A PROJECT REPORT ON

Marathi Document Summarizer using Graph Based Model

SUBMITTED TOWARDS THE
PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

BACHELOR OF ENGINEERING (Computer Engineering)

BY

Student Name : Sanket Sutar	Exam No:
Student Name : Prathamesh Chaudhari	Exam No:
Student Name : Piyush Patil	Exam No:
Student Name : Rupesh Deshmukh	Exam No:

Under The Guidance of

Prof. Priti Vaidya

Department of Computer Engineering
K. K. Wagh Institute of Engineering Education & Research
Hirabai Haridas Vidyanagari, Amrutdham, Panchavati,
Nashik-422003
Savitribai Phule Pune University

A. Y. 2020-21 Sem I

K. K. Wagh Institute of Engineering Education and Research
Department of Computer Engineering

CERTIFICATE

This is to certify that the Project Titled

Marathi Document Summarizer using Graph Based Model

Submitted by

Student Name : Sanket Sutar Exam No:

Student Name : Prathamesh Chaudhari Exam No:

Student Name : Piyush Patil Exam No:

Student Name : Rupesh Deshmukh Exam No:

is a bonafide work carried out by Students under the supervision of Prof. Guide Name and it is submitted towards the partial fulfillment of the requirement of Bachelor of Engineering (Computer Engineering) Project during academic year 2020-21.

Prof. Priti Vaidya
Internal Guide
Department of Computer Engineering

Prof. Dr. S. S. Sane
Head
Department of Computer Engineering

Abstract

Manual summarization of large documents of texts is tedious and error prone. Also, the results in such kind of summarization may lead to different results for a particular document. Thus, Automatic text summarization has become important due to the tremendous growth of information and data. It chooses the most informative part of text and forms summaries that reveal the main purpose of the given document. It yields summary produced by summarization system which allows readers to comprehend the content of document instead for reading each and every individual document. So, the overall intention of Text Summarizer is to provide the meaning of text in less words and sentences. Summarization can be categorized as : Abstractive summarization and Extractive summarization. This case study is based on an extractive concept implemented on the studied models. Numerous automatic text summarization systems are handy today for English and other foreign languages.

But when it comes to Indian languages, we observe inadequate number of automatic summarizers. Our efforts in this direction are mainly for developing automatic text summarizer for Marathi Language. We look forward to evaluate the obtained summaries using ROUGE metric.

Acknowledgments

First and foremost, we would like to thank our project guide, **Prof. Priti Vaidya**, for her guidance and support. We will forever remain grateful for the constant support and guidance extended by our guide, in making this project stage-1 report. Through our many discussions, she helped us to form and solidify ideas.

With a deep sense of gratitude, we wish to express our sincere thanks to, **Prof. Dr. S. S. Sane** for his immense help in planning and executing the works in time. My grateful thanks to the departmental staff members for their support.

We would also like to thank our wonderful colleagues and friends for listening our ideas, asking questions and providing feedback and suggestions for improving our ideas.

Sanket Sutar
Prathamesh Chaudhari
Piyush Patil
Rupesh Deshmukh
(B.E. Computer Engg.)

INDEX

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 PROJECT IDEA

- Text Summarization is a technique of condensing actual text into abstract form which provides same meaning and information as provided by actual text. It chooses the most informative part of text and forms summaries that reveal the main purpose of the given document. It yields summary produced by summarization system which allows readers to comprehend the content of document instead for reading each and every individual document.
- So, the overall intention of text summarizer is to provide the meaning of text in less words and sentences. Summarization systems can be sorted into two categories: Abstraction-based summarization and Extraction-based summarization.
- Extractive summaries involve extracting appropriate sentences from the source text in sequential manner. The appropriate sentences are extracted by applying statistical and language reliable features to the input text. But there is limit in extraction. The extracted phrases and sentences are in chronological order. While, abstractive text summaries are formed by enacting natural language understanding concepts.
- This kind of summarizer generally, incorporates terms that do not exist in the document. It aims to imitate methods used by humans, such as representing a concept that is available in the original article in a better and more comprehensive way. It is effective summarizer however, it is very difficult to implement.

1.2 MOTIVATION OF THE PROJECT

- Various automatic text summarization systems are accessible for most often used languages. Most of these text summarization systems are for English and other foreign languages. Moreover, technical documentation is often minimal or even absent.
- When it comes to Indian languages, automatic summarization systems are very limited. Very little research and work has been done in text summarization for

the Indian language Marathi (an Under-Resourced language).

1.3 LITERATURE SURVEY

- Various fields make use of text summarization systems like, education field, social media (news articles, twitter, facebook messages), search engines, bio-medical field, government offices, researcher, etc [2].

- Virat V. Giri and et al. reviewed text summarizers based on various Indian languages and their performances. They studied and proposed summarization method for Marathi in detail wherein Marathi stemmer, Marathi proper name list, EnglishMarathi noun list, Marathi keywords extraction, Marathi rule based named entity recognition etc. for pre-processing of text followed by processing of text [1].

- Sheetal Shimpikar and et al. studied various techniques of text summarization for various Indian languages [2]. Sunitha C and et al. worked on Abstractive summarization methods that are used for Indian languages. They explained Abstractive summarization technique, classified in two approaches such as structure based approach and semantic based approach [3]. Hamzah Noori Fejer and et al. gave a major contribution by proposing a combined approach of clustering technique and extracting keyphrases.

  They have proposed a new approach of clustering which combines hierarchical and k-means clustering. The results obtained from their experiments proved the proposed model gives better performance when compared with existing ones [4]. An unsupervised approach for Marathi stemmer has been discussed by Mudassar Majgaonker and et al. [6].

- The present work on text summarization of Marathi text with question based system using rule based stemmer technique or generating question, we used rule based approach of abstractive text summarization and POS tagger, NER tools and rule based stemmer. Here Marathi text is taken as input, on it POS tagger is applied and then questions are generated for the given input as per Marathi language rules by Deepali K. Gaikwad and et al. At this stage they have framed rules of stemmer only for who ´ ´type questions [5]. Thus it can be extended to learning all What type questions too.

- Mangesh Dahale proposed text summarizer using inverted indexes [9]. Jayshri Patil and et al. reviewed different approaches of Named Entity Recognition (NER) and discussed issues and challenges arising in Indian languages [8]. Pooja Pandey and et al. discussed extraction of root words using morphological analyzer for devanagari script [11]. Aishwarya Sahani and et al. contributed to automatic text categorization of Marathi language documents [7].

  Rafael Ferriera et. Al used four dimensional graph based model for text summarization which relies on four dimensions(similarity,semantic similarity,co-reference,discourse information) to create the graph [16]. Federico Barrios et. al used variations in similarity measures along with TextRank for summarization [15]. Our work includes use of TextRank along with positional distribution of sentence scores and considering thematic similarity which gave promising results.

CHAPTER 2

PROBLEM DEFINITION AND SCOPE

2.1 PROBLEM STATEMENT

- To imitate methods used by humans, such as representing a concept that is available in the original article in a better and more comprehensive way. It is effective summarizer however, it is very difficult to implement.

2.1.1 Goals and objectives

- To find informative sentences.
- To measure similarities is also a crucial issue in sentence clustering based summarization approach.
- To provide the meaning of text in less words and sentences.

2.1.2 Statement of scope

- To summarize the Marathi document by retaining the appropriate sentences based on features.
- To generate score of sentences and high scored sentences in a specific order of input text are considered for final summary.

2.2 MAJOR CONSTRAINTS

- Very little work has been done for constructing a text summarizer for Marathi language. Marathi language is morphologically very rich.
- It is required to study its morphology and pre-process the document before extracting features and then process those features which are important in each of the sentences.

2.3 METHODOLOGIES OF PROBLEM SOLVING AND EFFICIENCY ISSUES

- Document Preprocessing.

- Feature Extraction.
- Sentence Scoring.
- Graph Scoring.
- Similarity
- Summarization

2.4 OUTCOME

- To abstractive summarization by including NLP features and implementing more scoring techniques. Use of semantic ranking can also be done to obtain meaningful summaries.

2.5 APPLICATIONS

- Editors
- Writers
- Media
- News Paper
- Social Media

2.6 SOFTWARE RESOURCES REQUIRED

Platform :

1. Operating System : Windows95/98/XP/VISTA/7/8/9/10.
2. IDE : Visual Studio
3. Programming Language : C# language

2.7 HARDWARE RESOURCES REQUIRED

Sr. No.	Parameter	Minimum Requirement
1	CPU Speed	2 GHz +
2	RAM	4 GB
3	Hard Disk	500 GB

Table 2.1: Hardware Requirements

CHAPTER 3

PROJECT PLAN

3.1 PROJECT ESTIMATES

3.1.1 Reconciled Estimates

The model followed is the Constructive Cost Model (COCOMO) for estimating the effort required in completing the project. Like all the estimation models, the COCOMO model requires sizing information. This information can be specified in the form of:

- Object Point(OP)
- Function Point(FP)
- Lines of Source Code(KLOC)

  For our project, we use the sizing information in the form of Lines of Source Code.
- Total Lines of Code for our project , KLOC= 6k(approx).
- Cost of each person per month, Cp= Rs.200/- (per person-hour).

1. **Equations :**

   - The initial effort (Ei) in man months is calculated using the equation:

   $$E = a * (KLOC)^b$$

   Where, $a = 3.0$,$b = 1.12$, for a semi-detached project
   E= Efforts in person-hour

   $$D = a * (E)^b$$

   Where, $a = 2.5$,$b = 0.32$, for a semi-detached project
   D= Duration of project in months

2. **Semi-detached project :**

   Project of moderate size and complexity, where teams with mixed experience levels must meet a mixed rigid and less than rigid requirements (project midway between embedded and organic types).

- Equation for calculation of Number of people required for completion of project, using the COCOMO model is :

  $$N = E/D$$

  Where, N= Number of people required
  E= Efforts in person-month
  D= Duration of project in months

- Equation for calculation of Cost of Project, using the COCOMO model is :

  $$C = D * Cp * hrs$$

  Where, C= Cost of project
  D= Duration in hours
  Cp= Cost incurred per person-hour
  Hrs= hours

- Efforts :

  $$E = 3.0 * (5.2)^1.12$$

  E= 22.31 person-months
  Total of 22.31 person-months are required to complete the project successfully.

- Number of people required for the project :
  N= 22.31/7
  N= 3.83
  N= 4 people
  Therefore 4 people are required to successfully complete the project on schedule.

- **Duration of project** :

  $$D = 2.0 * (E)^0.32$$

D= 6.75 months

The approximate duration of project is 7 months.

- **Cost of project :**

  C= 4*30*210= 25200/-

  Therefore, the cost of project is 25200/-(approx.)

3.1.2 Project Resources

1. **People :** 4 Members required

2. **Hardware :**

   - Hard Disk 40GB and above
   - RAM 1 GB and above
   - Processor P4 and above
   - Camera

3. **Software :**

   - Software: Microsoft VS 2010
   - Tools: .Net Framework Tools
   - Database: SQL 2008

3.2 RISK MANAGEMENT W.R.T. NP HARD ANALYSIS

3.2.1 Risk Identification

1. Have top software and customer managers formally committed to support the project?
   - No
2. Are end-users enthusiastically committed to the project and the system/product to be built?
   - No
3. Are requirements fully understood by the software engineering team and its customers?
   - Yes
4. Have customers been involved fully in the definition of requirements?
   - No
5. Do end-users have realistic expectations?
   - Yes
6. Does the software engineering team have the right mix of skills?
   - Yes
7. Are project requirements stable?
   - Yes
8. Is the number of people on the project team adequate to do the job?
   - Yes
9. Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?
   - Yes

3.2.2 Risk Analysis

ID	Risk Description	Probability	Impact		
			Schedule	Quality	Overall
1	If document not properly get inputted	Low	Low	High	High
2	Clustering or matching not done proper	Low	Low	High	High
3	If summarization not done proper	Low	Low	High	High

Table 3.1: Risk Table

Probability	Value	Description
High	Probability of occurrence is	$> 75\%$
Medium	Probability of occurrence is	$26 - 75\%$
Low	Probability of occurrence is	$< 25\%$

Table 3.2: Risk Probability definitions [1]

Impact	Value	Description
Very high	$> 10\%$	Schedule impact or Unacceptable quality
High	$5 - 10\%$	Schedule impact or Some parts of the project have low quality
Medium	$< 5\%$	Schedule impact or Barely noticeable degradation in quality Low Impact on schedule or Quality can be incorporated

Table 3.3: Risk Impact definitions [1]

3.2.3 Overview of Risk Mitigation, Monitoring, Management

Following are the details for each risk.

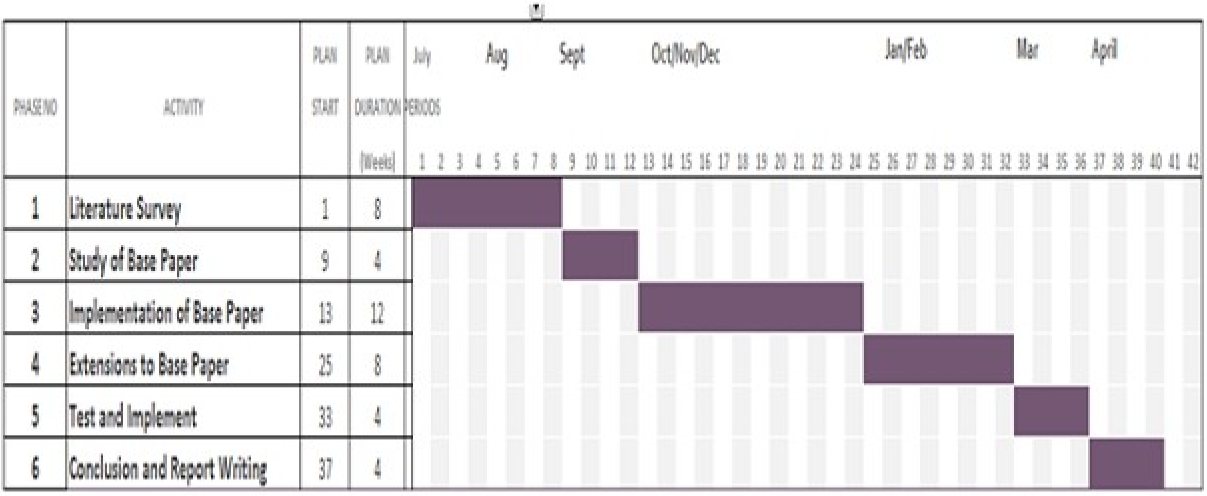
Risk ID	1
Risk Description	If document not properly get inputted.
Category	Input and document Gathering.
Source	Proper Document are to be given input.
Probability	Low
Impact	High
Response	Mitigate
Strategy	Proper Marathi document will give proper output.
Risk Status	Occurred

Risk ID	2
Risk Description	Clustering or matching not done proper.
Category	Coding
Source	Proper coding of clustering is to be done properly.
Probability	Low
Impact	High
Response	Mitigate
Strategy	Better coding algorithm and input data will solve issue.
Risk Status	Identified

Risk ID	3
Risk Description	If summarization not done proper.
Category	Coding
Source	Summarization and graph based matching algorithm will solve issue.
Probability	Low
Impact	Very High
Response	Accept
Strategy	Pattern matching algorithm and summarization will solve issue.
Risk Status	Identified

3.3 PROJECT SCHEDULE

3.3.1 Timeline Chart



PHASE NO	ACTIVITY	PLAN START	PLAN DURATION (Weeks)
1	Literature Survey	1	8
2	Study of Base Paper	9	4
3	Implementation of Base Paper	13	12
4	Extensions to Base Paper	25	8
5	Test and Implement	33	4
6	Conclusion and Report Writing	37	4

Figure 3.1: Timeline Chart

3.4 TEAM ORGANIZATION

The manner in which staff is organized and the mechanisms for reporting are noted.

3.4.1 Team structure

4 Person per team

- Sanket Sutar
- Prathamesh Chaudhari
- Piyush Patil
- Rupesh Deshmukh

3.4.2 Management reporting and communication

Period	Work Plan Topics	Work Done Space
2=7 to 7=7	Start of Project Topic search	Start of project topic search
9=7 to 14=7	Discussion on various topics with Guide	Discussion on various topics with guide
16=7 to 21=7	Discussion on various topics with Guide	Finalization of project topic
23=7 to 28=7	Finalization of topics	Searching on project related to topic
20=8 to 25=8	Finalization of project contents	Finalization of project contents
27=8 to 1=9	Preparation of Synopsis	Submission of Synopsis
3=9 to 8=9	Submission of Synopsis	Finding History of topics
10=9 to 15=9	Finding History of topic	Creating & Finalizing block diagram of project
17=9 to 22=9	Creating and Finalizing block	diagram Study of C# language in Detail
24=9 to 29=9	C# language learning	Coding for Stage I
1=10 to 6=10	Coding of module I	Coding for Stage I
8=10 to 13=10	Coding of module I	Coding for Stage I
15=10 to 20=10	Preparation of PPT and project report in latex	Preparation of PPT and project report in latex
22=10 to 27=10	Presentation of project stage I	Preparation of PPT and project report in latex
29=10	Submission of project report	Presentation of project stage
30=10	Coding of module I	Submission of Project Report
1=1 to 5=1	Coding for Module II	
7=1 to 12=1	Coding for Module III	
14=1 to 19=1	Coding for Module IV	
21=1 to 26=1	Evaluating performance of system	

Period	Work Plan Topics	Work Done Space
28=1 to 2=2	Comparison of various retrieval system based on project result	
4=2 to 9=2	Trial and error method applications	
11=2 to 16=2	Removing errors if present in overall system	
18=2 to 23=2	Finalizing project	
25=2 to 2=3	Preparation of project report	
4=3 to 9=3	Preparation of PPT	
11=3 to 16=3	Preparation of paper base on project result	
18=3 to 23=3	Presentation to Guide	
25=3 to 30=3	Submission of Project Report	

Table 3.4: Management reporting and communication

CHAPTER 4

SOFTWARE REQUIREMENT SPECIFICATION

4.1 INTRODUCTION

4.1.1 Purpose and Scope of Document

A Software requirements specification (SRS), a requirements specification for a soft- ware system, is a complete description of the behavior of a system to be developed and may include a set of use cases that describe interactions the users will have with the software. In addition it also contains non-functional requirements. Non- functional requirements impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

The software requirements specification document enlists all necessary requirements that are required for the project development. To derive the requirements we need to have clear and thorough understanding of the products to be developed. This is prepared after detailed communications with the project team and customer. A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform. An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real- world situations.

Parameters such as operating speed, response time, availability, portability, maintainability, footprint, security and speed of recovery from adverse events are evaluated. Methods of defining an SRS are described by the IEEE (Institute of Electrical and Electronics Engineers) specification 830-1998.There are many good definitions of System and Software Requirements Specifications that will provide us a good basis upon which we can both define a great specification and help us identify deficiencies in our past efforts. There is also a lot of great stuff on the web about writing good specifications. The problem is not lack of knowledge about how to create a correctly formatted specification or even what should go into the specification.

4.1.2 Overview of responsibilities of Developer

- Coordinate with team members.
- Assign various tasks to team members.
- Work in developing project plan, budget and schedule.
- Track project progress regularly and report to guide.
- Ensuring that project is completed within given budget and timeline.
- Conducts risk management analysis.

  Coordinates documentation, testing, and training efforts related to project plan.

4.2 USAGE SCENARIO

4.2.1 User profiles

- **User 1 :**

  This type of user will be users who want to classify sports videos.

4.2.2 Use-cases

All use-cases for the software are presented. Description of all main Use cases using use case template is to be provided.

Sr No.	Use Case	Description	Actors	Assumptions
1	Input	In this usecase we will take dataset and document as an input.	Dataset, System, User	Proper input and dataset is been taken.
2	Training Model	To train dataset	Dataset, System	Proper training model is been worked.
3	Preprocessing	Document preprocessing	System	Proper processing of document is been done.
4	Scoring	To match data with input document and generate score.	System, Trained dataset	Proper documentation is been done based on input and sentence matching
5	Summarization	To generate summary	System, user	To generate summarization based on inputs.

Table 4.1: Use Cases

4.2.3 Use Case View

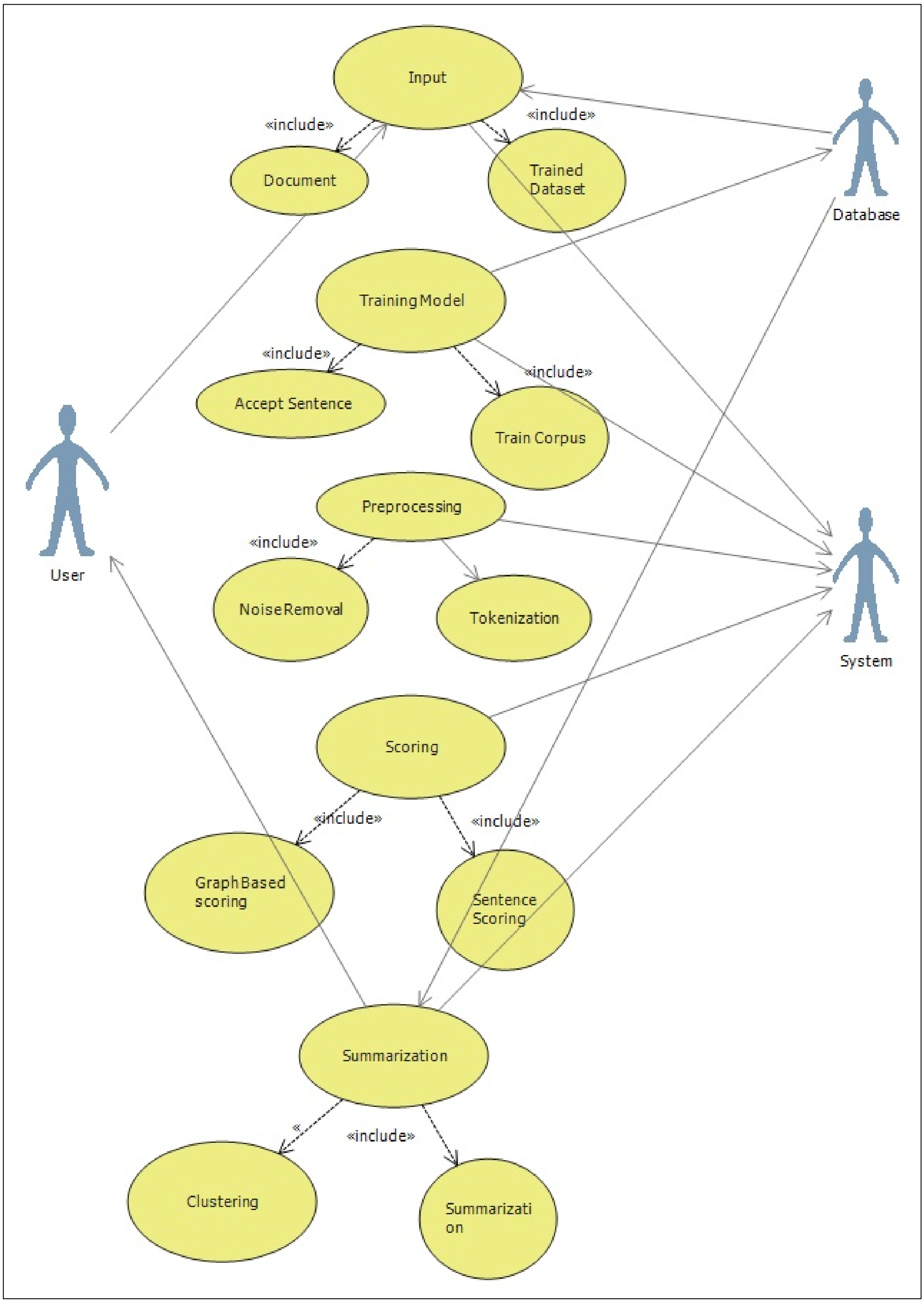


Figure 4.1: Use case diagram

4.3 DATA MODEL AND DESCRIPTION

4.3.1 Data Description

- Document Preprocessing.
- Feature Extraction.
- Sentence Scoring.
- Graph Scoring.
- Similarity.
- Summarization.

4.3.2 Data objects and Relationships

Data objects and their major attributes and relationships among data objects are described using an ERD- like form.

4.4 FUNCTIONAL MODEL AND DESCRIPTION

4.4.1 Data Flow Diagram

4.4.1.1 Level 0 Data Flow Diagram

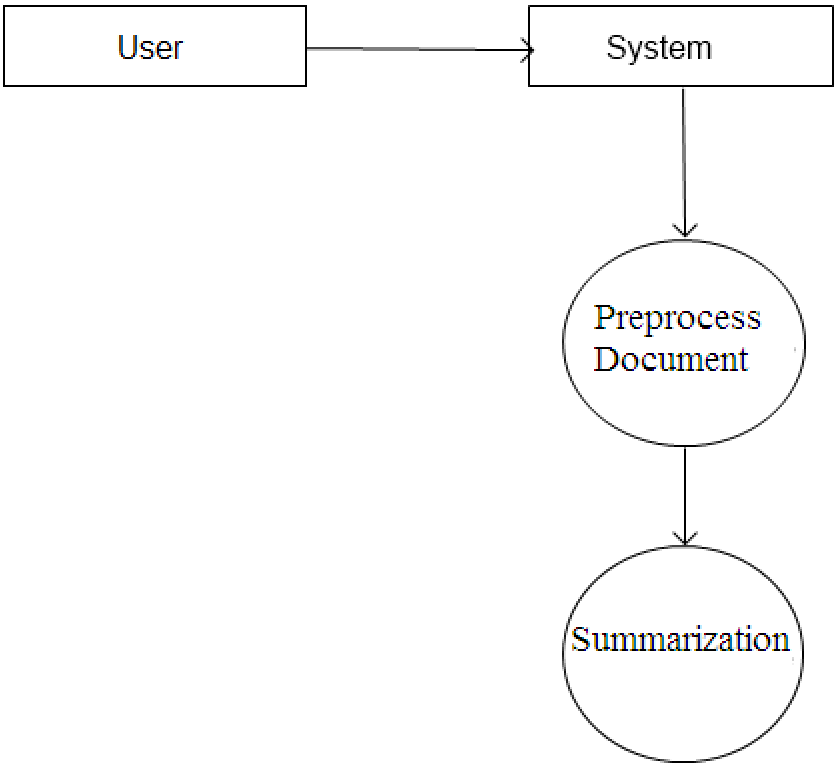


Figure 4.2: Level 0 Data Flow Diagram

4.4.1.2 Level 1 Data Flow Diagram

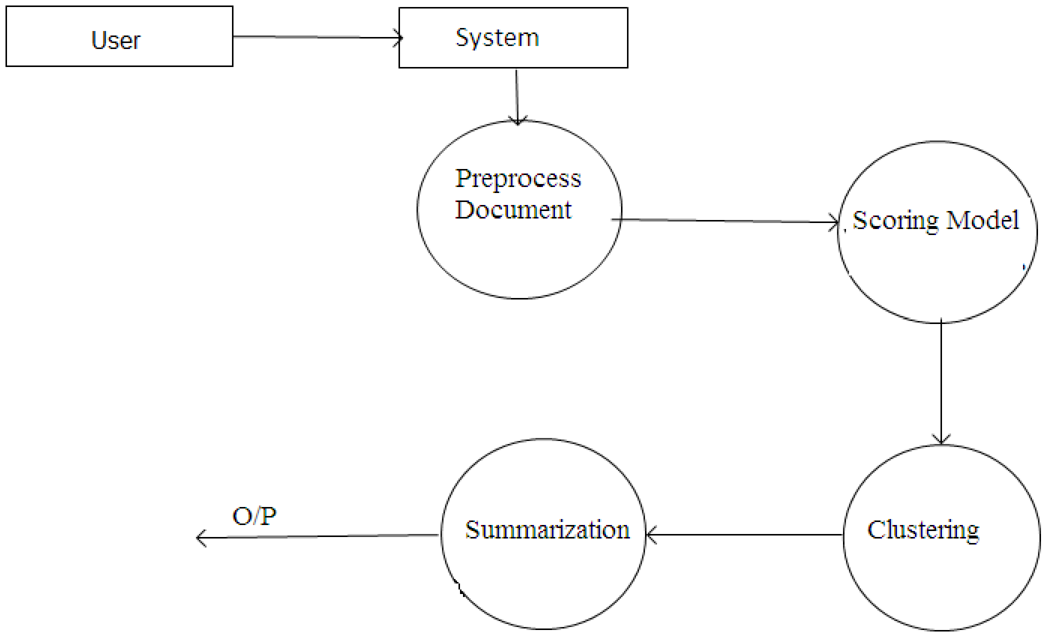


Figure 4.3: Level 1 Data Flow Diagram

4.4.2 Description of functions

A description of each software function is presented. A processing narrative for function n is presented.(Steps)/ Activity Diagrams. For Example Refer 4.4

The flow of main activities function are been performed in sequence.

1. Feature Extraction.
2. Sentence Scoring.
3. Graph Scoring.
4. Similarity.

4.4.3 Activity Diagram:

The Activity diagram represents the steps taken.

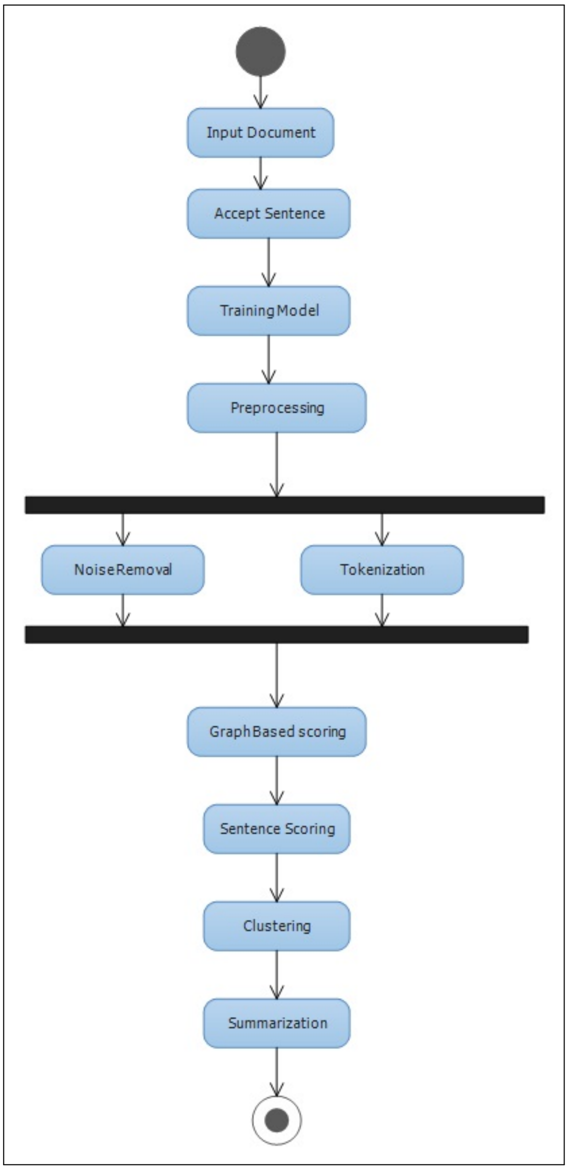


Figure 4.4: Activity diagram

4.4.4 Non Functional Requirements:

1. **User Interfaces :**

   System GUI will be the user interface.

2. **Hardware Requirements :**

   - Processor : Core i3 or more with speed 2 GHz
   - RAM : 4GB+
   - Hard Disk : 500 GB

3. **Software Resources Required :**

- Operating System: Windows95/98/XP/VISTA/7/8/9/10.
- Technology: .Visual Studio
- Language : C# language

4. **Communication Interface :**

   GUI will be our communication interface.

5. **Performance requirements :**

   Performance requirements for proposed system are as follows :

   - System will perform if proper dataset is been provided.
   - Will result better if it has proper training and proper documentation input is given.

6. **Safety requirements :**

   No safety requirements is been needed as our system is purely software oriented.

7. **Security requirements :**

   We are using login authentication methods for data privacy.

8. **Database Requirements :**

   System will be using food with nutrition's dataset.

9. **Legal Requirements :**

   System has to use legal version of ..net framework for installation process.

4.4.5 State Diagram:

State Machine Diagram

Fig.4.5 example shows the State diagram of System. The State are represented in ovals and state of system gets changed when certain events occur.

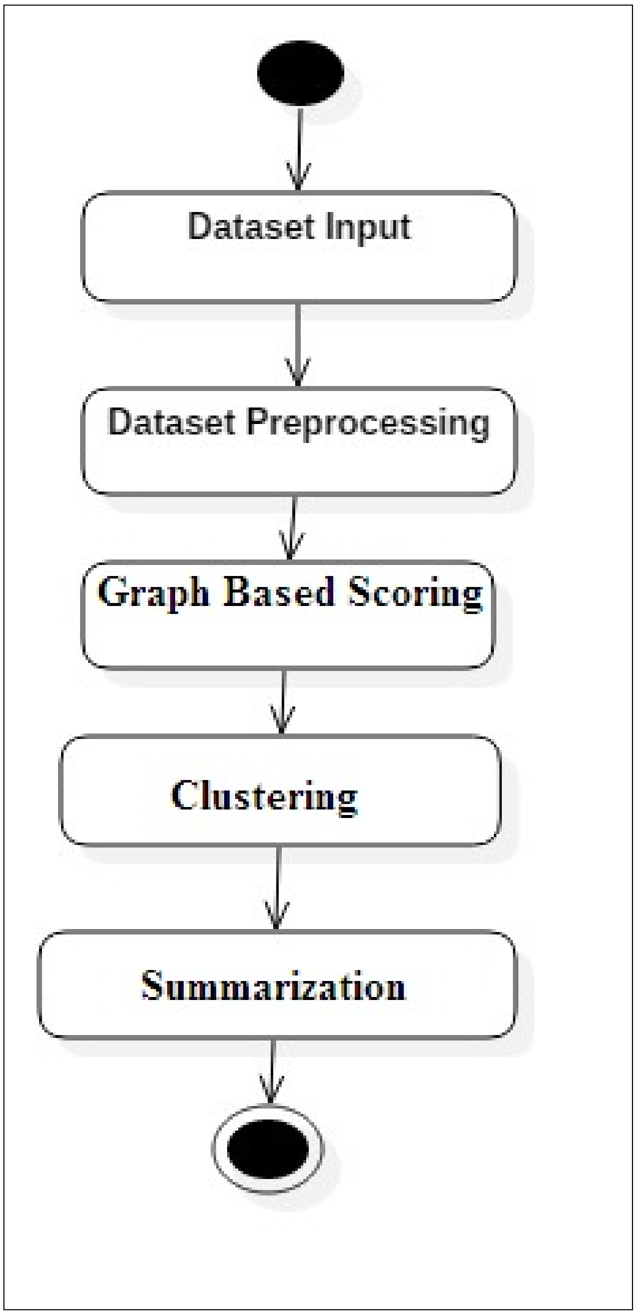


Figure 4.5: State Machine diagram

4.4.6 Design Constraints

To make GUI user friendly so as, new user can easily operate system. System will be usable based on its input and output language.

4.4.7 Software Interface Description

- **Communication Interface :**
  GUI will be our communication interface.

CHAPTER 5

DETAILED DESIGN DOCUMENT

5.1 INTRODUCTION

Text Summarization is a technique of condensing actual text into abstract form which provides same meaning and information as provided by actual text. It chooses the most informative part of text and forms summaries that reveal the main purpose of the given document. It yields summary produced by summarization system which allows readers to comprehend the content of document instead for reading each and every individual document.

So, the overall intention of text summarizer is to provide the meaning of text in less words and sentences. Summarization systems can be sorted into two categories:

- Abstraction-based summarization
- Extraction-based summarization

5.2 ARCHITECTURAL DESIGN

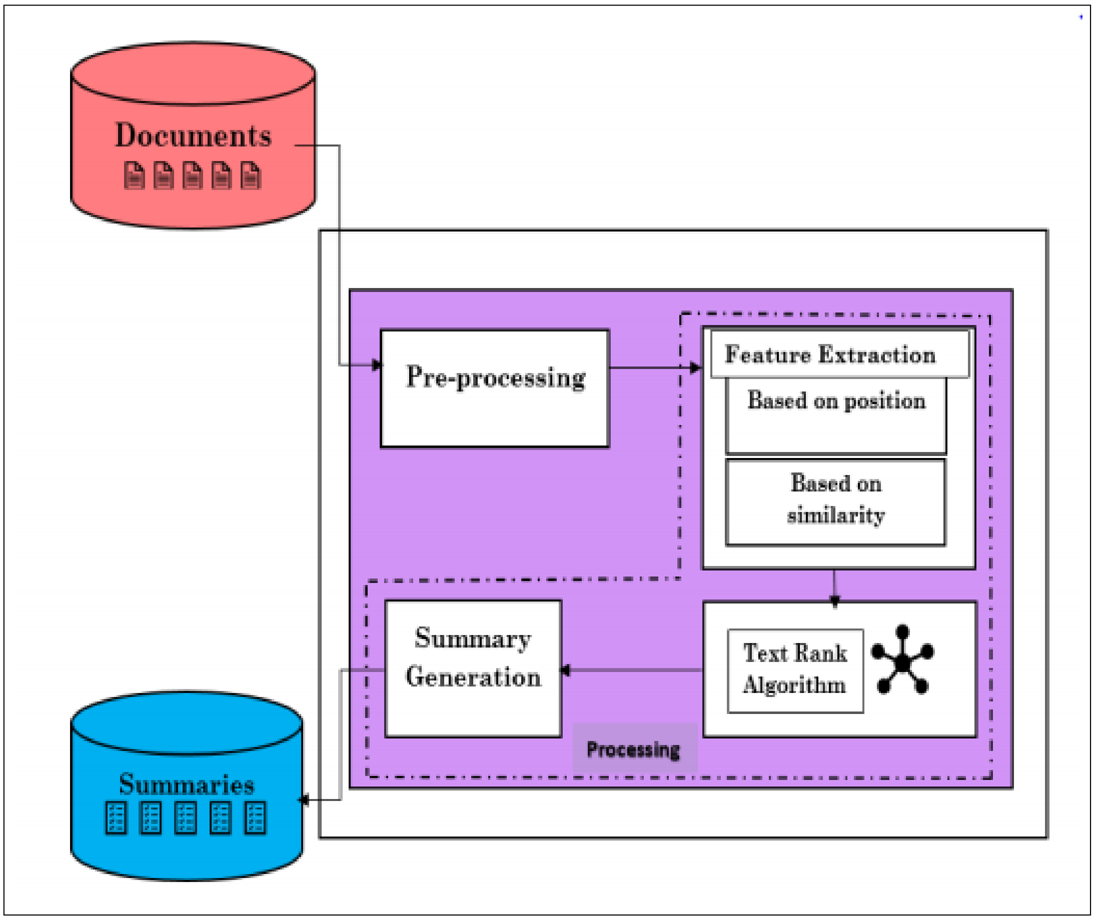


Figure 5.1: Architecture diagram

1. **Dataset :**

   - Multiple documents from dataset will be used to extract texts from documents on different subjects, such as education, politics and news articles. We have 634 documents based on news articles discussing the different topic in Marathi language.

   - The EMILLE (Enabling Minority Language Engineering) which includes monolingual, parallel and annotated corpora for Asian Languages including marathi is used for obtaining multi documents. The system can be divided into two broad stages: Pre-Processing and Processing stage.

2. **Pre-processing Stage :**

Pre-processing stage is essential in text summarization. It results into pre-processed data, which is ideally fit for processing stage. In general pre-processing stage consists of steps to remove punctuation marks, tokenization, stop word removal, stemming, etc. In this section we will discuss various steps used in pre-processing stage.

- **Boundary identification and punctuation marks removal:**

  Every sentence ends with a punctuation mark depending on the nature of sentence, whether interrogative, exclamatory, imperative or declarative. Also, use of quotation marks (”,’), commas(,), special characters(&,*,—) and symbols(#,), etc. is frequent.

  But when it comes to extract important words for processing stage, we need to eliminate these punctuation marks. Hence, we use techniques for removal of punctuation marks. The output of this step is punctuation marks free sentences in the document.

- **Stop words elimination :**

  Frequently occurring non essential words for processing in text summarization are generally termed as stop words. In marathi language, we use stop words like shivay, ase, eetar etc. in day to day use. We should eliminate them for obtaining meaningful context while processing the documents. The output of this sentence is stop words free sentences in the document.

- **Stemming and lemmatization :**

  The process of obtaining stem / radix or root word for morphological variants present in the documents. Lemmatization identifies lemma of a word. It is mapping of verbs into their infinitive and nouns into their singular form. Methods used for constructing stemmers include : Rule based-Porter’s Stemmer, Husk stemmer, Unsupervised stemming, suffix stripping-Lovins stemmer, Dawson stemmer, N gram method, HMM method, YASS(Yet Another Suffix Stripper) stemmer etc.

3. **Feature extraction :**

The features like SOV (Subject Object Verb - Experimental) verification, sentence positional value (POS tagging), TF-ISF (Term Frequency/ Inverse Sentence Frequency) or TF-IDF (Term Frequency/ Inverse Document Frequency) are extracted from pre-processed sentences. Sentences are further ranked on basis of features extracted.

5.3 DATA DESIGN

A description of all data structures including internal, global, and temporary data structures, database design (tables), file formats.

5.3.1 Internal software data structure

Stop words file and features are the internal data structure.

5.3.2 Global data structure

Features and Trained dataset are the global data structure.

5.3.3 Temporary data structure

Input document file are the temporary data structure.

5.3.4 Database description

https://github.com/prratadiya/marathi-news-document-dataset

Context :

Collection of 100 news articles in Marathi along with their extractive text summaries.

CHAPTER 6

DATASET AND EXPERIMENTAL SETUP

6.1 CONTENT

A collection of 100 news articles in Marathi language along with their corresponding summaries. The summaries are extractive in nature and have been acknowledged by a language expert. Results obtained by baseline approaches on this dataset after basic preprocessing steps (stop word removal, suffix stemming) were as follows :

Acknowledgements :

Recipe information lifted from:
https://github.com/prratadiya/marathi-news-document-dataset

CHAPTER 7

SUMMARY AND CONCLUSION

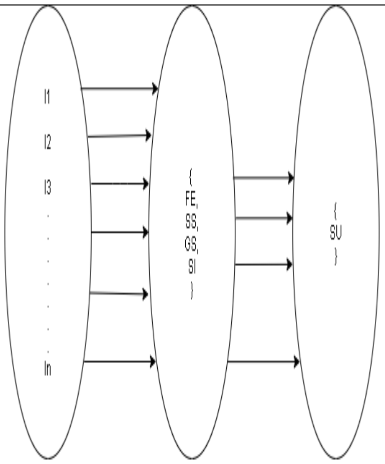
With the tremendous increase in the amount of content accessible online, there is a need of fast and effective automatic summarization system. The most important steps in this system approach are feature extraction, scoring and graph generation. This system can be used in various fields like education, in search engines to improve their performances, for Marathi news clustering, Question generation purpose and many other application oriented areas, etc.

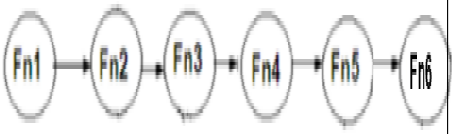
REFERENCES

[1] R. S. Pressman, *Software Engineering (3rd Ed.): A Practitioner's Approach*. New York, NY, USA: McGraw-Hill, Inc., 1992.

ANNEXURE A

MATHEMATICAL MODEL

Sr No.	Description	UML design observations
1.	**Problem description**	
	• Document Preprocessing. • Feature Extraction. • Sentence Scoring. • Graph Scoring. • Similarity • Summarization Let the system be described by *S*, $S = DI, FE, SS, GS, SI, SU$	Where, S: is a System. DI: Document Input. FE: Feature Extraction. SS: Sentence Scoring. GS: Graph Scoring. SI: Similarity. SU: Summarization.
2.	**Activity**	
	$D = d1, d2, \ldots\ldots\ldots\ldots\ldots.., dn$ $F = f1, f2, \ldots\ldots\ldots\ldots\ldots\ldots, fn$ $Y = FE, SS, GS, SI, SU$	Where, D is the set of extracted Input Document. F is the set of Function. Y is a set of techniques use for Marathi Document Summarization.
3.	**Vein diagram**	
		Where, S: is a System DI: Document Input. FE: Feature Extraction. SS: Sentence Scoring. GS: Graph Scoring. SI: Similarity. SU: Summarization.

Sr No.	Description	UML design observations
4.	**State diagram**	
		Fn1: Document Input. Fn2: Feature Extraction. Fn3: Sentence Scoring. Fn4: Graph Scoring. Fn5: Similarity. Fn6: Summarization.

5. **Functional Dependencies**

	Fn1	Fn2	Fn3	Fn4	Fn5	Fn6
Fn1	1	0	0	0	0	0
Fn2	0	1	0	0	0	0
Fn3	0	0	1	0	0	0
Fn4	0	0	0	1	0	0
Fn5	0	0	0	0	1	0
Fn6	0	0	0	0	0	1

- Fn1: Document Input.
- Fn2: Feature Extraction.
- Fn3: Sentence Scoring.
- Fn4: Graph Scoring.
- Fn5: Similarity.
- Fn6: Summarization.

Table A.1: Mathematical Model

ANNEXURE B

PLAGIARISM REPORT

ANNEXURE C

PAPER PUBLISHED (IF ANY)

ANNEXURE D

SPONSORSHIP DETAIL (IF ANY)