A PROJECT REPORT ON

TEXT SUMMARIZATION USING MACHINE LEARNING

Mini project submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY
(2018-2022)
BY

M.SURYA PRAKASH	**18241A1238**
S. SRIDHAR	**18241A1257**
D.RISHIKESH	**18241A1252**
K. MANUSH	**18241A1226**

Under the Esteemed guidance
of
Dr.V.PRASHANTHI,
Associative Prof
Dept of IT.

DEPARTMENT OF INFORMATION TECHNOLOGY
GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
HYDERABAD

CERTIFICATE

This is to certify that it is a bonafide record of Mini Project work entitled **"TEXT SUMMARIZATION USING MACHINE LEARNING"** done by **M. SURYA(18241A1238),S.SRIDHAR(18241A1257),D.RISHIKESH(18241A1252),K.MANUSH(18241A1226),** of **B.Tech (IT)** in the Department of Information Technology, **Gokaraju Rangaraju Institute of Engineering and Technology** during the period 2018-2022 in the partial fulfillment of the requirements for the award of degree of **BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY** from GRIET, Hyderabad.

Dr.V.Prashanthi ,
Associative prof
(Internal project guide)

Dr.K.Prasanna Lakshmi ,
Head of the Department

(Project External)

ACKNOWLEDGEMENT

We take the immense pleasure in expressing gratitude to our Internal guide**, Dr.V.Prashanthi, Associate Prof, dept of IT**, GRIET. We express our sincere thanks for her encouragement, suggestions and support, which provided the impetus and paved the way for the successful completion of the project work.

We wish to express our gratitude to **Dr. K. Prasanna Lakshmi**, **P. Gopala Krishna,** our Project Co-coordinators **K.Swanthana**, for their constant support during the project.

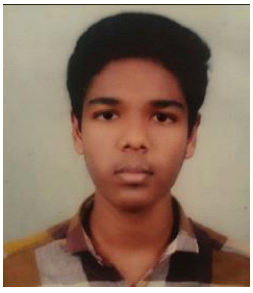
We express our sincere thanks to **Dr. Jandhyala N Murthy,** Director, GRIET**,** and **Dr. J. Praveen,** Principal, GRIET**,** for providing us the conductive environment for carrying through our academic schedules and project with ease.

We also take this opportunity to convey our sincere thanks to the teaching and non-teaching staff of GRIET College, Hyderabad.

Email:manushkumar007@gmail.com
Contact No:7659092420
Address:Nandipet, Nizamabad.

Email.suryaprakashmone98@gmail.com
Contact No: 9666619257
Address: SouthTeachersColony, Kavali

Email: deshettyrishironaldo@gmail.com
Contact No: 9100510535
Address: KPHB Colony, Hyderabad.



Email: sripathisridhar04@gmail.com
Contact No: 9505457914
Address: Vidyanagar, Karimnagar

DECLARATION

This is to certify that the project entitled "**TEXT SUMMARIZATION USING MACHINE LEARNING"** is a bonafide work done by us in partial fulfillment of the requirements for the award of the degree **BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY** from Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad.

We also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites, books and paper publications are mentioned in the Bibliography.

This work was not submitted earlier at any other University or Institute for the award of any degree.

M.SURYA PRAKASH 18241A1238

S.SRIDHAR 18241A1257

D.RISHIKESH 18241A1252

K.MANUSH 18241A1226

TABLE OF CONTENTS

11. LIST OF FIGURES

ABSTRACT

A summary is a text that produced from one or more texts, that conveys important information in original text, and it is in a shorter form. In this new era ,where tremendous information is available on the internet, it is most important to provide the improved mechanism to extract the information quickly and most efficiently. It is very difficult for human beings to manually extract the summary of a large paragraphs .Text Summarization is the process of creating a short ,accurate and fluent summary of a longer text document or any wiki. Automatic summarization system is used to reduce the user's time in reading the whole information available on web. To retrieve useful knowledge within a reasonable time period in wikipedia, it must be summarized. Reducing a text with a computer program in order to create a summary that retains the most important points in the original text. There are many reasons and uses for a summary of a larger document or paragraphs. One example that might come readily to mind is to create a concise summary of a long news article, but there are many more cases of text summaries that we may come across everyday. Text Summarization can be divided into Abstractive , Extractive summarization. Abstractive method is a best approach to deal with required information. With this the access time for searching will be improved.

Keywords : Abstractive summarization, Wikipedia, flask

Domain : Machine learning, Natural language processing techniques

1. INTRODUCTION

1.1 Introduction to Project

A summary is a text that produced from one or more texts, that converts important information in original text, and it is in a shorter form Text Summarization is the process of creating a short, accurate and fluent summary of a longer text document or any Wikipedia's URL.

There are many reasons and uses for a summary of a larger document or paragraphs. One example that might come readily to mind is to create a concise summary of a long news article, but there are many more cases of text summaries that we may come across every day.

Agenda: minimising a text with a computer code in order to create a summary that gives most important points from the original text.

1.2 Existing Systems

The model for text summarization in an extractive summarization follows semantic representation of sentences. Count based techniques like TF-IDF Vectorizer, pretrained word embedding techniques like Word2Vec and pretrained SOTA transformers like BERT or its variants are used for better capture context.

We will initially collect the large amount of text from any language resource using text corpus. By using sentence tokenizer we will whole text into sentences and words. Then TF-IDF-Vectorizer which is a statistical-measure which defines how relevant a word can weigh in content throughout the web, which is referred to as corpus. Then, with the help of sentence re-organizer, it will generate the required number of sentences are selected with highest rank in the sentence scoring which will make sure that no consecutive sentences are selected. Finally, the summary with shortening created with relevant information of original text.

This model is a complicated and a lengthy mechanism is present in it. We have to be more careful when we have to deal with this model. We are preferring to a simple model for text summarization is preferred.

1.3 Proposed System

The proposed system is followed by a text and then Sentence Segmentation, Tokenization (words, sentences), removing Stopwords tasks are applied from NLTK. A Flask application is created. It is a web framework and a python module that lets you develop web applications easily.

A web page is created by html for an easy approach. With that we can just paste any URL and get the result with important points.

These projects select only paragraphs in the contents because, important information is mainly stored in paragraphs.

Fig 1 Extractive Summarization

Fig 1 shows that our system working is based on the extractive summarization of the text summarization. We are identifying important phrases or sentences from the original text and extract only these phrases from the text. These extracted sentences would be the summary. So, text summarizing using same words which are keywords to the sentence from the original text for forming a shorter text is an extractive summarization.

2. REQUIREMENT ENGINEERING

2.1 Hardware Requirements

- Processor – i5 and above (64-bit OS).
- Memory – 4GB RAM (Higher specs are recommended for high performance)
- Input devices – Keyboard, Mouse

2.2 Software Requirements

- Windows/Mac
- Visual studio code
- Python3
- NLTK algorithm
- Flask , pandas , requests libraries

3. LITERATURE SURVEY

During late 1950 text summarization using NLP was introduced and before that time it is based on the statistical methods which is published in the year 1958. Those methods mainly involved selecting large blocks of text for generating relative and rational abstracts for the same. Furthermore, this work progressed to better and more persuasive results with the help of graph based ranking model for text processing and with Maximal Marginal Relevance (MMR) criterion as detailed. Meanwhile, evaluation measures like Bilingual Evaluation Understudy (BLEU) were invented that determined how well an automatic summary covered the matter, present in an original text. They also differentiated datasets like the DUC series, Medline, TAC series and many other were developed so that comparison and contrasting of various summarization methods would be possible for any kind of large text.

Extractive text summarization is a popular field within natural processing, which has resulted with a wide range of different methods and techniques that are to be found in the literature. Generally, extractive text summarization process used widely in any model to summarize the given text. But gradually, the practice of abstractive text summarization has brought a momentous change in this field. This might be because of the techniques that are being used under the structure based methods and the semantic based methods that condenses a text more strongly than the extraction methods. It chiefly involves converting a text into pre-processed form before finally converting it to a summary, as explained before for a particular Indian language. There are comparatively lesser works done in this process as it requires the usage of natural language generation technology, which is harder to develop. But most of the cases extractive summarization is used comparatively abstractive summarization.

Since there has been many improvements offered in the literature, many other kinds of summaries methods has also been proposed. The purpose behind the text summarization seems to be one of the most crucial aspects while choosing the algorithm. The most thriving technique from all, has been the usage of deep neural networks that are tremendously powerful machine learning models and can achieve excellent performance in sequence learning, because of their parallel computation. The Recurrent Neural Network (RNN) which is a natural generalization of feedforward neural networks to sequences that computes a sequence of outputs on the base of a given set of inputs. This is rightly elucidated in before which also debunks how RNN proves to be inefficient eventually, because of the resulting long term dependencies which is why the concept of Long Short-Term Memory (LSTM), that learns problems with long range temporal dependencies has found out to be better. As far as our research work goes, we have used encoder-decoder architecture for text summarization using Keras and Tensorflow libraries for neural networks and NIPS conference articles for the dataset. Special attention is devoted to evaluation of summarization systems, as future research on summarization is strongly dependent on progress in this area.

4. TECHNOLOGY

4.1 ABOUT PYTHON

Python is powerful and fast, plays well with others, is user friendly and easy to learn, and is open source. It is an all-around valuable programming language used in Dialog flow. It is used as a base for the most prominent Abased programming in light of its versatility, straightforwardness and longstanding reputation. Python is an interpreter, high-level, general-purpose programming language.

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which

encourages program modularity and code reuse. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

4.2 APPLICATIONS OF PYTHON

One significant advantage of learning Python is that it’s a general-purpose language that can be applied in a large variety of projects. Below are just some of the most common fields where Python has found its use:

- Data science
- Scientific and mathematical computing
- Web development
- Computer graphics
- Basic game development
- Mapping and geography (GIS software)

4.3 PYTHON IS WIDELY USED IN DATA SCIENCE

Python’s ecosystem is growing over the years and it’s more and more capable of the statistical analysis.

It’s the best compromise between scale and sophistication (in terms OD data processing).

Python emphasizes productivity and readability.

Python is used by programmers that want to delve into data analysis or apply statistical techniques (and by devs that turn to data science).

There are plenty of Python scientific packages for data visualization, machine learning, natural language processing, complex data analysis and more. All of these factors make Python a great tool for scientific computing and a solid alternative for commercial packages such as MatLab. The most popular libraries and tools for data science are:

4.3.1 PANDAS

Pandas name is derived from the term "panel data", an econometrics term for multidimensional structured data sets. It is a library for data manipulation and analysis. The library provides data structures and operations for manipulating numerical tables and time series. It is also known as "Python Data Analysis Library"

4.3.2 NUMPY

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.This is a fundamental package for scientific computing with Python, adding support for large, multi-dimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays.

4.3.3 MATPLOTLIB

Matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib allows you to generate plots, histograms, power spectra, bar charts, error charts, scatterplots, and more.

4.3.4 SCIKIT-LEARN

Scikit-learn are a machine learning library. It features various classification, regression and clustering algorithms including support vector machines,

random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

4.3.5 CSV

The so-called CSV (Comma Separated Values) format is the most common import and export format for spreadsheets and databases. The csv module implements classes to read and write tabular data in CSV format. It allows programmers to say, "write this data in the format preferred by Excel," or "read data from this file which was generated by Excel," without knowing the precise details of the CSV format used by Excel. Programmers can also describe the CSV formats understood by other applications or define their own special-purpose CSV format.

4.4 Dataset Description

Text Summarization is nothing but as we know summarizing the text into short format which will make easy to read and understand the required information from a large amount of the data or the text. There are two kinds of text summarization namely Abstractive and Extractive summarization. Text summarizing using semantic understanding that is creating our own sentences or words to summarize a given text is Abstractive Summarization. Text summarizing using same words which are keywords to the sentence from the original text for forming a shorter text.

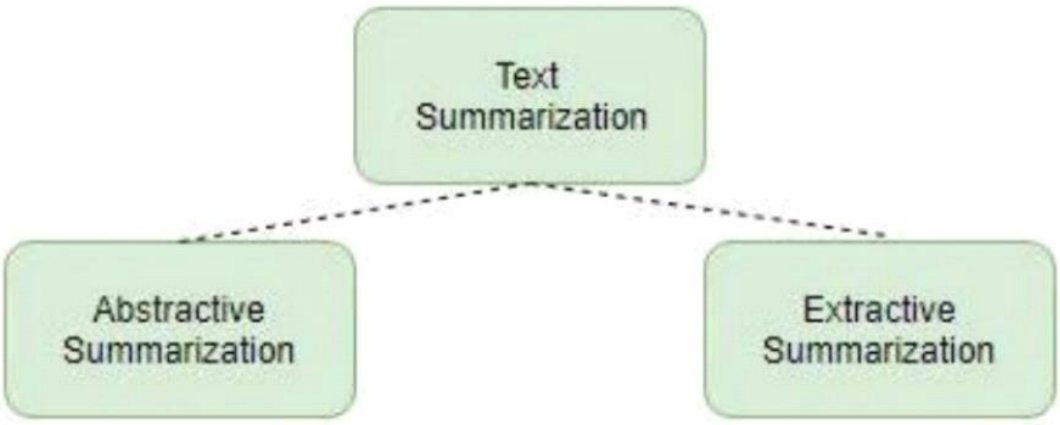


Fig 2 Classification of Text Summarization

Fig 2 shows that there are two kinds of text summarization namely Abstractive and

Extractive summarization. Text summarizing using semantic understanding that is creating our own sentences or words to summarize a given text is Abstractive Summarization. Text summarizing using same words which are keywords to the sentence from the original text for forming a shorter text.

4.4.1 Extractive Summarization

In Extractive Summarization, we are identifying important phrases or sentences from the original text and extract only these phrases from the text. These extracted sentences would be the summary, this shows that it will going to focus on using extractive methods in our model. This method functions by identifying important sentences or excerpts from the text and reproducing them as part of the summary. In this approach, no new text is generated, only the existing text is used in the summarization process.

4.4.2 Working of a NLTK Algorithm

Fig 4.4.2 describes the flow chart of our algorithm in which it shows the each process involved in our algorithm which works one after the other. We will take a text from any chrome of any Wikipedia and we copy the URL link of it. Then it will divide the given text into sentences and words which are present in human readable form that is in the form of natural language. It is done by using sentence segmentation. After that with the help of tokenization, it will protect our data by replacing with the main keywords in the sentence.

Stop words stores common words like is, was, any commas, full stops, repeating words and many other in it and removes from original sentences of a text. It will calculate the frequency of each keywords or main words in a text by using NLTK algorithm, so for which the frequency is high we select those words and forms the sentence of a text. Finally, we generate the summarization of a text by arranging them in a sequence using sentence reorganizer. Thus, we get the required summarized text.

Tokenization: Tokenization is process of dividing the stream of text entered by the user into individual group of words/ phrases/ symbols which give a meaningful sentence. This process of division is the initial step which is further taken as the next level for the preprocessing. Tokenization is also referred to as text segmentation or lexical analysis.

Stop Words: Stop Words are also referred as the “Bag of Words”. This bag of words will commonly occuring for example ‘a’, ‘an’, ‘the’, etc. words. These are explicitly recognized by the search engines and made to be ignored to contribute effective

meaning in the understanding of the text or the document.

NLTK is a platform or a package that provides a set of natural lang algorithms. It helps the computer to analysis, preprocess and understand the written text. Tokenization, Stopwords, Segmentation, Stemming etc. are the algorithms.

Removal of Stop Words: The Removal of stop words is the process where the most commonly occurring words in the document are extracted and using these words would mean to contribute very little meaning or contribute no help in the tagging system.

5. DESIGN REQUIREMENT ENGINEERING

Concept of uml :

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.ML stands for Unified Modeling Language.UML is different from the other common programming languages such as C++, Java, COBOL, etc.UML is a pictorial language used to make software blueprints.There are a number of goals for developing UML but the most important is to define some general purpose modeling language, which all modelers can use and it also needs to be made simple to understand and use.

UML DIAGRAMS:

5.1 Use case Diagram for user and summarization tool :

A use case diagram in the Unified Modeling language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals(represented as use case),and any dependencies between those cases. The main purpose of a use case diagram

is to show system functions are performed for which actor. Roles of the actors in the system can be depicted.

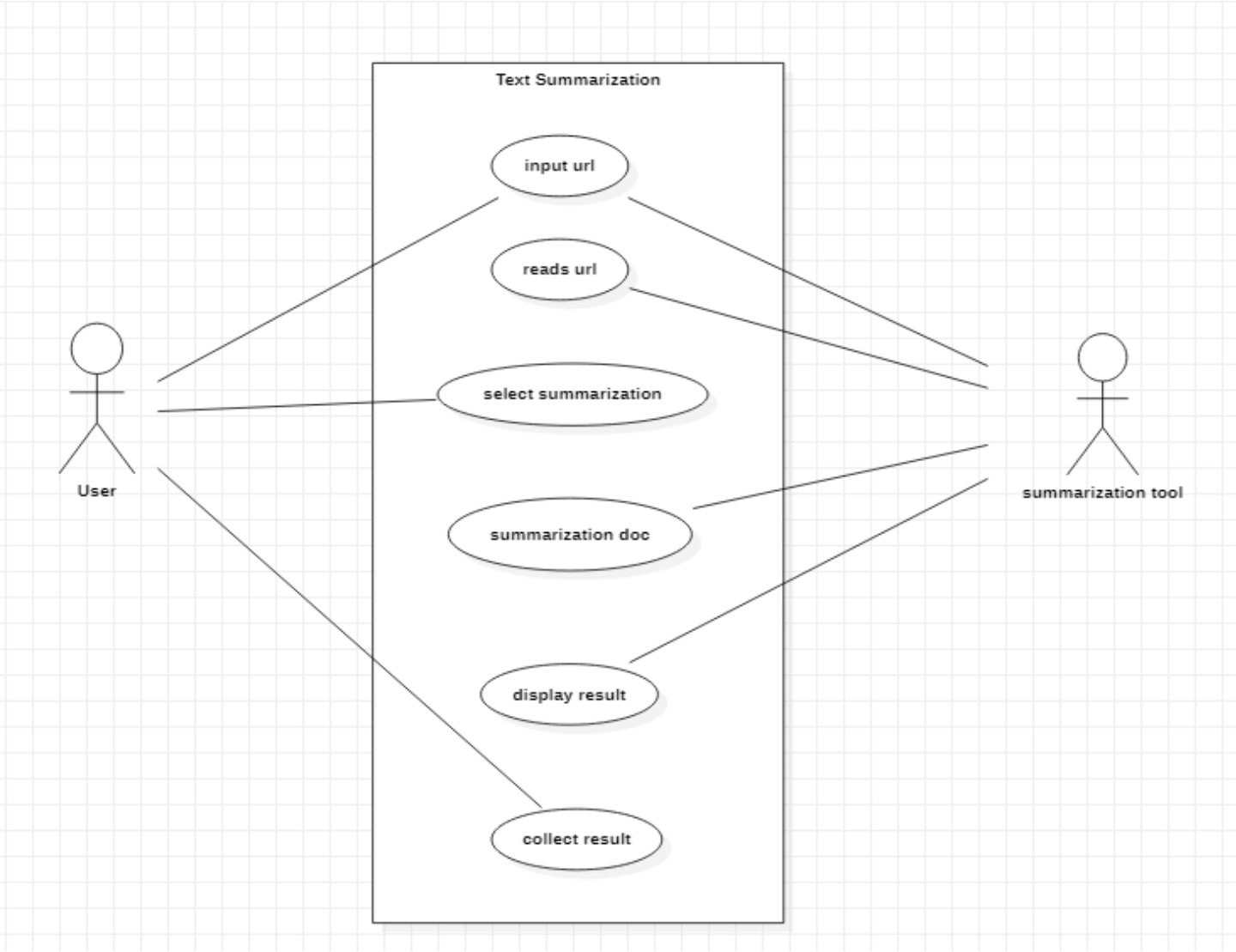


Fig 3 Use Case Diagram

Fig 3 shows the use case diagram of our system which describes the interaction between actors which are the one will interact with the subjects. In our project there are mainly two actors involved in it namely User and Summarization tool. In this diagram we will going to see the interaction involved to them.

Initially, User gives the input URL in which our tool reads it. Then User selects summarization in which the given tool will summarize it. Finally, Summarization tool will display result, then user will collect it which is our required output. So, these diagram helps our model the interaction between the system and the user.

5.2 Sequence Diagram for client and NLTK algorithm:

A sequence diagram in UML is a kind of interaction diagram which shows how each process of the system operates with one another and in what order. It is a constructed as a message sequence chart. Sequence diagrams are sometimes called event diagrams or timing diagrams.

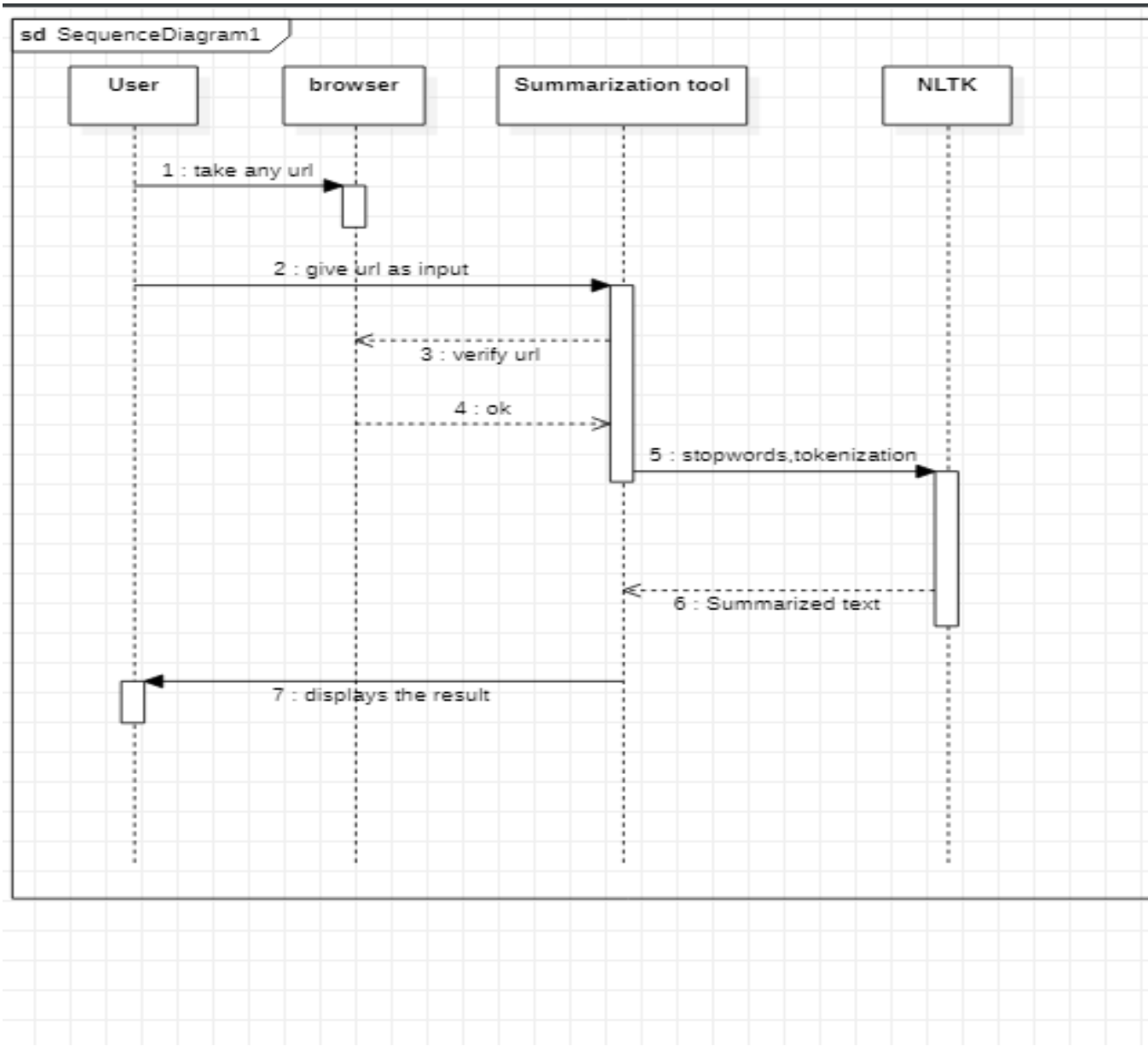


Fig 4 Sequence Diagram

Fig 4 shows the Sequence diagram of our system which is an interaction diagram in which some sequence of information flow from one object to another object in a specified order to represent the time order of a process. It aims at a specific functionality of a model.

There are mainly four objects in our project, they are as follows namely User, Browser, Summarization tool and NLTK algorithm. The information will flow from this object from one by one. Initially, User take any URL as input and Browser will give URL as input. With the help of summarization tool, it will verify URL, if it is ok, then it continues further. Then NLTK algorithm which contains in bulit stopwords, tokenization, etc. in it will perform its process in order to get summarized text. Finally, it will display the result to the user. So these diagram represents the sequence of our information flowing from one object to another object.

5.3 Activity diagram for user and server:

Activity diagram is another important behavioral diagram in uml diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.

Fig 5 shows the Activity diagram of our system which helps the model the workflow of a system from one activity to another involving different components or states like initial, final & activity states, etc. It represents the execution of the system.

We will take a HTTP server, then go to HTML page. Then it displays input from URL. If our given input data if verified, then it continues otherwise it goes back that is backtrack to input & retries again that means a loop is involved in it. If it is success, it will load text using our NLTK algorithm. Then the algorithm will browse it and gives the summarized text. So, it is one of the UML diagrams which will make logical representation of a model in which it involves branching, loops, conditions, etc

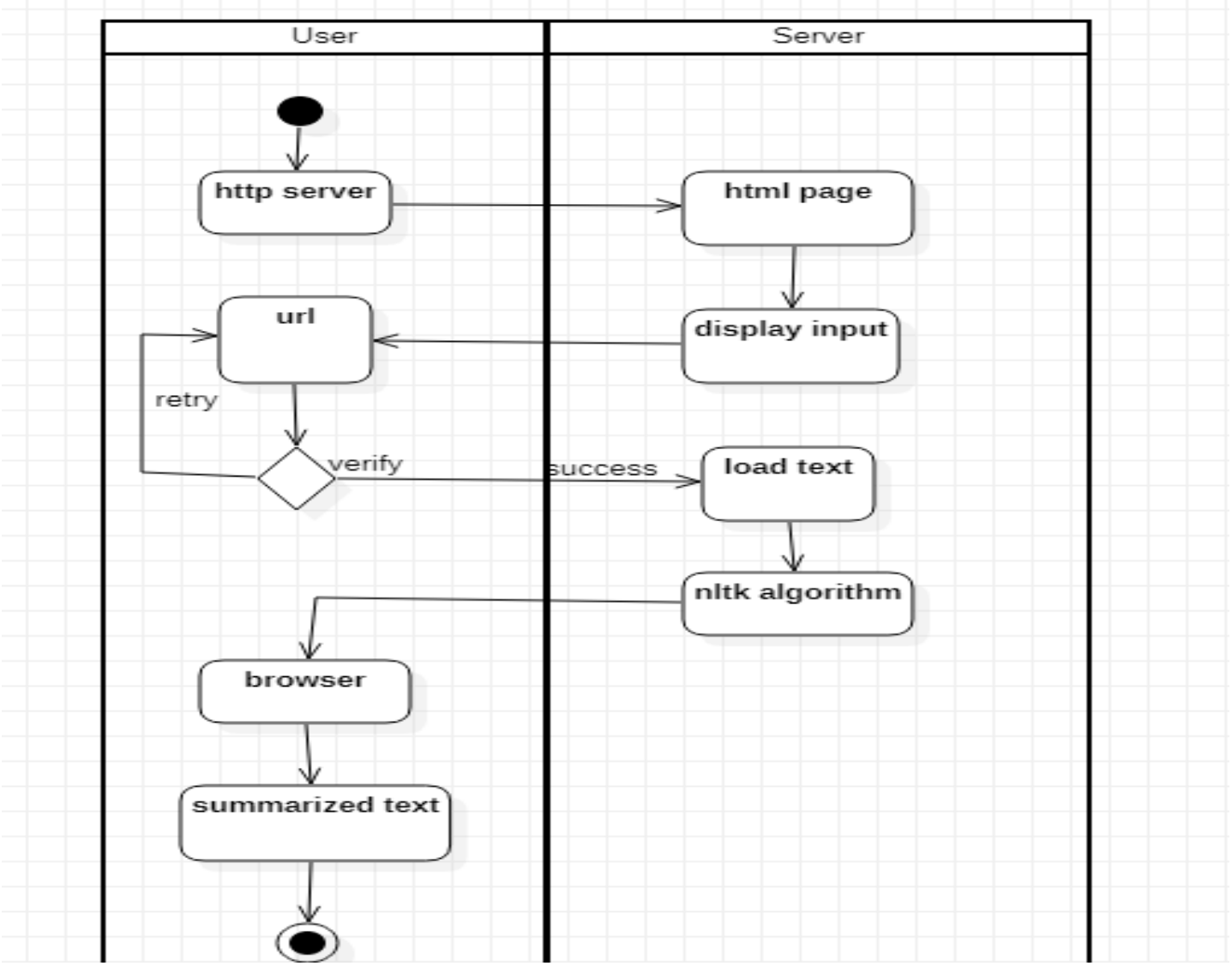


Fig 5 Activity Diagram

5.4 State Chart Diagram for user and browser:

State Chart diagram will define different states of an object during its lifetime. It describes the behaviour of systems. It requires that system described is composed of a finite no of states. Flow of control from one state to another state. It also describes the flow from one state to another state. This will normally shows how the state of an object changes in its lifetime. It also a logical view of

functionality in the model/project which contains paths, loops, conditions, etc. in it.

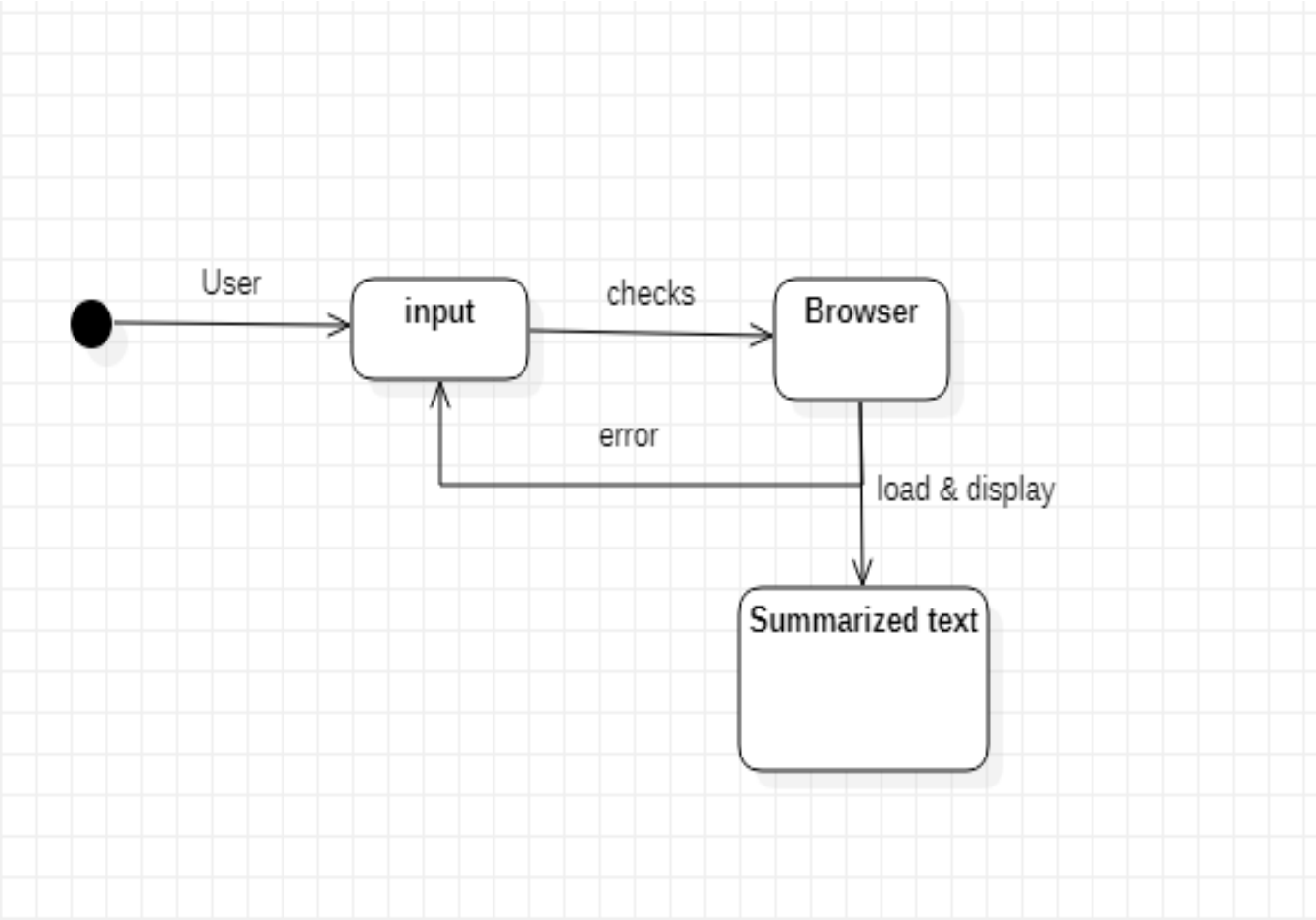


Fig 6 State Chart Diagram

Fig 6 shows the State Chart diagram of our system which will define different states of an object during its lifetime. It also describes the flow from one state to another state. It normally shows how the state of an object changes in its lifetime. It also a logical view of functionality in the model/project which contains paths, loops, conditions, etc. in it.

In these diagram, initial state would be our User which gives input. Then our Browser will check our input, if it has any error, it again goes back/backtrack towards our input that is it contains a loop. If it is ok, then it loads & display the

result which our desired output which is the summarized text. So, State Chart diagram mainly captures how a system has a state & how it behaves in that state from which it will go to another state.

5.6 Architecture

Two major text summarization approaches:
1.Abstractive Summarization 2.Extractive Summarization.

Abstractive Summarization which selects words on semantic understanding and even includes extra words. Extractive Summarization which extracts most important & meaningful sentences from text and forms a summary.

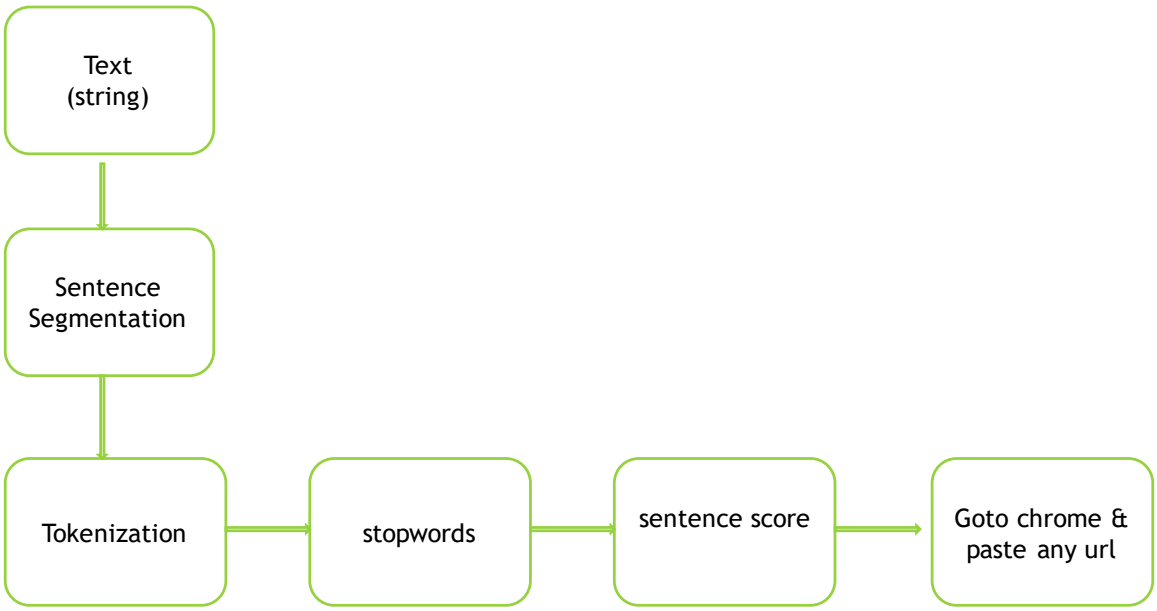


Fig 7 System Architecture

Fig 7 shows the architecture of our system in which it describes all the process, methods, functions and many more which involve in our model from the input to output. The user can give any type of large data or text as input through a URL link. Here we are using a platform/toolkit called NLTK. It contains many algorithms. The following are the processes which are involved in our algorithm and they are as follows.

The given input which is generally a text will be processed in our algorithm.

Using sentence segmentation the text is divided into sentences followed by words, these will lead to individual words. This separation of words from sentences is called word tokenization. Stopwords plays a better role in reducing the text as it removes the unecessary words and repeated words like this, is, are etc are like noises in the text. Frequency score is caluculated to find out how many times the word is repeated. With this we can reduce the count of a word by dividing it with its frequency number and then sentence score is caluculated. Finally, we get the required output which is summarized text from the given input, it shows the required result to the user. So, these are the main processes involved in our architecture and it is based on the extractive summarization of the text summarization.

6. IMPLEMENTATION

6.1 Modules Used

Tkinter: This module is used for building GUI and comes inbuilt with Python. This module comes built-in with Python.

Wikipedia: As we all know Wikipedia is a great source of knowledge just like Geeks for Geeks we have used the Wikipedia module to get information from Wikipedia or to perform a Wikipedia search. To install this module type the below command in the terminal.

Web browser: To perform Web Search. This module comes built-in with Python.

Requests: Requests is used for making GET and POST requests. To install this module type the below command in the terminal.

Beautiful Soup: Beautiful Soup is a library that makes it easy to scrape information from web pages. To install this module type the below command in the terminal.

NLTK: This is a platform for building python programs which works on normal human readable language data, it applies statical natural language processing. It contains libraries like tokenization, semantic reasoning and many.

Pandas: It is a software library for python language used for data manipulation and analysis. It is used mainly for big calculations or an bigger data, it has also Numpy in it.

Flask: Flask is a web developing application in which runs our input format that is our text data. It is a popular python web framework.

6.2 Modules

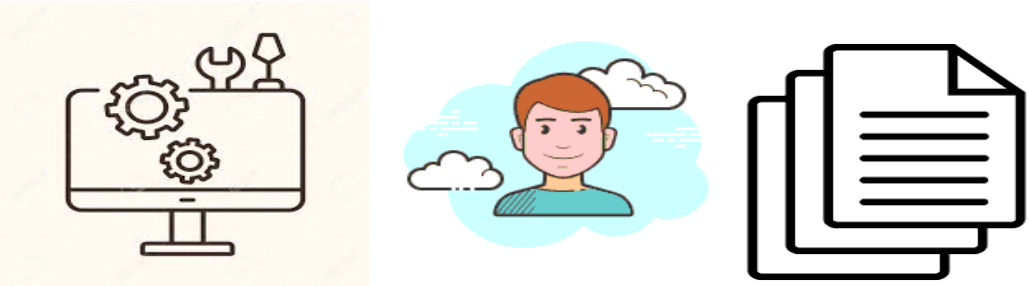


Fig 8 Modules involved in our system

Fig 8 describes the modules which are involved in our model and there are User module and System module. A collection of build settings and source files which allows us to divide our project into different units of functionality. Our project has mainly two types of modules involved in which one module as a dependency on another module. They are:

1) User Module
2) System Module

6.3 User Module

Fig 9 User Module

Fig 9 shows the main purpose of User Module is to allow the users to login in, register and log out. The member which has some certain abilities to create project is a User Module. It is beginning module of many projects including us. User module will the rights for those who want to access a particular software related to their project. This module will be in most of the projects since it is the initial functionality

of any project to describe it in the effective manner.

User browse a webpage which is a http server, go to HTML page and then copies the URL link. After that user will paste the URL link in the box which contains upload the link. If it is a valid URL link of a browser then it will continue otherwise it goes back to the user interface.

It is all about the role of user/client and its operations and what he/she is capable of the following things:

- A HTML page is created for an easy approach. This is the front end of the project.
- This webpage consists of two components:
  - INPUT: User/client paste any URL of any wikipedia in that text area of input.
  - SUBMIT: There will be a submit button which is present on next to input. When the button is clicked it redirects the user to another webpage and displays the result of the Summarized text.

6.4 System Module

Fig 10 System Module

Fig 10 describes it is a System module which defines all about the internal process as a back end of the system. There are two major approaches in these module namely: 1. Abstractive 2 .Extractive.

This module consists of the following things:

- By using Flask app, it can get the contents whatever present in that given URL. As it has two methods: GET, POST
- The text is loaded in a template document. It focuses only on paragraphs.
- It divides the paras into an individual sentences and then into a list of words.
- Removes stop words. Now frequency of each word is calculated for reducing the repeated words in the text.
- Displays the summarized text in a webpage.

6.5 Sample code

This is the main code which contains how the summarization process is happened……

```
from flask import Flask, render_template,request
import requests
from bs4 import BeautifulSoup
import nltk
import pandas as pd
app = Flask(__name__)
def get_wiki_content(url):
    req_obj = requests.get(url)
    text = req_obj.text
    soup = BeautifulSoup(text)
    all_paras = soup.find_all("p")
    wiki_text = ''
    for para in all_paras:
        wiki_text += para.text
    return wiki_text
def top10_sent(url):
    required_text = get_wiki_content(url)
    stopwords = nltk.corpus.stopwords.words("english")
    sentences = nltk.sent_tokenize(required_text)
    words = nltk.word_tokenize(required_text)
    word_freq = {}
    for word in words:
        if word not in stopwords:
            if word not in word_freq:
```

```
            word_freq[word] = 1
        else:
            word_freq[word] += 1
    max_word_freq = max(word_freq.values())
    for key in word_freq.keys():
        word_freq[key] /= max_word_freq
    sentences_score = []
    for sent in sentences:
        curr_words = nltk.word_tokenize(sent)
        curr_score = 0
        for word in curr_words:
            if word in word_freq:
                curr_score += word_freq[word]
        sentences_score.append(curr_score)
sentences_data = pd.DataFrame({"sent":sentences, "score":sentences_score})
    sorted_data = sentences_data.sort_values(by = "score", ascending = False).reset_index()
top10_rows = sorted_data.iloc[0:11,:]
    return " ".join(list(top10_rows["sent"]))
@app.route("/", methods = ["GET", "POST"])
def index():
    if request.method == "POST":
        url = request.form.get("url")
        url_content = top10_sent(url)
        return url_content
    return render_template("index.html")
if __name__ == "__main__":
    app.run(debug=True)
```

**For the front-end, one can use any type of html with their respective styles.

7. SOFTWARE TESTING

7.1 Unit Testing:

Unit testing is carried out for testing modules constructed from the

system design. Each part is compiled using inputs for specific modules. Every module are assembled into a larger unit during the unit testing process. Testing has been performed on each phase of project design and coding. The testing of module interface is carried out to ensure the proper flow of information into and out of the program unit while testing. The temporarily generated output data is ensured that maintains its integrity throughout the algorithm's execution by examining the local data structure. Finally, all error-handling paths are also tested.

7.2 Integration Testing:

We usually perform system testing to find errors resulting from unanticipated interaction between the sub-system and system components. Software must be tested to detect and rectify all possible errors once the source code is generated before delivering it to the customers. For finding errors, series of test cases must be developed which ultimately uncover all the possibly existing errors. Different software techniques can be used for this process. These techniques provide systematic guidance for designing test that exercise the internal logic of the software components and exercise the input and output domains of a program to uncover errors in program function, behaviour and performance. We test the software using two methods: White Box testing: Internal program logic is exercised using this test case design techniques. Black Box testing: Software requirements are exercised using this test case design techniques. Both techniques help in finding maximum number of errors with minimal effort and time.

7.3 Acceptance Testing:

The testing process is a part of broader subject referring to verification

and validation. We must acknowledge the system specifications and try to meet the customer's requirements and for this sole purpose, we must verify and validate the product to make sure everything is in place. Verification and validation are two different things. One is performed to ensure that the software correctly implements a specific functionality and other is done to ensure if the customer requirements are properly met or not by the product. Verification of the project was carried out to ensure that the project met all the requirement and specification of our project. We made sure that our project is up to the standard as we planned at the beginning of our project development.

7.4 Testing on our System:

Every model involves coding part and testing part since they are the two things for completing a system successfully. In our code it contains both html code and python code for summarization of the text. The algorithm we are using is the NLTK algorithm that is Natural Language Toolkit in which it undergoes two types of software testing White Box testing and Black Box testing. We have already discussed the modules used in our project these modules will undergo the unit testing in our system.

The URL link of a text which is our input and summarized text from given input is our output involves various kinds of testing in our system. Integration testing will be there for checking the software requirements of our system, it sees whether our system has capacity to get the required output. It will test RAM, OS, libraries required for process the code, coding platform whether it has capability to get the code without any errors especially compile time errors and many other software requirements. From assigning the path of input, adding required header files and libraries, different variables for each process involved in our algorithm, in built html code and this code involving in the main code is observed by the testing, if any kind of errors occurs it will be done by the integration testing. If everything goes well in our code that is sentence segmentation function in which this function defines division of sentences and then words int the main code. The tokenization function in which this function defines for calculating the value of each word in the main code. Stop words are defined under the main code itself, it is a variable in which it stores

unnecessary words before executing the other functions in the code. A function will form sentences by giving importance to the most valued words which our given by the tokenizer function. After completion of the testing, if it has no errors in the entire code that is both the html code and the main code, it will generate the output for the user. In this way, testing in our system is validated and verified as we planned at the beginning of our system.

8. RESULTS

This interface contains a input option and a submit button. Anyone can give any Wikipedia URL and kick the submit button.

Input:

https://en.wikipedia.org/wiki/Ratan_Tata

Output:

[9] He studied at the Campion School, Mumbai till the 8th class, followed by Cathedral and John Connon School, Mumbai and at Bishop Cotton School in Shimla,[10] and, in 1955, graduated from Riverdale Country School in New York City. [5] Born in 1937, he is a scion of the Tata family, and son of Naval Tata who was later adopted by Ratanji Tata, son of Jamsetji Tata, the founder of Tata Group. [22] Ratan Tata resigned his executive powers in the Tata group on 28 December 2012, upon turning 75, appointing as his successor, Cyrus Mistry, the 44-year-old son of Pallonji Mistry of the Shapoorji Pallonji Group, the largest individual shareholder of the group and related by marriage. [60] In October 2016, Tata Sons removed Cyrus Mistry as its chairman, nearly 4 years after he took over the reins of the over $100 billion conglomerate, Ratan Tata made a comeback, taking over the company's interim boss for 4 months. "[36] In one of the most dramatic developments in the recent past, the board of directors of Tata Group on 24 October 2016 voted for the removal of its chairman Cyrus Mistry with immediate effect and made Ratan Tata the interim chairman, and in February 2017, Mistry was removed as a director for Tata Sons. Tata invested personal savings in Snapdeal – one of India's leading e-commerce websites –and, in January 2016, Teabox, an online premium Indian Tea seller,[27] and CashKaro.com, a discount coupons and cash-back website. [33][34][35] Tata Motors rolled out the first batch of Tigor Electric Vehicles from

its Sanand Plant in Gujarat, regarding which Ratan Tata said, "Tigor indicates a willingness to fast-forward India's electric dream. He was also chairman of Tata Group, from 1990 to 2012, and again, as interim chairman, from October 2016 through February 2017, and continues to head its charitable trusts. He got Tata Tea to acquire Tetley, Tata Motors to acquire Jaguar Land Rover, and Tata Steel to acquire Corus, in an attempt to turn Tata from a largely India-centrist group into a global business. He continues to head the main two Tata trusts Sir Dorabji Tata Trust and Sir Ratan Tata Trust and their allied trusts, with a combined stake of 66% in Tata Sons, Tata group's holding company. Ratan Tata was born in Bombay, now Mumbai, on 28 December 1937,[7] and is the son of Naval Tata (born in Surat).

9. CONCLUSION AND FUTURE ENHANCEMENTS

9.1 Conclusion

Text summarizer is a web application which helps in summarizing the text. With the help of python and visual studio code, it will become easy to express the important information in the final summary. The access time for information searching will be improved. From this we can conclude that most our model is based on extractive methods. As we know time is most valuable thing in the world it will reduce our time and gain the relevant information from a large texts or data. The access time for information searching will be improved. It removes the wastage or repeated data**.** From this, we can conclude that most of the all summarization techniques are based on the Extractive methods.

9.2 Future enhancements

In this project, NLP and NLTK algorithm takes a very important role in new machine human interface. When we look at some of the products based on the technologies of it, we can see that they are very much advanced but very useful as well. There are many languages spoken around the globe, thus it is difficult in building a model to predict accurate results. This problem gets more complicated when we think of different people speaking the same language but in different styles. This approach has boosted the probabilities of content words. This approach has further enhanced the probabilities of content words in the above proposed system. Significant improvements in perplexity have been observed in topic specific and domain independent models.

10. BIBLIOGRAPHY

[1] Alterman R. and Bookman AL., "Some computational experiments in summarisation" in Discourse Processes 13 pp 143-174. 1990.

[2] Hearst M., "Multi-Paragraph Segmentation of Expository Text", Proceedings of ACL-94, Las Cruces, New Mexico, 1994.

[3] Sparck Jones K., Discourse modeling for automatic summarizing, Technical Report No. 290, University of Cambridge Computer Laboratory, 1993.

[4] Teufel, S. and Moens, M, "Sentence extraction as a classification task", in Mani, I., and Maybury, M., eds., Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization, Madrid, Spain, 1997.

[5]https://www.analyticsvidhya.com/blog/2019/06/comprehensive-guide-text-summarization-using-deep-learning-python/

[6]**https://towardsdatascience.com/the-secret-guide-to-human-like-text-summarization-fcea0bfbe801**

[7]**https://analyticsindiamag.com/hands-on-guide-to-extractive-text-summarization-with-bertsum/**

[8] **https://github.com/surya737/mini-prooject**