

A Project Report on
“CREATION OF PRESENTATIONS WITH
FEATURE SPECIFIC MULTI-DOCUMENT TEXT
SUMMARIZATION”

Submitted in partial fulfillment of the requirement for
Degree in Bachelor of Engineering (Information Technology)

By
VRISHALI BHOR (501109)
KEVEN SEBASTIAN (501123)
VARUN SAXENA (501148)
PRAVIN HODGE (500714)

Guided by:
Mrs. POONAM BARI



Department of Information Technology
Fr. Conceicao Rodrigues Institute of Technology
Sector 9A, Vashi, Navi Mumbai – 400703
University of Mumbai
2014-2015

CERTIFICATE

This is to certify that the project entitled

CREATION OF PRESENTATIONS WITH FEATURE SPECIFIC MULTI-DOCUMENT TEXT SUMMARIZATION

Submitted By

VRISHALI BHOR

KEVEN SEBASTIAN

VARUN SAXENA

PRAVIN HODGE

In partial fulfillment of degree of **B.E. in Information Technology** for term work
of the project is approved.

External Examiner

Internal Examiner

Internal Guide

Head of the Department

Principal

Date: - / /2015

College Seal

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We 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.

VRISHALI BHOR (501109)

KEVEN SEBASTIAN (501123)

VARUN SAXENA (501148)

PRAVIN HODGE (500714)

Date: / /2015

ABSTRACT

In this technological era, almost all the educational institutes and corporate industries make use of presentations to convey information to the targeted audience. However, extracting useful information and making presentations is a time consuming process. There is abundance of data available on the internet. Choosing useful sources of data and then summarizing the provided contents requires sincere effort. It is an attempt to develop an automatic procedure designed to extract the information from multiple text documents written about the same topic using clustering and feature specific algorithm. Resulting summary allows individual users, such as professional information consumers, to quickly familiarize themselves with information contained in a large collection of documents. The software takes the documents containing information from the user as the input for summarization. The output is in the form of a presentation containing the gist of the topic under consideration.

INDEX

Sr. No.	Topic	Page No.
1	Introduction	1
	1.1 Introduction	2
	1.2 Automatic Text Summarization	2
	1.2.1 Clustering	2
	1.2.2 Feature Specific Summarization	3
	1.3 Objective	4
	1.4 Overview of Proposed System	4
2	Literature Survey	6
	2.1 Introduction	7
	2.2 Existing systems related to Text Summarization	7
	2.3 Existing systems related to Presentation makers	11
3	Proposed System	13
	3.1 Introduction	14
	3.2 Problem Statement	14
	3.3 Scope	14
	3.3.1 Goals and Objectives	14
	3.3.2 Features and Advantages	15
	3.4 System Requirements	15
	3.4.1 Minimum Software Requirements	15
	3.4.2 Minimum Hardware Requirements	15
	3.5 Time Line Chart	16
	3.5.1 Time Line Chart for Semester VII	16
	3.5.2 Time Line Chart for Semester VIII	16
4	Design	17
	4.1 Introduction	18

	4.2 Architectural Design	18
5	Implementation	21
	5.1 Introduction	22
	5.2 Pseudocode and Algorithms	22
	5.3 Flow Charts	33
6	Conclusion and Future Scope	39
	1. Conclusion	40
	2. Future Scope	40
	Appendix A: Snapshots	41
	Appendix B: Data Set	44
	References	47
	Acknowledgement	48
	Paper Published and Certificates	49

List of Figures and Images

Chapter. No.	Figure Name	Page No.
3	3.5.1 Timeline chart for Semester VII	16
	3.5.2 Timeline chart for Semester VIII	16
4	4.1 The Proposed System	18
5	5.1 Methodology	22
	5.2 Preprocessing	33
	5.3 Clustering	34
	5.4 Score Generation based of feature profile	35
	5.5 Summary Generation	36
	5.6 Presentation Creation	36
Appendix A	7.1 Login Form	42
	7.2 Main Form	43
	7.3 Final Presentation	43
	Paper Publish	49

List of Tables

Chapter No.	Figure Name	Page No.
5.	5.5 Test Cases	37

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

A presentation, as the name suggests, is a tool used for presenting and explaining an idea or a particular topic to the targeted audience. It is typically a demonstration, lecture, or speech meant to inform or persuade the listeners. The corporate and educational institutes use presentations extensively as a tool to convey their knowledge. However, creating a good presentation is a time consuming process. It requires a huge amount of information to be extracted and analyzed which can sometimes be tedious and tiring. To address this problem, we have proposed to create software that automates the process of creating presentation with the help of data mining concepts. It works in works two phases- multi-document summarization and creation of presentation from summarized information. This chapter highlights the basic concepts of text summarization using clustering and feature specific methodologies and creation of presentations. It also puts light on the objective and overview of our proposed system.

1.2 AUTOMATIC TEXT SUMMARIZATION

Due to the growth of internet, the amount of information available at one's expense has increased, which in turn, has increased the problem of information overload. Thus, Summarization is the need of the era. According to [1], 'A summary can be defined as a text that is produced from one or more texts, that contains a significant portion of the information in the original text(s), and that is no longer than half of the original text(s)'. Automatic text summarization is the technique, where a computer summarizes a text. A text is entered into the computer and a summarized text is returned, which is a non-redundant extract from the original text. Summaries should preserve the essence of the subject of its source(s). A summary should typically be short, precise and to the point. There are two methods of summarization, extractive and abstractive.

1. Extractive methods work by selecting a subset of existing words, phrases, or sentences in the original text to form the summary. Extractive summarization is mainly concerned with what the summary content should be, usually relying solely on extraction of sentences.
2. Abstractive methods build an internal semantic representation and then use natural language generation techniques to create a summary that is closer to what a human might generate. Such a summary might contain words not explicitly present in the original.

1.2.1 CLUSTERING

Cluster analysis is a multivariate method which aims to classify a sample of subjects (or objects) on the basis of a set of measured variables into a number of different groups such that similar subjects are placed in the same group. It can be defined as “Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups [1].” Due to this feature, clustering plays an important role in data mining concept and to extract the important and summarized information from a huge amount of data. Clustering is used to identify themes or subtopics of common information, because multiple documents relating to a particular topic are likely to contain redundant information in addition to information unique to each document. Once themes have been known, a representative passage in each theme is selected and included in the summary.

A brief overview on clustering used in text summarization is:

1. There are various algorithms used for clustering. In the process of clustering, similar objects are grouped together.
2. Thus it leads to increase in inter cluster distances and minimizes the intra cluster distances.
3. As a result various clusters are formed.
4. These clusters are arranged in an order.
5. Further these clusters are required for mining the data and creating a summarized text from a document.

1.2.2 FEATURE SPECIFIC SUMMARIZATION

Feature based text summarization is used for ordering sentences in a cluster by analyzing the features of each sentence. On capturing sentence-specific features we compute the feature profile of each sentence [2]. In the proposed method we consider 6 features for ranking sentences. They include title feature, sentence length, term weight, sentence position, proper noun, and numerical data.

A brief overview of the above features is as follows:

1. Title feature: It represents the number of words in a sentence that match with the words in the title. Higher the score of title feature, higher is the ranking of the sentence [3].
2. Sentence length: It is used to identify normalized length of sentences in a source. Shorter sentences are not expected to be a part of the summary.

3. Term weight: It depicts the number of occurrences of a term in a document. The term weight corresponds to the importance of the term in context with the subject under consideration.
4. Sentence Position: According to this feature, the opening and closing sentences usually contain the gist of the topic and hence are given more importance.
5. Proper noun: It is the number of proper nouns the sentence contains. Having a higher score in the feature denotes higher importance of the sentence.
6. Numerical data: Sentences having numerical data are usually necessary to be included in the summary. This feature is the ration of numerical data in a sentence and the length of the sentence.

1.3 OBJECTIVE

Today, the current system of text summarization technique is a complex issue that uses the concept of data mining. Thus creating a summary is not an easy task. Also, the current system creates only the summary of the document but does not provides anything in creating the presentation. The objective of the system proposed, is to develop software that accepts multiple documents as input to provide a concise and to-the-point presentation (as well as a summary) that conveys the content in the source documents in a presentable manner. The source documents are initially clustered to obtain groups of related sentences. Feature profiles are generated for ranking of these sentences which then form a part of the presentation. The software completely automates the process of creating presentations by choosing the topic for each slide on its own. However, it also allows the user to personalize the presentation in case the user wants to.

1.4 OVERVIEW OF THE PROPOSED SYSTEM

In our proposed system, the input to the system is a collection of documents. Depending on the user's choice, the system provides output in the form of a concise summary or a presentation covering the important points of the subject. The proposed approach produces an extractive summary by selecting salient sentences from the documents cluster wise. All the relevant documents are grouped together into clusters by using threshold-based document clustering approach. Based on feature profile salient sentences from each cluster are identified and ranked according to their weights of importance. Based on the ranking of sentences, sentences are selected

and ordered. The system then iteratively extracts one sentence at a time, and pastes it on the appropriate slide. The user is also provided with a facility to insert an image(s) in the slide by simply browsing in the directories or selecting from a range of pictures that the software retrieves from Google.

The proposed approach can be decomposed into seven sub processes:

- 1. Data preprocessing:** From the collection of documents, the boundaries of sentences are identified and the documents are split into sentences. Sentences are in turn split into words. Frequently occurring insignificant words called functional words or stop words like “a”, “the”, “of” are removed because they do not contribute to the meaning of the sentence. [2]
- 2. Documents representation:** In this step, the cleaned and noise free data is subjected for documentation representation which is further subjected for data clustering.
- 3. Clustering of data:** In this, the represented documents are clustered on the basis of the objects that form the same group.
- 4. Score generation based on feature profile:** Based on feature profile salient sentences from each cluster are identified and ranked according to their weights of importance.
- 5. Summary generation:** In this stage, based on the ranking of the clusters, summary is generated.
- 6. Presentation creation:** It is the final stage in which the summarized text is subjected to creation of presentation. During this process, the user is also provided with a facility to insert an image(s) in the slide by simply browsing in the directories or selecting from a range of pictures that the software retrieves from Google.

CHAPTER 2

LITERATURE

SURVEY

2.1 INTRODUCTION

This chapter highlights the different systems related to text summarization and presentation generation. The advantages and disadvantages of each of them are also explained below.

2.2 EXISTING SYSTEMS RELATED TO TEXT SUMMARIZATION

2.2.1 H. P. LUHN

This system could only have a single-document input and has a domain specific to technical articles. Term filtering and word frequency is carried out in which the words with low frequency were removed. Luhn proposed that the importance of a particular word is based on the frequency of the word in that article. The output of Luhn's work was called abstract or 'auto abstract' but the correct term is extract because sentences from the source were included in the summary.

Advantages:

- 1) The auto-abstracts have a high degree of reliability, consistency and stability.
- 2) The auto-abstract is the first example of a machine generated equivalent of a completely intellectual task in the field of literature evaluation.

Disadvantages:

- 1) It lacked sophistication.
- 2) In some cases due to the author's style of writing sentences of lower significance were selected.

2.2.2 H. P. EDMUNDSON

This system can only be used for single documents and it produces document extracts. The system was introduced in the year 1969 and it was domain specific to articles. The previous systems were based on two features which were word frequency and positional importance. This system was based on two additional features: Cue words i.e. the presence of pragmatic words such as 'significant', 'impossible' or 'hardly' and giving these words cue weights and Skeleton of the document i.e. to identify if the sentence is a title or a header. The system uses a research methodology which includes procedures for compiling dictionaries, setting control parameters and comparing the automatic extract to a manual extract.

Advantages:

- 1) With the use of two additional features the system has become more accurate and provides a reliable extract.
- 2) The new research methodology used has helped in summarizing documents and will be fruitful to the future systems.

Disadvantages:

- 1) The system is not suitable if the source input is a long document.
- 2) The features and methodology used is costly and consumes a lot of operational time.
- 3) The major disadvantage is in the method of inputting text since the system is unable to automatically capture mathematical and chemical symbols in machine form.

2.2.3 BARZILAY AND ELAHADAD

This system was introduced in the year 1997 to produce a summary from a single document. It uses a new algorithm which is based on lexical chains. Lexical chains are sequences of related terms grouped together by text cohesion relationships (e.g. synonymy or homonymy). This system works in the following manner: the original text is segmented, lexical chains are constructed, strong chains are identified and then the significant sentences are extracted.

Advantages:

- 1) This system produces a notable improvement above a commercially available summarizer both in precision and in recall.
- 2) The results indicate the strong potential of lexical chains as a knowledge source for sentence extraction.

Disadvantages:

- 1) Extracted sentences contain anaphora links to the rest of the text.
- 2) This system does not provide any way to control the length and level of detail of the summary.
- 3) Results are significantly better for the 10%-length summaries than for the 20%.

2.2.4 SUMMARIST

This system creates a robust automated text summarization system based on the equation $\text{summarization} = \text{topic identification} + \text{interpretation} + \text{generation}$. This system is used for both a research tool and as an engine to produce summaries. It produces both extracts and abstracts for arbitrary English and other-language text.

Advantages:

- 1) It produces both extracts and abstracts.
- 2) This system is a multi-lingual system.
- 3) It uses many of the previous techniques such as lexical chains, position in the text, location, topic identification and cue phrases.

Disadvantages:

- 1) This system does not take into account the combination of semantic and statistical techniques which is one of the most complex tasks for all natural language processing.
- 2) The system would be very accurate and reliable with these techniques.

2.2.5 SUMMONS

SUMMONS stands for SUMMarizing Online NewS articles. This system is used for both single and multi-document summarization. It is domain specific to online news. This system summarizes full text inputs using templates using a message understanding system. This system can summarize a series of news articles on the same event. This system consists of two components a lexical chooser and the Functional Unification Formalism (FUF).

Advantages:

- 1) It was successful in demonstrating the feasibility of generating summaries of a series of news articles, highlighting the changes over time.
- 2) To provide more confidence to the user the system provided sources that agree on the facts. Combining information from various incomplete sources to form a complete report was also one of the functions of the system.

Disadvantages:

- 1) Statistical techniques were not used in this system; these techniques would have increased the robustness and vocabulary of the system.
- 2) This system was not suitable for multiple languages.
- 3) This system lacked the ability to assess and change the interpretation component of the system.

2.2.6 CENTRIFUSER

This system was developed in the year 2001. It takes inputs from multiple documents and gives the output in the form of extracts. The input documents are health-care articles and hence it is domain specific to only health-care. This system produces query driven summaries. This system uses a specific tool for clustering which is the SIMFINDER tool. This system is based on document topic tree which means each document will be represented by a tree data structure. The system provides three types of outputs that are designed to help the user understand the documents.

Advantages:

- 1) By collecting and analyzing both video and audio data on user's interactions with the system, this system is able to characterize those aspects of WWW interfaces that are useful to patients and their families who are seeking health information in response to questions.
- 2) This system is used for coding in the categories of user comments which helps in finding the areas of improvement for the system.

Disadvantages:

- 1) The system is not as efficient when compared to the other search engines.
- 2) In this system it is a bit complicated to move to the next step as compared to other systems.

2.2.7 MEAD

This system was developed at the University of Michigan in 2001. It can produce both single and multi-document extractive summaries. The system is based on the idea of centroid-based feature and other features such as position and overlap with the first sentence. It uses CIDR Topic Detection and Tracking system to identify all the articles related to an emerging event. This system was only domain specific to news articles.

Advantages:

- 1) This system had the features such as to find the length of the sentence in words, to know if the length of a sentence is above or below certain threshold, to find the position of the sentence in the document.

2.2.8 CATS

This system was developed in the year 2005. It uses single as well as multi-document input. This system is domain specific only to news articles. This system analyzes which information in the document is important in order to include it in the summary. In this system statistical techniques are used to compute a score for each sentence as well as temporal expressions and redundancy is solved.

Advantages:

- 1) In this system, statistical techniques are used which were not used in the previous systems.
- 2) This system is also known as the Answering Text Summarizer.

2.3 EXISTING SYSTEMS RELATED TO PRESENTATION MAKERS**2.3.1 MICROSOFT OFFICE POWERPOINT**

PowerPoint is a part of the Microsoft Office Suite of productivity programs. It allows people to create a series of single page slides that contain information to be presented.

Advantages:

- 1) It allows the presenter to include graphs, images and movies in the presentation making the information both understandable and memorable.
- 2) The slides also include speaker's notes to remind the speaker of the important points that may not have fit into the slide.

Disadvantages:

- 1) While a simple graph can make a small table of numbers more clear, a complex set of graphs made from a complex table can make everything much more difficult to understand.

2.3.2 PREZI

Prezi is a cloud-based (SaaS) presentation software and storytelling tool for presenting ideas on a virtual canvas. The product employs a zooming user interface (ZUI), which allows users to zoom in and out of their presentation media, and allows users to display and navigate through information within a 2.5D or parallax 3D space on the Z-axis[4].

Advantages:

- 1) Prezi is supported by multiple devices and is easy to learn and use.
- 2) It is cost effective and one can also work on the presentation offline.

Disadvantages:

- 1) It is template driven and cannot necessarily customize slides.

CHAPTER3

PROPOSED

SYSTEM

3.1 INTRODUCTION

This system describes the problem statement, scope, features and objectives of the proposed system. It also specifies the minimum hardware and software requirements of the system long with the timeline charts.

3.2 PROBLEM STATEMENT

A presentation plays an important role when it comes to represent an idea or to explain the topic to the audience. However, the current system of creating a presentation technique is a complex issue. Knowing the importance of creating a presentation, it has become important to modify it and make the system automated. Till the present situation, creating a presentation for a presenter is not an easy task. It is time consuming, tedious and effort making. The presenter has to go through multiple documents, create a small summary in which all the highlighted points are covered and then create the presentation. This problem can be solved by developing a software tool that automates the method of creating a presentation. Thus, a new system is proposed in which the software accepts multiple documents as input in order to provide a summary and from that summary a concise and to-the-point presentation is created (that conveys the content in the source documents in a presentable manner). From the multiple documents, various clusters of data or sentences are created and the feature profiles are generated.

3.3 SCOPE

The application is developed on the basis of two types of requirements of the people that is summarization of text and creation of presentations. Summarization of the text deals with knowing the basic idea of the entire topic. It is like giving brief information over the entire subject. On the other hand, creating a presentation involves presenting information in an effective manner. In a presentation, only the important points are highlighted which help in the understanding of the topic. Our system deals with both these concepts thus providing an easy way for gathering and delivering information.

3.2.1 OBJECTIVES

1. To automate the process of creating presentations

2. Increased accuracy in summarization of text
3. Multipurpose features of text summarization and presentation creation
4. To ease the work of people who deal with presentations and summaries on a day to day basis.

3.2.2 FEATURES AND ADVANTAGES

1. Easy to use
2. Makes use of internet as per user's requirement
3. Maintains a separate database to store important details of the presentation or summarized text within the system.
4. Cost effective and can be easily downloaded from various site.

3.4 SYSTEM REQUIREMENTS

3.3.1 MINIMUM SOFTWARE REQUIREMENTS

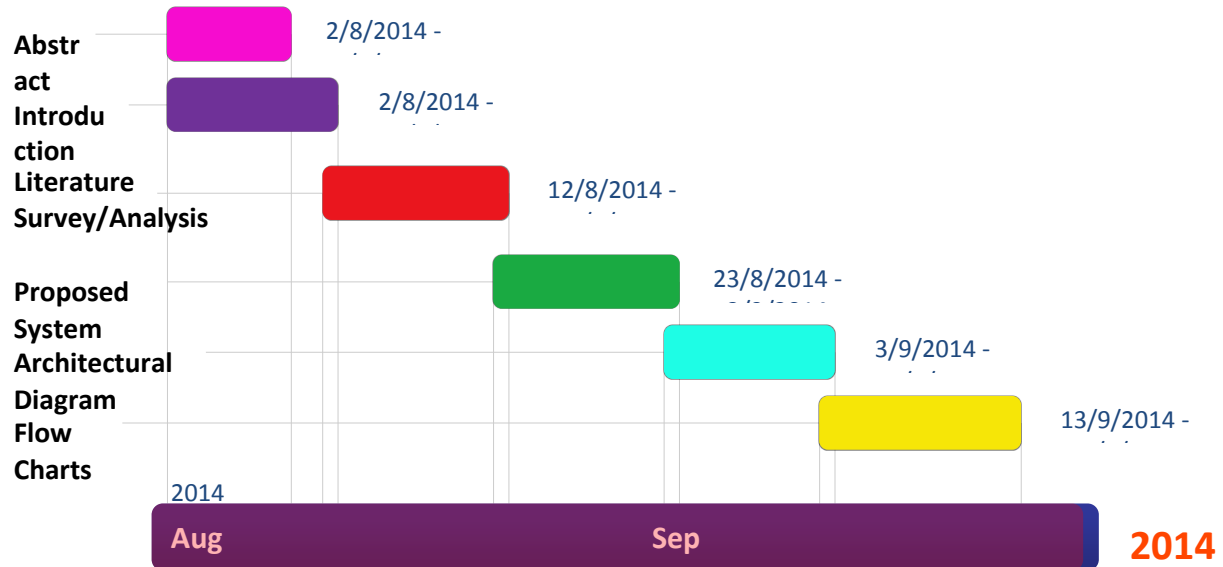
1. JDK 1.6 or higher for Java Support
2. Apache Tomcat for JSP support

3.3.2 MINIMUM HARWARE REQUIREMENTS

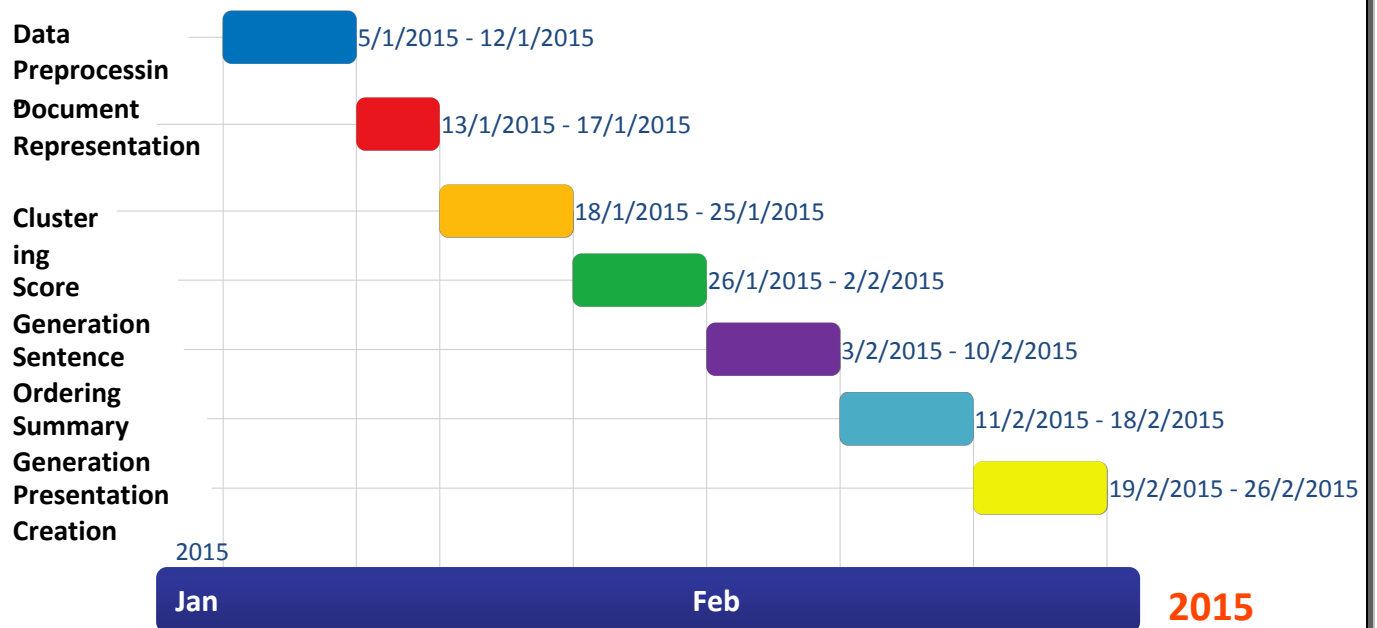
1. Intel CORE2DUO processor
2. RAM 2GB
3. Hard disk storage 1 GB

3.5 TIMELINE CHARTS

3.5.1 SEMESTER VII



3.5.2 SEMESTER VIII



CHAPTER 4

DESIGN

4.1 INTRODUCTION

This chapter explains in brief the architectural design of the proposed system. The internal working of each step in the architecture is also explained.

4.2 ARCHITECTURAL DESIGN

Initially, the user has to provide the system with input documents. The documents can be of file format doc, docx or txt. These files have to be first converted to text files. For this purpose, Java provides various facilities for reading, writing and creating files.

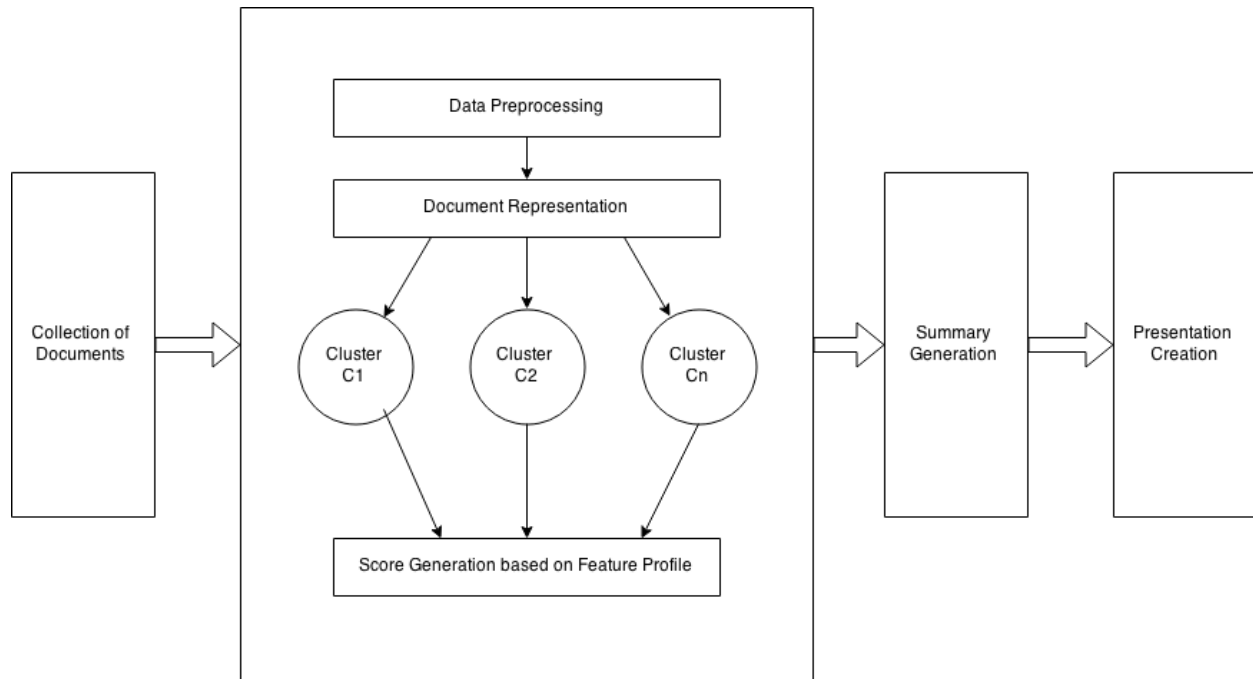


Figure 4.1: The Proposed system

4.2.1 DATA PREPROCESSING

A preprocessing procedure consists of various steps. We include the following steps for our system:

1. Lexical analysis:

The aim of this step is to identify words in the documents. Digits, hyphens, punctuation marks and case of the letters are the primary concern of this step.

2. Elimination of stop words:

A stop word can be a word without meaning in a specific language or it can be token that does not have any linguistic meaning. Examples of stop words in English are ‘a’, ‘the’, ‘is’ etc. [5]

3. Stemming:

A stem is the portion of the word which is left after removal of its affixes. Stems are thought to be useful for improving searching of terms because they reduce variants of the same root word to common concept. [5]

4.2.2 DOCUMENT REPRESENTATION

After preprocessing, we obtain a collection of words from the set of documents. This collection of words is known as vocabulary. It is represented as a two dimensional matrix wherein rows represent documents and columns represent words. The concept of TF-IDF (Term Frequency-Inverse Document Frequency) is used to fill the matrix to reflect the count of a particular word in a specific document. Each entry in this matrix is an integer count.

4.2.3 CLUSTERING OF DATA

After the process of document representation, the sentences are spilt into a matrix representation. In clustering, data is grouped on the basis of Cosine similarity. In this method, the clusters are created on the basis of dissimilarities or the distances between the data. This is the most straightforward way of computing distances between objects in a multi-dimensional space.

4.2.4 SCORE GENERATION BASED ON FEATURE PROFILE

In this step, the documents in a particular cluster are split into sentences. They are scored on the basis of the features of title, sentence length term weight, sentence position, presence of proper noun and numerical data. Higher the score of a sentence more is the importance of the sentence.

4.2.5 SUMMARY GENERATION

On the basis of clustering and feature profile system, score is generated which leads to ordering of the sentences. In this step, a summary is generated by extracting the highly ranked sentences.

4.2.6 PRESENTATION CREATION

In this stage, slides are generated based on the summary that is generated. Sentences are extracted from the summary and inserted into the slides. This is done by using Apache POI. Each slide is titled as 'Summary'.

CHAPTER 5

IMPLEMENTATION

5.1 INTRODUCTION

This chapter describes the implementation of the proposed system in terms of the block diagram, algorithms and flow charts.

5.2 PSEUDOCODE AND ALGORITHMS

5.2.1 STEMMING

INPUT: filerules String, documents in the format doc, docx and txt

OUTPUT: Processed document

1. Constructor

```
ruletable := filerules into Vector
Set preStrip to false
Set filerules to rules
If pre.equals with "/"p"
    Set preStrip to true
EndIf
Call method ReadRules with filerules
```

2. ReadRules

```
Initialise ruleCount to 0
Initialise j to 0
    Initialise line to " "
        While ( line )
            RuleCount++;
            Set j to 0
            Create new String
            Set rule to ""
            For (; j < line.length() && line.charAt(j) != ' '; j++)
```

```

        Set rule
    EndFor
    Call method ruleTable.addElement with rule
EndWhile

```

```

ch='a'

```

```

For j is 0, j is less than 25, j increments by 1

```

```

EndFor

```

```

For j is 0, j is less than ruleCount minus 1, j increments by 1

```

```

    While ( ( String ) )

```

```

        Ch++;

```

```

        Set charCode with ch of ruleIndex to j

```

```

    EndWhile

```

```

EndFor

```

3. FirstVowel

```

    If (i < last && !vowel(word.charAt(i), 'a')) {

```

```

        i++;

```

```

    EndIf

```

```

    If i is not equal to 0

```

```

        For (; i < last && !vowel(word.charAt(i), word.charAt(i - 1)); i++);

```

```

    EndFor

```

```

    EndIf

```

```

    If i < last

```

```

        Return i

```

```

    Else

```

```

        Return last

```

```

    EndIf

```

4. stripSuffixes

For pll is 0, pll +1 is less than stem.length and stem.charAt with pll plus 1 is greater than or equal to 'a' and stem.charAt with pll plus 1 is less than or equal to 'z', pll increments by 1

 If pll

 Set Continue to -1

 EndIf

EndFor

Initialise pfv to FirstVowel with stem, pll

Initialise iw to stem.length minus 1

While Continue is not equal to -1

 Set Continue to 0

 Initialise ll to stem.charAt with pll

 Int prt;

 If ll is greater than or equal to 'a' and ll is less than or equal to 'z'

 Set prt to position in ruleIndex

 Else Set prt to -1

 If prt is equal to -1

 Set Continue to -1

 If Continue is equal to 0

 Set rule to ruleTable.elementAt with prt as String

 While Continue is equal to 0

 Set ruleok to 0

 If rule.charAt with 0 is not equal to ll

 Set Continue to -1

 Set ruleok to -1

 EndIf

 Initialise ir to 1

Set iw to pll minus 1

 While ruleok is equal to 0

 If rule.charAt with ir is greater than or equal to '0' and


```

rule.charAt with ir is less than or equal to '9'
    Set ruleok to 1
Else if rule.charAt with ir is equal to '*'
    If intact
        Ir++;
        Set ruleok to 1
    Else
        Set ruleok to -1
    EndIf
Else if rule.charAt with ir is not equal to stem.charAt with
iw
    Set ruleok to -1
Else if iw is less than or equal to pfv
    Set ruleok to -1
Else
    Ir++;
    Iw--;
EndIf
EndWhile
If ruleok is equal to 1
    Int xl;
    For xl is 0, rule.charAt ( ir plus xl plus 1 ) is less than '.' or
rule.charAt with ir plus xl plus 1 is greater than '>', xl increments by 1
        Set xl to -rule.charAt with ir as pll
    EndFor
    If pfv is equal to 0
        If xl
            Set ruleok to -1
        EndIf
    Else if ( xl )
        Set ruleok to -1

```

```

        EndIf
    EndIf
    If ruleok is equal to 1
        Set intact to false
        Set pll to -rule.charAt with ir as pll
        Ir++;
        Set stem to stem.substring with 0, pll plus 1
        While ir
            Set stem
            Ir++;
            Pll++;
            System.out.println("keyword
suffix:"+stem.toString());
        EndWhile
        If rule.charAt with ir is equal to '.'
            Set Continue to -1
        Else
            Set Continue to 1
        EndIf
    Else
        Prt++;
        Set rule to ruleTable.elementAt with prt as String
        If rule.charAt with 0 is not equal to ll
            Set Continue to -1
        EndIf
    EndIf
EndWhile
EndIf
EndWhile

```

5. stripPrefixes

Initialise last to prefixes.length

For i is 0, i is less than last, i increments by 1

If str.startsWith with position i in prefixes and str.length is greater than position i
in prefixes

Set str to str.substring with position i in prefixes

System.out.println(" keyword:" + str);

EndIf

EndFor

System.out.println("keyword':" + str);

6. stripAffixes

if str.length() > 3 and preStrip

str = stripPrefixes(str);

if (str.length() > 3) {

str = stripSuffixes(str);

return str

5.2.2 COSINE SIMILARITY

INPUT: Output of stemming

OUTPUT: cosineValue Double

1. CosineSimilarity

Initialise allUniqueTokenSet by calling getUniqueTerms with allTokenSet

commonTerms = (termsinStr1 + termsinStr2) - allUniqueTokenSet.length

where termsinStr1= str1UniqueTokenSet.length

termsinStr2=str2UniqueTokenSet.length

double cosineValue = (commonTerms / (Math.sqrt(termsinStr1) * Math.sqrt(termsinStr2))
* 100)

2. **getUniqueTerms**

```
Create new ArrayList
For i is 0, i is less than strTokenSet.length, i increments by 1
  Initialise isDistinct to false
  For j is 0, j is less than i, j increments by 1
    If position i in strTokenSet is equal to ( position j in strTokenSet )
      Set isDistinct to true
      System.out.println(" common values=" + strTokenSet);
    EndIf
  EndFor
  If not isDistinct
    System.out.println(" strTokenSet[i]: " + strTokenSet[i]);
  EndIf
EndFor
Create new char
Copying the contents of the 2d array to a new 1d array
  Set i of allchar to tempAL.get with i
EndFor
Return allchar
```

5.2.3 **CLUSTERING**

INPUT: Output of CosineSimilarity Algorithm

OUTPUT: Cluster of documents

```
Do while docAL.size is greater than 0
  Initialise threshold to 0
  For j is 1, j is less than docAL.size, j increments by 1
    Call method cosines.add with cosineValue
  EndFor
  Initialise clone to cosines.clone as ArrayList
```

```

Call method Collections.sort with clone
MaxcosineValue := max
mincosineValue := min
Threshold /= docAL.size() - 1;
For j is 1, j is less than docAL.size, j increments by 1
    Initialise vectorData1 to docAL.get with j as String
    Initialise compareDoc to docHashMap.get with vectorData1 as String
    Initialise cosineValue to cosines.get with j minus 1 as Double
        If not clusterAL.contains with vectorData1
            Call method clusterAL.add with vectorData1
        EndIf
        productAL.remove(productAL.get(j));
    Else
        Call method pendingAL.add with vectorData1
    EndIf
EndFor
Call method finalclusterAL.add with clusterAL
EndDo
Return finalclusterAL

```

5.2.4 FEATURE SPECIFIC ALGORITHM

INPUT: Output of clustering algorithm

OUTPUT: Sentence weight

1. Constructor

Initialise sentence_position_Threshold to 0

Initialise numof_sentence to 0

Final double threshold = 0.1;

Call method Hm_paramweight.put with "tf", 4

Call method Hm_paramweight.put with "pnf", 3

Call method Hm_paramweight.put with "ndf", 2

Call method `Hm_paramweight.put` with "spf", 1
Set `max_weight_keyword` to `Hm_paramweight.get` with "tf" multiplied by
`Hm_paramweight.get` with "pnf" multiplied by `Hm_paramweight.get` with "ndf" multiplied
by `Hm_paramweight.get` with "spf"
Set `sentence_position_Threshold` to `Math.round` with `numSentences` multiplied by
`threshold`
If `sentence_position_Threshold < 1`
 Set `sentence_position_Threshold` to 1

1. **Procedure: getAllfeaturesweight**

```
public static HashMap<String, Integer> getAllfeaturesweight(int fid, HashMap<String,
Integer> paramweight) throws SQLException, ClassNotFoundException
Create new HashMap
Int tf1, spf1, pnf1, ndf1, key_weight = 0;
Int tfval, spfval, pnfval, ndfval;
String tf, spf = null, pnf = null, ndf = null;
Set tfval to paramweight.get with "tf"
Set spfval to paramweight.get with "spf"
Set pnfval to paramweight.get with "pnf"
Set ndfval to paramweight.get with "ndf"
Initialise sql to " select keywords
,title_feature,sentence_position,proper_noun,numericalword from keyword_data where
fileid="" plus fid plus ""
Initialise rs to stmt.executeQuery with sql
While rs.next
    Initialise keyword to rs.getString with "keywords"
    Set tf to rs.getString with "title_feature"
    Set spf to rs.getString with "sentence_position"
    Set pnf to rs.getString with "proper_noun"
    Set ndf to rs.getString with "numericalword"
    If tf.equals with "true"
```

```

        Set tf1 to tfval * 1
    Else
        Set tf1 to tfval * 0
    EndIf
    If spf.equals with "true"
        Set spf1 = spfval * 1
    Else
        Set spf1 to spfval * 0
    EndIf
    If pnf.equals with "true"
        Set pnf1 to pnfval * 1
    Else
        Set pnf1 to pnfval * 0
    EndIf
    If ndf.equals with "true"
        Set ndf1 to ndfval * 1
    Else
        Set ndf1 to ndfval * 0
    EndIf
    Set key_weight to tf1 plus spf1 plus pnf1 plus ndf1
    Call method keyword_weight.put with keyword, key_weight
EndWhile
Return keyword_weight

```

2. **Procedure: Isnumeric**

```

public static boolean Isnumeric(String str)
For i is 0, i is less than str.length, i increments by 1
    Set c to str.charAt with i
    If Character.isDigit with c
        Set isnumeric to true
        break;

```

EndFor
Return isnumeric

3. Procedure: sentence_position

```
public static ArrayList<Integer> sentence_position(int extraction_threshold, int  
numsentence)  
Create new ArrayList  
For i is 0, i is less than extraction_threshold, i increments by 1  
    Call method sentences_high.add with i plus 1  
    Call method sentences_high.add with numsentence minus i  
EndFor  
Return sentences_high
```

5.3 FLOW CHARTS

Considering a document provided by the user as input, we first split the documents into words. These words are checked for stop words which are removed. Thereafter, stemming is performed so as to remove variants of the same root word. For example, ‘detection’ and ‘detecting’ are variants of the word ‘detect’. Figure 5.1 represents the process of preprocessing. The output of this step is a preprocessed document.

The preprocessed data is subjected to clustering and we obtain clusters of related document. This process is depicted in the figure 5.2.

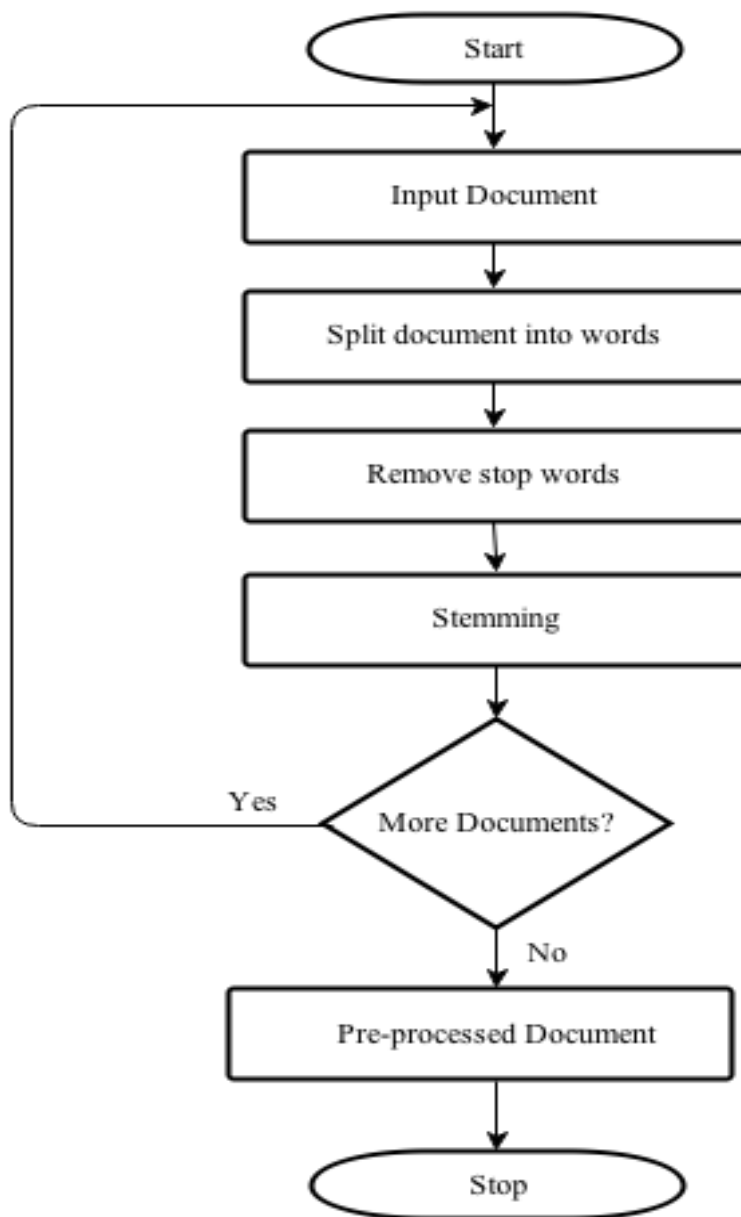


Figure 5.1: Preprocessing

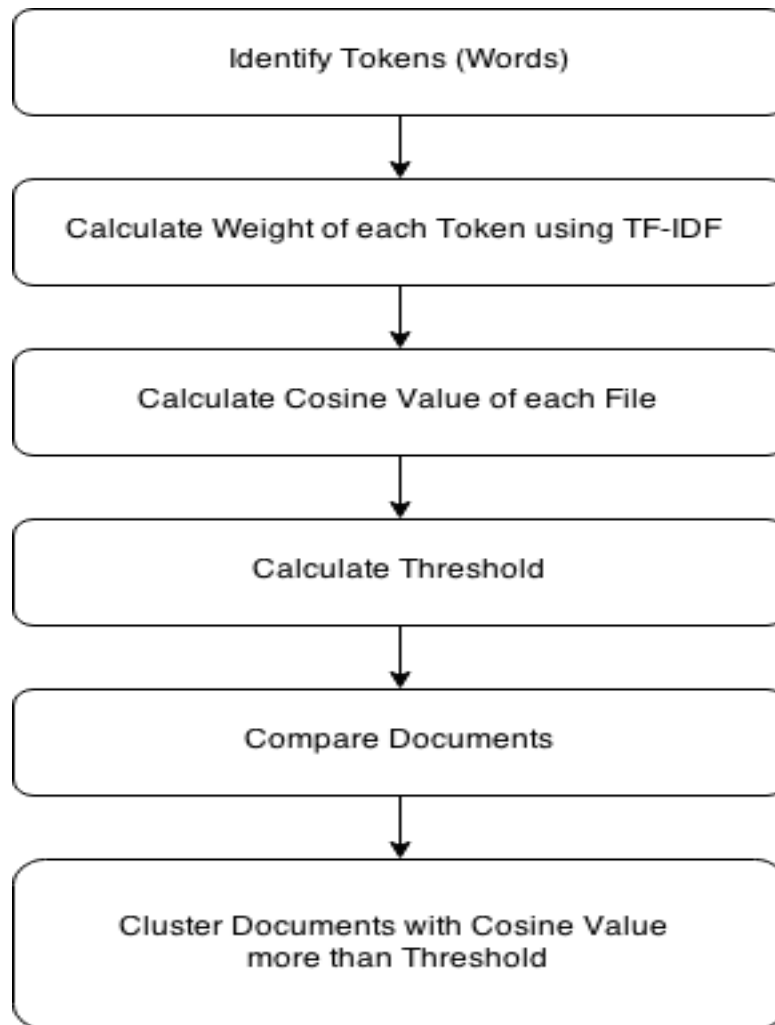


Figure 5.2: Clustering

After clustering, we prepare the feature profile sentences. The score is incremented based on the features of title, sentence length, term weight, sentence position, numerical data and proper nouns. This process is shown in figure 5.4. We create a database of sentences with decreasing order or their scores. After sentence ordering we then perform summary generation which is depicted in the figure 5.5. After generation of summary we create presentation by the process depicted in figure 5.6.

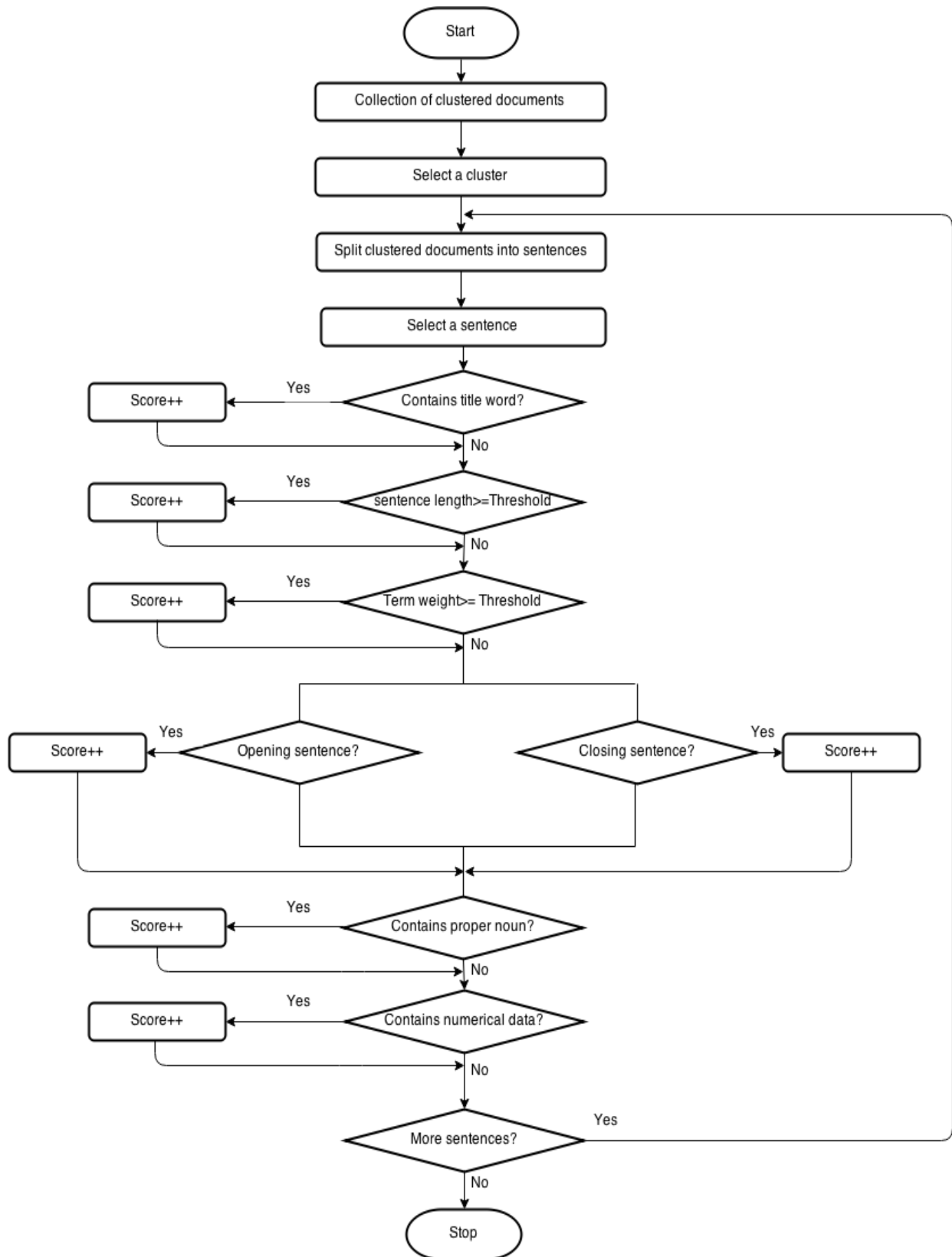


Figure 5.3: Score generation based on feature profile

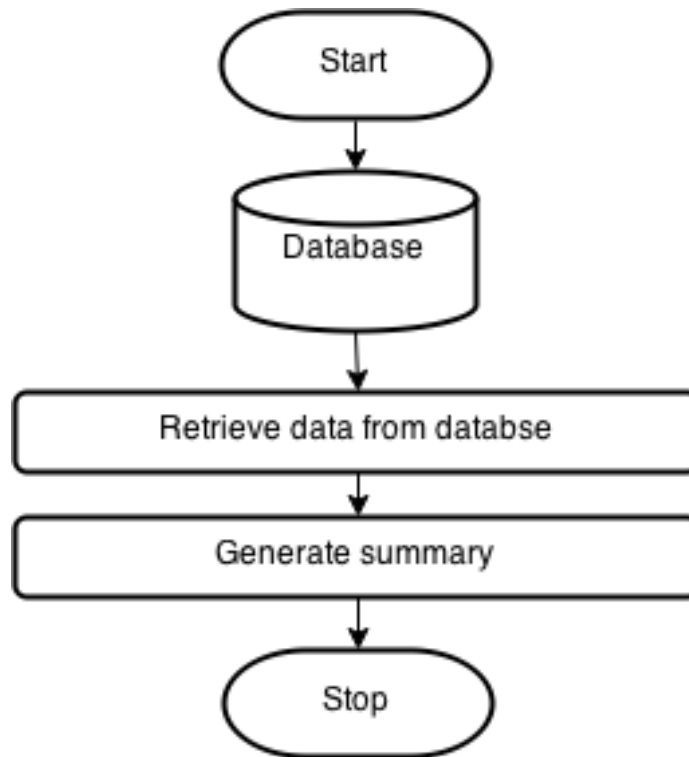


Figure 5.4: Summary Generation

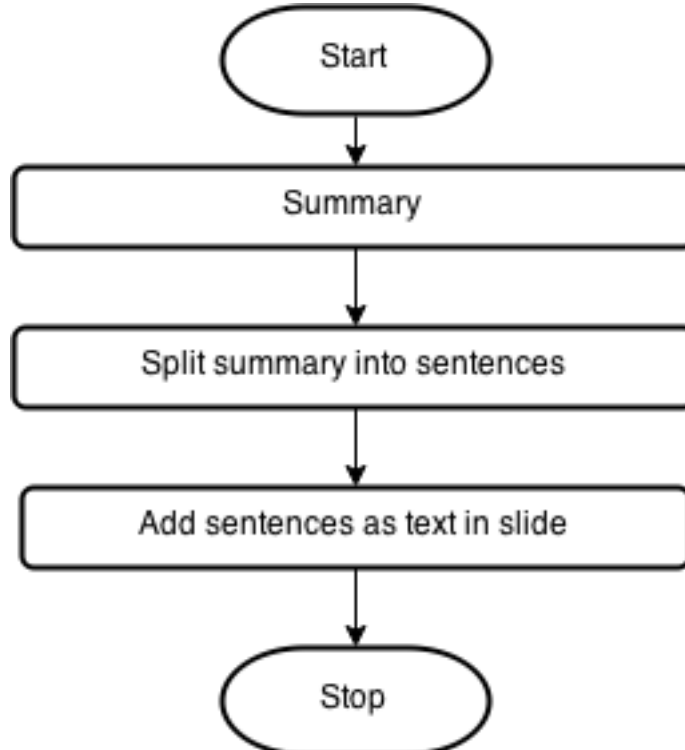


Figure 5.5: Presentation Creation

5.4 TEST CASES

Test case ID	Test case Execution Steps	Expected result	Tester	Test result
TC1	Open the software running in java platform	First window for login is displayed	Keven	Pass
TC2	Enter the login Id and password with is case sensitive.	New window is obtained	Keven	Pass
TC3	Select the multiple documents from the system conferencing	Intake four types of documents	Varun	Pass
TC4	Limit the number of multiple documents	Various types of documents are taken	Varun	Pass
TC5	Create the clusters	Various clusters are created	Vrishali	Pass
TC6	Select the cluster which the user want.	User selects the required cluster	Vrishali	Pass
TC7	Display the content in the selected cluster	Displays the selected documents in the given cluster	Pravin	Pass
TC8	Click to create the summary	Summary is generated	Pravin	Pass
TC9	View the summary	The summary is successfully displayed in the system	Vrishali	Pass

TC10	Save the summary	The summary is saved in the required location provided by the user	Vrishali	Pass
TC11	Creation of presentation	Presentation is created automatically in the root directory	Varun	Pass
TC12	Save the presentation	Presentation is saved in the required location by the user.	Varun	Pass

CHAPTER 6

CONCLUSIONS

AND FUTURE

SCOPE

6.1. CONCLUSIONS

Due to the evolution of Internet, abundant information is available at our expense. However, this leads to the problem of information overload. The project proposed and developed is an attempt to solve this problem by automating the process of text summarization as well as text presentation creation. In this system, multiple documents are taken into the account and from that multiple documents a single summary is generated. It consists of two phases; in the first half there are three major steps that are data pre-processing, clustering and feature specific. Using these steps, summary is generated automatically. In the second half, the system is dealing with presentation creation. In order to place the summary in a presentation format apache poi which is a jar file is used. Thus, a presentation is created automatically. The systems developed provide a way to automate the process of text summarization and then the presentation. Thus, the proposed system eases the work of professionals, professors, and students etc. who deal with presentations on a regular basis.

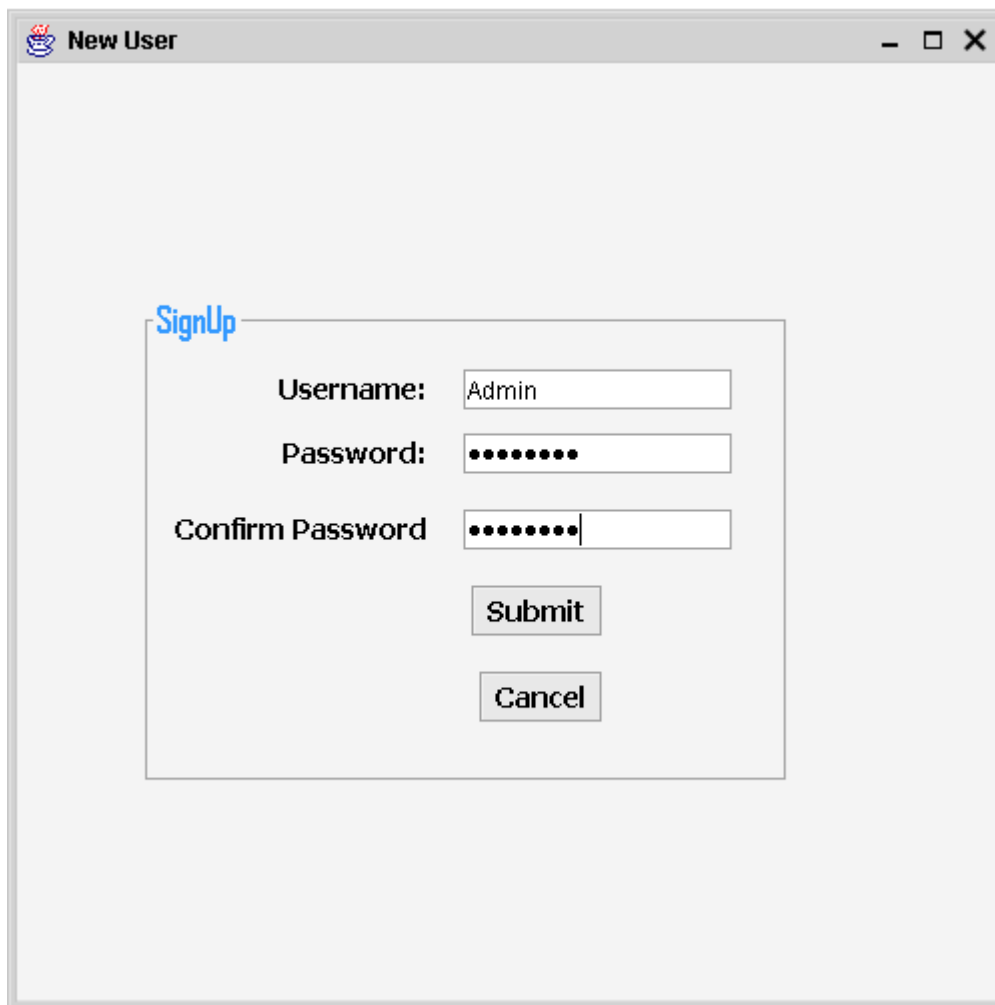
6.2. FUTURE SCOPE

1. Insertion of images, graphs etc. as the current system deals only with text documents.
2. Insertion of more types of text formats like excel format, pdf format etc.
3. Provision for formatting the presentation within the system in order to make it more interactive.

APPENDIX A: SNAPSHOTS

It is the Graphical User Interface of the system. It consists of login form, main form and a final presentation form. The login form is to validate the system to the correct user to provide security and the login form is case sensitive. The main form consists of two sections, that is, source and summary. In source the user has to insert the data and clusters will be formed. And after clustering, on the basis of score a summary is generated. Further in the last form, in the root directory, a text presentation is generated using that summary.

Login form:



The image shows a graphical user interface window titled "New User". Inside the window, there is a "SignUp" section. This section contains three input fields: "Username:" with the text "Admin", "Password:" with masked characters (dots), and "Confirm Password" with masked characters (dots). Below these fields are two buttons: "Submit" and "Cancel".

Figure7.1:Login Form

Main Form:

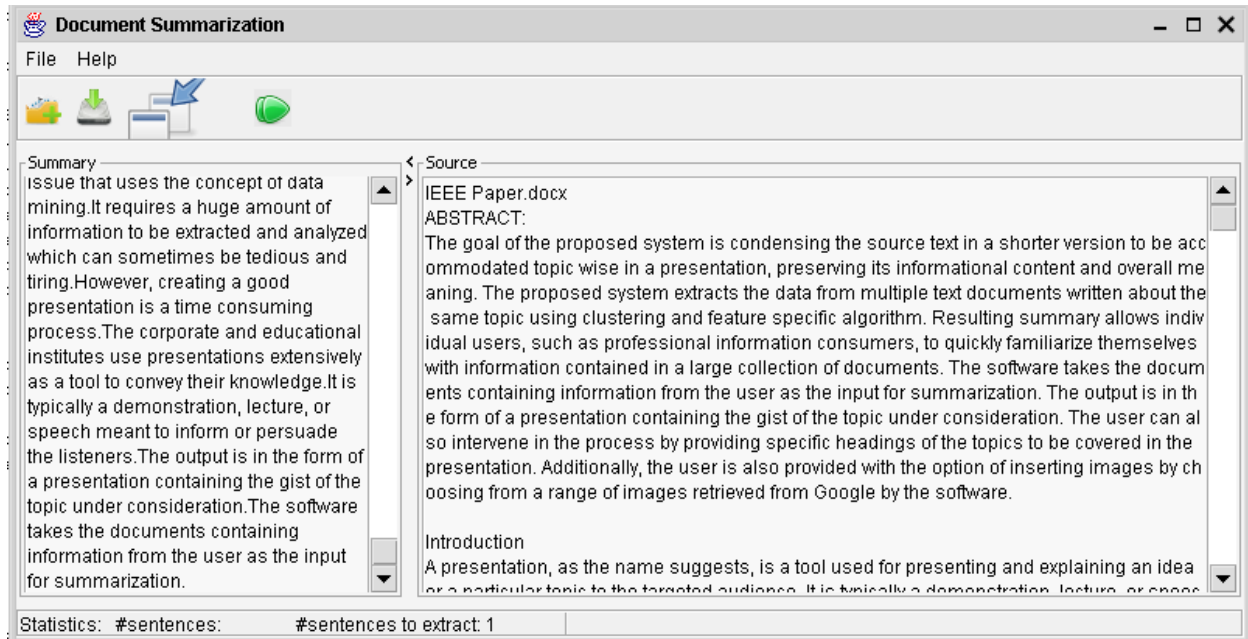


Figure 7.2: Main Form

Final Presentation:

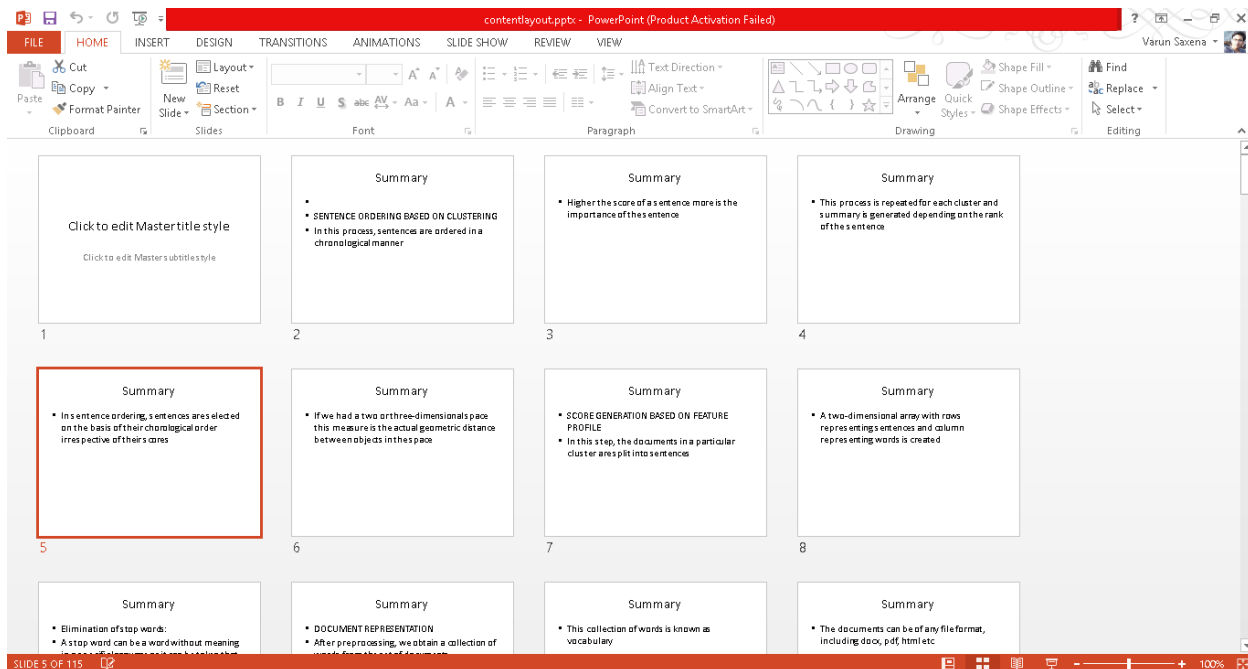


Figure 7.3: Final Presentation

APPENDIX B: DATASET

FILE 1: C++.doc

C is a general-purpose, high-level language that was originally developed by Dennis M. Ritchie to develop the UNIX operating system at Bell Labs. C was originally first implemented on the DEC PDP-11 computer in 1972. In 1978, Brian Kernighan and Dennis Ritchie produced the first publicly available description of C, now known as the K&R standard. The UNIX operating system, the C compiler, and essentially all UNIX applications programs have been written in C. The C has now become a widely used professional language for various reasons.

FILE 2: CInfo.doc

C was originally first implemented on the DEC PDP-11 computer in 1972. In 1978, Brian Kernighan and Dennis Ritchie produced the first publicly available description of C, now known as the K&R standard. C is a structure oriented programming language. The main disadvantage in C is that it does not support classes and objects. To overcome this, C extension is developed called as C++.

FILE 3: Stck.txt

A stack is an abstract data type which is also known as LIFO (last in, first out). Push and pop are the operations that stack perform. In push, element is added to the collection and pop is to remove the last element. In stack, the first element "popped off" a stack in series of pushes and pops is the last element that was pushed in the sequence. There are certain occasions in which the stack is full and does not contain enough space to accept an entity to be pushed, the stack will enter in overflow state. If the stack is empty and the element is popped then the stack will enter in underflow state.

FILE 4: StckInfo.docx

A stack or LIFO (last in, first out) is an abstract data type that serves as a collection of elements. The operations performed are push that is to add an element to the collection and pop is to remove the last element that was added. The term LIFO stems from the fact that, using these operations, the first element "popped off" a stack in series of pushes and pops is the last element that was pushed in the sequence. This is equivalent to the requirement that, considered as a linear data structure, or more abstractly a sequential collection, the push and pop operations occur only at one

end of the structure, referred to as the top of the stack. A stack may be implemented to have a bounded capacity. If the stack is full and does not contain enough space to accept an entity to be pushed, the stack is then considered to be in an overflow state. The pop operation removes an item from the top of the stack. A pop either reveals previously concealed items or results in an empty stack, but, if the stack is empty, it goes into underflow state, which means no items are present in stack to be removed. A stack is a restricted data structure, because only a small number of operations are performed on it. The nature of the pop and push operations also mean that stack elements have a natural order. Elements are removed from the stack in the reverse order to the order of their addition.

REFERENCES

- [1] Hovy, E. H. Automated Text Summarization. In R. Mitkov (ed), The Oxford Handbook of Computational Linguistics, chapter 32, pages 583-598. Oxford University Press, 2005.

- [2] A. Kogilavani and Dr. P. Balasubramani, International journal of computer science & information Technology (IJCSIT) Vol.2, No.4, August 2010 DOI : 10.5121/ijcsit.2010.2409 99 Clustering and Feature specific sentence extraction based summarization of multiple documents.

- [3] Ladda Suanmali et. al. Automatic Text Summarization Using Feature-Based Fuzzy Extraction

- [4] Lendeneva Yulia et. al. Effect of Preprocessing on Extractive Summarization with Maximal Frequent Sequences National Polytechnic Institute Center for Computing Research

ACKNOWLEDGEMENT

It gives us great pleasure in acknowledging the support and help of our principal **Dr. Rollin Fernandes** for giving us the time and resources for completion of our final year project.

We would like to thank **Prof. H. K. Kaura**, HOD of Computer Science and Information Technology Department for his support and guidance.

It is with immense gratitude that we acknowledge the support, advice and help of our supervisor, **Mrs. Poonam Bari**, whose expertise, understanding, and patience, added considerably to our project development skills. We are thankful for her aspiring guidance, invaluable constructive criticism and friendly advice during the project work.

We would also like to express our gratitude to **Mrs. Archana Shirke** and **Mrs. Dhanashree Hadsule**, Project coordinator for Information Technology Department, who supported us throughout the course of this project.

We share the credit of our work with the Faculties of I.T. Department for suggesting ideas pertaining to various areas of I.T., which greatly helped us to decide on the objectives of our project.

Yours sincerely,

VRISHALI BHOR

KEVEN SEBASTIAN

VARUN SAXENA

PRAVIN HODGE

PAPER PUBLISHED AND CERTIFICATES



Sinhgad Technical Education Society's

SINHGAD INSTITUTE OF MANAGEMENT & COMPUTER APPLICATION

MCA DEPARTMENT

Organizes



2nd National Conference

Innovations in IT and Management

27th and 28th February 2015

Sponsored by

AIMS

ASSOCIATION OF
INDIAN MANAGEMENT SCHOOLS

Certificate

This is to certify that Dr. / Prof. / Mr. / Ms. Varun Saxena
of F.C.R.I.T, Vashi, Navi Mumbai
has Attended / Participated & Presented Paper titled CREATION OF PRESENTATIONS WITH
FEATURE SPECIFIC MULTI-DOCUMENT TEXT SUMMARIZ in the Two days 2nd National Conference
on "*Innovations in IT and Management*" (NC²TM-15) held on 27th & 28th February 2015.

Dr. S. D. Mundhe
Director - SIMCA - MCA

Dr. Milind Marathe
Director - SIMCA - MBA



Sinhgad Technical Education Society's

SINHGAD INSTITUTE OF MANAGEMENT & COMPUTER APPLICATION

MCA DEPARTMENT

Organizes



2nd National Conference

Innovations in IT and Management

27th and 28th February 2015

Sponsored by

AIMS

ASSOCIATION OF
INDIAN MANAGEMENT SCHOOLS

Certificate

This is to certify that Dr. / Prof. / Mr. / Ms. Keven Sebastian

of F.C.R.I.T, Vashi, Navi Mumbai

has Attended /Participated & Presented Paper titled CREATION OF PRESENTATIONS WITH FEATURE SPECIFIC MULTI-DOCUMENT TEXT SUMMARIZATION in the Two days 2nd National Conference

on "*Innovations in IT and Management*" (NC²TM-15) held on 27th & 28th February 2015.

Dr. S. D. Mundhe
Director - SIMCA - MCA

Dr. Milind Marathe
Director - SIMCA - MBA



Sinhgad Technical Education Society's

SINHGAD INSTITUTE OF MANAGEMENT & COMPUTER APPLICATION

MCA DEPARTMENT

Organizes



2nd National Conference

Innovations in IT and Management

27th and 28th February 2015

Sponsored by

AIMS

ASSOCIATION OF
INDIAN MANAGEMENT SCHOOLS

Certificate

This is to certify that Dr. / Prof. / Mr. / Ms. Urishali Bhor
of F.C.R.I.T, Vashi, Navi Mumbai
has Attended / Participated & Presented Paper titled CREATION OF PRESENTATIONS WITH FEATURE
SPECIFIC MULTI-DOCUMENT TEXT SUMMARIZATION in the Two days 2nd National Conference
on "Innovations in IT and Management" (NCITM-15) held on 27th & 28th February 2015.

Dr. S. D. Mundhe
Director - SIMCA - MCA

Dr. Milind Marathe
Director - SIMCA - MBA



Sinhgad Technical Education Society's

SINHGAD INSTITUTE OF MANAGEMENT & COMPUTER APPLICATION

MCA DEPARTMENT

Organizes



2nd National Conference

Innovations in IT and Management

27th and 28th February 2015

Sponsored by

AIMS

ASSOCIATION OF
INDIAN MANAGEMENT SCHOOLS

Certificate

This is to certify that Dr. / Prof. / Mr. / Ms. Pravin Hodge
of F.C.R.I.T., Vashi, Navi Mumbai
has Attended / Participated & Presented Paper titled CREATION OF PRESENTATIONS WITH FEATURE
SPECIFIC MULTI-DOCUMENT TEXT SUMMARIZATION in the Two days 2nd National Conference
on "*Innovations in IT and Management*" (NC²TM-15) held on 27th & 28th February 2015.

Dr. S. D. Mundhe
Director - SIMCA - MCA

Dr. Milind Marathe
Director - SIMCA - MBA