion

9.1 Results

This project has a lot of scope for the future development and innovation. As of now system is trained for five domains that the application supports and can be extended to identify each and every other domain present so that people from all research domain can use this software. For now system supports only 5 domains namely networking, cloud computing, data mining.

This application can be trained with other then research papers like news etc. which can classify the new in some or the other way.

The results that are obtained are immensely good. We have obtained an accuracy of 81%. The software developed is able to predict the documents domain correctly from the test data set provided.

User can seamlessly use the software by just selecting the documents testing path from the application and search the particular domain, and the results will be available.

Improvement can be made in reducing the time required to predict and annotate the documents present in the image. This can be improved and scaled so that the results can be obtained faster. This is achieved by using concepts like multi-threading etc.

10. Retrospective

Literature survey is most important thing for PHD student when it comes to research. Most of the student waste their time finding out the relevant papers for their research. Over system helps to finds their interested papers is a mixed set of dataset.

The project went very well .The formation the cluster and removing and updating the corpus was well planned and executed. The NLP tool we used was well suited for my project work. We used java and Sql which we had known from good time which made over coding and integration easy.

There were few challenges that we faced during the whole process:-

Finding the dataset for training the system of all domain was a tuff job. Finding the right NLP tool that suits over system. Integration of all the module was tuff job.

Extraction the data from pdf format took a while to figure it out.

We got to know new technology which we were unaware of .we were introduce to text mining from the scratch , were we learnt new algorithms of datamining and text mining .

Learnt few concept of NLP.

11. Bibliography

- [1]Chen Wenliang, Chang Xingzhi, and Wang Huizhen, “Automatic Word Clustering for Text Categorization Using Global Information” Copyright ACM 2004.
- [2]Dino Ienco Rosa Meo “Exploration and Reduction of the Feature Space by Hierarchical Clustering” Dipartimento di Informatica, Universit`a di Torino, Italy.
- [3]Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques, Second Edition, University of Illinois at urbanachampaing 2006.
- [4]George Forman “Feature Selection: We've barely scratched the surface” Published in IEEE Intelligent Systems, November 2005.
- [5]Ajay S. Patil, B.V. Pawar, Automated Classification of Web Sites using Naive Bayesian Algorithm, IMECS vol-1, 2012.
- [6]Ms. Darshna Navadiay, Mr. Mehul Parikh, Ms. Roshni Patel, Constructure Based Web Page Classification, International Journal of Computer Science and Management Research, Vol 2
- [7]Huan Liu, “Evolving Feature Selection” Published by the IEEE Computer Society 2003.
- [8]Jinxu Chen¹ Donghong Ji¹ Chew Lim Tan “Unsupervised Feature Selection for Relation Extraction” Institute for Infocomm Research 2002.
- [9]Martin H.C. Law and Mario A.T. Figueiredo “Simultaneous Feature Selection and Clustering Using Mixture Models” iEEE 2004.
- [10]Tao Liu and Shengping Liu “An Evaluation on Feature Selection for Text Clustering” Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

[11]Yanjun Li Congnan Luo, “Text Clustering with Feature Selection by Using Statistical Data” IEEE 2008.

[12]Zhao, Y. and Karypis, G., “Clustering Algorithms for Document Datasets”, Data Mining and Knowledge Discovery [C].10(2), pp.141-168, 2005.

[13]Liu, F. and Xiong, L., “Survey on text clustering algorithm -Research present situation of text clustering algorithm” IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS), pp.196-199, 2011.

[14]Zhao, Y., and Karypis, G., “Evaluation of Clustering algorithms for document datasets”, International Conference on Information and Knowledge Management, McLean,Virginia, United States, pp.515-524, 2002.