Project Report

on

Automatic Summarization of User Reviews

In partial fulfillment of requirements for the degree

of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

Submitted by:

Piyush Mali [19100BTIT06588]

Raghav Sood [19100BTIT06595]

Rishika Jain [19100BTIT06604]

Under the guidance of

Prof. Sujit K Badodia
Prof. Manorama Chouhan

DEPARTMENT OF INFORMATION TECHNOLOGY
SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY
SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA, INDORE

JULY-DEC 2022

SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA, INDORE
SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

DECLARATION

We here declare that work which is being presented in the project entitled "**Automatic Summarization of User Reviews**" in partial fulfillment of degree of **Bachelor of Technology in Information Technology** is an authentic record of our work carried out under the supervision and guidance of **Prof. Sujit K Badodia** and **Prof. Manorama Chouhan** Asst. Professor of Information Technology. The matter embodied in this project has not been submitted for the award of any other degree.

Student 1Signature

Student 2 Signature

Student 3Signature

Date:

SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA, INDORE
SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

PROJECT APPROVAL SHEEET

Following team has done the appropriate work related to the "**Automatic Summarization of User Reviews**" in partial fulfillment for the award of **Bachelor of Technology in Information Technology** of "SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY" and is being submitted to SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA, INDORE.

Team:
1. Piyush Mali (19100BTIT06588)
2. Raghav Sood (19100BTIT06595)
3. Rishika Jain (19100BTIT06606)

Internal Examiner **External Examiner**

SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA, INDORE
SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that **Piyush Mali, Raghav Sood and Rishika Jain** working in a team have satisfactorily completed the project entitled "**Automatic Summarization of User Reviews**" under the guidance of **Prof. Sujit K Badodia** and **Prof. Manorama Chouhan** in the partial fulfillment of the degree of **Bachelor of Technology in Information Technology** awarded by SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY affiliated to SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA, INDORE during the academic year **July2019-Dec 2023**.

Prof. Sujit K Badodia
Prof. Manorama Chouhan
Project Guide

Prof. Rahul Choudhary
Project Coordinator

Dr. Jigyasu Dubey
Head, Department of Information Technology

ACKNOWLEDGEMENT

We are grateful to a number of persons for their advice and support during the time of complete our project work. First and foremost, our thanks goes to **Dr. Jigyasu Dubey** Head of the Department of Information Technology and **Prof. Sujit K Badodia** and **Prof. Manorama Chouhan** the mentor of our project for providing us valuable support and necessary help whenever required and also helping us explore new technologies by the help of their technical expertise. His direction, supervision and constructive criticism were indeed the source of inspiration for us.

We would also like to express our sincere gratitude towards our Director **Dr. Anand Rajavat** for providing us valuable support.

We are really indebted to **Prof. Rahul Choudhary,** project coordinator for helping us in each aspect of our academic's activities. We also owe our sincere thanks to all the **faculty members** of Information Technology Department who have always been helpful.

We forward our sincere thanks to all **teaching and non-teaching staff** of Information Technology department, SVVV Indore for providing necessary information and their kind co-operation.

We would like to thanks our parents and family members, our classmates and our friends for their motivation and their valuable suggestion during the project. Last, but not the least, we thank all those people, who have helped us directly or indirectly in accomplishing this work. It has been a privilege to study at SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA, INDORE.

ABSTRACT

Text summarization is a technique for creating a precise, concise summary of long texts while concentrating on the passages that convey important information and maintaining the overall meaning. The goal of automatic text summarization is to reduce lengthy articles into shorter versions because doing it manually would be time-consuming and expensive. Machine learning algorithms may be used to examine papers and find the sections that contain pertinent facts and information before producing the necessary summary paragraphs.

The "Automatic Summarization of User Reviews" project serves as a template for creating a summary of all customer reviews. Because there is a wealth of material available online for any topic, condensing the pertinent data into a summary would benefit both producers and a large number of other users. Our model is required since there are more people shopping online and more options available to them. Before making a purchase, the user needs to know how reliable the product was when it was utilized by real customers.

Our programmed uses all of the product reviews as input to provide a succinct summary that closely follows the flow of all the evaluations. The reviews are first divided into sentences. It has been preprocessed to eliminate superfluous punctuation and leave just words with significance. Sentences are given relative weighting based on certain common characteristics, and evaluations are presented and categorized based on the salient scores assigned to each of the sentences.

LIST OF FIGURES

TABLE OF CONTENT

1. INTRODUCTION

1.1 Introduction

In a world where the internet is booming with vast volumes of data every day, the ability to automatically summarize data is a critical issue. Summaries of lengthy papers, news articles, and even dialogues can improve and speed up how much material we consume. Natural Language Processing (NLP), which has attracted a lot of interest in recent years, has generated a lot of interest in Automatic Text Summarizations.[1] Massive data transmission and data collection have suddenly entered our society. According to a survey by International Data Corporation (IDC), the amount of data generated online is predicted to rise from 4.4 ZB in 2013 to 180 ZB in 2025. That's a ton of information! IDC predicts that by 2025, there will be 180 zettabytes of digital data generated year worldwide, up from 4.4 zettabytes in 2013. That's a ton of details! IDC predicts that by 2025, there will be 180 zettabytes of digital data flying annually around the globe, up from 4.4 zettabytes in 2013.[2] That's a lot of details! ML algorithms that can automatically summarize lengthy texts and provide accurate summaries that elegantly express the intended information are required because there is so much data generated and migrating online.

What is Natural Language Processing (NLP)?

A subset of AI is NLP. that manages interactions between computers and people using natural language. The ultimate goal of NLP is to effectively read, decode, comprehend, and comprehend human language. [1] To extract meaning from human language, the majority of NLP approaches rely on machine learning. The size of the global natural language processing market is anticipated to reach $29.5 billion by 2025, growing at a market growth rate of 20.5% CAGR throughout the projected period.[3][4] Natural language processing employs a variety of ways for deciphering human language, from algorithmic and rules-based methods to statistical and machine learning techniques. Because text- and voice-based data, like actual applications, vary greatly, a wide variety of techniques are required.[5] Tokenization, sorting, lemmatization/stemming, voice tagging, language detection, and semantic link identification are all fundamental NLP activities.

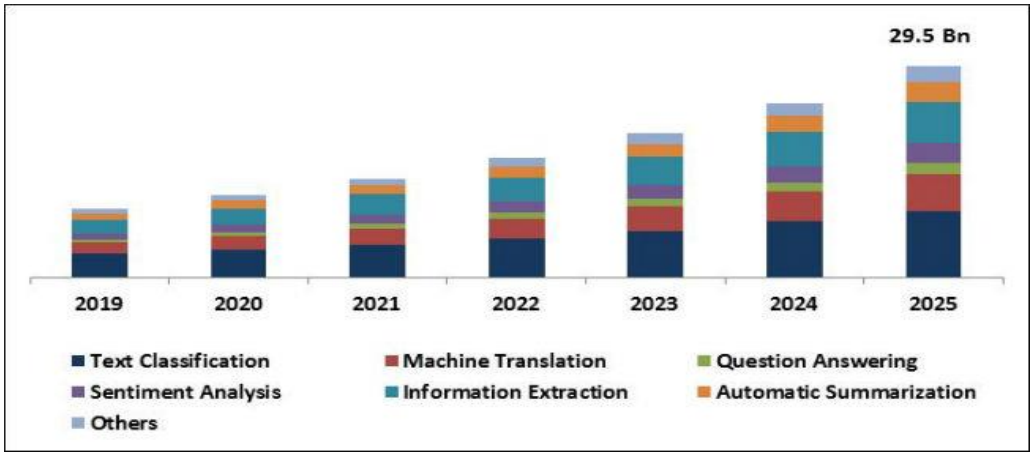


Fig 1.1 Global Natural Language Processing Market Size

1.2 Problem Statement

- Using NLP (Natural Language Processing) techniques, the challenge is to create and construct a model that may automatically summarize customer reviews for a product from an e-commerce website. The model must also be evaluated for correctness.
- NLP is challenging because of the complexity of processing human language. Natural language communication follows principles that are difficult for a machine to comprehend.
- Some of these laws may seem abstract and high-level, as when someone employs a sarcastic comment to convey information. However, some of these regulations may be of a low degree.

- Use the letter "s," for instance, to denote a number of objects. The use of machines to translate human language is fraught with numerous other issues.

- Due to the ambiguous nature of human language, there are no set, precise rules that can be taught to the computer. Instead, it is the ambiguity and imprecision of natural languages that make NLP challenging for machines to apply.

- Research and contrast the various text summarizing techniques. to find weaknesses in currently available text-summarization software. Create and put into use a model or tool for automatically summarizing user reviews.

1.3 Need for the proper System

Online has grown significantly. However, it can be challenging for marketers and business analysts to comprehend client concerns due to the sheer volume of customer evaluations that are placed on websites like Amazon.com. In this presentation, we outline a method for automatically summarising customer reviews and discuss the initial findings of our study on Amazon.com product reviews.We also evaluated those results on based on two metrics: Model Efficiency and Time efficiency. Our research, we hope, will advance the methods and comprehension of customer review summaries and will be advantageous to web marketers, business intelligence, and company owners alike. Study in the fields of e-commerce and text mining.

1.4 Objective

- Study and compare existing text summarization methods.
- To identify gaps in existing tools used for Automatic text summarization.
- Design and implement a tool or model for automatic summarization of user reviews.

1.5 Modules of the system

Extractive Summarization: In this process, we focus on the vital information from the input sentence and extract that specific sentence to generate a summary.[4] There is no generation of new sentences for summary, they are exactly the same that is present in the original group of input sentences.

Abstract Summarization: This is the opposite of Extractive summarization where it takes an exact sentence to generate a summary.[4] Abstract Summarization focuses on the vital information of the original group of sentences and generates a new set of sentences for the summary. This new sentence might not be present in the original sentence.

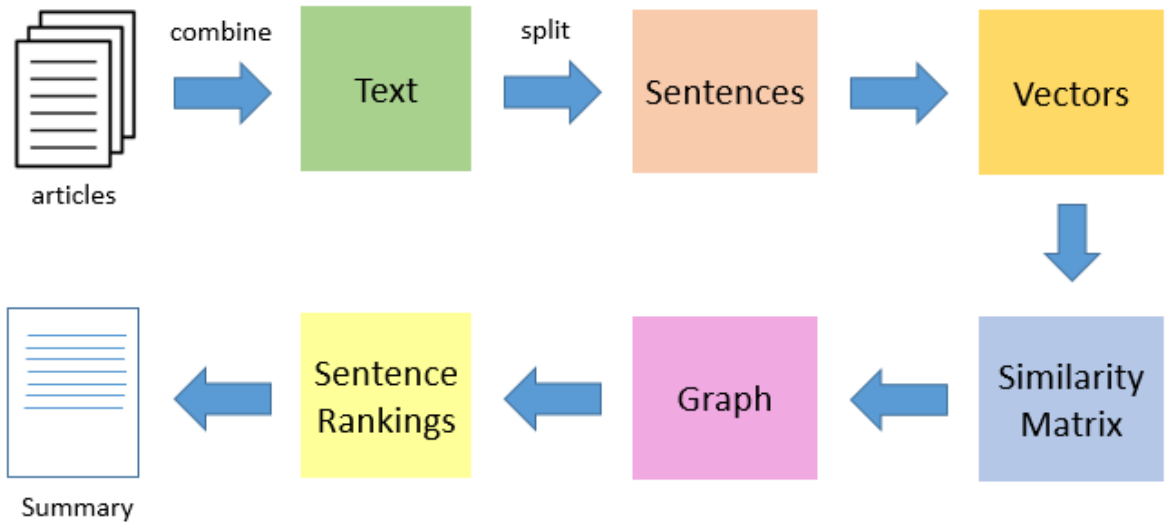


Fig 1.2 Text summarization

1.6 Scope

We know the Internet has increased user interaction; the number of consumer reviews written online has grown significantly. However, it can be challenging for marketers and business analysts to comprehend client concerns due to the sheer volume of customer evaluations that are placed on websites like Amazon.com. In this presentation, we outline a method for automatically summarising customer reviews and discuss the initial findings of our study on Amazon.com product reviews. We also evaluated those results on based on two metrics: Model Efficiency and Time efficiency. Our research, we hope, will advance the methods and comprehension of customer review summaries and will be advantageous to web marketers, business intelligence, and company owners alike. Study in the fields of e-commerce and text mining.

2. LITERATURE SURVEY

2.1 Existing System

Text summarization has a lot of research devoted to it, which has resulted in development of various techniques to do the same. The process of text summarization is broadly divided into two types – extractive and abstractive. Here, the review will be based on the extractive side of the process. Some of the most notable mentions include, Term Frequency-Inverse document frequency method, LSA based method, Cluster based method, Summarization using neural networks and fuzzy logic, machine learning approach, and graph-based approach. While the first method is built upon the idea of term frequencies to determine prospective words and phrase to be included in the summary, fuzzy logic does the same by using parameters of similarity to title, key-words and phrases and sentence length on the given text. Machine learning approaches focus on classifying prospective summary sentences, using different classifiers, like Naive Bayes classifier, while deep learning is used to train neural networks for finding out the same. Cluster Method goes a step further in producing summary for multi-documents on different themes.Finally, a graph-based method, after employing per-processing, pruning, stop word removal, gives a tree with sentences attached as nodes, and, through the weights of the edges, shows what sentences are more likely to be included in the summary

2.2 Proposed system

As the different techniques came out, each had its advantages and its disadvantages. Some performed better on particular domains, for example automobile articles, while others had a better overall performance. Keeping this in mind it was decided to study the effectiveness of different machine learning and deep learning models along with finding out which metrics are more influential in forming a summary. By finding what produces the most coherent summary through the research, it is aimed to aid future endeavors in the process of summarization and, subsequently, building a more accurate model. To train the model, a manually compiled dataset from different sources and domains was used. The decision to use of varied domains was an attempt to avoid bias towards particular topics and give the model equal exposure to different domains. The dataset comprises of individual sentences from each article and whether or not it was included in the summary.

2.3 Feasibility study

The Project" Automatic Summarization of User Reviews" aims to provide an efficient and enhanced summarization tool for the users that can be used to perform automatic text summarization of all the users to determine the reliability of a product based on the past experiences of the customers who have used the product previously.

Since there are a large number of reviews present for a product and a user does not have the time to go through all of them, our model will aim to summarize all the reviews and pick only the Top 10 (both negative and positive out of those.

This will help the user in making a faster decision. In this project, we'll tend to summarize the reviews posted by the users on the e-commerce giant Amazon for the mobile phones purchased by them through the website The dataset used will be Prompt cloud which contains approx. 4 lakhs reviews.

We'll be analyzing our results based on certain metrics like Model Efficiency and Time Efficiency. The accuracy of the summarizer will also be evaluated.

The mixed Reviews present in the dataset will first be segregated into positive and negative and then a score will be given to each of those positive and negative reviews. The Top 10 reviews with the highest score will be presented to the user based on which a user will be able to conclude its analysis and make an informed decision.

2.3.1 Technical Feasibility

Technical feasibility is a standard practice for companies to conduct feasibility studies before commencing work on a project. Businesses undertake a technical feasibility study to assess the practicality and viability of a product or service before launching it. Whether you are working as a product engineer, product designer or team manager, there may be plenty of situations in your career where you have to prepare a technical feasibility study. In this Project, we use Language: Python and Libraries are NLTK, Textblob, Vader, NumPy, Pandas, Networkx. And Algorithm used: TextRank: To Rank Sentences according to its importance in text, based on Google Page Rank Algorithm. And Glove Model: To Create Word Embeddings. We've used Stanford's GloVe 100d word embeddings for our project.

2.3.2 Economical Feasibility

Economic feasibility is the cost and logistic outlook for this project. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. The economical study analyses data to determine whether the cost ultimately profitable to the user. This project requires the User Reviews Data. So our Project" Automatic Summarization of User Reviews" aims to provide an efficient and enhanced summarization tool for the users that can be used to perform automatic text summarization of all the users to determine the reliability of a product based on the past experiences of the customers who have used the product previously. In this project, we'll tend to summarize the reviews posted by the users on the e-commerce giant Amazon for the mobile phones purchased by them through the website The dataset used will be Prompt cloud which contains approx. 4 lakhs reviews. Due to this, it is economically feasible.

2.3.3 Operational Feasibility

Assessing Operational feasibility is to gain an understanding of whether the proposed system is to solve the user problems, or take advantage of the opportunities or not. Is important to understand how the new systems will fetch into the current day-to-day operations of the organizations. Operational feasibility studies are generally utilized to process, Evaluation, Implementation, and Resistance. Python also enables developers to roll out programs and get

prototyping running, making the development process much faster. Once a project is on its way to becoming an Analytical tool or application, it can be ported to more sophisticated languages such as R or Java if necessary.

3. REQUIREMENT ANALYSIS

3.1Method used for Requirement analysis

While most participants completely agree the most for aspect-based summaries when we combined the ratings of completely agreed and agreed statistical summary where the most favored when the participants were asked to write a short summary of the rating based on provided criteria.

System runs on any regular PC/Laptop If the user intends on training the model on a different dataset, then a machine with 4 GB or more RAM and processor of 2.7 GHz or more is required. Commercial servers, workstations, and other high-end PCs may have more than one physical processor. Windows 7 Professional, Enterprise, and Ultimate allow for two physical processors, providing the best performance on these computers. Windows 7 Starter, Home Basic, and Home Premium will recognize only one physical processor.

3.2Data Requirements

The dataset obtained from Kaggle in prompt cloud repository which contains more than 4 lakhs reviews of cellphones bought on amazon by the users we'll look into it to find useful insights with Reviews, Price, and their Relationship

Field of Feature sets: -

- Product Title
- Brand
- Price
- Rating
- Review Text
- Number of People who find those reviews helpful

Dataset can be found from the link below:
https://www.kaggle.com/PromptCloudHQ/amazon-reviews- unlocked-mobile-phones

3.3Functional Requirements

- User will give a text file needed to be summarized, and the summary will be displayed in a new page.

- The model has to be trained on a set of documents along with their corresponding summaries. A summary will include all the sentences that give the main idea of the

text.

3.4Non-Functional Requirements

- Application must be simple and easy to use.
- Application must be intuitive and simple in the way it displays all relevant data and relationships.
- The application must inform the user in situations of a crash or error Should be up for working as and when required.

3.5 System Specification

3.5.1 Hardware specification

- System runs on any regular PC/Laptop.
- If the user intends on training the model on a different dataset, then a machine with 4 GB or more.
- RAM and processor of 2.7 GHz or more is required.

3.5.2 Software Specification

- Any modern web browser.
- Language Used: Python environment
- Libraries/Toolkits: NLTK, Sci-kit Learn , Keras , Pandas , Numpy
- Web frame Used: Flask

4. DESIGN

4.1 Software Requirements Specification

4.2.1 Supplementary Specifications

- **Tech Stack**

  Language: Python

  Libraries: NLTK, TextBlob, Vader, NumPy, Pandas, Networkx

- **Algorithm used:**

  TextRank: To Rank Sentences according to its importance in text, based on Google Page Rank Algorithm.

  Glove Model: To Create Word Embeddings. We've used Stanford's GloVe 100d word embeddings for our project.

4.2.2 Use Case Model

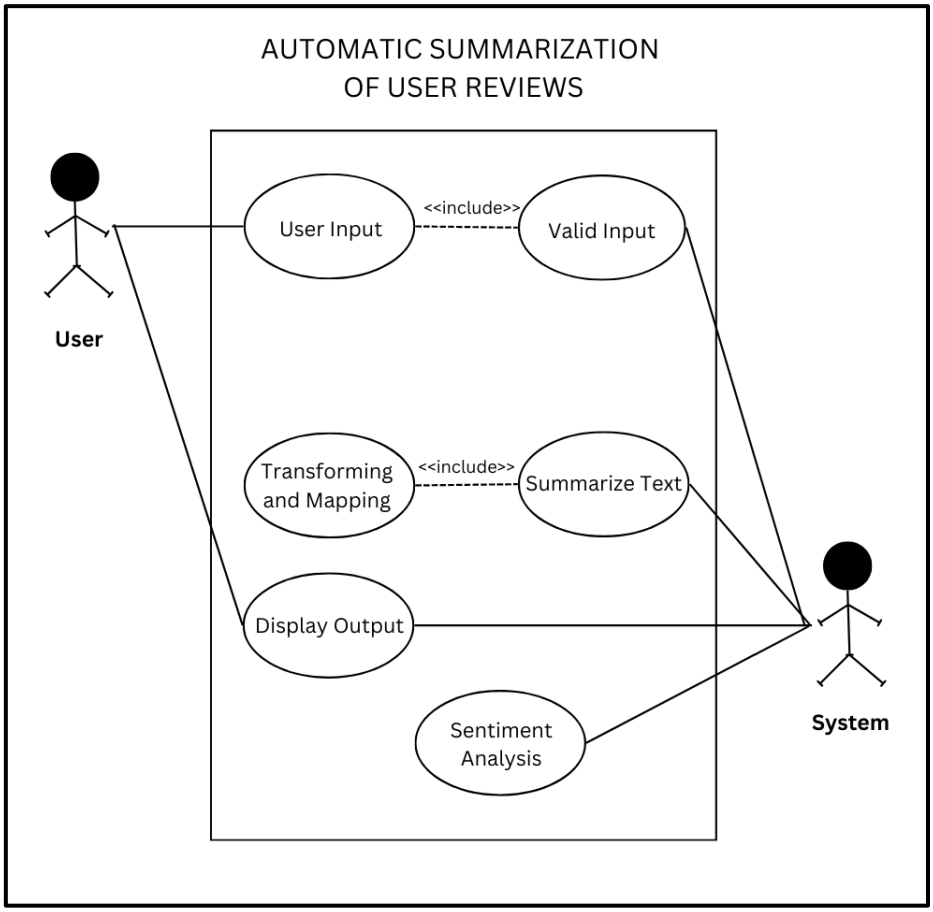


Fig 4.1.2 Use Case Model

4.2Conceptual diagram

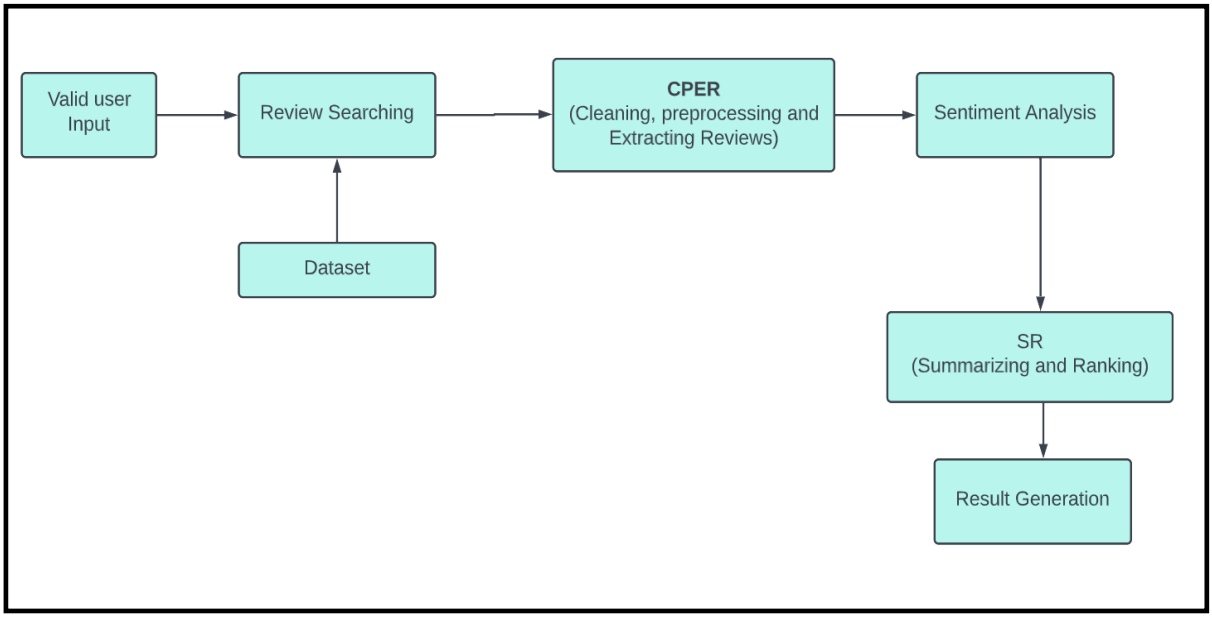


Fig 4.2 Conceptual Diagram

4.3Data flow Diagram (Level 0,1)

DFD Level-0

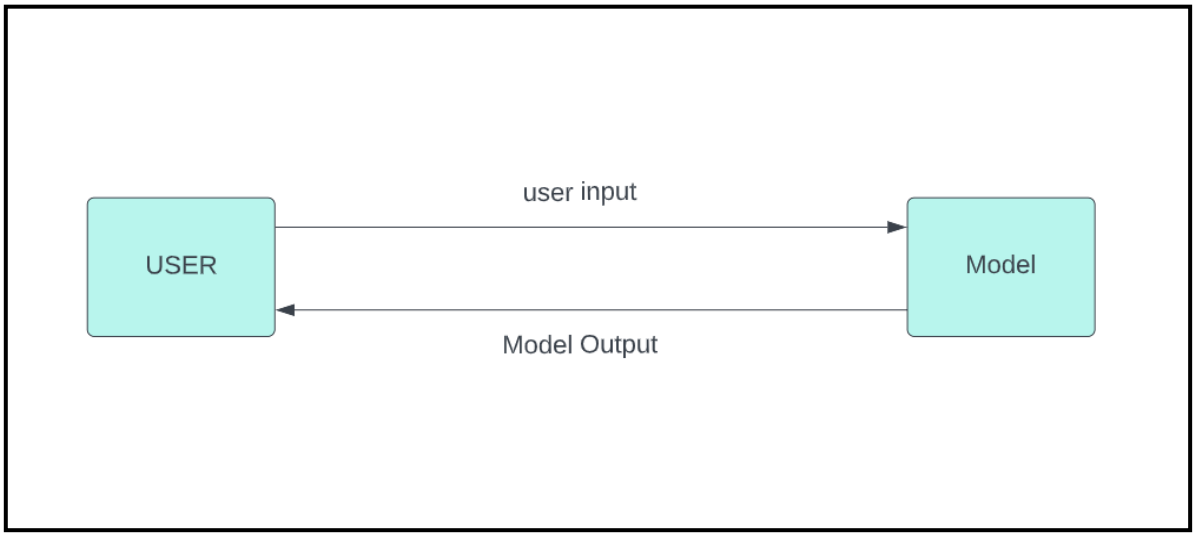


Fig 4.3 (a) DFD Level 0

DFD Level-1

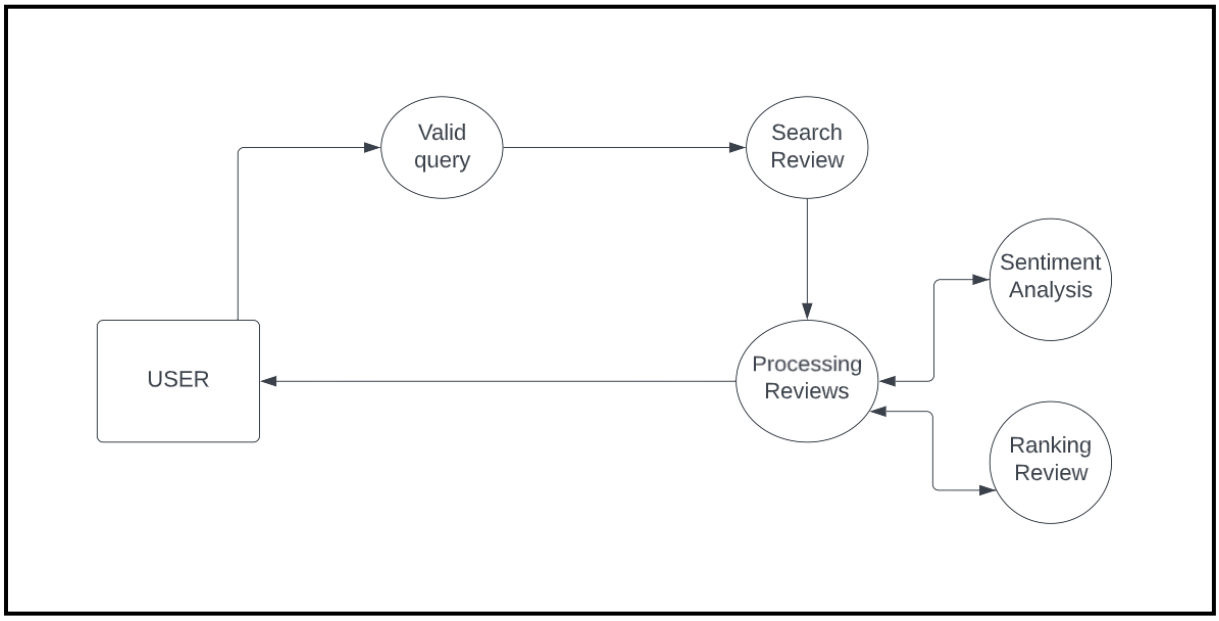


Fig 4.3 (b) DFD Level 1

5. SYSTEM MODELING

5.1 Detailed Class Diagram

Class diagram is a static diagram and it is used to model the static view of a system. The static view describes the vocabulary of the system. Class diagram is also considered as the foundation for component and deployment diagrams. Class diagrams are not only used to visualize the static view of the system but they are also used to construct the executable code for forward and reverse engineering of any system.

Generally, UML diagrams are not directly mapped with any object-oriented programming languages but the class diagram is an exception. Class diagram clearly shows the mapping with object-oriented languages such as Java, C++, etc. From practical experience, class diagram is generally used for construction purpose.

In a nutshell it can be said, class diagrams are used for −

- Describing the static view of the system.
- Showing the collaboration among the elements of the static view.
- Describing the functionalities performed by the system.
- Construction of software applications using object-oriented languages.

Class Diagram of our Project:

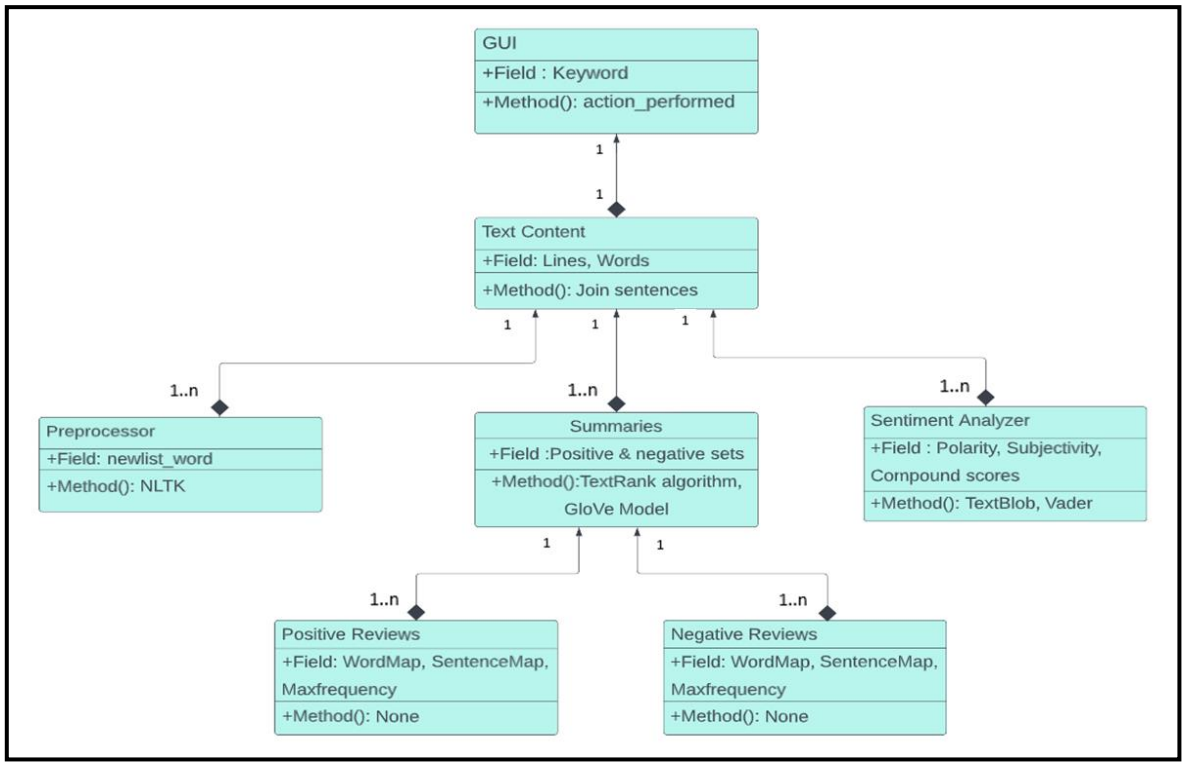


Fig 5.1 Class Diagram

5.2 Interaction Diagram

5.2.1 Sequence Diagram

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

Purpose of a Sequence Diagram

- To model high-level interaction among active objects within a system.
- To model interaction among objects inside a collaboration realizing a use case.
- It either models' generic interactions or some certain instances of interaction.

Sequence Diagram

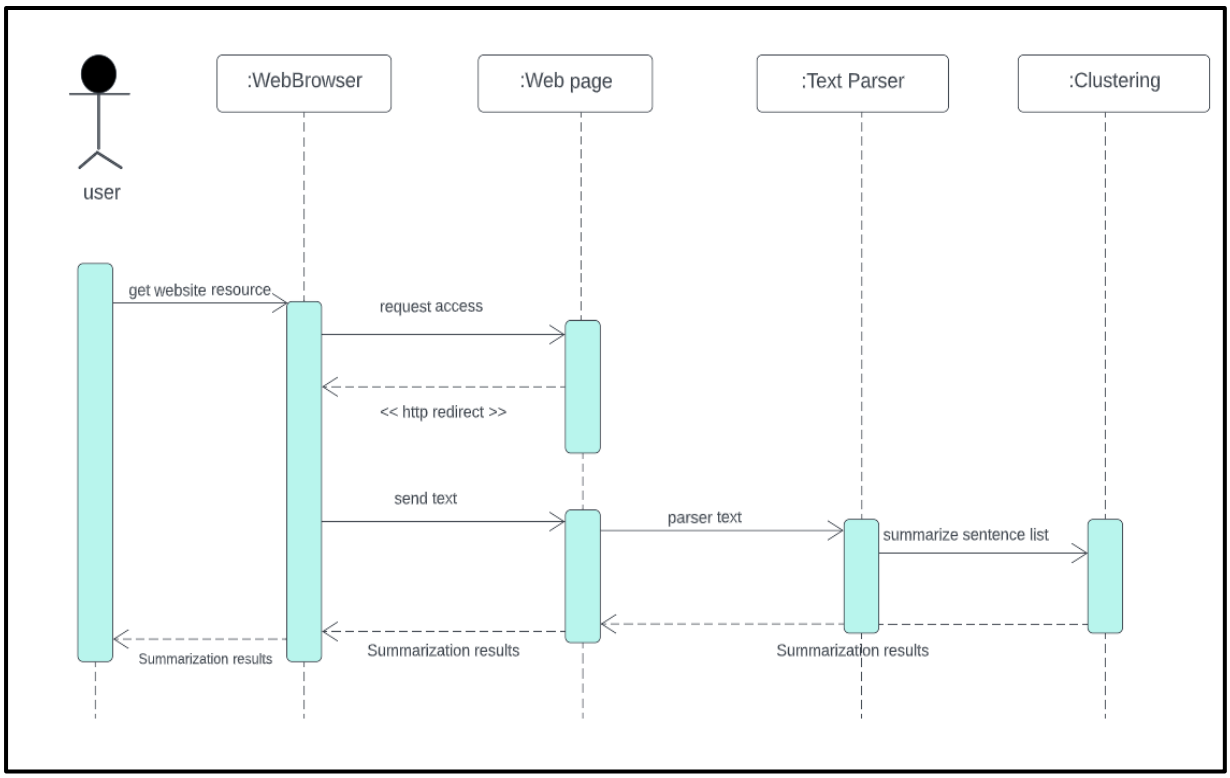


Fig 5.2 Sequence Diagram

5.3 Activity Diagram

Activity diagram is suitable for modeling the activity flow of the system. An application can have multiple systems. Activity diagram also captures these systems and describes the flow

from one system to another. This specific usage is not available in other diagrams. These systems can be database, external queues, or any other system.

We will now investigate the practical applications of the activity diagram. From the above discussion, an activity diagram is drawn from a very high level. So it gives high level view of a system. This high-level view is mainly for business users or any other person who is not a technical person. This diagram is used to model the activities which are nothing but business requirements. The diagram has more impact on business understanding rather than on implementation details.

Activity diagram can be used for −

- Modeling work flow by using activities.
- Modeling business requirements.
- High level understanding of the system's functionalities.
- Investigating business requirements at a later stage.

Activity Diagram

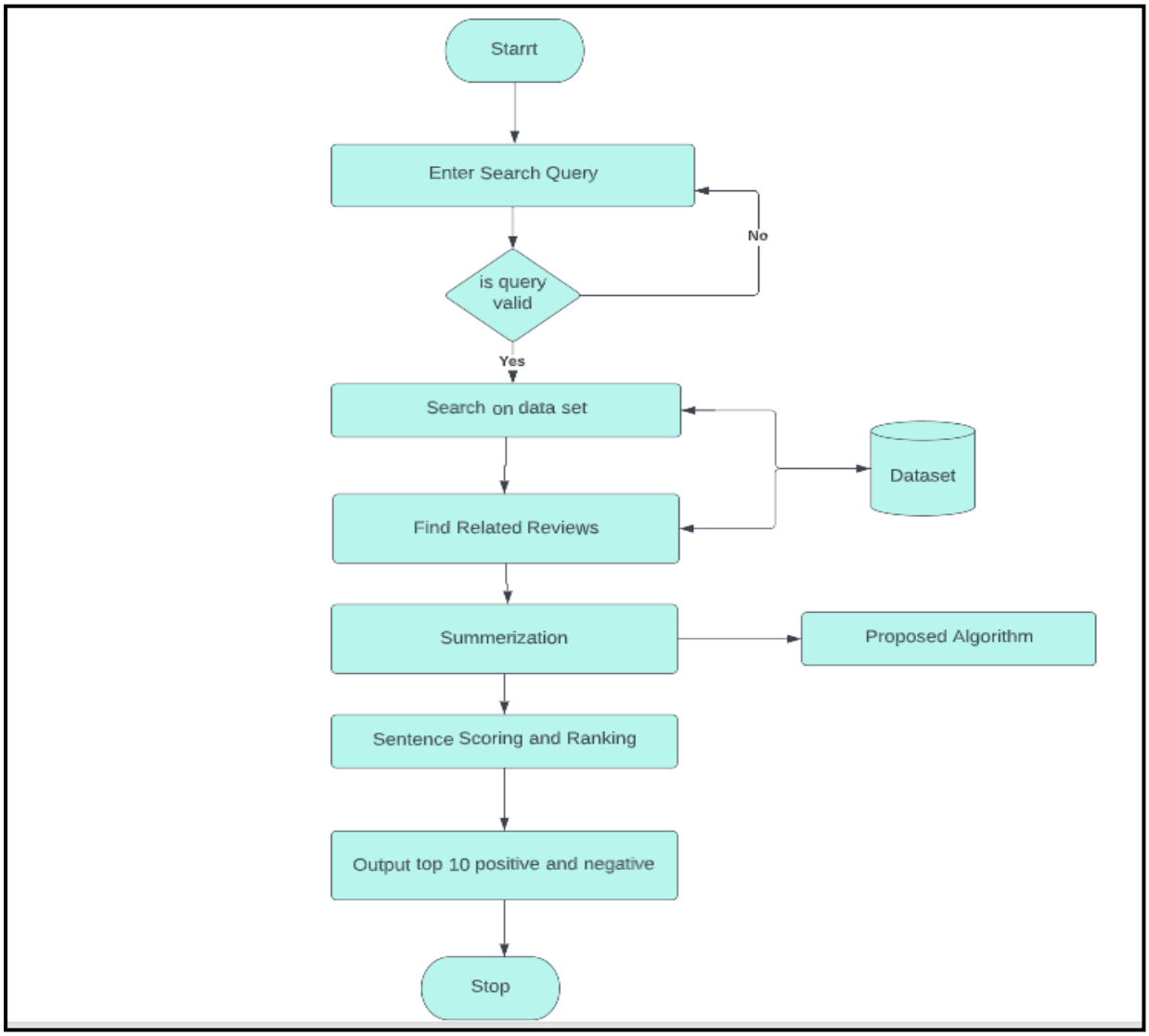


Fig 5.3 Activity Diagram

5.4 Component Diagram

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

Thus, from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc. Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

The purpose of the component diagram can be summarized as −

- Visualize the components of a system.
- Construct executables by using forward and reverse engineering.
- Describe the organization and relationships of the components.

Component diagram

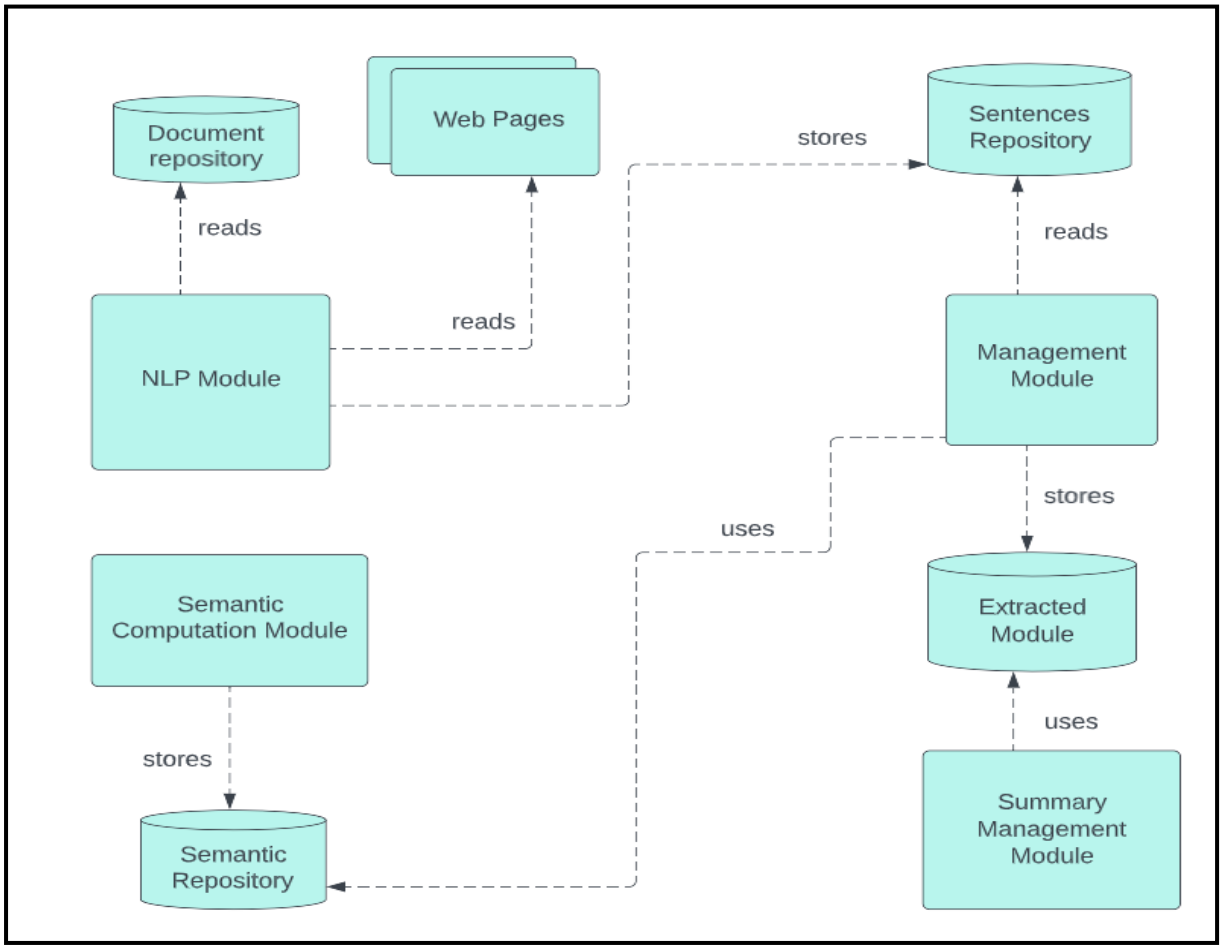


Fig 5.4 Component Diagram

5.5 Structure Chart

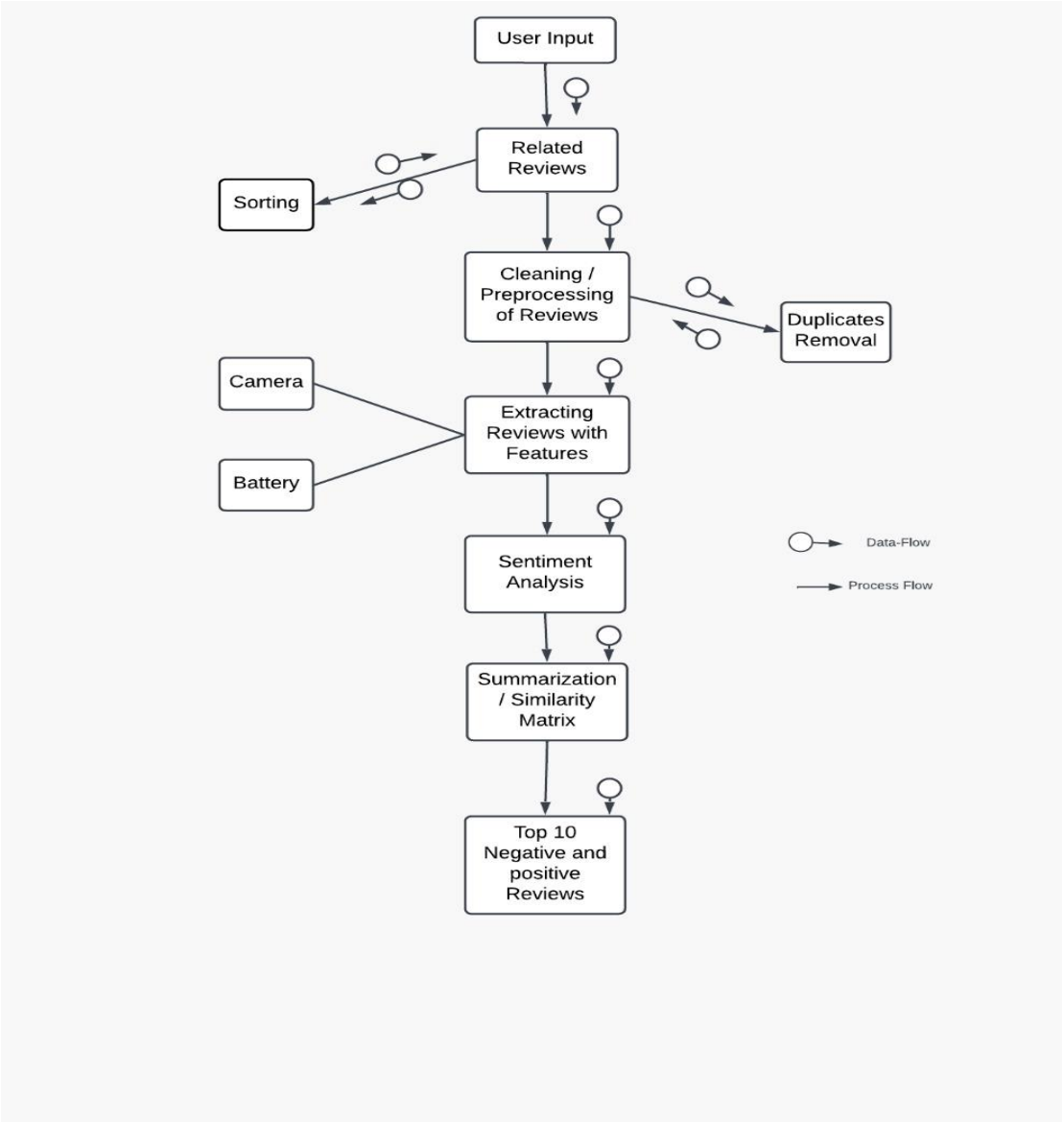


Fig 5.5 Structure Chart

5.6 State Chart

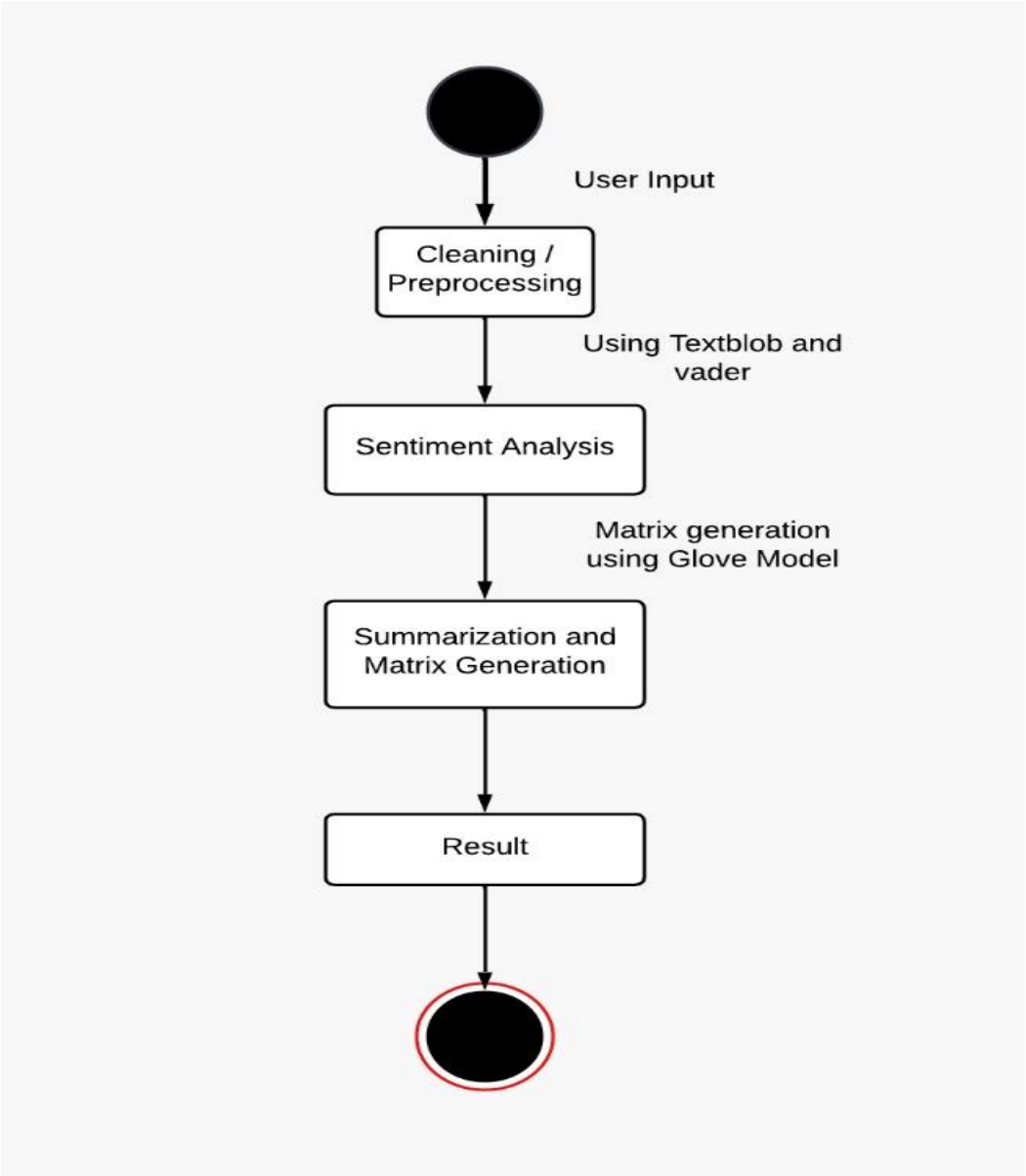


Fig 5.6 State Chart

5.7 Object Diagram

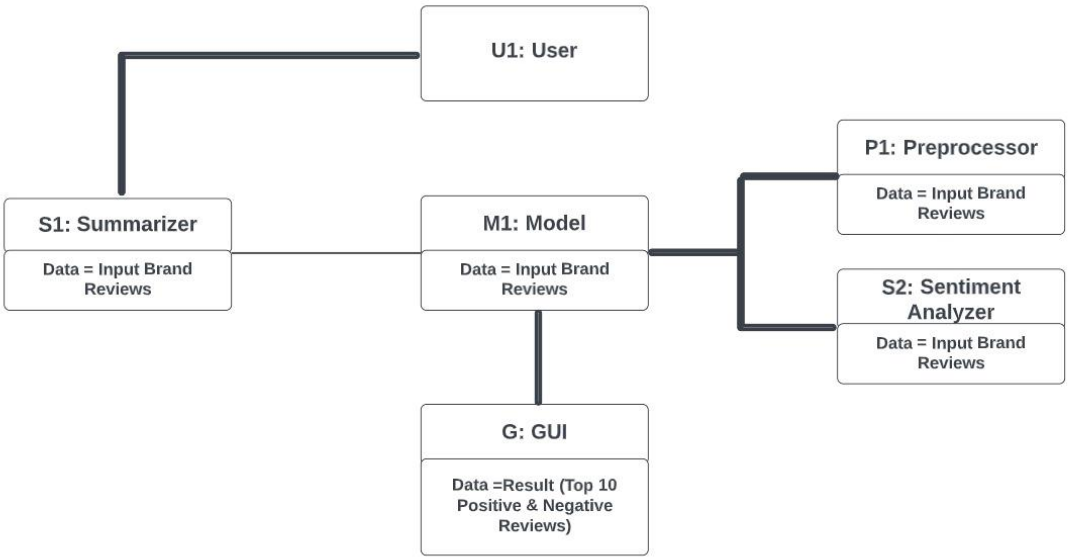


Fig 5.7 Object Diagram

5.8 Test Plans and Implementation Images

5.8.1 Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product it is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests.[11] Each test type addresses a specific testing requirement.

TYPES OF TESTS

1. UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated.[11] It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

2. INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program.[11] Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

3. VALIDATION TESTING

An engineering validation test (EVT) is performed on first engineering prototypes, to ensure that the basic unit performs to design goals and specifications.[11] It is important in identifying design problems, and solving them as early in the design cycle as possible, is the key to keeping projects on time and within budget. Too often, product design and performance problems are not detected until late in the product development cycle — when the product is ready to be shipped. The old adage holds true: It costs a penny to make a change in engineering, a dime in production and a dollar after a product is in the field.

Verification is a Quality control process that is used to evaluate whether or not a product, service, or system complies with regulations, specifications, or conditions imposed at the start of a development phase. Verification can be in development, scaleup, or production. This is often an internal process.

Validation is a Quality assurance process of establishing evidence that provides a high degree of assurance that a product, service, or system accomplishes its intended requirements. This often involves acceptance of fitness for purpose with end users and other product stakeholders.

4. SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.[11] System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s).

• System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

• System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS).

• System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

Module Under Test	Text file summarization
Description	When the client program is executed, it takes text file as input and process it further.
Output	Calculate the sentence score using different methods and gives the relevant summary.
Remarks	Test Successful.

Some Test Results:

• Short Input - While performing the testing for smaller inputs we get an error of minimum value where it denotes about the word's frequency is not greater than required frequency to calculate the summary.

• Foreign Language - While giving input in any language, it successfully performs the summarization process and a meaningful summary is obtained.

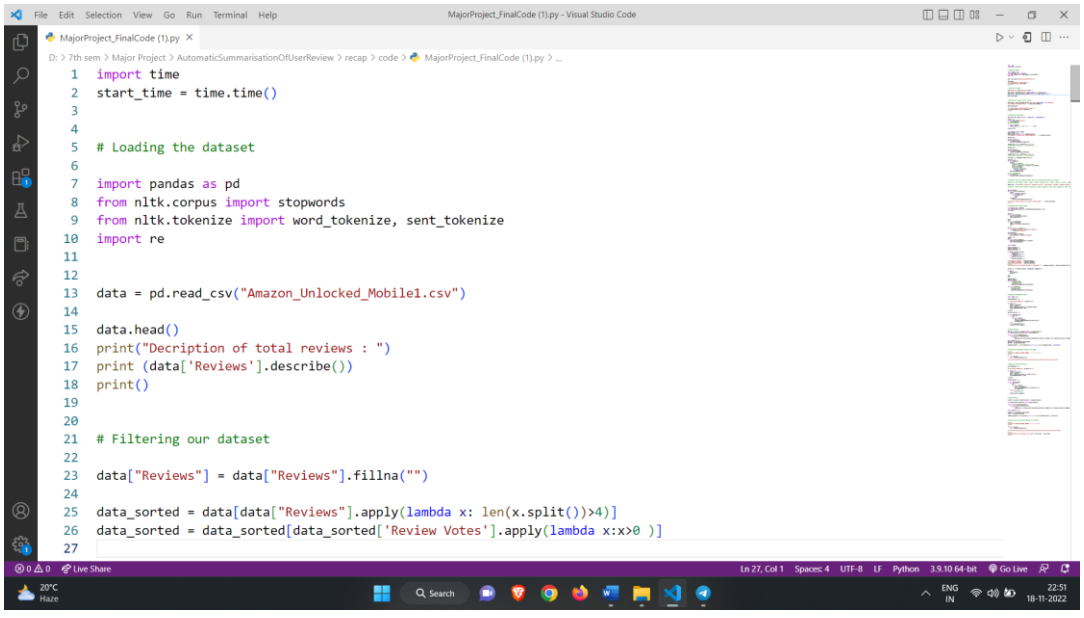
•Improper URL - If the given URL hasn't a defined and a sequential data which can be summarize then it displays the error as mentioned below since the web scrapper can't get the exact data from the URL from which our summary could be generated.

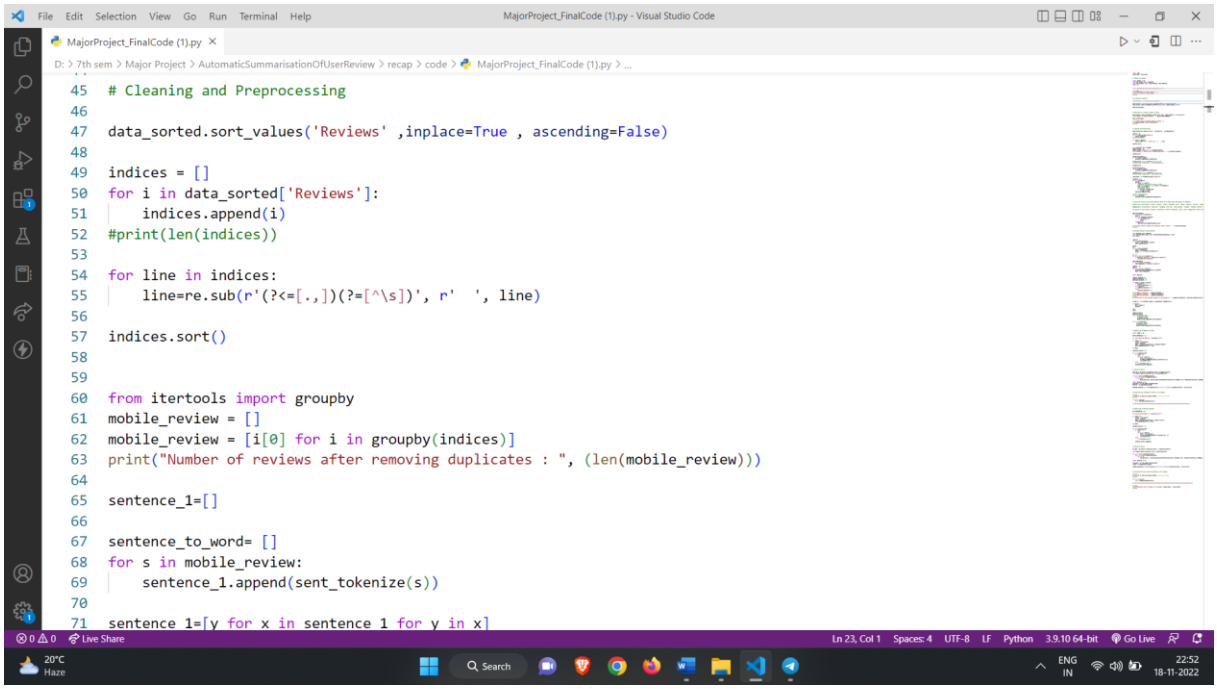
• Illogical Text - If any illogical or meaningful text is given as an input, then the summary won't come as it will not make sense to generate a summary of punctuation marks or any stop words. As given below the output is generated where it shows that the given text could be stop words which gets eliminated in the pre-processing phase of summarization.

• Repeated Text - If the repeated text is given as input to generate the summary, then the summary will be obtained but it will also be in repeated manner since the text are repeating due to which the program can't differentiate between the meaning of the generated summary. So based on the repeated input, summary is generated.

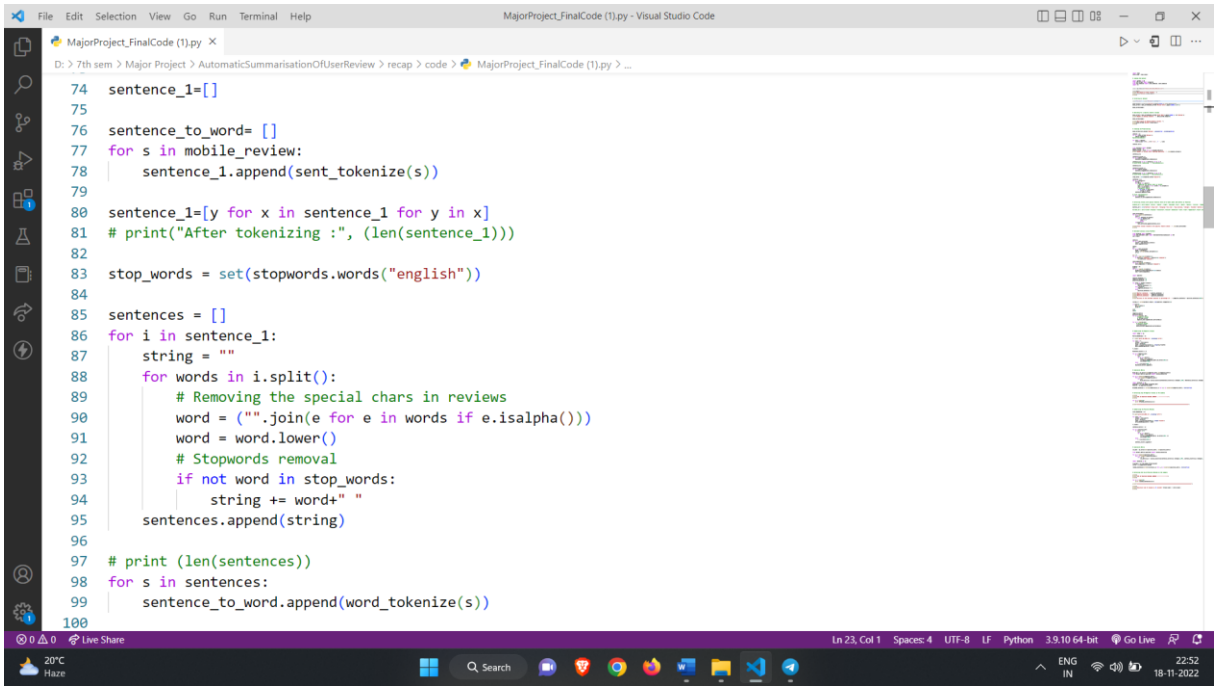
Test ID	Test case	Description	Expected Result	Status
T01	Short Input	Input is too short	Should not generate summary	Pass
T02	Foreign Language	Input is in different language other than English	Summary should be generated	Pass
T03	Improper URL	If the data cannot be scrapped from the given URL	Cannot extract data	Pass
T04	Repeated Text	When the same text is given number of times	Summary for the text repeating multiple times should only be shown once on the output screen	Fail
T05	Illogical input	Meaningless inputs like symbols, punctuation marks etc.	It should not generate a summary and should show an error message	Pass

5.8.2 Implementation Images

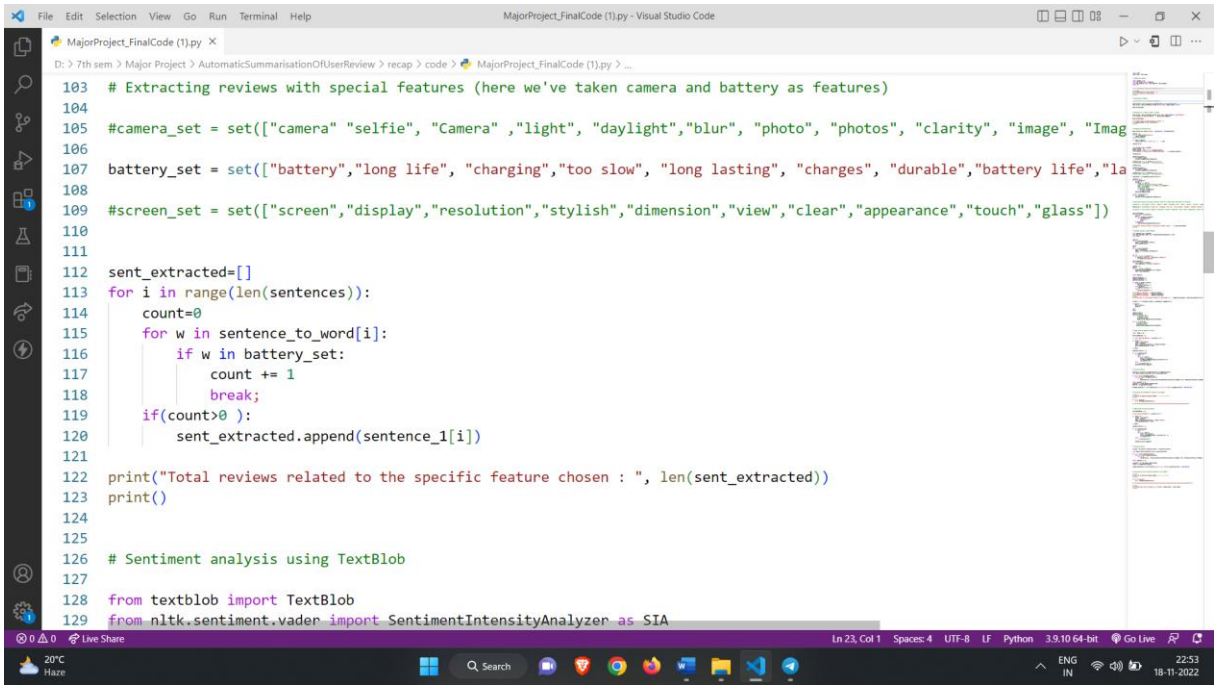




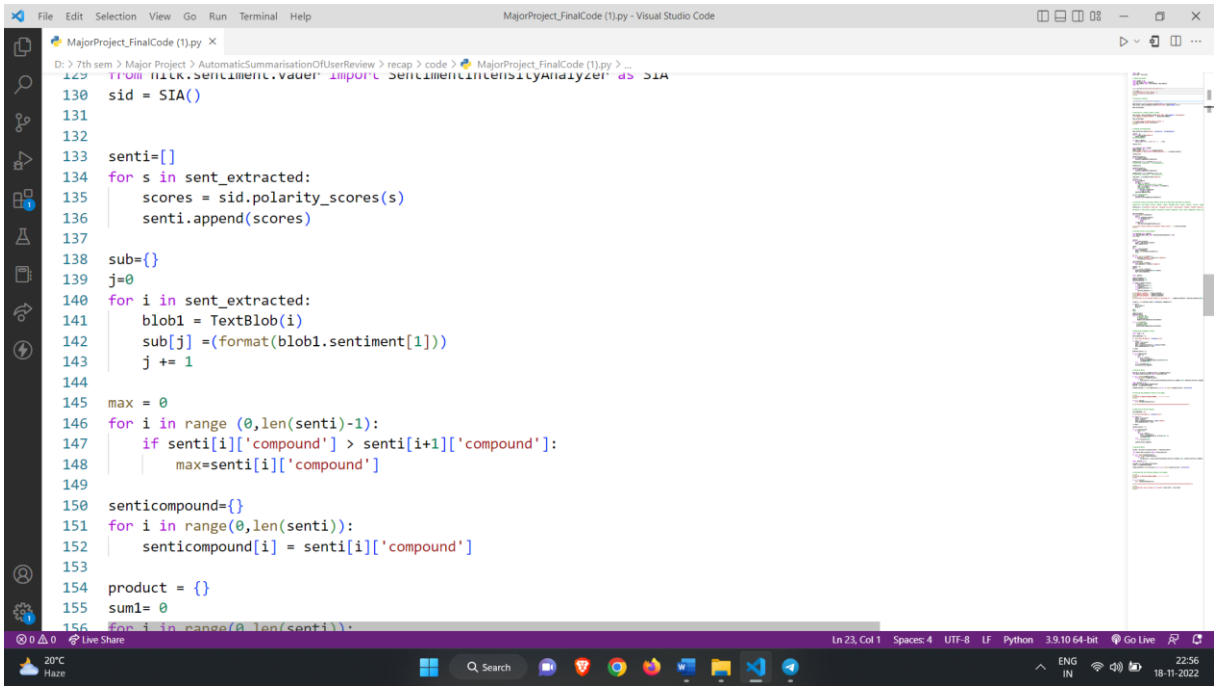
```
# Cleaning and Preprocessing

data_sorted.sort_values('Reviews' ,inplace=True , ascending=False)

indices = []
for i in data_sorted['Reviews']:
    indices.append(i)
#print(len(indices))

for line in indices:
    line=re.sub(r'(?<=[.,])(?=[^\s])', r' ', line)

indices.sort()


from itertools import groupby
mobile_review = []
mobile_review = [i[0] for i in groupby(indices)]
print("Number of reviews after removing duplicates : ", (len(mobile_review)))

sentence_1=[]

sentence_to_word= []
for s in mobile_review:
    sentence_1.append(sent_tokenize(s))

sentence_1=[y for x in sentence_1 for y in x]
```



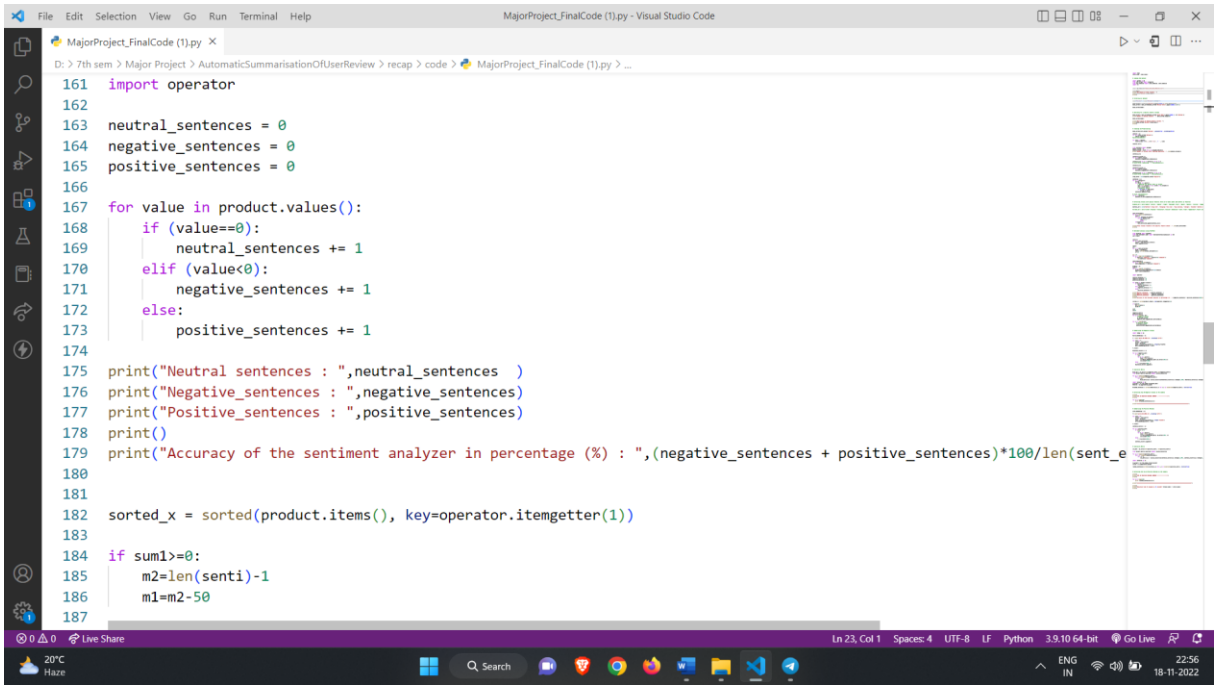
```
sentence_1=[]

sentence_to_word= []
for s in mobile_review:
    sentence_1.append(sent_tokenize(s))

sentence_1=[y for x in sentence_1 for y in x]
# print("After tokenizing :", (len(sentence_1)))

stop_words = set(stopwords.words("english"))

sentences = []
for i in sentence_1:
    string = ""
    for words in i.split():
        # Removing the special chars in reviews
        word = ("".join(e for e in words if e.isalpha()))
        word = word.lower()
        # Stopwords removal
        if not word in stop_words:
            string += word+" "
    sentences.append(string)

# print (len(sentences))
for s in sentences:
    sentence_to_word.append(word_tokenize(s))
```



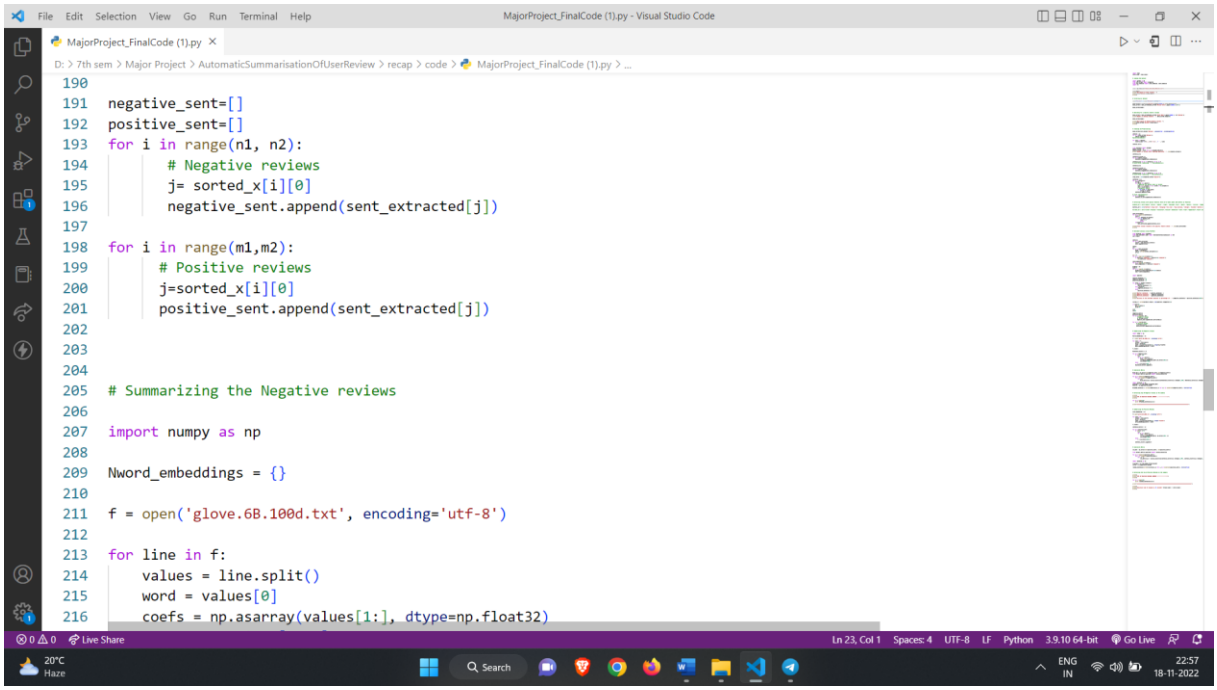
```
# Extracting reviews with special features (here we've taken camera and battery as features)

#camera_set = set(["camera" "selfie", "Camera" ,"light", "daylight","blur", "photo", "photos", "clarity", "image", "Imag

battery_set = set(["battery","long life", "charging","too slow", "long lasting", "charges", "durable","battery life","la

#screen_set = set(["screen","display","resolution","stylish","dimension","view","clear","appearance","touch","glass"])


sent_extracted=[]
for i in range(len(sentences)):
    count=0
    for w in sentence_to_word[i]:
        if w in battery_set:
            count += 1
            break;
    if(count>0 ):
        sent_extracted.append(sentence_1[i])

print("Total reviews related to the specific feature chosen : ", len(sent_extracted))
print()


# Sentiment analysis using TextBlob

from textblob import TextBlob
from nltk.sentiment.vader import SentimentIntensityAnalyzer as SIA
```



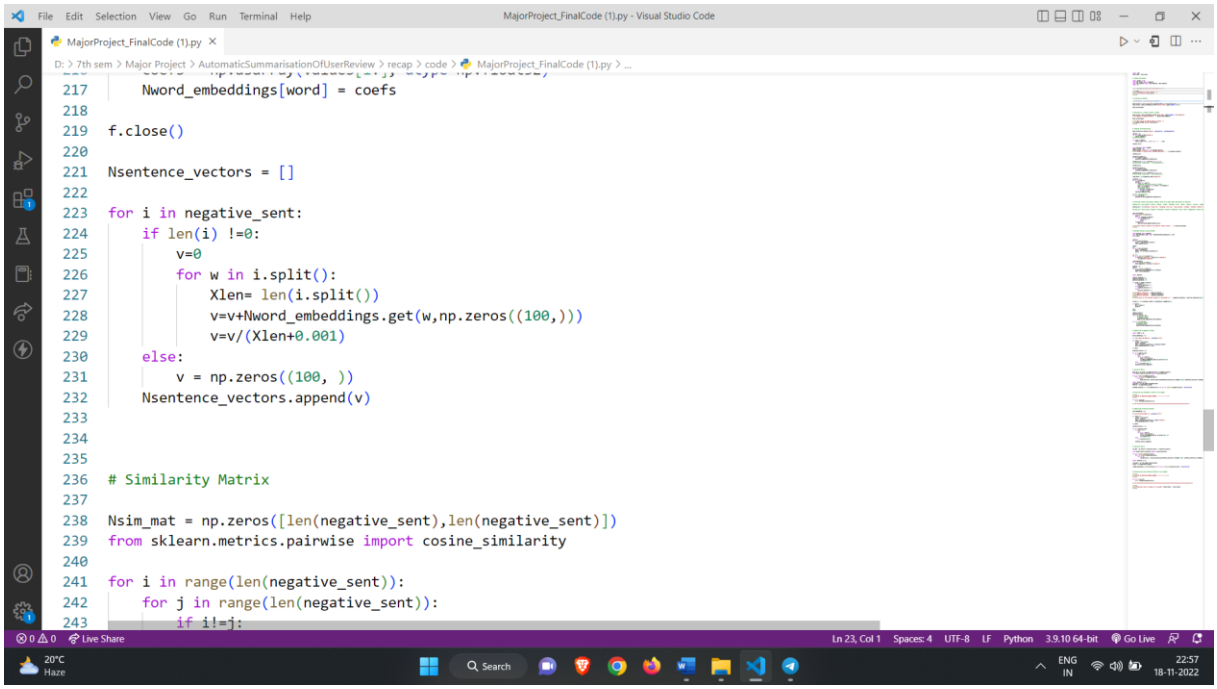
```
sid = SIA()


senti=[]
for s in sent_extracted:
    scores = sid.polarity_scores(s)
    senti.append(scores)

sub={}
j=0
for i in sent_extracted:
    blob1 = TextBlob(i)
    sub[j] =(format(blob1.sentiment[1]))
    j += 1

max = 0
for i in range (0,len(senti)-1):
    if senti[i]['compound'] > senti[i+1]['compound']:
        max=senti[i]['compound']

senticompound={}
for i in range(0,len(senti)):
    senticompound[i] = senti[i]['compound']

product = {}
sum1= 0
```



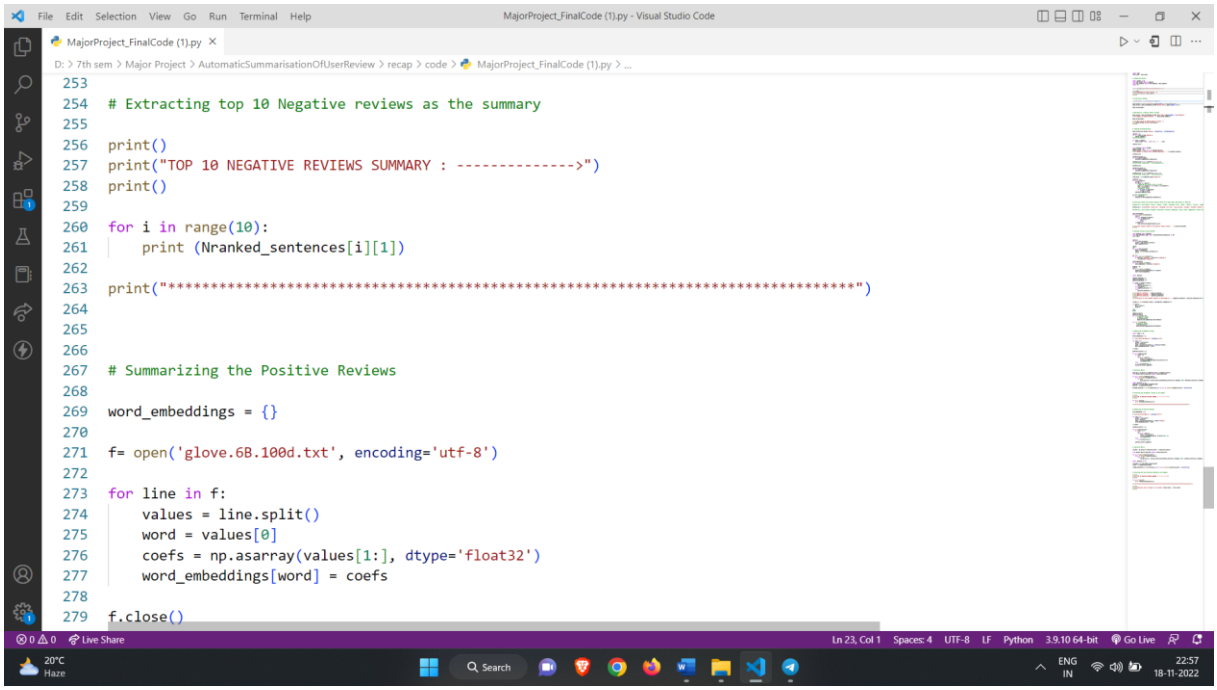
```
import operator

neutral_sentences = 0
negative_sentences = 0
positive_sentences = 0

for value in product.values():
    if (value==0):
        neutral_sentences += 1
    elif (value<0):
        negative_sentences += 1
    else:
        positive_sentences += 1

print("Neutral sentences : ",neutral_sentences  )
print("Negative_sentences : ",negative_sentences)
print("Positive_sentences : ",positive_sentences)
print()
print("Accuracy of the sentiment analyzer in percentage (%) : ",(negative_sentences + positive_sentences)*100/len(sent_e


sorted_x = sorted(product.items(), key=operator.itemgetter(1))

if sum1>=0:
    m2=len(senti)-1
    m1=m2-50
```



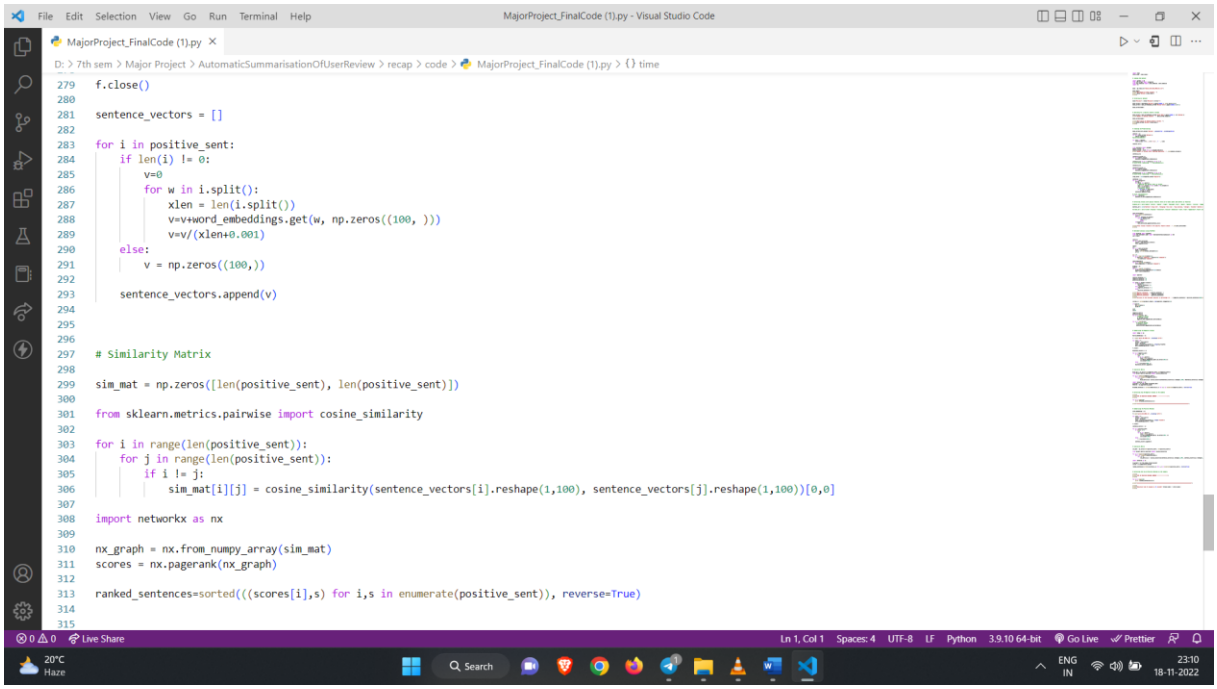
```
negative_sent=[]
positive_sent=[]
for i in range(n1, n2):
        # Negative reviews
        j= sorted_x[i][0]
        negative_sent.append(sent_extracted[j])

for i in range(m1,m2):
       # Positive reviews
       j=sorted_x[i][0]
       positive_sent.append(sent_extracted[j])


# Summarizing the Negative reviews

import numpy as np

Nword_embeddings = {}

f = open('glove.6B.100d.txt', encoding='utf-8')

for line in f:
    values = line.split()
    word = values[0]
    coefs = np.asarray(values[1:], dtype=np.float32)
```



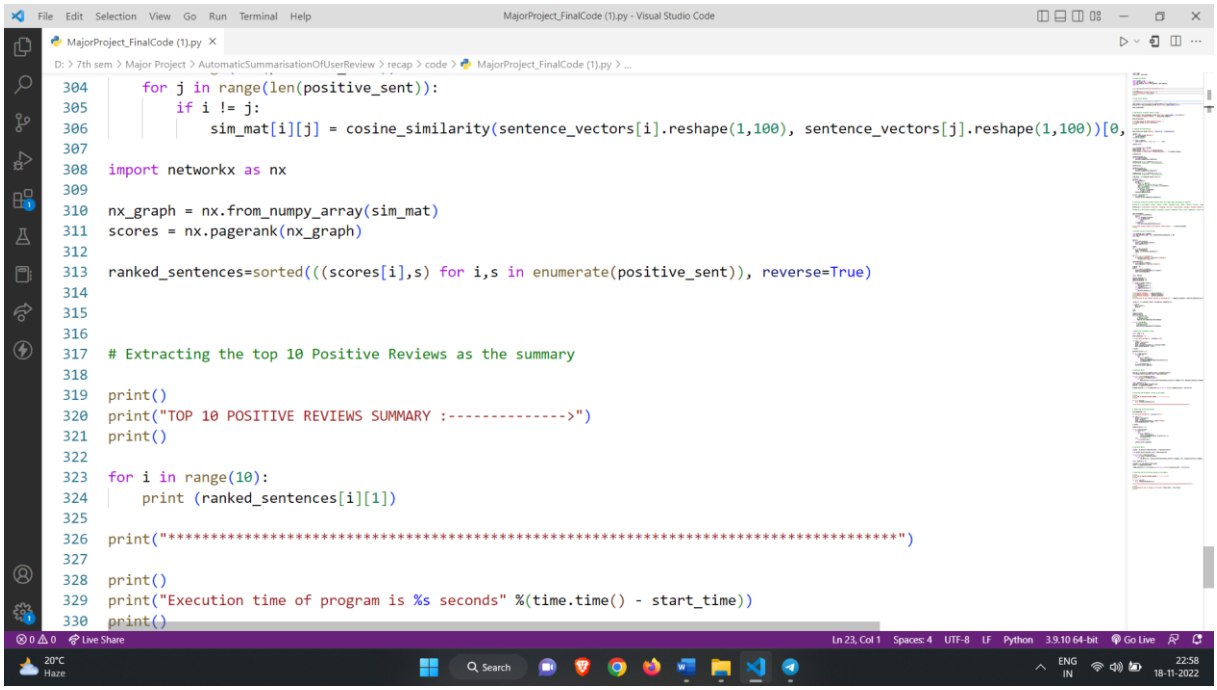
```
    Nword_embeddings[word] = coefs

f.close()

Nsentence_vectors = []

for i in negative_sent:
    if len(i) !=0:
        v=0
        for w in i.split():
            Xlen= len(i.split())
            v=v+Nword_embeddings.get(w,np.zeros((100,)))
            v=v/(Xlen+0.001)
    else:
        v = np.zeros((100, ))
    Nsentence_vectors.append(v)



# Similarity Matrix

Nsim_mat = np.zeros([len(negative_sent),len(negative_sent)])
from sklearn.metrics.pairwise import cosine_similarity

for i in range(len(negative_sent)):
    for j in range(len(negative_sent)):
        if i!=j:
```



```

# Extracting top 10 Negative reviews as the summary

print()
print("TOP 10 NEGATIVE REVIEWS SUMMARY : ---------->")
print()

for i in range(10):
    print (Nranked_sentences[i][1])

print("********************************************************************************")



# Summarizing the Positive Reviews

word_embeddings = {}

f= open('glove.6B.100d.txt', encoding='utf-8')

for line in f:
    values = line.split()
    word = values[0]
    coefs = np.asarray(values[1:], dtype='float32')
    word_embeddings[word] = coefs

f.close()
```

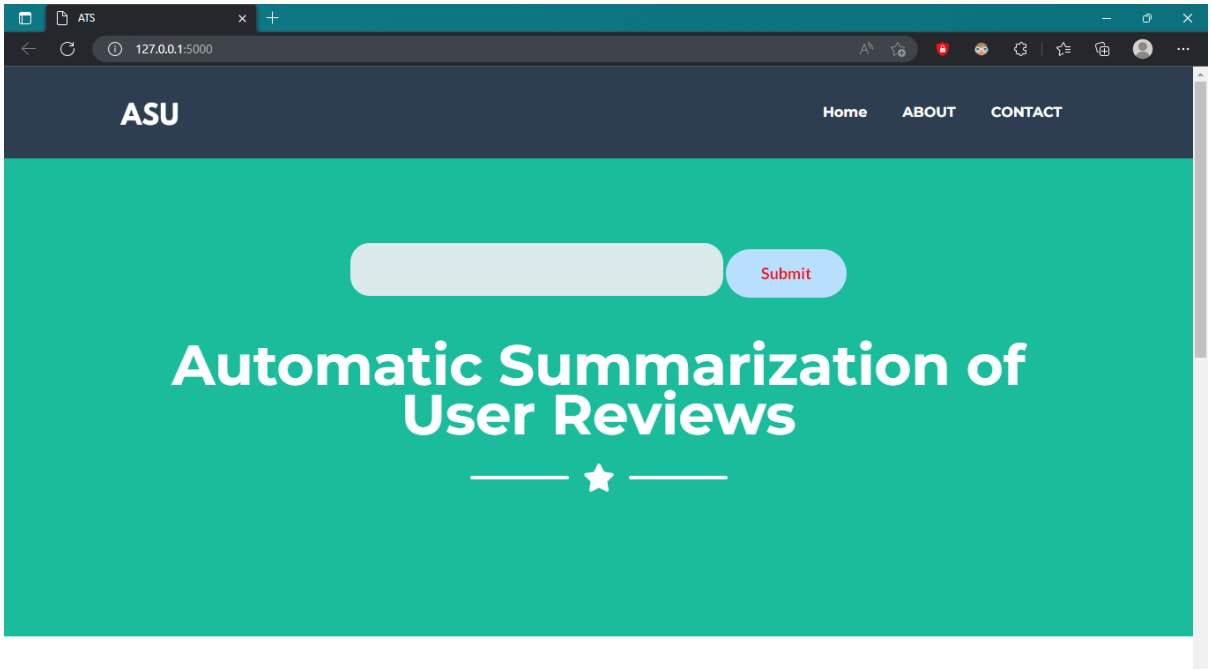


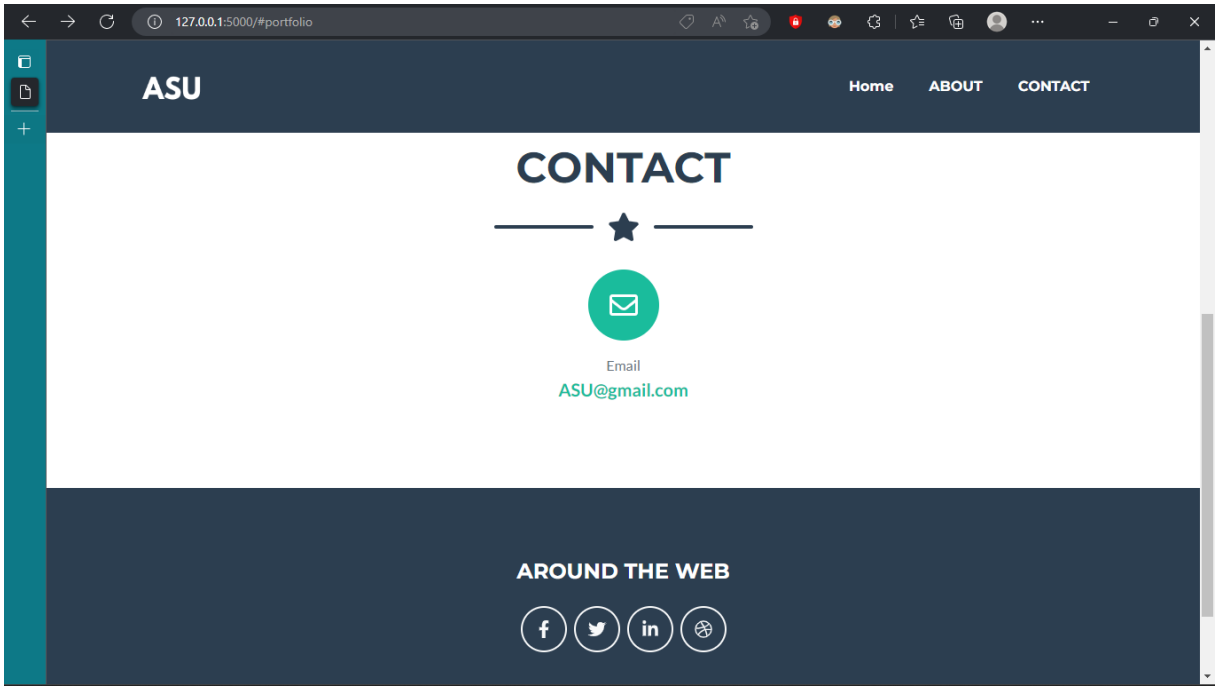
```
f.close()

sentence_vectors = []

for i in positive_sent:
    if len(i) != 0:
        v=0
        for w in i.split():
            xlen = len(i.split())
            v=v+word_embeddings.get(w, np.zeros((100, )))
            v=v/(xlen+0.001)
    else:
        v = np.zeros((100,))

    sentence_vectors.append(v)


# Similarity Matrix

sim_mat = np.zeros([len(positive_sent), len(positive_sent)])

from sklearn.metrics.pairwise import cosine_similarity

for i in range(len(positive_sent)):
    for j in range(len(positive_sent)):
        if i != j:
            sim_mat[i][j] = cosine_similarity(sentence_vectors[i].reshape(1,100), sentence_vectors[j].reshape(1,100))[0,0]

import networkx as nx

nx_graph = nx.from_numpy_array(sim_mat)
scores = nx.pagerank(nx_graph)

ranked_sentences=sorted(((scores[i],s) for i,s in enumerate(positive_sent)), reverse=True)
```

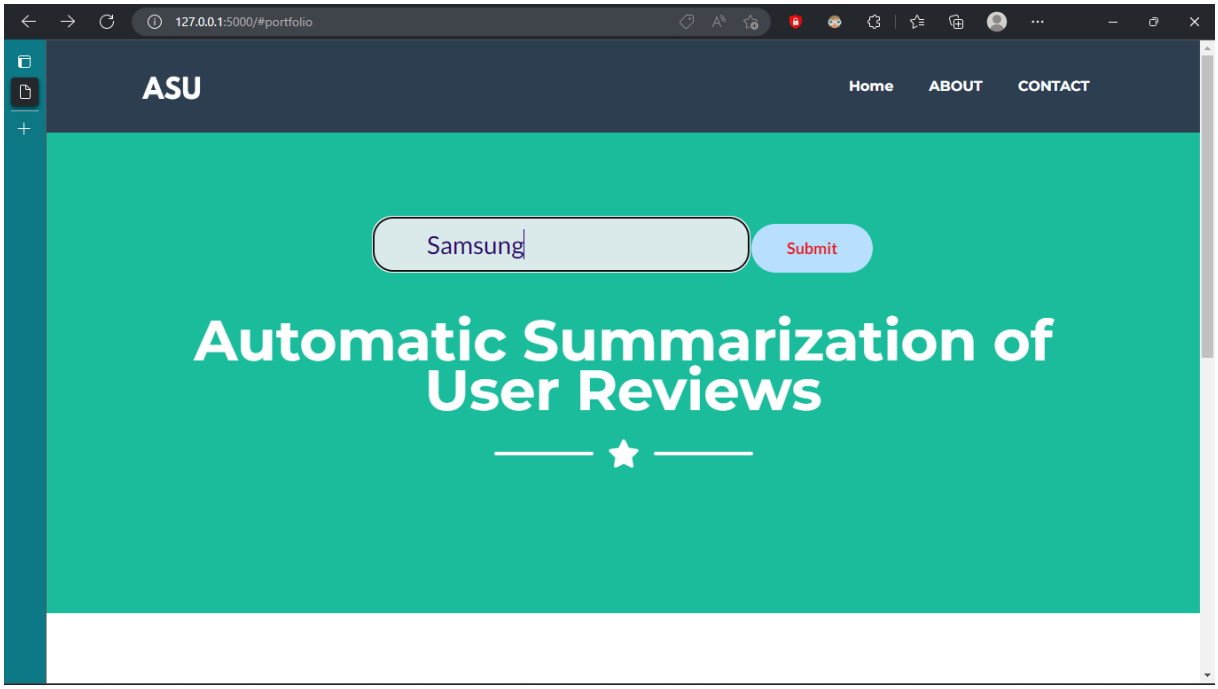


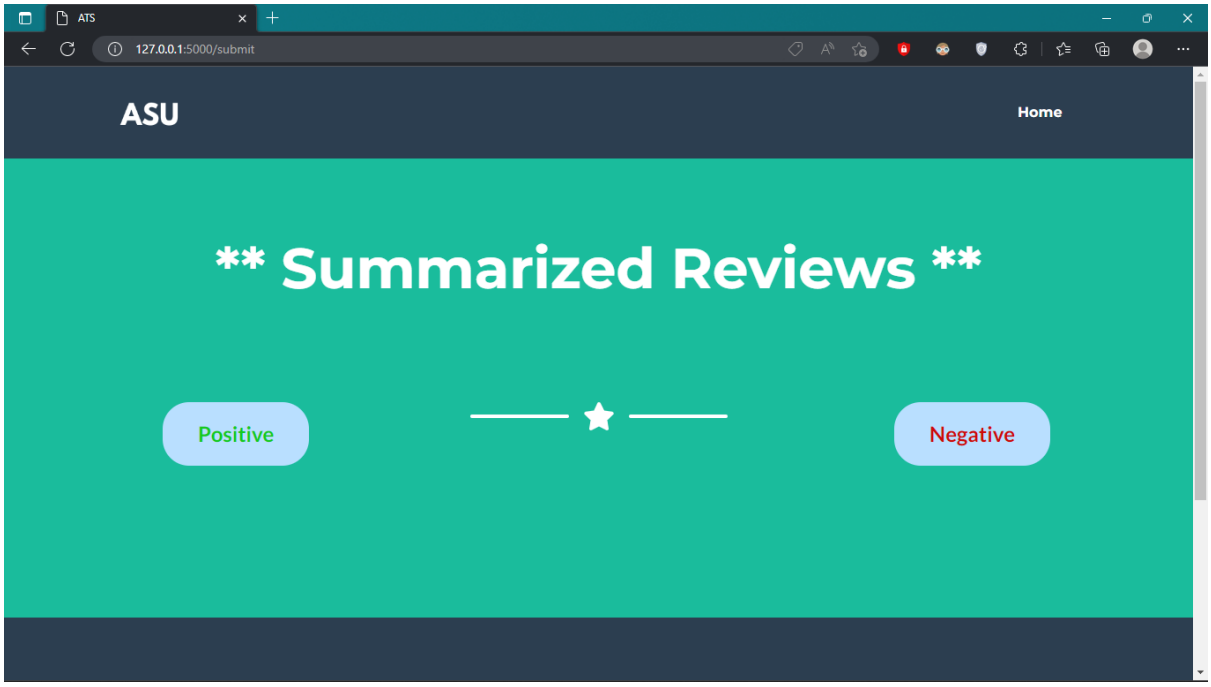
```
    for j in range(len(positive_sent)):
        if i != j:
            sim_mat[i][j] = cosine_similarity(sentence_vectors[i].reshape(1,100), sentence_vectors[j].reshape(1,100))[0,

import networkx as nx

nx_graph = nx.from_numpy_array(sim_mat)
scores = nx.pagerank(nx_graph)

ranked_sentences=sorted(((scores[i],s) for i,s in enumerate(positive_sent)), reverse=True)



# Extracting the top 10 Positive Reviews as the summary

print()
print("TOP 10 POSITIVE REVIEWS SUMMARY :-------------->")
print()

for i in range(10):
    print (ranked_sentences[i][1])

print("*********************************************************************************************")

print()
print("Execution time of program is %s seconds" %(time.time() - start_time))
print()
```

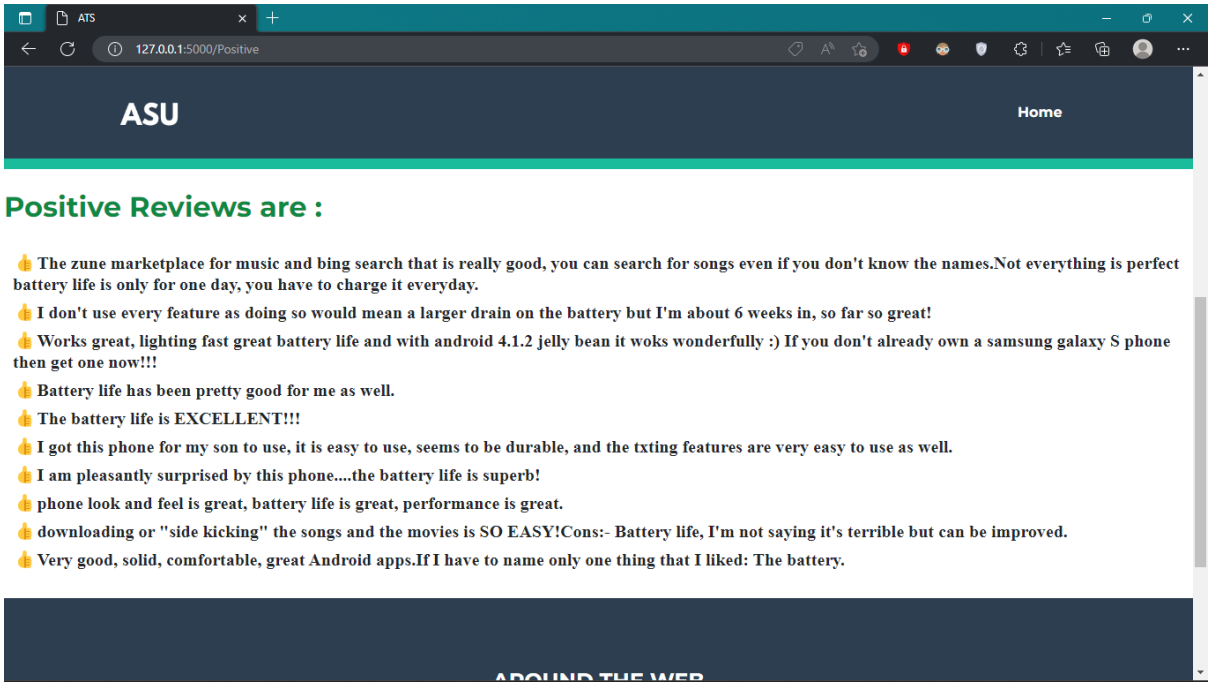
Front End Screenshots:

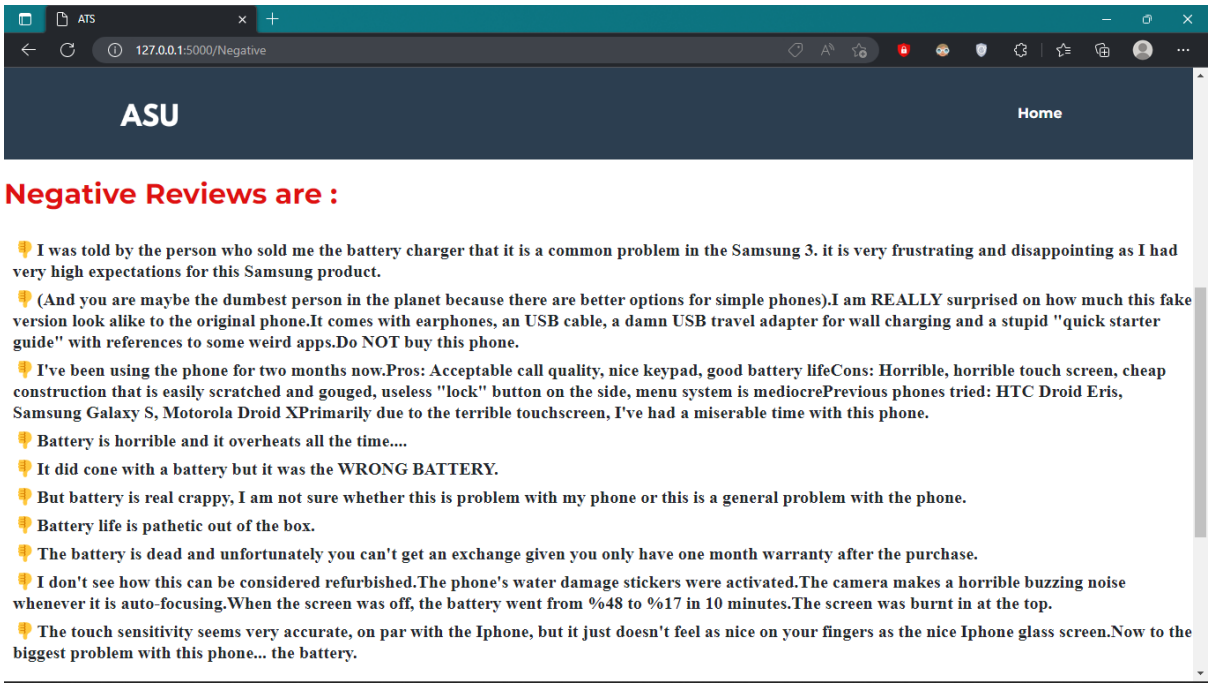





127.0.0.1:5000/#portfolio
ASU
Home
ABOUT
CONTACT
Samsung
Submit
Automatic Summarization of User Reviews


ATS
127.0.0.1:5000/submit
ASU
Home
** Summarized Reviews **
Positive
Negative


ATS
127.0.0.1:5000/Positive
ASU
Home
Positive Reviews are :
👍 The zune marketplace for music and bing search that is really good, you can search for songs even if you don't know the names.Not everything is perfect battery life is only for one day, you have to charge it everyday.
👍 I don't use every feature as doing so would mean a larger drain on the battery but I'm about 6 weeks in, so far so great!
👍 Works great, lighting fast great battery life and with android 4.1.2 jelly bean it woks wonderfully :) If you don't already own a samsung galaxy S phone then get one now!!!
👍 Battery life has been pretty good for me as well.
👍 The battery life is EXCELLENT!!!
👍 I got this phone for my son to use, it is easy to use, seems to be durable, and the txting features are very easy to use as well.
👍 I am pleasantly surprised by this phone....the battery life is superb!
👍 phone look and feel is great, battery life is great, performance is great.
👍 downloading or "side kicking" the songs and the movies is SO EASY!Cons:- Battery life, I'm not saying it's terrible but can be improved.
👍 Very good, solid, comfortable, great Android apps.If I have to name only one thing that I liked: The battery.


ATS
127.0.0.1:5000/Negative
ASU
Home
Negative Reviews are :
👎 I was told by the person who sold me the battery charger that it is a common problem in the Samsung 3. it is very frustrating and disappointing as I had very high expectations for this Samsung product.
👎 (And you are maybe the dumbest person in the planet because there are better options for simple phones).I am REALLY surprised on how much this fake version look alike to the original phone.It comes with earphones, an USB cable, a damn USB travel adapter for wall charging and a stupid "quick starter guide" with references to some weird apps.Do NOT buy this phone.
👎 I've been using the phone for two months now.Pros: Acceptable call quality, nice keypad, good battery lifeCons: Horrible, horrible touch screen, cheap construction that is easily scratched and gouged, useless "lock" button on the side, menu system is mediocrePrevious phones tried: HTC Droid Eris, Samsung Galaxy S, Motorola Droid XPrimarily due to the terrible touchscreen, I've had a miserable time with this phone.
👎 Battery is horrible and it overheats all the time....
👎 It did cone with a battery but it was the WRONG BATTERY.
👎 But battery is real crappy, I am not sure whether this is problem with my phone or this is a general problem with the phone.
👎 Battery life is pathetic out of the box.
👎 The battery is dead and unfortunately you can't get an exchange given you only have one month warranty after the purchase.
👎 I don't see how this can be considered refurbished.The phone's water damage stickers were activated.The camera makes a horrible buzzing noise whenever it is auto-focusing.When the screen was off, the battery went from %48 to %17 in 10 minutes.The screen was burnt in at the top.
👎 The touch sensitivity seems very accurate, on par with the Iphone, but it just doesn't feel as nice on your fingers as the nice Iphone glass screen.Now to the biggest problem with this phone... the battery.

6. CONCLUSION & FUTURE WORK

6.1 Limitations of Project

Assessing rundowns (either consequently or physically) is a troublesome undertaking. The fundamental issue in assessment originates from the difficulty of building a standard against which the consequences of the frameworks that must be thought about. Further, it is exceptionally elusive out what a right summary is on the grounds that there is an opportunity of the framework to produce a superior summary that is not the same as any human summary which is utilized as an estimation to precise result. There are certain challenges in extractive method of text summarization. Extractive summarization lacks the readability of the text produced.

6.2 Future Enhancement

The status, and state, of automatic summarizing has radically changed through the years. It has specially benefit from work of other asks, e.g. information retrieval, information extraction or text categorization. Research on this field will continue due to the fact that text summarization task has not been finished yet and there is still much effort to do, to investigate and to improve. Definition, types, different approaches and evaluation methods have been exposed as well as summarization systems features and techniques already developed. In the future we plan to contribute to improve this field by means of improving the quality of summaries, and studying the influence of other neighbor tasks techniques on summarization. In future work abstractive methods can be implemented. In abstractive method build an internal semantic representation and then use natural language generation techniques to create a summary.

7 BIBLIOGRAPHY

7.1 References

1. M. F. Mridha, A. A. Lima, K. Nur, S.. Das, M. Hasan and M. M. Kabir, "A Survey of Automatic Text Summarization: Progress, Process and Challenges," in IEEE CAccess, volume 9, pp. 156043-156070, 2021, doi:10.1109/ACCESS.2021.3129786. (2021)

2. Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, Hoda K. Mohamed,Automatic text summarization: A comprehensive survey, Expert Systems with Applications,Volume 165,113679 ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2020.113679 (2021)

3. Kumar, Y., Kaur, K. & Kaur, S. Study of automatic text summarization approaches in different languages. *ArtifIntell Rev* 54, 5897–5929 https://doi.org/10.1007/s10462-021-09964-4 (2021)

4. Atif, Naomie and Yogan. P ,Genetic semantic Graph Approach for multi document abstractive Summarization. Fifth International Conference on Digital Information Processing and Communications (ICDIPC)(2015).

5. GloVe: Global Vectors for Word Representation: Jeffrey Pennington, Richard Socher, Christopher D. Manning https://nlp.stanford.edu/projects/glove/ (2014)

6. Tingting Wei, Yonghe Lu, Huiyou Chang, Qiang Zhou, Xianyu Bao. "A semantic approach for text clustering using WordNet and lexical chains", Expert Systems with Applications, 2015

7. Gupta, Vishal, and Gurpreet Singh Lehal. "A Survey of Text Summarization Extractive Techniques", Journal of Emerging Technologies in Web Intelligence, 2010.

8. Pranitha Reddy, R.C. Balabantaray. "Improvisation of the Document Summarization by Combining the IR Techniques with “Code-Quantity Memory and Attention” Linguistic Principles", Procedia Technology, 2012

9. Hennig, Leonhard. "Content Modeling f or Automatic Document Summarization", Technische Universität Berlin, 2011.

10. Pandimurugan. "A survey of software testing in ref actoring based software models", International Conf erence on Nanoscience Engineering and Technology (ICONSET 2011), 11/2011

11. https://www.softwaretestinghelp.com/types-of-software-testing/

12. AdhikaWidyassari, S. R. (2020). Review of automatic text summarization techniques & methods. Journal of King Saud University - Computer and Information Sciences, 18.

13. Amigó E, G. J. (2005). a framework for the evaluation of text summarization systems.proceedings of the 43rd annual meeting on association for computational linguistics. ACL ’05.

14. Antiqueira L, O. O. (2007). A complex network approach to text summarization. Information Sciences.

15. B. Cretu, Z. C. (2002). Automatic summarization based on sentence extraction. International Journal of Applied Electromagnetic and mechanics.

16. Brownlee, J. (2019, August 7). A Gentle Introduction to Text Summarization. Retrieved from https://machinelearningmastery.com/: https://machinelearningmastery.com/gentleintroduction-text-summarization/

17. ChangjianFanga, D. M. (2016, March 5). Wordsentence co-ranking for automatic extractive text summarization. Retrieved from https://www.sciencedirect.com/: https://www.sciencedirect.com/science/article/a bs/pii/S0957417416306959?via%3Dihub

18. Conroy, J. M. (2001). Text summarization via hidden markov models. Proceedings of SIGIR'01.
19. D. Gillick, K. R. (2009). A global optimization network for meeting summarization. Proc. IEEE Int. Conf. Acoust, 1-4.

20. Darji, H. (2020, January 8). Text SummarizationKey Concepts. Retrieved from https://medium.com/: https://medium.com/@harshdarji_15896/textsummarization-key-concepts-23df617bfb3e

21. Evans, D. K. (2005). Similarity-based multilingual multidocumentsummarization.Technical Report CUCS-014-05