SAVITRIBAI PHULE PUNE UNIVERSITY

A PROJECT REPORT ON

ABSTRACTIVE DOCUMENT SUMMARIZATION USING MACHINE LEARNING APPROACH

SUBMITTED TOWARDS THE
PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

BACHELOR OF ENGINEERING
(Computer Engineering)

BY

Mr. PURUSHOTTAM NIMJE	Exam No: B190734320
Mr. MAYUR KARDILE	Exam No: B190734276
Mr. PRATIK BACHCHE	Exam No: B190734210
Ms. ANUJA PARVE	Exam No: B190734302

Under The Guidance of

Prof. Kanchan Pradhan

DEPARTMENT OF COMPUTER ENGINEERING

JSPM's BHIVARABAI SAWANT INSTITUTE OF TECHNOLOGY AND RESEARCH

GAT NO:720 /1 and 2, NAGAR ROAD, WAGHOLI,PUNE-412207

JSPM's BHIVARABAI SAWANT INSTITUTE OF TECHNOLOGY AND RESEARCH

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the Project Entitled

ABSTRACTIVE DOCUMENT SUMMARIZATION USING MACHINE LEARNING APPROACH

Submitted by

Mr. PURUSHOTTAM NIMJE	Exam No: B190734320
Mr. MAYUR KARDILE	Exam No: B190734276
Mr. PRATIK BACHCHE	Exam No: B190734210
Ms. ANUJA PARVE	Exam No: B190734302

is a bonafide work carried out by Students under the supervision of Prof. Kanchan and it is submitted towards the partial fulfillment of the requirement of Bachelor of Engineering (Computer Engineering).

Prof. Kanchan Pradhan
Internal Guide
Dept. of Computer Engg.

Dr. G. M. Bhandari
H.O.D
Dept. of Computer Engg.

Dr. T. K. Nagaraj
Principal
JSPM's Bhivarabai Sawant Institute of Technology and Research

Signature of Internal Examiner

Signature of External Examiner

PROJECT APPROVAL SHEET

A Project Title

Abstractive Document Summarization Using Machine Learning Approach

Is successfully completed by

Mr. PURUSHOTTAM NIMJE	Exam No: B190734320
Mr. MAYUR KARDILE	Exam No: B190734276
Mr. PRATIK BACHCHE	Exam No: B190734210
Ms. ANUJA PARVE	Exam No: B190734302

at

DEPARTMENT OF COMPUTER ENGINEERING

JSPM's Bhivarabai Sawant Institute of Technology and Research

SAVITRIBAI PHULE PUNE UNIVERSITY,PUNE

ACADEMIC YEAR 2022-2023

Prof. Kanchan Pradhan	Dr. G. M. Bhandari
Internal Guide	H.O.D
Dept. of Computer Engg.	Dept. of Computer Engg.

Acknowledgments

This is a great pleasure immense satisfaction to express my deepest sense of gratitude thanks to everyone who has directly or indirectly helped me in completing my Seminar work successfully..

I express my gratitude towards project guide ***Prof.Kanchan Pradhan*** *and* ***Dr.G.M. Bhandari****, Head of Department of Computer Engineering, Jspm's, Bhivarabai sawant Institute of Technology and Research, Wagholi, Pune, who guided encouraged me in completing the Seminar work in the scheduled time. I would like to thank our Principal* ***Dr.T.K. Nagaraj****, for his extended support.*

No words are sufficient to express my gratitude to my family for their unwavering encouragement. I also thank all my friends for being a constant source of my support.

PURUSHOTTAM NIMJE
MAYUR KARDILE
PRATIK BACHCHE
ANUJA PARVE
(B.E. Computer Engg.)

Abstract

We present a novel divide-and-conquer strategy for neural summary of large documents. Our solution makes use of the document's discourse structure and sentence similarity.Divide the task into smaller summarization challenges.We divide a lengthy document and its summary into numerous source-target pairs, which are then used to train a model. learns to sum up each section of the document individually After The partial summaries are then combined to form a final comprehensive summary. In contrast to the conventional approach, we may break down the challenge of long document summary into smaller pieces and simpler problems, reducing computational complexity and increasing the number of training examples while also reducing There is noise in the target summaries. There are two publicly available datasets.

Keywords: Summarization, Machine Learning, Classification, NLP (Natural Language Processing) etc.

Contents

List of Figures

List of Tables

Chapter 1

Synopsis

1.1 Project Title

Abstractive Document Summarization Using Machine Learning Approach

1.2 Project Option

Internal Project

1.3 Internal Guide

Prof. Kanchan Pradhan

1.4 Technical Keywords (As per ACM Keywords)

1. **NLP:-**
   NLP, or Natural Language Processing, is a field of computer science that focuses on enabling computers to understand, interpret, and generate natural language, such as English, Spanish, or French. NLP involves the use of algorithms and techniques to analyze and manipulate large amounts of text data, including speech, text documents, and social media posts. Some common NLP techniques include sentiment analysis, text classification, entity recognition, and machine translation. NLP has a wide range of applications, from chatbots and virtual

assistants to language translation, text summarization, and data analysis. It is a rapidly evolving field with many exciting developments, including the use of deep learning and neural networks to improve the accuracy and effectiveness of NLP algorithms

2. **Generate Summary:-**
   Generating a summary involves creating a shorter, condensed version of a larger piece of text, such as an article, report, or research paper. The goal of summarization is to distill the most important information from the original text and present it in a concise and easy-to-read format. There are different types of summarization techniques, including extractive and abstractive summarization. Extractive summarization involves selecting and combining sentences from the original text to create a summary, while abstractive summarization involves generating new sentences that capture the key information from the original text. Summarization can be useful in a variety of contexts, such as news articles, legal documents, and academic papers, where it can help readers quickly understand the main points without having to read the entire text. It is also an important tool in natural language processing and machine learning, where it is used to help computers better understand and analyze text data.

3. **Document Summarization:-**
   Document summarization is a process of condensing a longer document into a shorter version while retaining its key information and meaning. It can be achieved through two main techniques: extractive and abstractive summarization. Extractive summarization involves selecting important sentences or phrases from the original document and combining them to create a summary. Abstractive summarization, on the other hand, involves generating new sentences that capture the meaning of the original document. Document summarization can be useful in various scenarios, such as news articles, legal documents, and scientific papers, where it can help readers quickly understand the main points without having to read the entire document. It is also an essential tool in natural language processing and machine learning, where it is used to help computers better analyze and understand text data.

1.5 Problem Statement

Our method uses the document's discourse structure and analyses sen- tence similarity to break the difficulty down into smaller summary prob- lems. A long text and its summary are divided into many source-target pairs, which are then used to train a model that learns to summarise each piece of the abstractive document independently. Following that, the component summaries are combined to form a final comprehensive overview.

1.6 Abstract

We present a novel divide-and-conquer strategy for neural summary of large documents. Our solution makes use of the document's discourse structure and sentence similarity.Divide the task into smaller summarization challenges.We divide a lengthy document and its summary into numerous source-target pairs, which are then used to train a model. learns to sum up each section of the document individually After The partial summaries are then combined to form a final comprehensive summary. In contrast to the conventional approach, we may break down the challenge of long document summary into smaller pieces and simpler problems, reducing computational complexity and increasing the number of training examples while also reducing There is noise in the target summaries. There are two publicly available datasets.
Keywords: Summarization, Machine Learning, Classification, NLP (Natural Language Processing) etc.

1.7 Goals and Objectives

- To develop a system that can automatically summarize a long document into a shorter summary with key information that captures the essence of the original document.

- To improve the efficiency of information processing and retrieval by reducing the amount of time and effort required to read through and extract relevant information from long documents.

- To create a tool that can help users quickly get an overview of the main points of a document, allowing them to make better-informed decisions about whether to read the full document or not.

- To ensure that the summary is accurate and informative, conveying the most important information from the document in a concise and readable manner.

- To develop a system that can handle a variety of document types, including research papers, news articles, legal documents, and more.

- To optimize the summary generation process to ensure that the system can produce high-quality summaries quickly and efficiently.

- To explore and integrate different techniques and methods, such as natural language processing (NLP), deep learning, and machine learning algorithms, to enhance the quality and accuracy of the summary generation process.

- To develop a user-friendly interface that allows users to easily input and summarize documents, and to display the summary in a clear and concise format.

1.8 Names of Conferences / Journals where papers can be published

- IJIRSET- International Journal of Innovative Research in Science Engineering Technology
  http://www.ijirset.com/volume-11-issue-12.html

- IJIRSET- International Journal of Innovative Research in Science Engineering Technology
  http://www.ijirset.com/volume-12-issue-4.html

1.9 Review of Conference/Journal Papers supporting Project idea

1. Paper name:- Text Summarization using Sentiment Analysis for DUC Data.
   Author:- Nidhika Yadav , Niladri Chatterjee
   Abstract:- Text summary is a technique for determining a text's main points. In the literature, various strategies have been presented, and many of these are currently in use in commercial systems. The basic purpose of this study is to see if sentiments can be used to summarise material effectively. We describe a text summarization technique based on the emotions associated with key phrases that is computationally efficient. Sentiment analysis is already in use for large-scale text data interpretation and opinion mining in a variety of fields. The current research shows that sentiment analysis can be utilised to summarise text as well. Our findings were compared to those of the conventional DUC2002 datasets.

2. Paper name:- Multi-document Summarization by using Text Rank and Maximal Marginal Relevance for Text in Bahasa Indonesia
   Author:- Dani Gunawan, Siti Hazizah Harahap, Romi Fadillah Rahmat
   Abstract:- The text summarizer eliminates unnecessary information by selecting the most important sentences. There's a chance that two or more crucial statements in a multi-document summary will contain the same information. If those sentences are included in the summary result, the information will be redundant. The goal of this research is to condense similar lines from a variety of sources containing similar information into a more concise text summary. This study divides a variety of internet news articles into six groups to achieve its goal. The articles are concatenated and pre-processed to create a clean text. Following the acquisition of clean text, the TextRank algorithm is used to extract the important sentences based on similarity measurement. This procedure produced the summary text.

3. Paper name:- Automatic Text Summarization by Local Scoring and Ranking for Improving Coherence
   Author:- P.Krishnaveni, Dr.S. R. Balasundaram.
   Abstract:- Due to the large amount of textual material available on the internet, machine-generated summarization has received a lot of attention. Manually summarising these online text documents is difficult for most people. As a result, we will require an automatic text sum-

marizer. Automatic Text Summarization (ATS) is defined as ”shortening the original text while preserving its information value and overall meaning.” Despite the fact that automatic text summarization has been studied since the 1950s, more cohesive and meaningful summaries still exist. The proposed technique generates an automatic feature-based extractive heading-wise text summarizer that improves the summary text’s coherence and, as a result, its understandability. It just summarises the input document using local scoring and ranking, and that’s it.

1.10 Plan of Project Execution

SR.NO	**Name/Title**	**Start Date**	**End Date**
1	Preliminary Survey	07/09/2022	23/09/2022
2	Introduction and Problem Statement	15/09/2022	20/09/2022
3	Literature Survey	15/09/2022	25/09/2022
4	Project Statement		
5	Software Requirement and Specification	12/09/2022	03/10/2022
6	System Design	14/092022	17/11/2022
7	Partial Report Submission	01/11/2022	28/11/2022
8	Architecture Design	01/12/2022	10/12/2022
9	Implementation	15/01/2022	20/02/2023
10	Deployement		
11	Testing	15/12/2022	20/02/2023
12	Paper Publish	11/03/2022	10/04/2023
13	Report Submission	21/04/2023	09/05/2023

Table 1.1: Project Execution Plan

Chapter 2

Technical Keywords

2.1 Area of Project

Machine Learning

2.2 Technical Keywords

1. **NLP:-**
   NLP, or Natural Language Processing, is a field of computer science that focuses on enabling computers to understand, interpret, and generate natural language, such as English, Spanish, or French. NLP involves the use of algorithms and techniques to analyze and manipulate large amounts of text data, including speech, text documents, and social media posts. Some common NLP techniques include sentiment analysis, text classification, entity recognition, and machine translation. NLP has a wide range of applications, from chatbots and virtual assistants to language translation, text summarization, and data analysis. It is a rapidly evolving field with many exciting developments, including the use of deep learning and neural networks to improve the accuracy and effectiveness of NLP algorithms

2. **Generate Summary:-**
   Generating a summary involves creating a shorter, condensed version of a larger piece of text, such as an article, report, or research paper. The goal of summarization is to distill the most important information from the original text and present it in a concise and easy-to-read format. There are different types of summarization techniques, including extractive and abstractive summarization. Extractive summarization

involves selecting and combining sentences from the original text to create a summary, while abstractive summarization involves generating new sentences that capture the key information from the original text. Summarization can be useful in a variety of contexts, such as news articles, legal documents, and academic papers, where it can help readers quickly understand the main points without having to read the entire text. It is also an important tool in natural language processing and machine learning, where it is used to help computers better understand and analyze text data.

3. **Document Summarization:-**
   Document summarization is a process of condensing a longer document into a shorter version while retaining its key information and meaning. It can be achieved through two main techniques: extractive and abstractive summarization. Extractive summarization involves selecting important sentences or phrases from the original document and combining them to create a summary. Abstractive summarization, on the other hand, involves generating new sentences that capture the meaning of the original document. Document summarization can be useful in various scenarios, such as news articles, legal documents, and scientific papers, where it can help readers quickly understand the main points without having to read the entire document. It is also an essential tool in natural language processing and machine learning, where it is used to help computers better analyze and understand text data.

Chapter 3

Introduction

With the massive growth of digital information and the increase in the amount of text data generated every day, there is an urgent need to develop automated techniques for summarizing documents. Document summarization is the task of reducing a large document into a concise summary that contains the most important information in the original text. Abstractive document summarization is a subfield of natural language processing (NLP) that aims to generate summaries that go beyond simple sentence selection by producing new phrases and sentences that capture the essence of the original text.

Abstractive summarization is challenging because it requires the system to understand the meaning and context of the input text and generate a summary that accurately reflects the most important information. Unlike extractive summarization, which selects sentences or phrases from the original text, abstractive summarization involves generating new sentences that may not appear in the original document.

Abstractive summarization is a relatively new area of research in NLP, and recent advances in deep learning have led to significant progress in the development of abstractive summarization systems. This paper provides a comprehensive overview of abstractive document summarization, including its background, key challenges, and recent advances.

Background:

The field of document summarization has a long history, with research dating back to the 1950s. Early research focused on extractive summarization, which involves selecting sentences or phrases from the original document to form a summary. However, extractive summarization has several limitations, including the inability to generate novel sentences and the risk of producing summaries that are similar to the original text.

Abstractive summarization, on the other hand, has the potential to generate summaries that are more concise and coherent than extractive summaries. However, abstractive summarization is a challenging task that requires the system to understand the meaning and context of the input text and generate a summary that accurately reflects the most important information.

Challenges:

Abstractive summarization poses several challenges, including:

1. Content selection: Abstractive summarization requires the system to identify the most important information in the input document and generate a summary that accurately reflects that information. This requires the system to have a deep understanding of the meaning and context of the text.

2. Novelty generation: Unlike extractive summarization, which relies on selecting sentences or phrases from the original text, abstractive summarization requires the system to generate novel sentences that may not appear in the original document.

3. Coherence: Abstractive summarization requires the system to generate a summary that is coherent and understandable, even if it includes new sentences that are not present in the original document. This requires the system to generate sentences that are grammatically correct and semantically meaningful.

Recent Advances:

Recent advances in deep learning have led to significant progress in the development of abstractive summarization systems. One of the most successful approaches to abstractive summarization is the transformer-based model, which uses a self-attention mechanism to capture long-term dependencies and generate high-quality summaries.

Another recent development in abstractive summarization is the use of reinforcement learning (RL) to train summarization models. RL allows the system to learn from its own mistakes by generating summaries and receiving feedback on their quality. This approach has shown promising results, particularly for generating longer summaries.

Conclusion:

Abstractive document summarization is a challenging but important problem in NLP. Recent advances in deep learning have led to significant progress in the development of abstractive summarization systems, with transformer-based models and RL-based training methods showing promising results. As the amount of digital information continues to grow, automated document summarization will become increasingly important, and abstractive summarization is likely to play a key role in this field.

3.1 Project Idea

Text Summarization is an important Natural Language Processing(NLP) task. Text summarization involves condensing a larger text into smaller sizes by preserving the meaning to convey the core message of the text. Basically, it wraps up long text into main points so that we can read any long text in short form.

3.2 Motivation

- Data compression and information comprehension, both of which are required for information science and retrieval, are inextricably linked to summarization. The ability to generate informative and well-written document summaries has the potential to increase the success of both information discovery technologies and human users looking to quickly scan vast amounts of paper for significant information. Indeed, despite being one of the least solved natural language processing (NLP) tasks, automatic summarization has recently been recognised as one of the most important.

3.3 Problem Statement

- Our method uses the document's discourse structure and analyses sentence similarity to break the difficulty down into smaller summary problems. A long text and its summary are divided into many source-target pairs, which are then used to train a model that learns to summarise each piece of the abstractive document independently. Following that, the component summaries are combined to form a final comprehensive overview.

Chapter 4

Literature Survey

1. The paper "Text Summarization using Sentiment Analysis for DUC Data" by Nidhika Yadav and Niladri Chatterjee explores the possibility of using sentiment analysis to effectively summarize text. The authors note that there are a number of text summarization techniques already available in the literature, and many of these are currently in use in commercial systems. However, the authors argue that sentiment analysis - a technique that is commonly used for large-scale text interpretation and opinion mining in a variety of fields - has the potential to be leveraged for summarization as well.

   The authors propose a computationally efficient text summarization technique that takes into account the emotions associated with key phrases. Their approach involves identifying important phrases in the text, then determining the sentiment associated with each phrase using a sentiment analysis algorithm. The authors then use this information to generate a summary that captures the overall sentiment of the text.

   To evaluate the effectiveness of their approach, the authors compare their results to those of conventional text summarization techniques using the DUC2002 dataset. Their findings suggest that sentiment analysis can be an effective tool for summarizing text, and that their approach can achieve comparable or better results than traditional methods.

2. The paper "Multi-document Summarization by using Text Rank and Maximal Marginal Relevance for Text in Bahasa Indonesia" by Dani

Gunawan, Siti Hazizah Harahap and Romi Fadillah Rahmat proposes a method for generating summaries from multiple texts in Bahasa Indonesia. The authors note that selecting the most important sentences is a crucial step in summarization, but there's a risk of redundancy if multiple crucial statements contain the same information. To overcome this issue, the authors propose a method that condenses similar lines from various sources containing similar information into a more concise text summary.

To achieve their goal, the authors divided a variety of internet news articles into six groups and concatenated them to create a clean text. Following the acquisition of clean text, the TextRank algorithm is used to extract important sentences based on similarity measurement. The authors claim that this method produces a summary that is more informative and contains fewer redundancies compared to other methods. This study could be particularly useful for summarizing large volumes of text in Bahasa Indonesia, which is an important language spoken in Southeast Asia.

In conclusion, the proposed method of multi-document summarization using Text Rank and Maximal Marginal Relevance has the potential to generate informative and concise summaries from multiple texts in Bahasa Indonesia. The authors' approach of condensing similar lines and avoiding redundancy could be particularly useful for summarizing large volumes of text. This study could pave the way for further research in the field of multi-document summarization and text analysis in Bahasa Indonesia.

3. The paper "Automatic Text Summarization by Local Scoring and Ranking for Improving Coherence" by P. Krishnaveni and Dr. S. R. Balasundaram highlights the need for machine-generated summarization of large amounts of textual material available on the internet. The authors propose a technique for automatic feature-based extractive heading-wise text summarization, which aims to improve the coherence and understandability of the summary text. The proposed technique uses local scoring and ranking to summarize the input document, preserving its information value and overall meaning.

   Despite the fact that automatic text summarization has been studied since the 1950s, the authors point out that more cohesive and meaningful summaries still exist. The proposed technique focuses on improving the coherence of summary texts by considering the relationships between the sentences in the input document. The local scoring and ranking approach used in the technique assigns scores to each sentence based on its importance, and then ranks the sentences in the order of importance to generate the summary. The technique also considers headings in the input document to ensure that the summary is generated in a heading-wise manner. Overall, the proposed technique has the potential to improve the efficiency and effectiveness of automatic text summarization.

4. The paper ”Text Summarization Model of Combining Global Gated Unit and Copy Mechanism” by S huxia Ren and Kaijie Guo discusses the challenge of ensuring accuracy in automatic text summarization, particularly with regards to Out-of-Vocabulary (OOV) issues. To address these concerns, the authors propose a new text summarization model that combines the advantages of extractive and abstractive summarization systems. Their proposed model incorporates a global gated unit and a copy mechanism (GGUC) to improve summarization quality. The experimental results showed that their model outperforms other text summary systems on LCSTS datasets, indicating its effectiveness in generating accurate and high-quality summaries.

   Overall, this paper highlights the ongoing efforts to improve the accuracy and quality of automatic text summarization using neural networks and attention mechanisms. By combining different approaches and techniques, such as the GGUC model proposed in this paper, researchers are continuing to advance the state-of-the-art in text summarization and make it more practical and useful for a variety of applications.

5. The paper ”Extractive Text Summarization Using Sentence Ranking” by J.N.Madhuri and Ganesh Kumar.R presents an automatic approach for text summarization using an extractive method. The paper proposes a statistical strategy that ranks the importance of each sentence in the input document and extracts the most relevant ones to form a concise summary. The proposed method assigns weights to each sentence based on several statistical features such as sentence length, word frequency, and sentence position. These weights are then used to rank

the sentences, and the most important sentences are extracted to form the summary.

The proposed method is evaluated on various datasets, and the results show that it outperforms existing extractive summarization techniques. The method achieves high ROUGE scores, which is a widely used metric to evaluate the quality of the summary. The paper also discusses the limitations of the proposed method, such as the inability to capture the semantic relationships between sentences, which is a common issue in extractive summarization. Overall, the proposed method provides an effective and efficient solution for extractive text summarization.

6. The paper "Text Summarization Using Text Frequency Ranking Sentence Prediction" by Meena S M, Asmitha R E and Ramkumar M P presents an approach to text summarization that involves analyzing the frequency of words in a given text and then using this information to predict which sentences contain the most important information. The authors argue that the use of text frequency ranking allows for a more accurate summarization of the text, as it ensures that the most frequently occurring words are highlighted in the summary. The approach involves several steps, including text preprocessing, sentence segmentation, and ranking based on text frequency.

   The authors also propose a tool called the Text Formatting and Reading Support Program (TFRSP) that implements their approach and can assist readers in summarizing text. TFRSP is designed to help readers overcome the challenge of information overload by summarizing the text and presenting the most important information in a concise and structured manner. The tool can also be used to improve readability by identifying and highlighting the most relevant sentences in the text. Overall, the paper presents a promising approach to text summarization and demonstrates the potential benefits of using text frequency ranking to improve the accuracy and readability of summarization.

7. The paper titled "A Divide-and-Conquer Approach to the Summarization of Long Documents" proposes a new approach for summarizing large documents using a divide-and-conquer strategy. The authors, Alexios Gidiotis and Grigorios Tsoumakas, take advantage of the discourse structure of a document and divide the summarization task

into smaller problems using sentence similarity. They create multiple source-target pairs by dividing the lengthy document and its summary, which are then used to train a model that can summarize each section of the document separately. The partial summaries are then combined to create a final summary of the entire document. This approach reduces computational complexity and increases the number of training examples, leading to more accurate summaries with reduced noise.

By dividing the challenge of long document summarization into smaller and simpler problems, the proposed approach reduces the computational cost and complexity of the summarization task, making it more feasible for large documents. The authors have demonstrated the effectiveness of their approach using experiments on benchmark datasets, which showed that their approach outperformed existing state-of-the-art methods for long document summarization. The divide-and-conquer approach proposed in this paper has the potential to revolutionize the field of document summarization and can be extended to other applications, such as text classification and sentiment analysis.

Chapter 5

Problem Definition and scope

5.1 Problem Statement

Our method uses the document's discourse structure and analyses sentence similarity to break the difficulty down into smaller summary problems. A long text and its summary are divided into many source-target pairs, which are then used to train a model that learns to summarise each piece of the abstractive document independently. Following that, the component summaries are combined to form a final comprehensive overview.

5.1.1 Goals and objectives

The main objective of a text summarization system is to identify the most important information from the given text and present it to the end users.

5.2 Methodologies of Problem solving and efficiency issue

Using NLP Algorithm we solve the efficiency issues.

5.3 Outcome

Autonomy abstracted Document summarising is a multi-step process with several sub-tasks. Every subtask has the ability to generate high-quality summaries. Abstractive document summarization requires identifying necessary paragraphs from a given document.

5.4 Applications

NLP text summarization is the process of breaking down lengthy text into digestible paragraphs or sentences. This method extracts vital information while also preserving the meaning of the text. This reduces the time required for grasping lengthy pieces such as articles without losing vital information

5.5 Hardware Resources Required

System Processors : Core2Duo
Speed : 2.4 GHz
Hard Disk : 150 GB

5.6 Software Resources Required

Operating system : 64 bit Windows 10.
Coding Language : Python
Design constraints : Spyder

Chapter 6

Project Plan

6.1 Project Plan

NO	TASKS	DURATION(DAYS
1	Group Formation	4
2	Decide Area of Interest	4
3	Search Topic	5
4	Topic Selection	5
5	Sanction Topic	5
6	Search Related Information	12
7	Understanding Concept	7
8	Search Essential Document(IEEE and White Paper, Software)	6
9	Problem Definition	2
10	Literature Survey	5
11	SRS	14
12	Project Planning	2

Table 6.1: Project Plan

6.2 Project Estimates

6.2.1 Reconciled Estimates

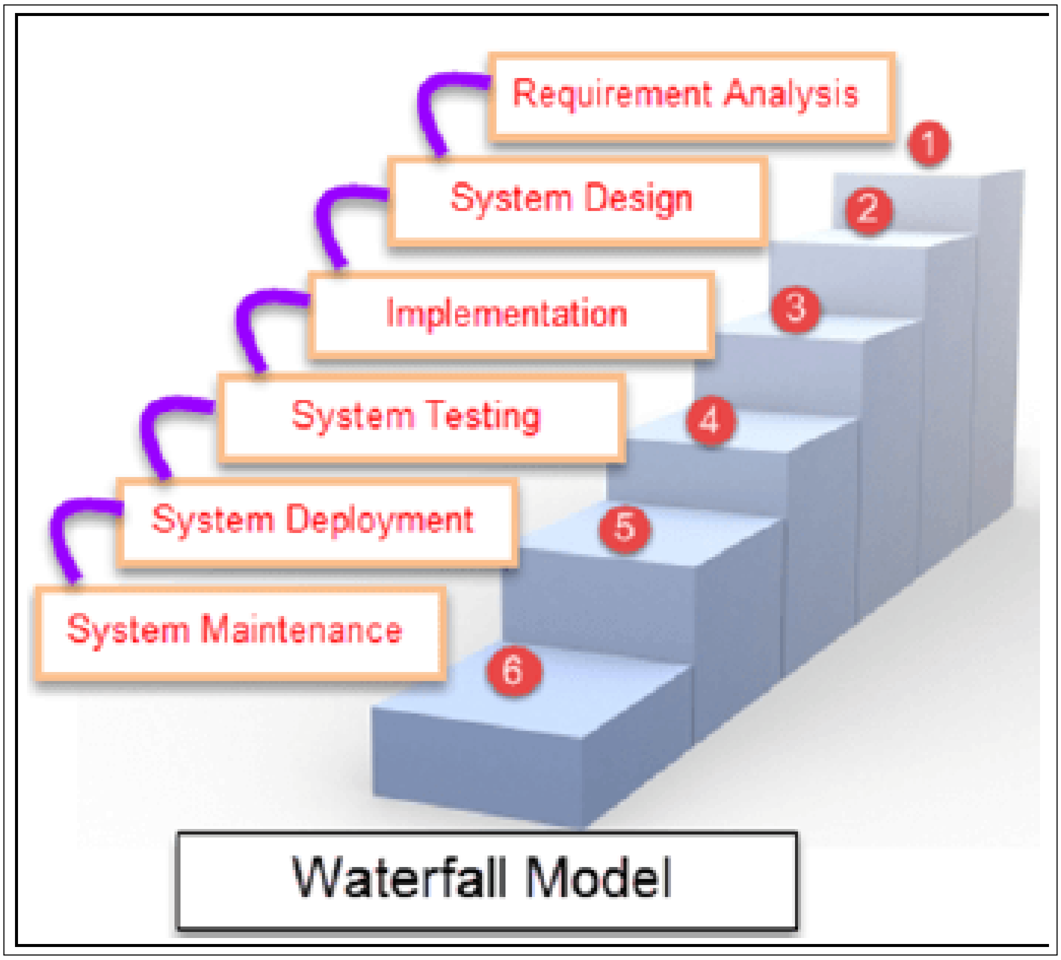


Figure 6.1: SDLC Model Diagram

1. **Requirement gathering and analysis:** In this step of waterfall we identify what are various requirements are need for our project such are software and hardware required, database, and interfaces.
2. **System Design:** In this system design phase we design the system which is easily understood for end user i.e. user friendly.We design some UML diagrams and data flow diagram to understand the system flow and system module and sequence of execution.

3. **Implementation:** In implementation phase of our project we have implemented various module required of successfully getting expected outcome at the different module levels. With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

4. **Testing:** The different test cases are performed to test whether the project module are giving expected outcome in assumed time. All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

5. **Deployment of System:** Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.

6. **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.
   All these phases are cascaded to each other in which progress is seen as flowing steadily downwards like a waterfall through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap.

6.2.2 Project Resource

Well configured Laptop, Spyder IDE, 2 GHZ CPU speed, 2 GB RAM, Internet connection.

6.3 Risk Management

1. **In appropriate dataset -** To overcome this risk we are trying to use well organized and complete dataset.

2. **Security-** To overcome and improving security we use multilevel security like access permissions of user.

6.3.1 Risk Identification

1. Have top software and customer managers formally committed to support the project?
   Ans- Not applicable.

2. Are end-users enthusiastically committed to the project and the system/product to be built?
   Ans- Not known at this time.

3. Are requirements fully understood by the software engineering team and its customers?
   Ans-Yes

4. Have customers been involved fully in the definition of requirements?
   Ans-Not applicable

5. Do end-users have realistic expectations?
   Ans-Not applicable

6. Does the software engineering team have the right mix of skills?
   Ans-yes

7. Are project requirements stable?
   Ans-Not applicable

8. Is the number of people on the project team adequate to do the job?
   Ans-Not applicable

9. Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?
   Ans-Not applicable

6.3.2 Risk Analysis

The risks for the Project can be analyzed within the constraints of time and quality

6.3.3 Risk Table

ID	Risk Description	Probability	Schedule	Quality	Overall
1	Correctness	low	low	high	low
2	Availability	high	low	high	high

Table 6.2: Risk Table

Risk Probability definitions:

Probability	Value	Description
High	Probability of occurence is	>75%
Medium	Probability of occurence is	26-75%
Low	Probability of occurence is	<25%

Table 6.3: Risk Probability

Risk Impact definitions:

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 6.4: Risk Impact

6.3.4 Overview of Risk Mitigation, Monitoring, Management:

Following are the details for each risk.

Risk ID	1
Risk Description	Loss of data
Category	Enviorment
Source	Shoulder Surfing
Probability	High
Impact	High
Response	Mitigate
Strategy	Break security
Risk Status	Occured

Table 6.5: Overview of Risk Mitigation, Monitoring, Man- agemen

6.4 Project Schedule

6.4.1 Project Task Set

Major Tasks in the Project stages are:

- Task 1: correctness
- Task 2: availability
- Task 3: integrity

6.4.2 Task Network

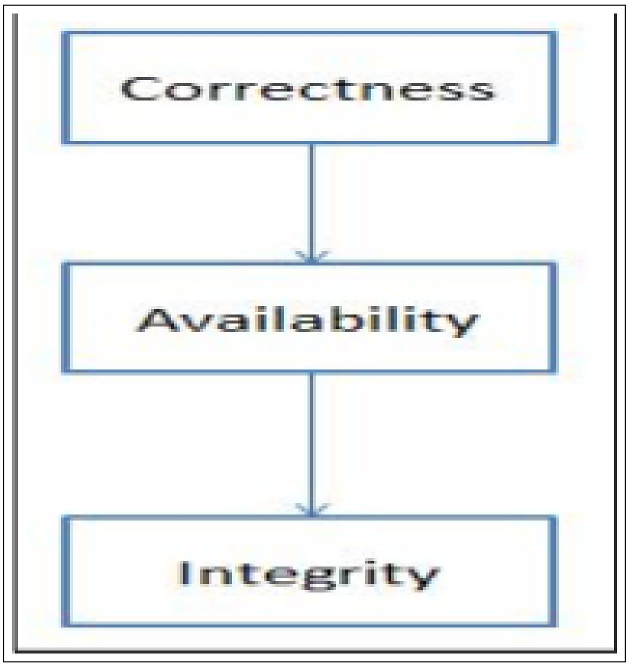


Figure 6.2: Task Network

6.4.3 Timeline Chart

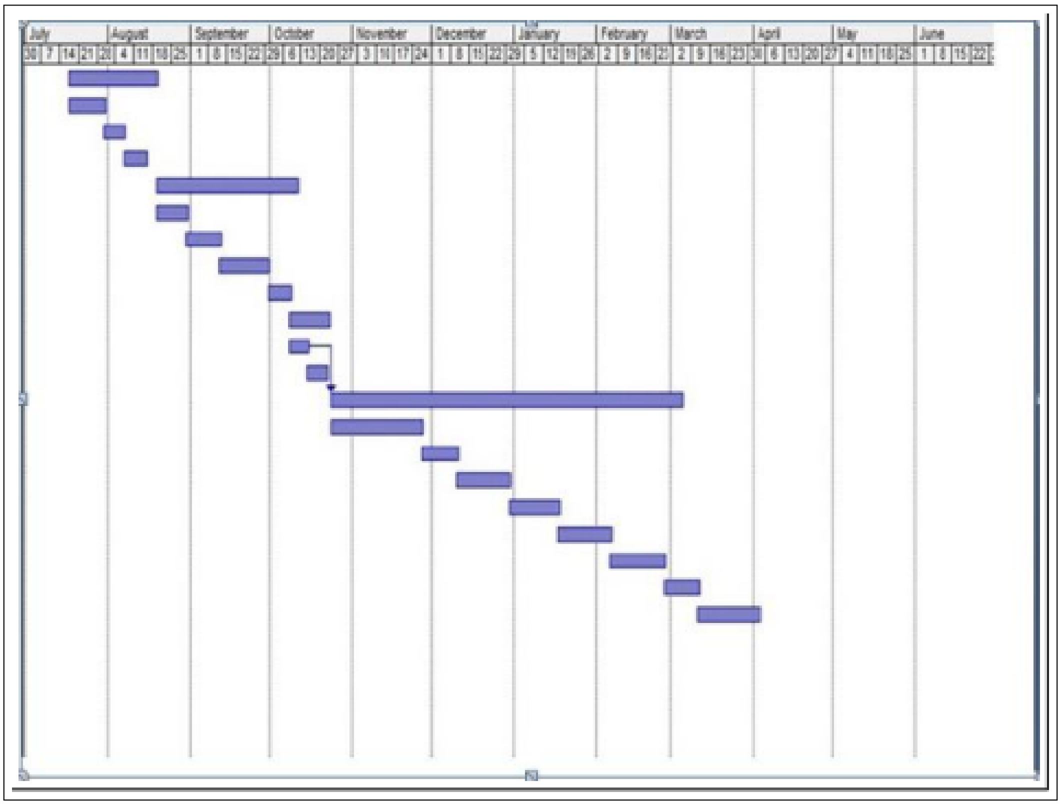


Figure 6.3: Timeline chart

6.5 Team Organization

Team consists of 4 members and proper planning mechanism are used and roles of each member are defined.

6.5.1 Team Structure

The team structure for the project is identified. There are total 4 members in our team and roles are defined. All members are contributing in all the phases of project.

6.5.2 Management Reporting And Communication

Well planning mechanisms are used for progress reporting and inter/intra team communication are identified as per requirements of the project.

Chapter 7

Software Requirement Specification

7.1 Introduction

7.1.1 Project Scope

The rapid growth of the field of information retrieval is a result of the world's digitization. To stay informed, people use a range of resources. People like information that is brief and to the point. There is a huge issue with reading news stories and internet evaluations or feedbacks that makes it difficult to form conclusions unless they are read entirely. As a result, the concept of text summarization emerges as a means of improving information retrieval.

7.1.2 User Classes and Characteristics

1. This is useful for generate document summary.

7.2 Assumptions and Dependencies

Assumptions:
We have used Python Technique.Input as Text as paragraph.
Dependencies:
We have used python libraries. Output to generate document summary.

7.3 Functional Requirements

Functinal Specification:

- The application is user friendly.
- It provides an easy interface to user.
- The accessibility or response time of the application should be fast.

Dependency and Constraints:

- End User application will be developed in Windows OS.
- All scripts shall be written in Python.
- Application design pattern shall be Singleton.

7.4 External Interface Requirements (If Any)

7.4.1 User Interfaces

Tkinter: Tkinter is Python's de-facto standard GUI (Graphical User Interface) package. Tools in Tkin- ter: Window: It refers to a rectangular area somewhere on the user's display screen. Top level window: A window that exists independently on the screen. It will be decorated with the standard frame and controls for the desktop manager. It can be moved around the desktop, and can usually be resized.

7.4.2 Hardware Interfaces

System Processors : Core2Duo
Speed : 2.4 GHz
Hard Disk : 150 GB

7.4.3 Software Interfaces

Operating system : 64 bit Windows 10.
Coding Language : Python
Design constraints : Spyder and Pycharm.

7.5 Nonfunctional Requirements

Non-functional Requirements Performance Requirement-

7.5.1 Performance Requirements

- The performance of the functions and every module must be well.
- The overall performance of the software will enable the users to work efficiently.
- Performance of detect Depression level should be fast.
- Performance of the providing virtual environment should be fast.

7.5.2 Safety Requirements

The application is designed in modules where errors can be detected and fixed easily. This makes it easier to install and update new functionality if required.

7.5.3 Software Quality Attributes

Software has many quality attribute that are given below:-

1. **Adaptability:** This software is adaptable by all users.
2. **Availability:** This software is freely available to all users. The availability of the software is easy for everyone.
3. **Maintainability:** After the deployment of the project if any error occurs then it can be easily maintained by the software developer.
4. **Reliability:** The performance of the software is better which will increase the reliability of the Soft- ware.
5. **User Friendliness:** Since, the software is a GUI application; the output generated is much user friendly in its behavior.
6. **Integrity:** Integrity refers to the extent to which access to software or data by unauthorized persons can be controlled.
7. **Security:** Users are authenticated using many security phases so reliable security is provided.
8. **Testability:** The software will be tested considering all the aspects.

7.6 Analysis Models: SDLC Model to be applied

The Waterfall Model is a step-by-step approach to software development. Each stage of the SDLC phase is intended to accomplish a specific goal. Winston Royce first presented it in 1970. This is how we will use it in our project. Because it is simple, straightforward, and uncomplicated, this paradigm is simple to understand and apply. It's simple to maintain because the model is tight, with clear deliverables and a review procedure for each phase. The waterfall paradigm works well for smaller projects with welldefined and well-known requirements.

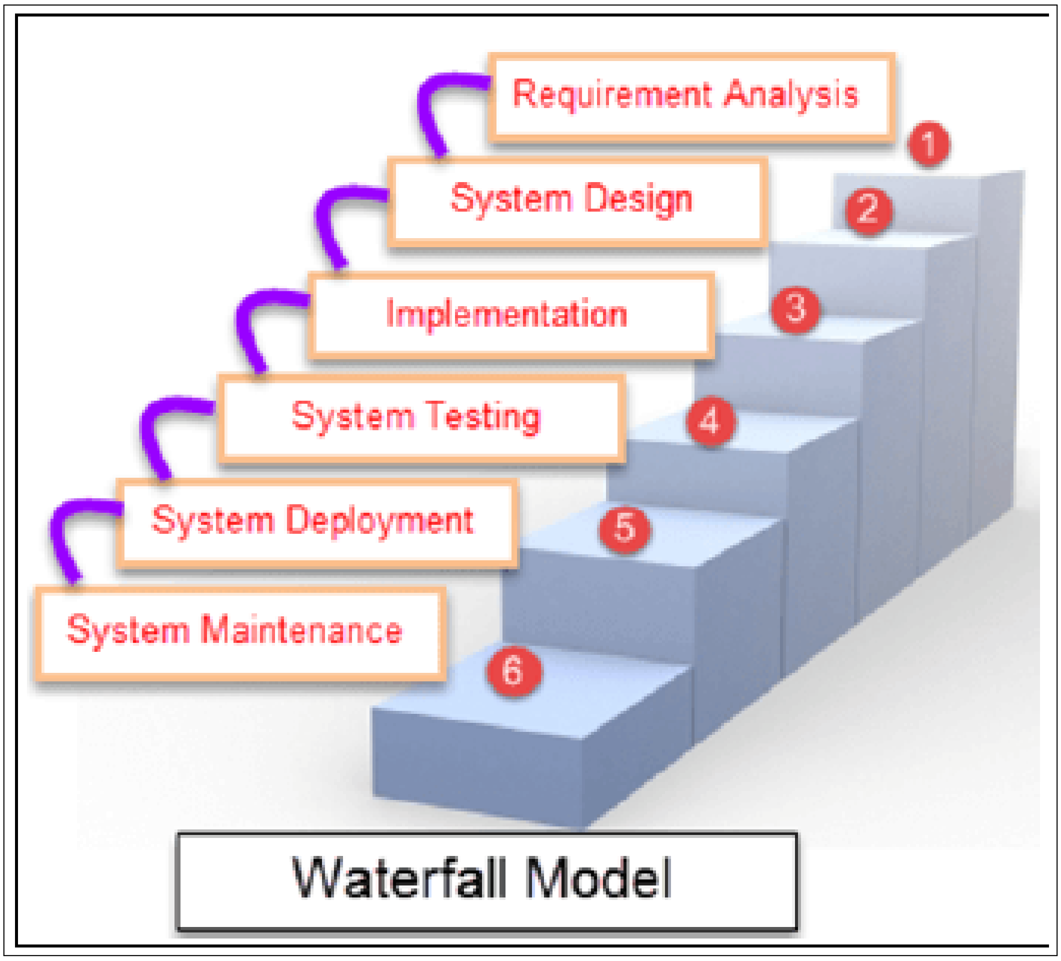


Figure 7.1: SDLC Model Diagram

7.7 System Implementation plan

SR.NO	TASK	DURATION (Days/Months)	Start Date	End Date
1	Group Formation	6days	31/07/2022	05/08/2022
2	Decide Area of Interest	3days	07/09/2022	09/09/2022
3	Search Topic	3days	09/09/2022	09/09/2022
4	Topic Selection	3days	07/09/2022	09/09/2022
5	Sanction Topic	3days	07/09/2022	09/09/2022
6	Search Related Information	2days	09/09/2022	10/09/2022
7	Understanding Concept	12days	12/09/2022	23/09/2022
8	Search Essential Document (IEEE White Paper, Software)	2days	09/09/2022	10/09/2022
9	Problem Definition	6days	15/09/2022	20/09/2022
10	Literature Survey	11days	15/09/2022	25/09/2022
11	SRS	8days	17/09/2022	24/09/2022
12	Project Planning	2month 4days	07/09/2022	10/11/2022
13	Modelling design	2month 4days	14/09/2022	17/11/2022
14	Technical Specification	22days	12/09/2022	03/10/2022
15	PPT	1month	10/10/2022	10/11/2022

Table 7.1: Project Plan

Chapter 8

System Design

8.1 System Architecture

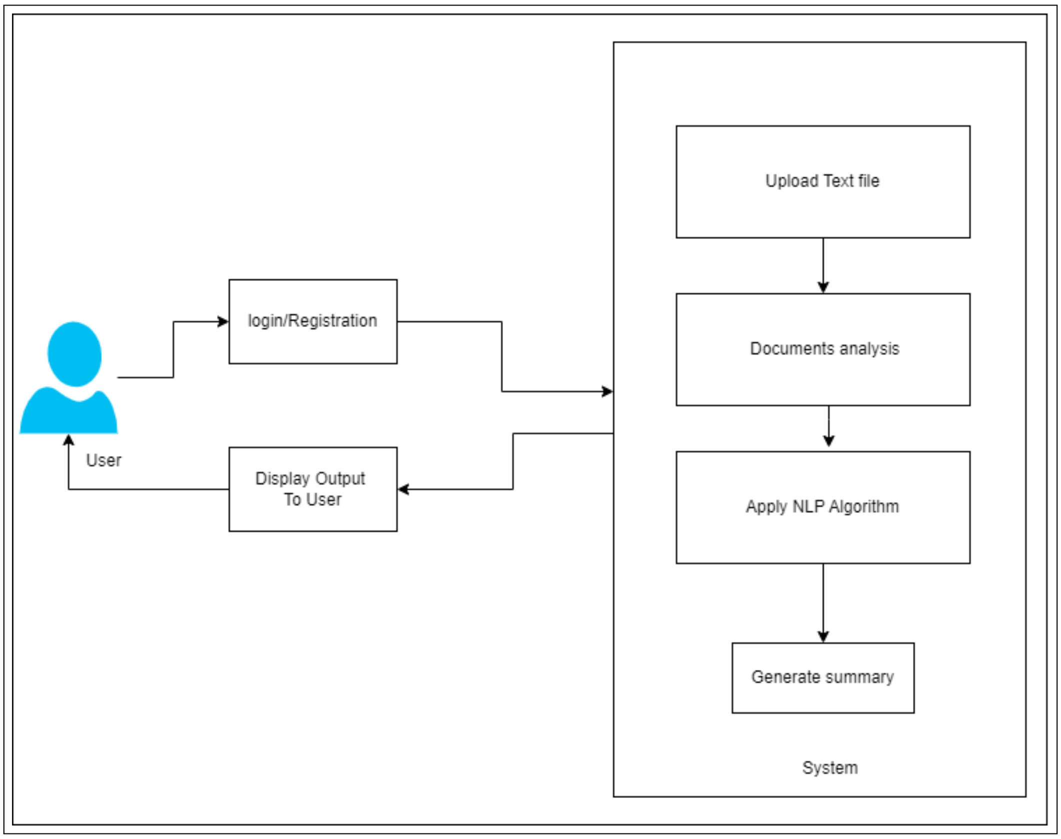


Figure 8.1: System Architecture

8.2 Data Flow Diagram

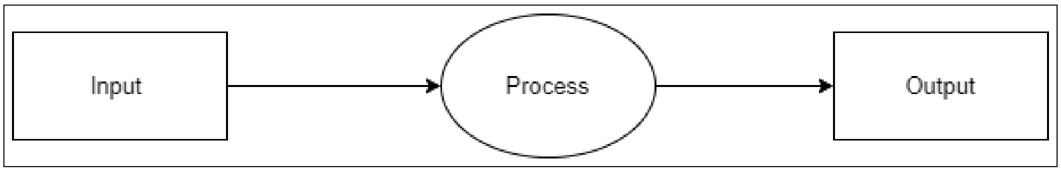


Figure 8.2: Project Plan

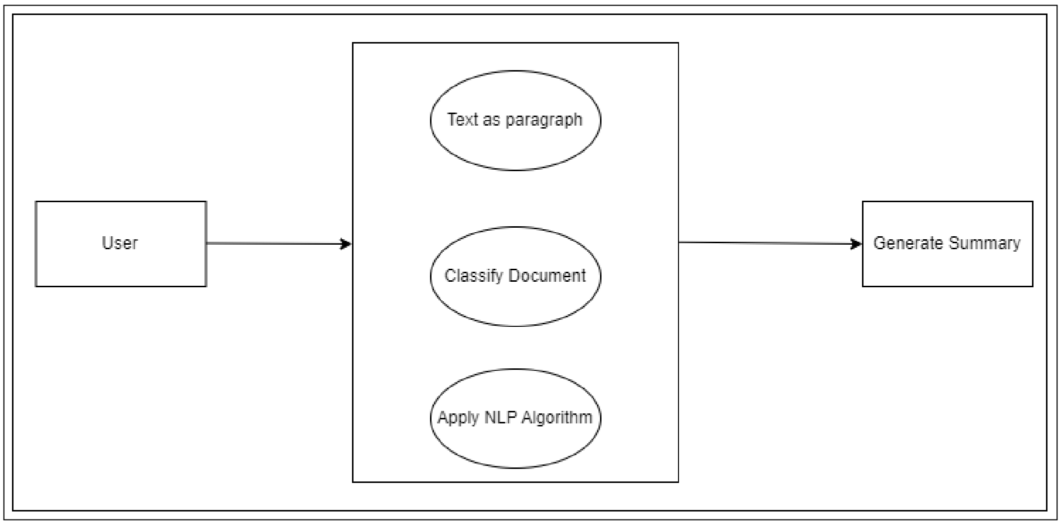


Figure 8.3: Project Plan

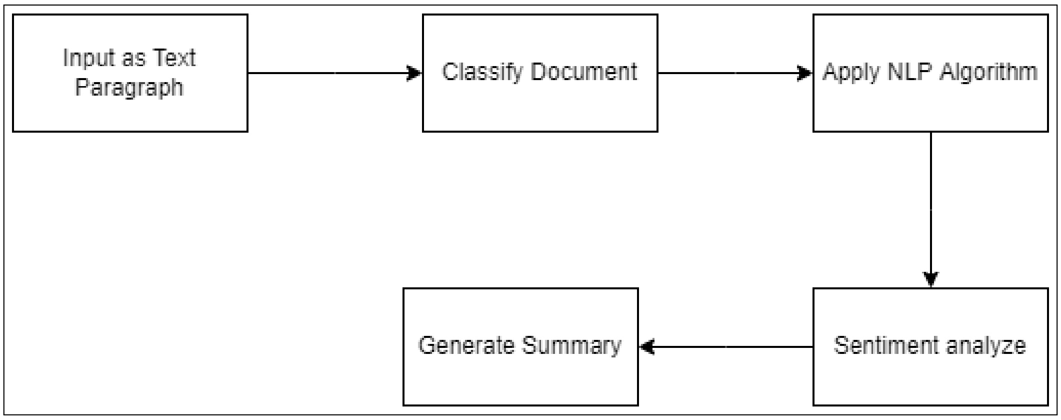


Figure 8.4: Project Plan

8.3 UML Diagram

The Unified Modeling Language (UML) is a programming language that is used to create software blueprints. The UML can be used to visualise, specify, build, and document software-intensive system artefacts. Although UML is process independent, it is best used in processes that are use case driven, architecture-centric, iterative, and incremental. There are a limited number of UML Diagrams available.

Use case Diagram.
Activity Diagram.
Sequence Diagram.
Class Diagram.

8.3.1 Use Case Diagram:

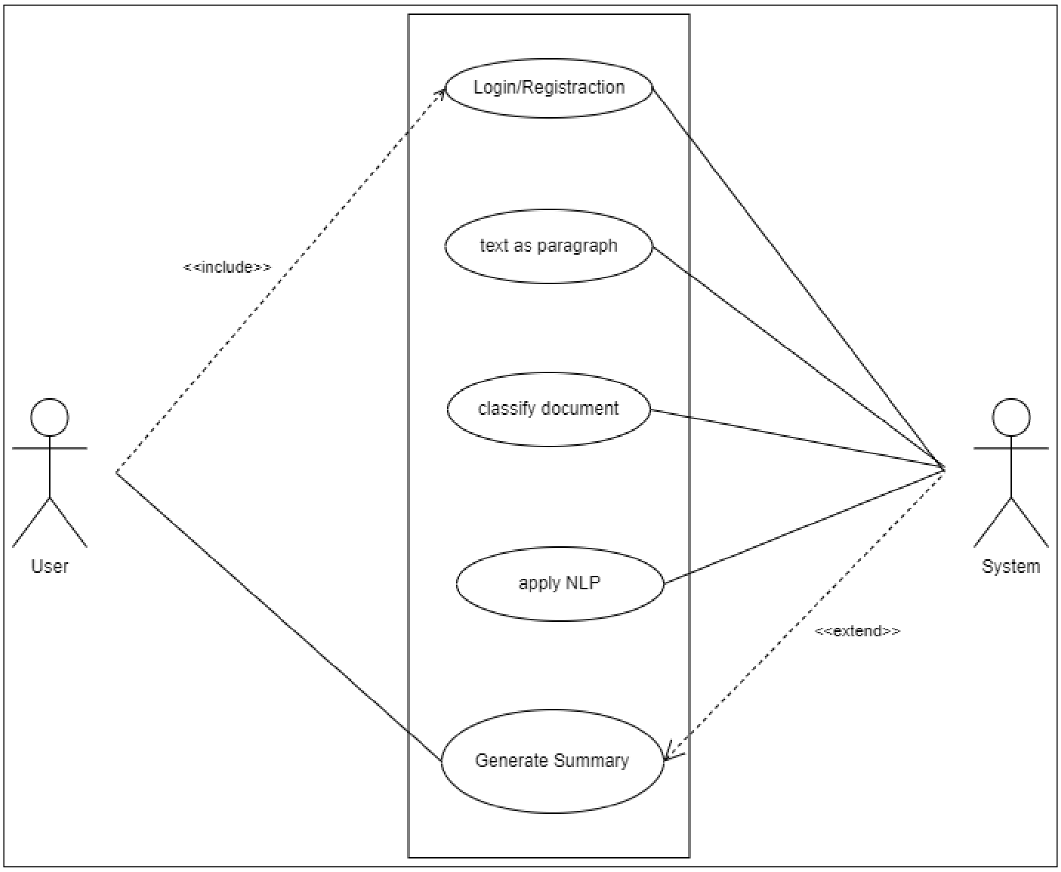


Figure 8.5: Use Case

8.3.2 Activity Diagram:

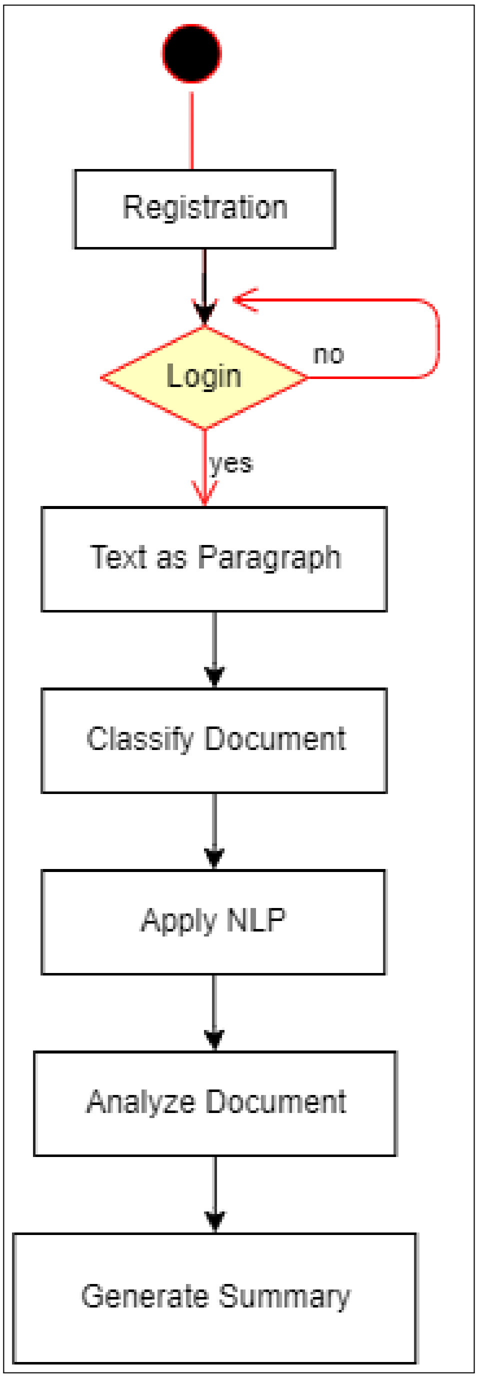


Figure 8.6: Activity Diagram

8.3.3 Sequence Diagram:

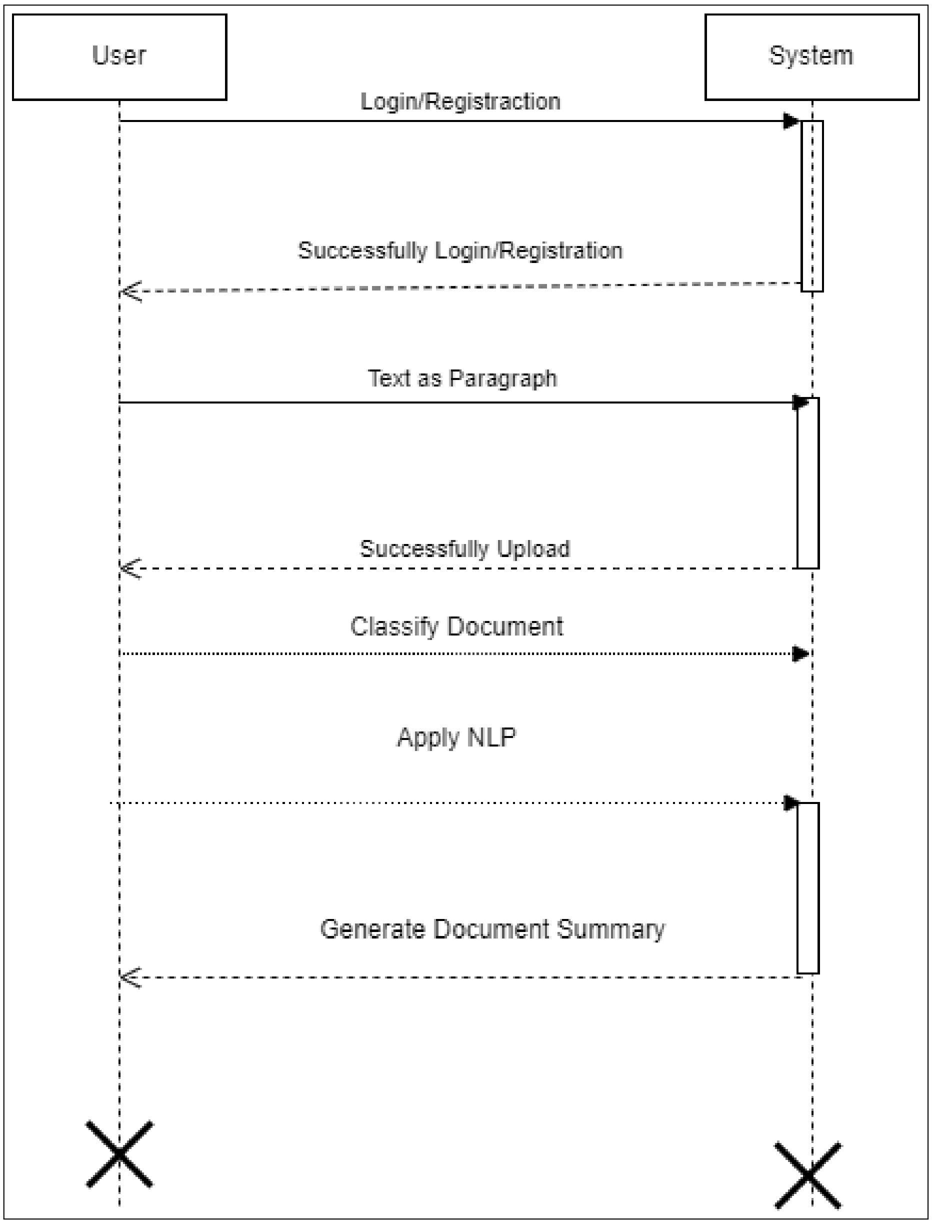


Figure 8.7: Sequence Diagram

8.3.4 Class Diagram:

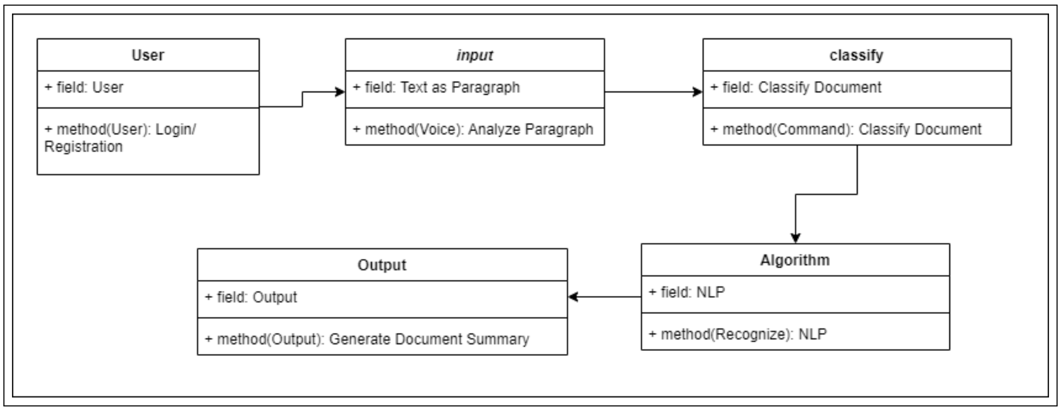

User
+ field: User
+ method(User): Login/ Registration
input
+ field: Text as Paragraph
+ method(Voice): Analyze Paragraph
classify
+ field: Classify Document
+ method(Command): Classify Document
Algorithm
+ field: NLP
+ method(Recognize): NLP
Output
+ field: Output
+ method(Output): Generate Document Summary

Figure 8.8: Class Diagram

Chapter 9

Project Implementation

9.1 Introduction

Automatic Document Summarization is the task of rewriting a document into its shorter form while still retaining its important content. The most popular two paradigms are extractive approaches and abstractive approaches. Extractive approaches generate summaries by extracting parts of the original document (usually sentences), while abstractive methods may generate new words or phrases which are not in the original document.

9.2 Tools and Technologies Used

Operating system : Windows 10.
Coding Language : Python.
IDE : Spyder.
Database : SQLite
Technologies Used
Anaconda Navigator(Spyder)
Libraries are used:

1. Keras
2. pillow
3. Tensorflow
4. Matplotlib
5. opencv

9.3 Methodologies/Algorithm Details

9.3.1 NLP Algorithm Used:

NLP algorithms are typically based on machine learning algorithms. Instead of hand-coding large sets of rules, NLP can rely on machine learning to automatically learn these rules by analyzing a set of examples (i.e. a large corpus, like a book, down to a collection of sentences), and making a statistical inference.
Step 1: Sentence segmentation
Step 2: Word tokenization
Step 3: Stop word analysis

Chapter 10

Test Cases

10.1 GUI Testing

Test Case	**Login Screen- Sign up**
Objective	Click on sign up button then check all required/mandatory fields with leaving all fields blank
Expected Result	All required/ mandatory fields should display with symbol”*”. Instruction line ”* field(s) are mandatory” should be displayed
Test case	Create a Password $\gg TextBox$ $ConfirmPassword \gg TextBox$
Objective	Check the validation message for Password and COnfirm Password field
Expected Result	Correct validation message should be displayed accordingly or ”Password and confirm password should be same” in place of ”Password mismatch”.

Table 10.1: GUI test case

10.2 Login Test Case

Test Case ID	**Test Case**	**Test Case I/P**	**Actual Result**	**Expected Result**	**Test Case Criteria (P/F)**
001	Enter the Wrong username or password click on submit button	Username or password	Error comes	Error should come	P
002	Enter the correct username and password click on submit button	username and password	Accept	Accept	P

Table 10.2: Login test case

10.3 Registration Test Case

Test Case ID	Test Case	Test Case I/P	Actual Result	Expected Result	Test Case Criteria (P/F)
001	Enter the number in username, middle name, last name in field	Number	Error comes	Error should comes	P
001	Enter the character in username, middle name, last name field	Character	Accept	Accept	P
002	Enter the invalid email id format in mail id field	kkgmail.com	Error comes	Error should comes	P
002	Enter the valid email id format in email id field	kk@gmail.com	Accept	Accept	P
003	Enter the invalid digit no in phone no field	99999	Error comes	Error should comes	P
003	Enter the 10 digit no in phone no field	9999999999	Accept	Accept	P

Table 10.3: Registration test case

10.4 System Test Cases

Test Case ID	**Test Case**	**Test Case I/P**	**Actual Result**	**Expected Result**	**Test Case Criteria (P/F)**
001	Store Xml file	Xml file	Xml file store	Error should come	P
002	Parse the xml file for conversion	Parsing	File got parse	Accept	P
003	Attribute identification	Check individual attribute	Identify attributes	Accepted	P
004	Weight analysis	Check weight	Analyze weight of individual attribute	Accepted	P
005	Tree formation	Form them tree	Formation	Accepted	P
006	Cluster evaluation	Check evaluation	Should check cluster	Accepted	P
007	Algorithm performance	Check evaluation	Should work algorithm properly	Accepted	P
008	Query formation	Check query correction	Should check query	Accepted	P

Table 10.4: System test case

Chapter 11

Result

11.1 GUI Main Page

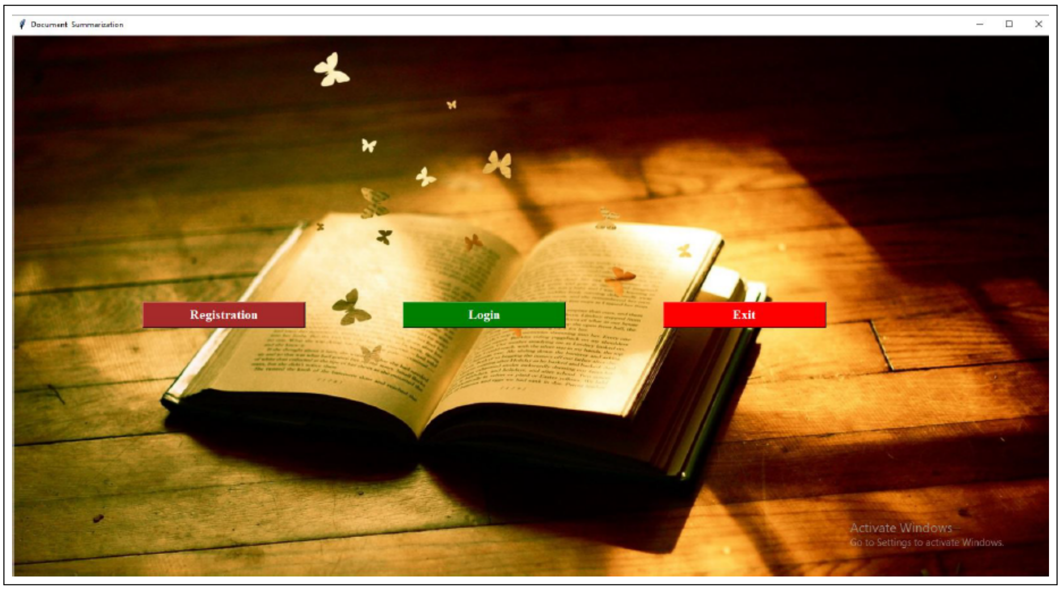
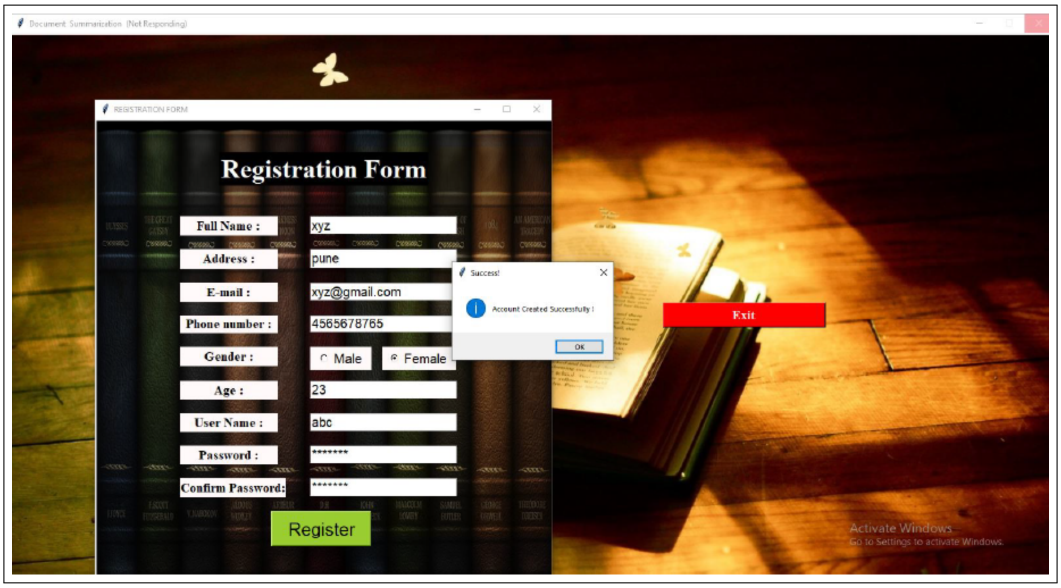


Figure 11.1: GUI Main Page

11.2 Registration


Document Summarization (Not Responding)
REGISTRATION FORM
Register
Exit
Success!
OK
Activate Windows
Go to Settings to activate Windows.

Registration Form

Full Name : xyz

Address : pune

E-mail : xyz@gmail.com

Phone number : 4565678765

Gender : ○ Male ● Female

Age : 23

User Name : abc

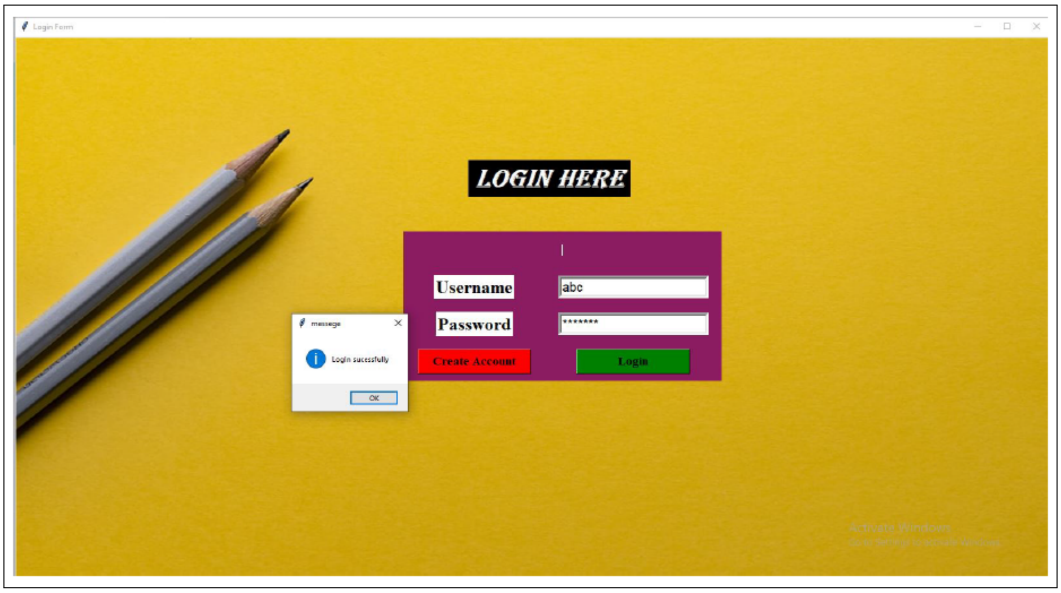
Password : *******

Confirm Password: *******

Account Created Successfully !

Figure 11.2: Registration

11.3 Login Page



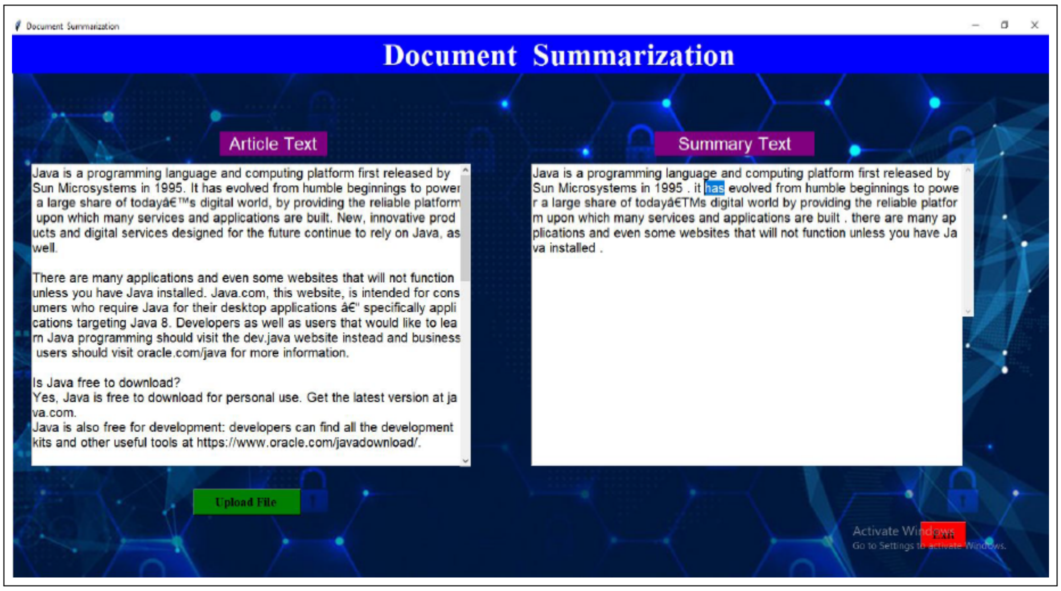
Username abc

Password *******

Login sucessfully

Figure 11.3: Login Page

11.4 Result



Java is a programming language and computing platform first released by Sun Microsystems in 1995. It has evolved from humble beginnings to power a large share of todayâ€™s digital world, by providing the reliable platform upon which many services and applications are built. New, innovative products and digital services designed for the future continue to rely on Java, as well.

There are many applications and even some websites that will not function unless you have Java installed. Java.com, this website, is intended for consumers who require Java for their desktop applications â€" specifically applications targeting Java 8. Developers as well as users that would like to learn Java programming should visit the dev.java website instead and business users should visit oracle.com/java for more information.

Is Java free to download?
Yes, Java is free to download for personal use. Get the latest version at java.com.
Java is also free for development: developers can find all the development kits and other useful tools at https://www.oracle.com/javadownload/.

Java is a programming language and computing platform first released by Sun Microsystems in 1995 . it has evolved from humble beginnings to power a large share of todayâ€TMs digital world by providing the reliable platform upon which many services and applications are built . there are many applications and even some websites that will not function unless you have Java installed .

Figure 11.4: Result

Chapter 12

Deployment and Maintenance

12.1 Installation and un-installation

Anaconda Navigator Installtion step:

1. Go to the Anaconda Website and choose a Python 3.
2. Locate your download and double click it.
3. Read the license agreement and click on I Agree.
4. Click on Next.
5. Note your installation location and then click Next.
6. This is an important part of the installation process.
7. Click on Next.
8. then you install the libraries pillow,opencv,keras,matplotlib,tensorflow

Anaconda Navigator Uninstalltion step:

1. go to the control panel select the anaconda navigator then Uninstall.

Chapter 13

Other Specifications

13.1 Advantages

1. Language fluency: Abstractive summarization can generate summaries that are more fluent and coherent than extractive summarization, which often produces disjointed and fragmented summaries.

2. Reducing the length of documents: Abstractive summarization can condense lengthy documents into shorter summaries, making it easier for readers to understand and comprehend the key points without having to read through the entire document.

3. Capturing the essence of the document: Abstractive summarization can capture the most important information and ideas contained in a document, providing a more complete and accurate representation of the content.

4. Handling new and unique content: Unlike extractive summarization, which can only reproduce content that is already present in the source document, abstractive summarization can generate summaries of new and unique content.

5. Customization: Abstractive summarization can be customized to produce summaries that are tailored to specific audiences or applications. This can be useful for generating summaries for different domains or for different languages.

13.2 Applications

1. News summarization: News organizations can use abstractive document summarization to generate short summaries of news articles for readers who are short on time or who want a quick overview of a story.

2. Legal document summarization: Law firms and legal departments can use abstractive document summarization to summarize lengthy legal documents, such as contracts, briefs, and court opinions, to quickly extract the key points.

3. Scientific paper summarization: Researchers can use abstractive document summarization to generate summaries of scientific papers, allowing them to quickly review a large number of papers and identify the ones that are most relevant to their research.

4. Business document summarization: Companies can use abstractive document summarization to summarize business reports, presentations, and proposals, helping executives quickly understand the main points without having to read through the entire document.

Chapter 14

Conclusion and Future Scope

14.1 Conclusion

Autonomy abstracted Document summarising is a multi-step process with several sub-tasks. Every subtask has the ability to generate high-quality summaries. Abstractive document summarization requires identifying necessary paragraphs from a given document. We suggested abstractive based text summary in this research, based on a statistical new approach based on sentence ranking, in which the summarizer selects phrases based on their rating. Abstract sentences are created as a summarised text that is then turned to audio. The proposed model outperforms the old technique in terms of accuracy.

14.2 Future Scope

We plan to extend the present work and analyze it by incorporating other features in addition to sentiments. The present method does not consider neutral sentiments. We plan to extend the work to use neutral sentiments in future. We wish to analyze sentences in more structured format, wherein the text.opinion is having various parts like name, features, opinions, relations, etc. which converts the unstructured text to a structured form.

Chapter 15

APPENDIX- B

(a) E. Sandhaus, “The New York Times annotated corpus.” Linguistic Data Consortium, Philadelphia 6.12, 2008, Art. no. e26752.

(b) C. Napoles, M. Gormley, and B. Van Durme, “Annotated gigaword,” in Proc. Joint Workshop Autom. Knowl. Base Construction Web-Scale Knowl. Extraction, 2012, pp. 95–100.

(c) M. Grusky, M. Naaman, and Y. Artzi, “Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Human Lang. Technol., 2018, pp. 708–719.

(d) A. Cohan et al., “A discourse-aware attention model for abstractive summarization of long documents,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Human Lang. Technol., 2018, pp. 615–621.

(e) S. Subramanian, R. Li, J. Pilault, and C. Pal, “On extractive and abstractive neural document summarizationwith transformer language models,” in Proc. 2020 Conf. Empirical Methods Natural Lang. Process., 2019, pp. 9308–9319.

Chapter 16

References

(a) R. Socher, “Boiling the information ocean,” 2020. [Online]. Available: http://tiny.cc/45ohlz

(b) S. Chopra, M. Auli, and A. M. Rush, “Abstractive sentence summarization with attentive recurrent neural networks,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Human Lang. Technol., 2016, pp. 93–98.

(c) R. Nallapati, B. Zhou, C. dos Santos, C. Gulcehre, and B. Xiang, “Abstractive text summarization using sequence-to-sequence RNNs and beyond,” in Proc. SIGNLL Conf. Comput. Natural Lang. Learn. Stroudsburg, PA, USA, 2016, pp. 280–290.

(d) A. See, P. J. Liu, and C. D. Manning, “Get to the point: Summarization with pointer-generator networks,” in Proc. Annu. Meet. Assoc. Comput. Linguist., 2017, pp. 1073–1083.

(e) R. Paulus, C. Xiong, and R. Socher, “A deep reinforced model for abstractive summarization,” in Proc. Int. Conf. Learn. Representations, 2018.

(f) Y. Liu and M. Lapata, “Text summarization with pretrained encoders,” in Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process., 2019, pp. 3721–3731.

(g) K. Song, X. Tan, T. Qin, J. Lu, and T. Y. Liu, “MASS: Masked sequence to sequence pre-training for language generation,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 5926–5936.

(h) L. Dong et al., “Unified language model pre-training for natural language understanding and generation,” inAdv. Neural Inform. Process. Syst., 2019, pp. 13 042–13 054.

(i) Y. Yan et al., “ProphetNet: Predicting future N-gram for sequencetosequence pre-training,” 2020, arXiv:2001.04063.

(j) K. M. Hermann et al., “Teaching machines to read and comprehend,” in Adv. Neural Inform. Process. Syst., 2015, pp. 1693–1701.

Chapter 17

Information of Project Group Members

(a) Name : Purushottam Ramdas Nimje

(b) Date of Birth : 28/06/2001

(c) Gender : Male

(d) Permanent Address : Plot No.55, Besa Vela Harishchandra Road, Behind Podar International School, Kh.No-191, My Dreamland Society, Nagpur, Maharashtra -440037

(e) E-Mail : purushottamnimje286@gmail.com

(f) Mobile/Contact No. : 9503802874

(g) Placement Details : No

(h) Paper Published : Yes

(a) Name : Mayur Anil Kardile

(b) Date of Birth : 01/09/2000

(c) Gender : Male

(d) Permanent Address : At Post Saradwadi, Taluka Shirur, District Pune

(e) E-Mail : mayurkardile2018@gmail.com

(f) Mobile/Contact No. : 9579458599

(g) Placement Details : No

(h) Paper Published : Yes

(a) Name : Pratik Bachche

(b) Date of Birth : 20/09/1996

(c) Gender : Male

(d) Permanent Address : Room No.05, Ram Naresh Dubey Chawl, Nivetiya Compound, Nivetiya Road, Malad East, Mumbai-400097

(e) E-Mail : pratiknb2027@gmail.com

(f) Mobile/Contact No. : 8850227674

(g) Placement Details : No

(h) Paper Published : Yes

(a) Name : Anuja Prakash Parve

(b) Date of Birth : 11/10/2001

(c) Gender : Female

(d) Permanent Address : Ambedkar Nagar Jintur Road near MSEB
Parbhani-431401

(e) E-Mail : anujaparve11@gmail.com

(f) Mobile/Contact No. : 8080936491

(g) Placement Details : No

(h) Paper Published : Yes