AN EFFECTIVE TEXT SUMMARIZER USING NATURAL LANGUAGE PROCESSING

A Main Project Report submitted in partial fulfillment of the

requirements for the awards of degree of

BACHELOR OF TECHNOLOGY
In
COMPUTER SCIENCE AND ENGINEERING

Submitted By

A. APARNA	**(16A91A05C4)**
S. HIMASRI	**(16A91A05H0)**
B. SAI PRADEEP	**(16A91A05C7)**
T. NAGENDRA	**(16A91A05H5)**

Under the esteemed guidance of

Mr. B.R.S.S. RAJU, MCA, M.Tech

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
ADITYA ENGINEERING COLLEGE (A)
Approved by AICTE, Permanently affiliated to JNTUK & Accredited by NBA, NAAC with ‘A’ Grade
Recognized by UGC under the sections 2(f) and 12(B) of the UGC act 1956
Aditya Nagar, ADB Road - Surampalem – 533437, E.G.Dist., A.P.,
2016-2020

ADITYA ENGINEERING COLLEGE (A)

Approved by AICTE, Permanently affiliated to JNTUK & Accredited by NBA, NAAC with 'A' Grade
Recognized by UGC under the sections 2(f) and 12(B) of the UGC act 1956
Aditya Nagar, ADB Road - Surampalem – 533437, E.G.Dist., A.P.,
2016-2020

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Project work entitled "**AN EFFECTIVE TEXT SUMMARIZATION USING NATURAL LANGUAGE PROCESSING**" is being submitted by

A. APARNA	**(16A91A05C4)**
S. HIMASRI	**(16A91A05H0)**
B. SAI PRADEEP	**(16A91A05C7)**
T. NAGENDRA	**(16A91A05H5)**

In partial fulfillment of the requirements for award of the B.Tech degree in Computer Science and Engineeringfor the academic year 2019-2020.

B.R.S.S.Raju

Project Guide
Mr. B.R.S.S.Raju, MCA, M.Tech
Assistant Professor,
Department of CSE

Head of the Department
Mrs.A.Vanathi, M.E.,(Ph.D),
Associate Professor
Department of CSE

External Examiner

DECLARATION

We hereby declare that the project entitled **“AN EFFECTIVE TEXT SUMMARIZATION USING NATURAL LANGUAGE PROCESSING”** is a genuine project. This work has been submitted to the **ADITYA ENGINEERING COLLEGE,** Surampalem affiliated to **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA** in partial fulfillment of the **B.Tech**degree**.** I/We further declare that this project work has not been submitted in full or part of the award of any degree of this or any other educational institutions.

BY

A. APARNA(16A91A05C4)

S. HIMASRI(16A91A05H0)

B. SAI PRADEEP(16A91A05C7)

T. NAGENDRA(16A91A05H5)

ACKNOWLEDGEMENT

It is with immense pleasure that we would like to express our indebted gratitude to our project supervisor **Mr. B.R.S.S. Raju, Assistant Professor,** who has guided us a lot and encouraged us in every step of the project work, her/his valuable moral support and guidance throughout the project helped us to a greater extent.

We wish to thank our Project Coordinator **Mrs. N. Akhila**, Sr.Assistant Professor in CSE Department for his/her support and valuable suggestions for the successful completion of this project.

Our deepest thanks to Our HOD **Mrs. A. Vanathi, Associate Professor, and Department of CSE** for inspiring us all the way and for arranging all the facilities and resources needed for our project.

We wish to thank our Dean academics,**Dr. S. Rama Sree** for her support and suggestions during our project work.

We owe our sincere gratitude to our Principal **Dr. M. Sreenivasa Reddy,** for providing a great support and for giving us the opportunity of doing the project.

We are thankful to our **College Management** for providing all the facilities in time to us for completion of our project.

Not to forget, **Lab Technicians, non-teaching staff and our friends** who have directly or indirectly helped and supported us in completing our project in time.

ABSTRACT

The technique in which a computer coded program can abridge the obtained text to a summary that contains the most essential points from the primary text is known as Automatic Text summarization. The need for automatic text summarization is paramount today as the surplus amount of research and data is available and accessible on the web. In recent years, condensing of information can be witnessed in various domains. Web search engines display excerpts on the results page from the data which is within the websites. The digital newsfeeds provide concise information, the essence of the long news stories, to the readers. Reading and identifying the relevant points are the two processes that are necessary to summarize the text and for the summary to be semantically comprehensive, natural language processing is employed. Web scraping is a methodology where a wealth of useful data, which in general is manually gleaned from various sources, is automatically procured by the system from the web for further implementations. In this process, the website is fetched first and then the data required is harvested from it which gets stored in either system's database or in spreadsheets. This text when given as the input to the summarization program gets identified and the location of all discrete sentences is detected. Extractive summarization picks up sentences directly from the document based on a scoring function to form a coherent summary. This method work by identifying important sections of the text cropping out and stitch together portions of the content to produce a condensed version. The salient words are quantified by their frequency and each sentence is ranked on the basis of the cumulative sum obtained from the normalized score of the salient words. The highest-scoring sentences are then conjoined to form the summary.

INDEX

Chapter No	Content	PageNo
1	**Intorduction**	**01**
	1.1Methodologies	02
	1.2Existing System	03
	1.3Proposed System	03
2	**Requirement Analysis**	**04**
	2.1 Hardware and Software Requirements	04
	2.2 Software Requirements Specification	05
	2.3 Goal	09
	2.4 Objective	09
	2.5 Functional Requirement	09
	2.6 Non Functional Requirement	10
3	**System Design**	**11**
	3.1 Data Dictionary	11
	3.2 Methodology	12
	3.3 UML Diagrams	22
4	**System Implementation**	**30**
	4.1 Selected Software	30

4.2 Sample Code 38

5 Testing 45

5.1 White box testing 45

5.2 Black box testing 45

5.3 Test cases 46

6 Screens and Outputs 47

6.1 Screenshots 47

6.2 Reports 52

7 Conclusion and Future Scope 53

7.1 Conclusion 53

7.2 Future Scope 54

Bibiliography 55

LIST OF FIGURES

Figure	Name of the Figure	PageNo
3.1	Natural Language Processing Venn Representation	12
3.2	Use case representation of text summarizer	23
3.3	Activity Diagram of text summarizer	25
3.4	Sequence Diagram for summarizing given text	26
3.5	Sequence Diagram for summarizing contents of given URL	27
3.6	Collaboration Diagram of summarizing given text	28
3.7	Collaboration Diagram of summarizing text in given URL	29
3.8	System Architecture of Text Summarizer	29
4.1	Installation of flask	41
4.2	Installation of spaCy	42
4.3	Installation of beautifulsoup	43
4.4	Installation of bs4	43
4.5	Installation of urllib	44

LIST OF TABLES

S.No	Table	Name of the Table	Page No
1	2.1	Usecase - User	06
2	2.2	Detailed Use Case Description for Summarizing Text	07
3	2.3	Detailed Use Case Description for Summarizing Text in URL	08
4	3.1	Data Dictionary	11
5	3.2	Weighted Frequencies of Tokens	18
6	3.3	Cummulative Sum of Sentences	19
7	4.1	Features of spaCy	37
8	5.1	Test Cases for Home Page	46
9	6.1	Reports	52

LIST OF SCREENSHOTS

S.No	Screenshot No	Name	Page No
01	6.1	Header of Home Page	47
02	6.2	Home Page	48
03	6.3	Text given as Input to Summarize	49
04	6.4	URL given as Input to Summarize	50
05	6.5	Output Screen	51

1. INTRODUCTION

The technique in which a computer coded program can abridge the obtained text to a summary that contains the most essential points from the primary text is known as Automatic Text summarization. The need for automatic text summarization is paramount today as the surplus amount of research and data is available and accessible on the web. In recent years, condensing of information can be witnessed in various domains. Web search engines display excerpts on the results page from the data which is within the websites. The digital newsfeeds provide concise information, the essence of the long news stories, to the readers. Reading and identifying the relevant points are the two processes that are necessary to summarize the text and for the summary to be semantically comprehensive, natural language processing is employed. Web scraping is a methodology where a wealth of useful data, which in general is manually gleaned from various sources, is automatically procured by the system from the web for further implementations. In this process, the website is fetched first and then the data required is harvested from it which gets stored in either system's database or in spreadsheets. This text when given as the input to the summarization program gets identified and the location of all discrete sentences is detected. Extractive summarization picks up sentences directly from the document based on a scoring function to form a coherent summary. This method work by identifying important sections of the text cropping out and stitch together portions of the content to produce a condensed version. The salient words are quantified by their frequency and each sentence is ranked on the basis of the cumulative sum obtained from the normalized score of the salient words. The highest-scoring sentences are then conjoined to form the summary.

1.1 Methodologies

1.1.1 Extractive Summarization

The extractive text summarization technique involves pulling key phrases from the source document and combining them to make a summary. The extraction is made according to the defined metric without making any changes to the texts. The extractive approach basically choose the various and unique sentences, sections and so forth make a shorter type of the first report. The sentences are estimated and chosen based on accurate highlights of the sentences. In the Extractive technique, we have to choose the subset from the given expression or sentences in given frame of the synopsis. The extractive outline frameworks depends on two methods i.e. - extraction and expectation which includes the arrangement of the particular sentences that are essential in the general comprehension the archive.

1.1.2 Abstractive Summarization

The abstraction technique entails paraphrasing and shortening parts of the source document. When abstraction is applied for text summarization in deep learning problems, it can overcome the grammar inconsistencies of the extractive method. The abstractive text summarization algorithms create new phrases and sentences that relay the most useful information from the original text just like humans do.

1.2 Existing System

In existing system, to summarize a large size text we have to go through the entire document line to line. We have to read it thoroughly, understand the meaning, decide on what points are actually important and need to be mentioned so that it precisely describes the entire text document. This is really a time taking process. To know the content of a URL we actually need to dig deep into it to understand how relevant it is to the information we are looking for. It takes away a lot of time. Manually going through the large text or the contents of a URL is a strenuous and tedious task. This is the existing system that is being followed to summarize a large text. A lot of effort and time are taken in this existing system.

1.3 Proposed System

Automatic text summarization is a web platform where users can give the large text that they want to be summarized. They can give the large text in the web page itself for it to be summarized or the URL of the website whose information is needed to be summarized can also be given. In both the cases summary is obtained and shown on the webpage itself. The whole text that is taken by the summarizer to summarize is shown on left side and the text that is summarized is shown on the right side of the webpage. Natural language processing techniques are used to obtain the summary from large texts. We need not shift through redundant and insignificant texts. It enhances readability of the document. Reading times of the actual large text and the summarized text is displayed right above the respective paragraphs. Reading time is the approximate time taken to read the paragraph. Reading time of actual text and summarized text varies greatly in most of the cases which shows the amount of time the text summarizer is saving for the user. Time elapsed is also displayed at the bottom which is the time that has been taken to summarize the text. Time elapsed is much less the time taken to summarize the text manually and hence saving a lot of effort along with time.

2. REQUIREMENT ANALYSIS

2.1 Hardware and Software Requirements

Hardware requirements for **Automatic Text Summarizer** are

- Processor: Minimum 1GHz. Recommended 2GHz or more.
- Ethernet connection (LAN) or a wireless adapter (Wi-Fi).
- Hard Drive: 2GB Free hard disk space
- Memory (RAM): Minimum 4GB.
- Operating system Windows 7 or newer.

Software requirements for **Automatic Text Summarizer** are

- HTML5
- CSS
- JavaScript
- Bootstrap
- Python
- Flask

HTML5, CSS is used for building static webpages. Flask is used to build the user interface of the page. Python is used for backend operations.

2.2 Software Requirements Specification

The following section provides an overview of the derived Software Requirements Specification (SRS) for the subject text Summarization. The document is presented and its intended audience outlined. Subsequently, the scope of the project specified by the document is given with a particular focus on what the resultant software will do and the relevant benefits associated with it. The nomenclature used throughout the SRS is also offered. To conclude, a complete document overview is provided to facilitate increased reader comprehension and navigation.

2.2.1 Vision

- For a user who wants to get a summary of a large text, the Text Summarizer provides a simple way to summarize the content.
- The Text summarizer is a webpage that saves a lot of time and effort of the user.
- The user will give the text or the URL of the website containing the text to be summarized in the Web User Interface.
- The summarized text is displayed on the Web User Interface.

2.2.2 Scope

The proposed web platform is for the user who wants to summarize their large text data. All they need to do is give the text that they want to summarize or the URL of the website containing information that is to be summarized. This helps the users save a lot of time and reduce the efforts that they put into summarizing large texts.

Any person who wants to get a summary of some text can use the Text Summarizer

- Exclusions:
- Text documents and Portable Document Format files cannot be uploaded.
- Images cannot be uploaded.
- It summarizes only text and not any other formats.

Assumptions:

- A user either gives the text to be summarized directly in the webpage or gives the URL of the website whose information is needed to be summarized.

2.2.3 System Functions

User

- Opens the website
- Gives text or URL of the website
- Clicks summarize

2.2.4 Detailed Software Requirements

Actor Name	User
Actor Id	TC01
Description	Gives text data or URL of website whose data needs to be summarized
Main Activities	Give text data or URL
Frequency of Use	Medium
Work Environment / Location	Browser
Number of Users	Any number of users

Table 2.1 Usecase - User

2.2.5 Detailed Use Case Description

<table>
<tr><td><b>Use Case Name</b></td><td>Summarize the text data</td></tr>
<tr><td><b>Use Case ID</b></td><td>UC1</td></tr>
<tr><td><b>Actor(s)</b></td><td>User</td></tr>
<tr><td><b>Goal</b></td><td>To get summary of large text data</td></tr>
<tr><td><b>Summary</b></td><td>The user clicks on the summarize to get the summary of text data</td></tr>
<tr><td><b>Preconditions</b></td><td>Requires internet connection</td></tr>
<tr><td><b>Main Flow</b></td><td><table>
<tr><td>1. The user enters the text data</td><td>1.1 The system displays the option summarize</td></tr>
<tr><td>2. User selects summarize</td><td>2.1 The system shows the summary of the text data</td></tr>
</table></td></tr>
<tr><td><b>Alternate Flows</b></td><td>The user can come out the window directly.</td></tr>
</table>

<table>
<tr><td><b>Use Case Name</b></td><td>Summarize the text data in website of given URL</td></tr>
<tr><td><b>Use Case ID</b></td><td>UC2</td></tr>
<tr><td><b>Actor(s)</b></td><td>User</td></tr>
<tr><td><b>Goal</b></td><td>To get summary of large text data in given website URL</td></tr>
<tr><td><b>Summary</b></td><td>The user clicks on the summarize to get the summary of text in website</td></tr>
<tr><td><b>Preconditions</b></td><td>Requires internet connection</td></tr>
<tr><td><b>Main Flow</b></td><td><table>
<tr><td>1. The user enters the URL</td><td>1.1 The system displays the option summarize</td></tr>
<tr><td>2. User selects summarize</td><td>2.1 The system shows the summary of the text in website URL</td></tr>
</table></td></tr>
<tr><td><b>Alternate Flows</b></td><td>The user can come out the window directly.</td></tr>
</table>

2.3 Goal

Much of the current work is focused in two major directions:

- Summarizing the text given or the content of the website URL given.
- Reduce efforts and save time of people to summarize manually.

2.4 Objective

The objective of our work is to:

- Summarize text without manually reading entire text.
- Web scraping the contents of the website and summarizing them.
- Display the reading times of original and summarized text.

2.5 Functional Requirement

The functional requirements describe the core functionality of the application.

2.5.1 Interface Requirement:

- Screen 1 to accept user input text data.
- Field 1 accepts Text data.
- Field 2 accepts URL of a website.
- Button 1 to send text data from Field 1 to backend.
- Button 2 to send URL from Field 2 to backend.

2.6 Non Functional Requirement

Non function requirement are those requirement of the system which are not directly concerned with specific functional delivered by the system. They may be related to emergent properties such as reliability, extendibility, usability, etc.

- To provide summary of text.
- To provide maximum accuracy.
- Provide reading time analysis.
- Ease of use.
- Availability
- Reliability
- Maintainability

3. SYSTEM DESIGN

3.1 Data Dictionary

TERM	DEFINITION
Admin	System admin is a person who is responsible for managing the whole system and who has full access to the system.
User	A person who is using the system but with a limited privilege
Data set	Collection of all the information monitored by this system.
Software Requirements Specification (SRS)	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. Admin is any person who is involved in the development process of the software.
Web User Interface	An application which can be accessed by user using any standard web browser and Internet Connection.
Admin	The person who can make changes to the software.
User	The person who interacts with the interface to get benefit.

Table 3.1 Data Dictionary

3.2 Methodology

3.2.1 Natural Language Processing

Natural Language Processing (NLP) is the intersection of Computer Science, Linguistics and Machine Learning that is involved with the interaction between computers and humans in natural language.

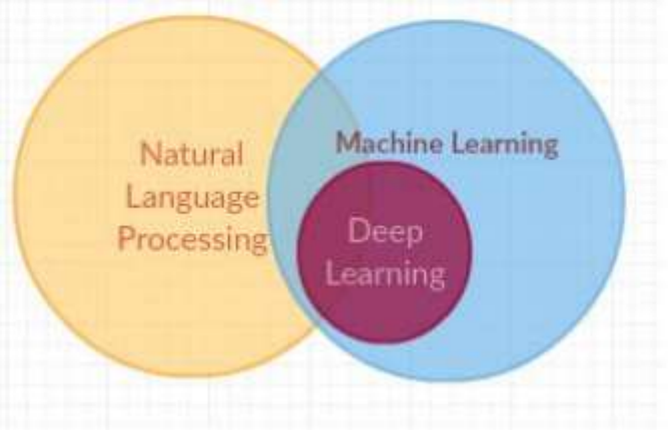


Fig 3.1 Natural Language Processing Venn Representation.

NLP is way toward empowering PCs to comprehend and deliver human dialect. Uses of NLP systems are utilized in separating of text, machine interpretation and Voice Agents like Alexa and Siri. NLP is one of the fields that are profited from the advanced methodologies in Machine Adapting, particularly from Profound Learning strategies. Regular Dialect Preparing method utilize the characteristic dialect toolbox for making the principle arrange in python tasks to work with human dialect data. This is simpler to-use by giving the interfaces to at least one than 40 corpora and dictionary resources, for portrayal, for part passages sentences and to get the words in its unique frame Marking, parsing, and glossary thinking for current reasoning quality basic dialect dealing with libraries, and for dynamic discourse.

3.2.2 Web Scraping

Web Scraping (also termed Screen Scraping, Web data Extraction, Web Harvesting, etc.) is a technique employed to extract large amounts of data from websites whereby the data is extracted and saved to a local file in your computer or to a database in table (spread sheet) format.

Data displayed by most websites can only be viewed using a web browser. The only option then is to manually copy and paste the data - a very tedious job which can take many hours or sometimes days to complete. Web Scraping is the technique of automating this process, so that instead of manually copying the data from websites, the Web Scraping software will perform the same task within a fraction of the time.

Here, in Text Summarizer we want our data to be fetched into backend directly. For this purpose we use urllib library. urllib is a Python module that can be used for opening URLs. It defines functions and classes to help in URL actions. With Python you can also access and retrieve data from the internet like XML, HTML, JSON, etc. You can also use Python to work with this data directly. From the data that is fetched from web scraping, raw text data in obtained and summarized.

3.2.3. Methodologies:

Step 1: It's a breeze

This step involves the process of breaking the given input paragraphs into a subset of sentences. In other words, the source text will disintegrate into different phrases.

Source text*:* The hardest part of your journey to success is dealing with people who will try to impose their limited views upon you. Those people think that they have a right to shut someone down and kill their dreams. Their self-limiting beliefs confine them to a mediocre life. They will tell you that you need to lower your expectations. They will tell you that you need to be more realistic. They will try to make you believe that you cannot achieve what you set your mind to. These are the kind of people you need to cut out of your life. When you are on your road to success, you should never let anyone tell you that you can't achieve your dreams. It is our individual passions, our dreams, and our actions that make us who we are and help us on our way to success. It's about learning to love yourself and constantly striving for success in such a way that other people could never diminish you.

After step 1:

1. The hardest part of your journey to success is dealing with people who will try to impose their limited views upon you.
2. Those people think that they have a right to shut someone down and kill their dreams.
3. Their self-limiting beliefs confine them to a mediocre life.
4. They will tell you that you need to lower your expectations.
5. They will tell you that you need to be more realistic.
6. They will try to make you believe that you cannot achieve what you set your mind to.
7. These are the kind of people you need to cut out of your life.
8. When you are on your road to success, you should never let anyone tell you that you can't achieve your dreams.
9. It is our individual passions, our dreams, and our actions that make us who we are and help us on our way to success.
10. It's about learning to love yourself and constantly striving for success in such a way that other people could never diminish you.

Step 2: Pre-processing of text

As there are unlimited sources of information in the modern world, the input document that we receive may not be in proper English format i.e. it might contain noise in it. The noise might be in the form of special characters, unwanted spaces, new line characters, stop words, etc. Thus, we perform the following operations in the input file to obtain only an informative part of the document:
Step 1: Remove all new line and carriage return characters
Step 2: Remove all brackets and special symbols with numbers
Step 3: Remove all commas, extra spaces and duplicate sentences.

In these steps removing all stop words from given input as per natural language. These stop word which does not give any meaningful information for the given context. For illustration, as if we are developing an emotion detection collection of words like "is", "am", and "the" which do not convey any information relate to that emotion. For instance, in this given sentence "I am feeling sad today", the beginning two words "I" and "am" this can be removed because these words does not providing emotion related message. Even though, "I" words is required as per importance of other reasons, to know who is feeling sad like identification. So there is not specific common universal stop word list which helps to remove it. Hence, it’s totally depends upon user defined application. In natural Language processing, for each word processing is required. By that it is suitable that has only those processed words in our performed text that are important in a context, by that we can save time in processing and results in a more robust NLP engine.

After step 2:

1. hardest part journey success dealing people who try impose limited views
2. people think right shut someone down kill dreams
3. self-limiting beliefs confine mediocre life
4. tell need lower expectation
5. tell need more realistic
6. try make believe cannot achieve set mind
7. people cut out life
8. road success never let anyone tell cannot achieve dreams
9. individual passions dreams actions make help way success
10. learning love yourself constantly striving success people never diminish

Step 3: Tokenization

It is the process of dividing the given input sentences into a set of discrete words.

After step 3:

{'hardest', 'part', 'journey', 'success', 'dealing', 'people', 'who', 'try', 'impose', 'limited', 'views', 'think', 'right', 'shut', 'someone', 'down', 'kill', 'dreams', 'self-limiting', 'beliefs', 'confine', 'mediocre', 'life', 'tell', 'need', 'lower', 'expectation', 'more', 'realistic', 'make', 'believe', 'cannot', 'achieve', 'set', 'mind', 'cut', 'out', 'road', 'never', 'let', 'anyone', 'individual', 'passions', 'actions', 'help', 'way', 'learning', 'love', 'yourself', 'constantly', 'striving', 'diminish'}

Step 4: Weighted frequencies

Weighted frequencies of all the words present in the tokenized list are obtained by dividing the frequency of each word by the frequency of the most repeated term.

Word	Frequency	Weighted frequency
hardest	1	0.25
part	1	0.25
journey	1	0.25
success	4	1
dealing	1	0.25
people	4	1
who	1	0.25
try	2	0.5
impose	1	0.25
limited	1	0.25
views	1	0.25
think	1	0.25
right	1	0.25
shut	1	0.25
someone	1	0.25
down	1	0.25
kill	1	0.25
dreams	3	0.75
self-limiting	1	0.25
beliefs	1	0.25
confine	1	0.25
mediocre	1	0.25
life	2	0.5
tell	2	0.5
need	2	0.5
lower	1	0.25
expectation	1	0.25
more	1	0.25
realistic	1	0.25
make	2	0.5

believe	1	0.25
cannot	2	0.5
achieve	2	0.5
set	1	0.25
mind	1	0.25
cut	1	0.25
out	1	0.25
road	1	0.25
never	2	0.5
let	1	0.25
anyone	1	0.25
individual	1	0.25
passions	1	0.25
actions	1	0.25
help	1	0.25
way	1	0.25
learning	1	0.25
love	1	0.25
yourself	1	0.25
constantly	1	0.25
striving	1	0.25
diminish	1	0.25

Table 3.2 Weighted frequencies of tokens

Step 5: Cut to the Chase

The summarizer method ultimately picks the top *k* most appropriate phrases to produce a concise and accurate summary

Original Sentence	Cumulative sum
The hardest part of your journey to success is dealing with people who will try to impose their limited views upon you	(0.25+0.25+0.25+1+0.25+1+0.25+0.5+0.25+0.25+0.25+0.25 = 4.5)
Those people think that they have a right to shut someone down and kill their dreams	(1+0.25+0.25+0.25+0.25+0.25+0.25+0.25+0.75 = 3.25)
Their self-limiting beliefs confine them to a mediocre life.	(0.25+0.25+0.25+0.25 +0.5 = 1.5)
They will tell you that you need to lower your expectations.	(0.5+0.5+0.25+0.25 = 1.5)
They will tell you that you need to me more realistic.	(0.5+0.5+0.25+0.25 = 1.5)
They will try to make you believe that you cannot achieve what you set your mind to.	(0.5+0.5+0.25+0.5+0.5+0.25+0.25 = 2.75)
These are the kind of people you need to cut out of your life.	(1+0.25+0.25+0.5 = 2)
When you are on your road to success, you should never let anyone tell you that you can’t achieve your dreams	(0.25+1+0.5+0.25+0.25+0.5+0.5+0.5+0.75 = 4.5)
It is our individual passions, our dreams, and our actions that make us who we are and help us on our way to success.	(0.25+0.25+0.75+0.25 +0.5+0.25+0.25+1 = 3.5)
It’s about learning to love yourself and constantly striving for success in such a way that other people could never diminish you.	(0.25+0.25+0.25+0.25 +0.25+1+1+0.5+0.25 = 4)

Table 3.3 Cumulative sum of sentences

Result:

The summary of the given source text is,

The hardest part of your journey to success is dealing with people who will try to impose their limited views upon you. When you are on your road to success, you should never let anyone tell you that you can’t achieve your dreams. It’s about learning to love yourself and constantly striving for success in such a way that other people could never diminish you

Number of characters in the input text: 889
Number of characters in the summary text: 360

Number of words in the input text: 167
Number of words in the summary text: 65

Number of sentences in the input text: 10
Number of sentences in the summary text: 3

Input text average reading time: 40 seconds
Summary text average reading time: 18 seconds

Time saved: 22 seconds

3.2.4 Applications:

The proliferation of the use of summaries is evident in the present information-overloaded world. Concise summaries aid easier searches and quick decisions. The following points are a few applications that are currently being implemented.

The mobile applications that provide compendious articles that summarize the entire news segment for people to remain updated on the current affairs while simultaneously conserving time.

1. Handbooks that provide outlines for students to prepare for their examinations primarily for revision.

2. Noting down the minutes of the meeting apart from recording the entire discussions.

3. The story-telling businesses including novels and movies display the excerpts that contain a message to the viewers in providing a hint on what is to be expected.

4. Abridgments for the plays or long novels leading to easy and quick understanding for the children.

5. Bulletins for the details that explain the volatility of the stock market and weather forecasts.

6. The movie reviews that are listed on various platforms prove beneficial to the audience in spending prudently.

3.3 UML Diagrams

3.3.1 Use case diagram

Use case diagram represent the overall scenario of the system. A scenario is nothing but a sequence of steps describing an interaction between a user and a system. Thus use case is a set of scenario tied together by some goal. The use case diagrams are drawn for exposing the functionalities of the system.

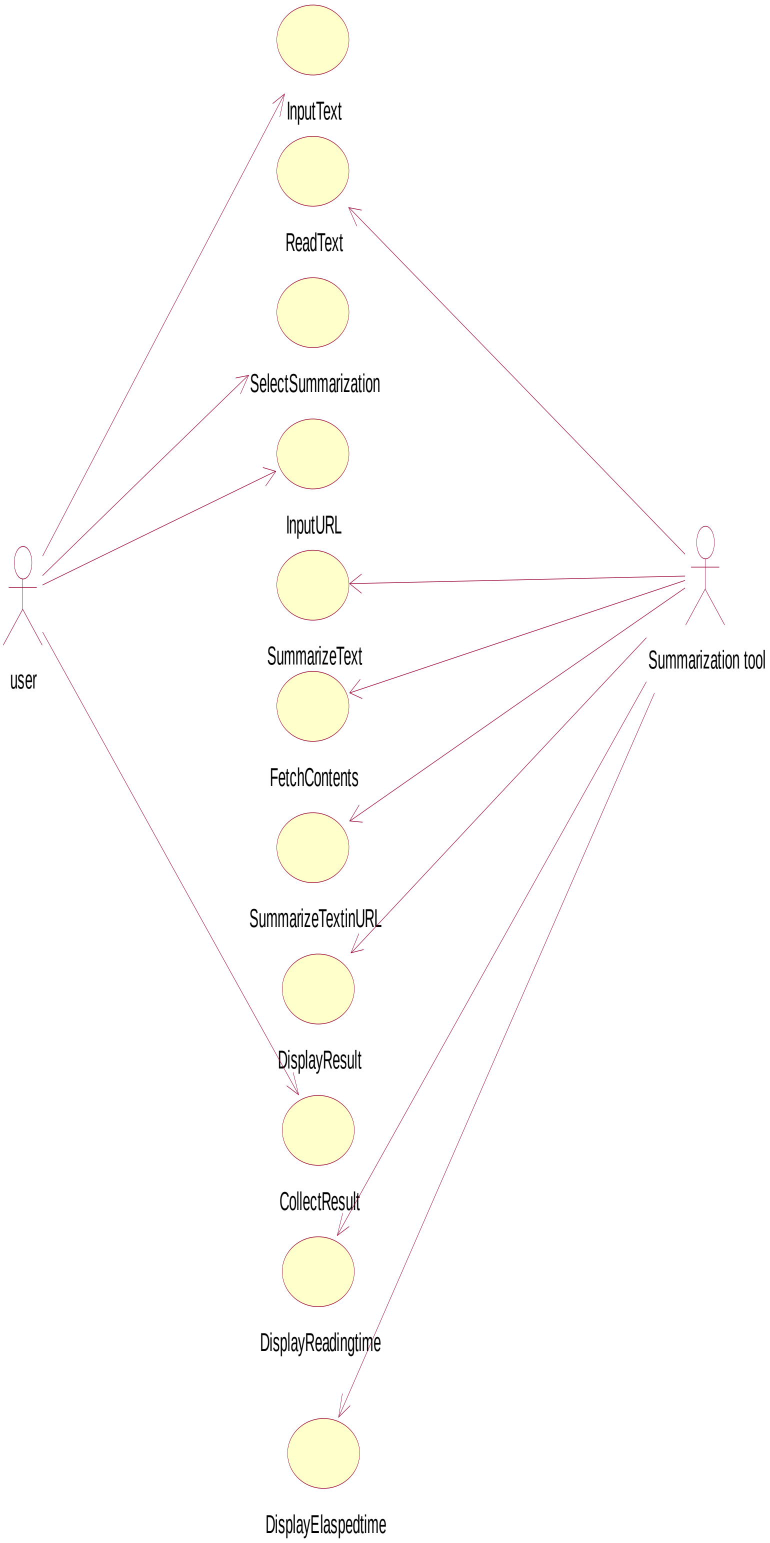


Fig 3.2 Use case representation of text summarizer

Use Cases

Draw use cases using ovals. Label the ovals with verbs that represent the system's functions.

Actors

Actors are the users of a system. When one system is the actor of another system,

Label the actor system with the actor stereotype.

Relationships

Illustrate relationships between an actor and a use case with a simple line.

For Relationships among use cases, use arrows labelled either "uses" or "extends."

3.3.2 Activity diagram

The activity diagram is a graphical representation for representing the flow of interaction within specific scenarios. It is similar to a flowchart in which various activities that can be performed in the system are represented.

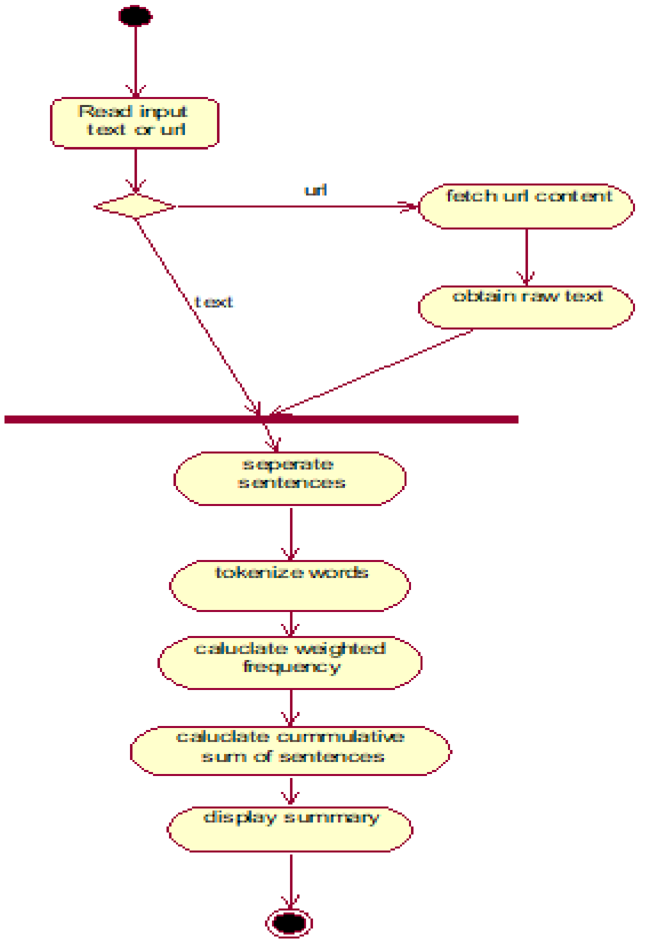


Fig 3.3 Activity diagram of text summarizer

3.3.3 Sequence Diagram

In the sequence diagram how the object interacts with the other object is shown. There are sequence of events that are represented by a sequence diagram.

It is a time oriented view of the interaction between objects to accomplish a behavioural goal of the system.

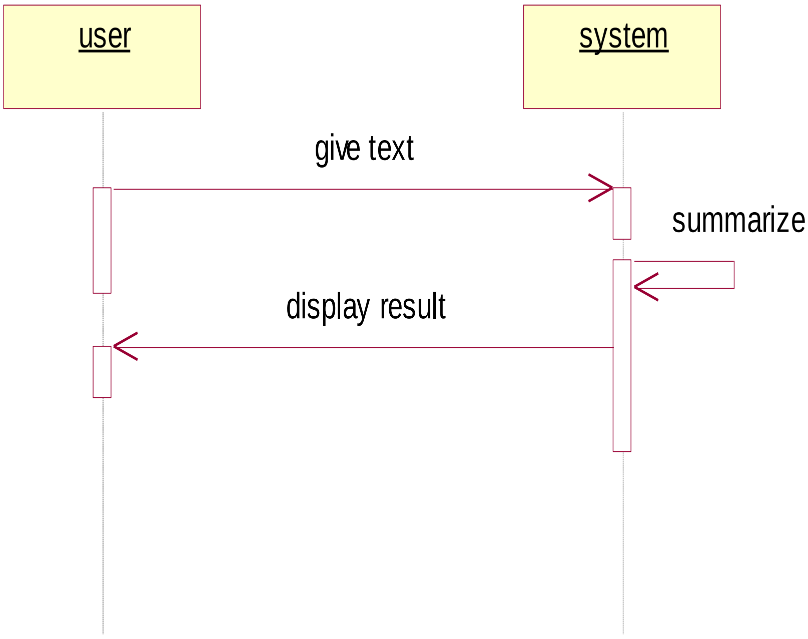


Fig 3.4 Sequence diagram for summarizing given text

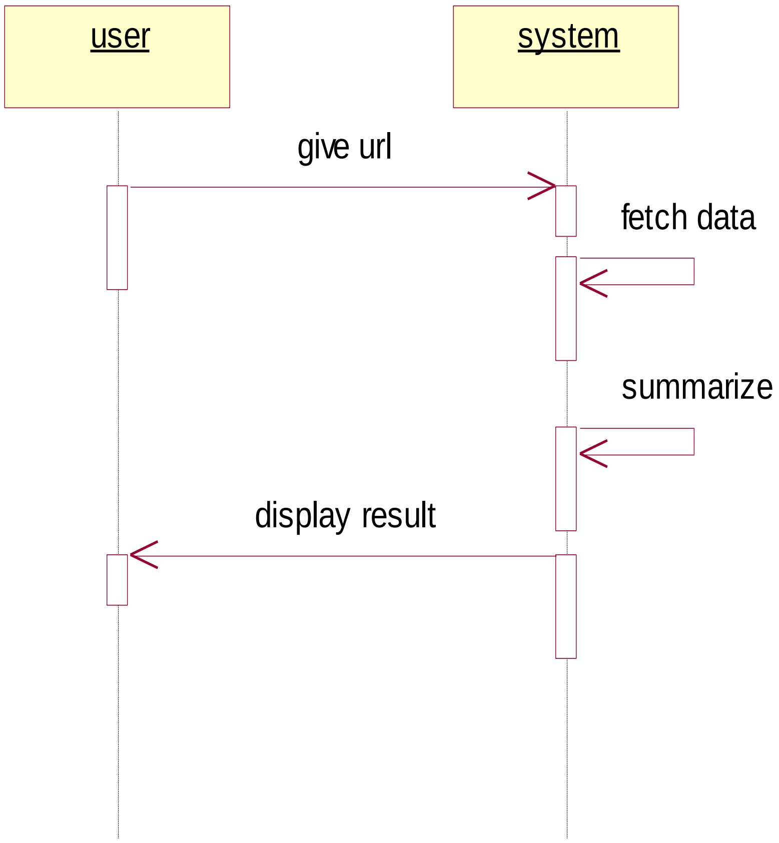


Fig 3.5 Sequence diagram for summarizing contents of given URL

As the name suggests, sequence diagrams describe the sequence of messages and interactions that happen between actors and objects. Actors or objects can be active only when needed or when another object wants to communicate with them. All communication is represented in a chronological manner. A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function

3.3.4 Collaboration Diagram

Collaboration is a collection of named objects and actors with links connecting them. They collaborate in performing some task.A Collaboration defines a set of participants and relationships that are meaningful for a given set of purposes. Collaboration between objects working together provides emergent desirable functionalities in Object-Oriented systems.

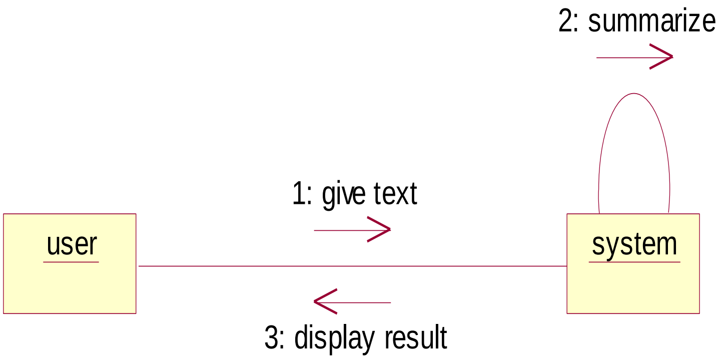


Fig 3.6 Collaboration Diagram of summarizing given text

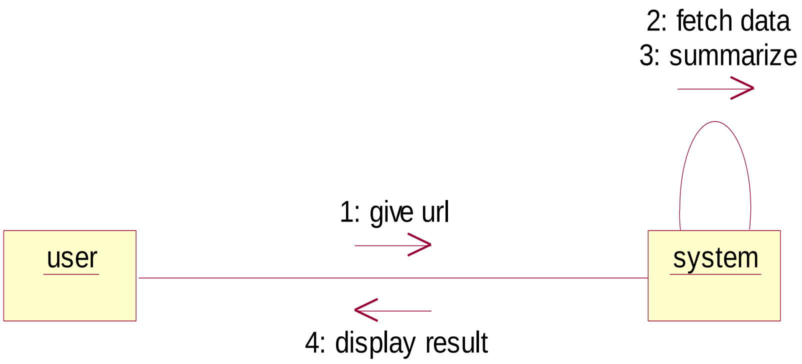


Fig 3.7 Collaboration Diagram of summarizing text in given URL

3.3.5 System architecture

The system architectural design is the design process for identifying the subsystems making up the system and framework for subsystem control and communication. The goal of the architectural design is to establish the overall structure of software system.

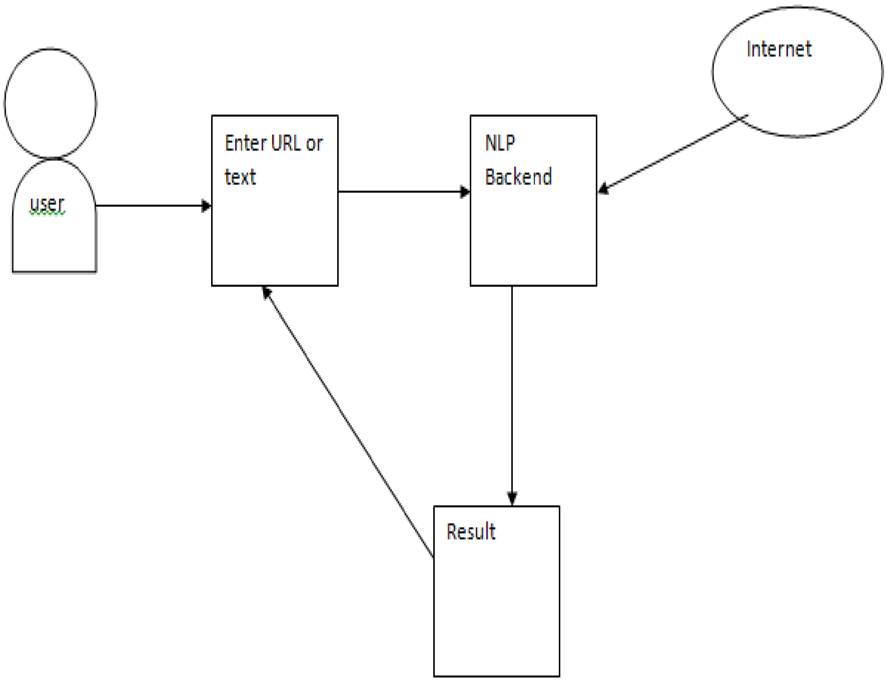


Fig 3.8 System Architecture of Text Summarizer

4. SYSTEM IMPLEMENTATION

4.1 Selected Software

This is a Web-based software, which contains many pages and hyperlinks. It handles effectively all the functions. You can switch from one page to another by clicking the hyperlinks. The user interface is created with PHP, HTML, CSS, and JavaScript.

4.1.1 Technologies:

- HTML5
- BOOTSRAP
- CSS
- JAVASCRIPT
- Python

HTML5:

HTML5 is the latest evolution of the standard that defines HTML. The term represents two different concepts. It is a new version of the language HTML, with new elements, attributes, and behaviours, and a larger set of technologies that allows the building of more diverse and powerful Web sites and applications. Designed to be usable by all Open Web developers, this reference page links to numerous resources about HTML5 technologies, classified into several groups based on their function.

Semantics: allowing you to describe more precisely what your content is.

Connectivity: allowing you to communicate with the server in new and innovative ways.

Offline and storage: allowing web pages to store data on the client-side locally and operate offline more efficiently.

Multimedia: making video and audio first-class citizens in the Open Web.

- 2D/3D graphics and effects: allowing a much more diverse range of
- presentation options.
- Performance and integration: providing greater speed optimization and better usage of computer hardware.
- Device access: allowing for the usage of various input and output devices.
- Styling: letting authors write more sophisticated themes.

BOOTSTRAP:

Bootstrap is a free front-end framework for faster and easier web development. Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plug ins. Bootstrap also gives you the ability to easily create responsive designs.

Features of BOOTSTRAP:

- **Easy to use**: Anybody with just basic knowledge of HTML and CSS can start using Bootstrap
- **Responsive features**: Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
- **Mobile-first approach**: In Bootstrap 3, mobile-first styles are part of the core framework
- **Browser compatibility**: Bootstrap is compatible with all modern browsers(Chrome, Firefox, Internet Explorer, Edge, Safari, and Opera)

CSS:

CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media.CSS saves a lot of work. It can control the layout of multiple web pages all at once. External style sheets are stored in CSS files.CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content. Separation of formatting and content also makes it feasible to present the same mark up page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices.

CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable

Advantages of CSS

- **CSS saves time** − You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- **Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So, less code means faster download times.
- **Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- **Superior styles to HTM**L − CSS have a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- **Multiple Device Compatibility** − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- **Global web standards** − Now HTML attributes are being deprecated and it is being recommended to use CSS. So, it’s a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

JAVASCRIPT:

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as Live Script, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name Live Script. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.
- Open and cross-platform

The merits of using JavaScript are:

- **Less server interaction** − You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
- **Immediate feedback to the visitors** − They don't have to wait for a page reload
- to see if they have forgotten to enter something.
- **Increased interactivity** − You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- **Richer interfaces** − You can use JavaScript to include such items as drag-and- drop components and sliders to give a Rich Interface to your site visitors.

PYTHON

Python is an interpreted high level, general purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Advantages/Benefits of Python

The diverse application of the Python language is a result of the combination of features which give this language an edge over others. Some of the benefits of programming in Python include:

1. Presence of Third Party Modules:

The Python Package Index (PyPI) contains numerous third party modules that make Python capable of interacting with most of the other languages and platforms.

2. Extensive Support Libraries:

Python provides a large standard library which includes areas like internet protocols, string operations, web services tools and operating system interfaces. Many high use programming tasks have already been scripted into the standard library which reduces length of code to be written significantly.

3. Open Source and Community Development:

Python language is developed under an OSI-approved open source license, which makes it free to use and distribute, including for commercial purposes.

Further, its development is driven by the community which collaborates for its code through hosting conferences and mailing lists, and provides for its numerous modules

4. Learning Ease and Support Available:

Python offers excellent readability and uncluttered simple-to-learn syntax which helps beginners to utilize this programming language. The code style guidelines, PEP 8, provide a set of rules to facilitate the formatting of code. Additionally, the wide base of users and active developers has resulted in a rich internet resource bank to encourage development and the continued adoption of the language.

5. User-friendly Data Structures:

Python has built-in list and dictionary data structures which can be used to construct fast runtime data structures. Further, Python also provides the option of dynamic high-level data typing which reduces the length of support code that is needed.

6. Productivity and Speed:

Python has clean object-oriented design, provides enhanced process control capabilities, and possesses strong integration and text processing capabilities and its own unit testing framework, all of which contribute to the increase in its speed and productivity. Python is considered a viable option for building complex multi-protocol network applications.

spaCy

spaCy is a free, open-source library for advanced Natural Language Processing (NLP) in Python. spaCy is designed specifically for production use and helps you build applications that process and “understand” large volumes of text. It can be used to build information extraction or natural language understanding systems, or to pre-process text for deep learning.

NAME	DESCRIPTION
Tokenization	Segmenting text into words, punctuations marks etc.
Part-of- speech (POS) Tagging	Assigning word types to tokens, like verb or noun.
Dependency Parsing	Assigning syntactic dependency labels, describing the relations between individual tokens, like subject or object.
Lemmatization	Assigning the base forms of words. For example, the lemma of “was” is “be”, and the lemma of “rats” is “rat”.
Sentence Boundary Detection (SBD)	Finding and segmenting individual sentences.
Named Entity Recognition (NER)	Labelling named “real-world” objects, like persons, companies or locations.
Entity Linking (EL)	Disambiguating textual entities to unique identifiers in a Knowledge Base.
Similarity	Comparing words, text spans and documents and how similar they are to each other.
Text Classification	Assigning categories or labels to a whole document, or parts of a document.
Rule-based Matching	Finding sequences of tokens based on their texts and linguistic annotations, similar to regular expressions.
Training	Updating and improving a statistical model’s predictions.
Serialization	Saving objects to files or byte strings.

Table 4.1 Features of spaCy

FLASK

Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more frequently than the core Flask program.

When you installed Flask, you also installed the flask command line script. Typing flask run will prompt the virtual environment's Flask package to run an HTTP server using the app object in whatever script the FLASK_APP environment variable points to. The script above also includes an environment variable named DEBUG that will be used a bit later.

4.2 Sample Code

- **Fetch text from URL**

```
def get_text(url):
page = urlopen(url)
soup = BeautifulSoup(page)
fetched_text=".join(map(lambdap:p.text,soup.find_all(")
))
return fetched_text
```

- **Given text analysis**

```
@app.route('/analyze',methods=['GET','POST'])
def analyze():
    start = time.time()
    if request.method == 'POST':
rawtext = request.form['rawtext']
final_reading_time = readingTime(rawtext)
```

```
final_summary = text_summarizer(rawtext)
summary_reading_time= readingTime(final_summary)
end = time.time()
final_time = end-start
return
render_template('index.html',ctext=rawtext,final_summary=final_su
mmary,final_time=final_time,final_reading_time=final_reading_ti
me,summary_reading_time=summary_reading_time)
```

- **Given URL text analysis**

```
@app.route('/analyze_url',methods=['GET','POST'])
def analyze_url():
        start = time.time()
        if request.method == 'POST':
                raw_url = request.form['raw_url']
                rawtext = get_text(raw_url)
                final_reading_time = readingTime(rawtext)
                final_summary = text_summarizer(rawtext)
summary_reading_time= readingTime(final_summary
                end = time.time()
                final_time = end-start
        return
render_template('index.html',ctext=rawtext,final_summary=final_su
mmary,final_time=final_time,final_reading_time=final_reading_ti
me,summary_reading_time=summary_reading_time)
```

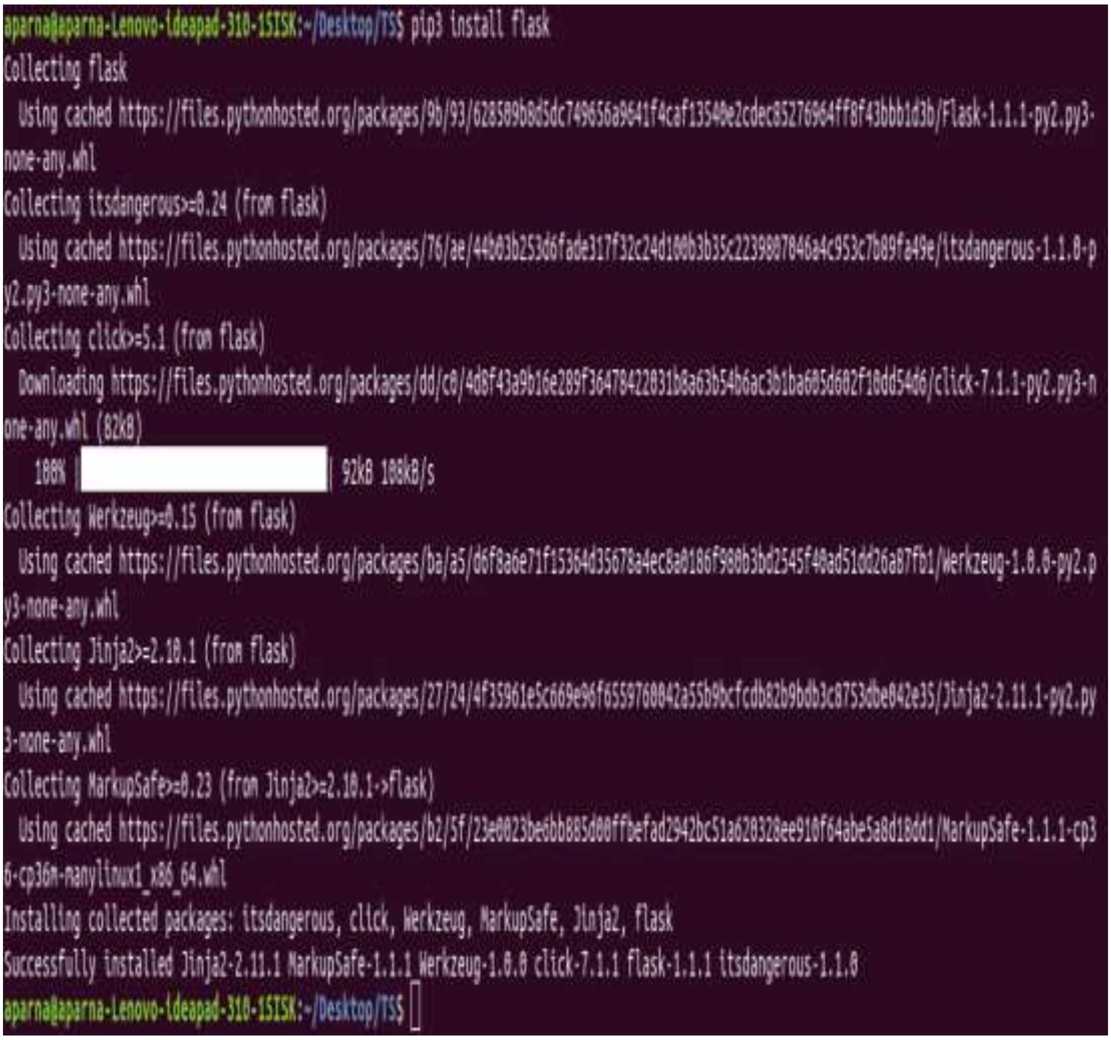
- **text summarizing**

```
def text_summarizer(raw_docx):
   raw_text = raw_docx
   docx = nlp(raw_text)
   stopwords = list(STOP_WORDS)
   # Build Word Frequency # word.text is tokenization in     spacy
  word_frequencies = {}
   for word in docx:
       if word.text not in stopwords:
           if word.text not in word_frequencies.keys():
               word_frequencies[word.text] = 1
           else:
               word_frequencies[word.text] += 1



   maximum_frequncy = max(word_frequencies.values())

   for word in word_frequencies.keys():
       word_frequencies[word]                    =                    (
word_frequencies[word]/maximum_frequncy)
   # Sentence Tokens
   sentence_list = [ sentence for sentence in docx.sents ]

   # Sentence Scores
   sentence_scores = {}
   for sent in sentence_list:
       for word in sent:
           if word.text.lower() in word_frequencies.keys():
               if len(sent.text.split(' ')) < 30:
                   if sent not in sentence_scores.keys():
                       sentence_scores[sent]=
                       word_frequencies[word.text.lower()]
                    else:
                       sentence_scores[sent]+=
                       word_frequencies[word.text.lower()]

summarized_sentences=nlargest(7,sentence_scores,
key=sentence_scores.get)
final_sentences= [ w.text for w in summarized_sentences]
summary = ' '.join(final_sentences)
       return summary
```

INSTALLATION STEPS

Installing Flask:

1. Open command prompt.
2. Give command pip3 install flask.
3. Click enter.

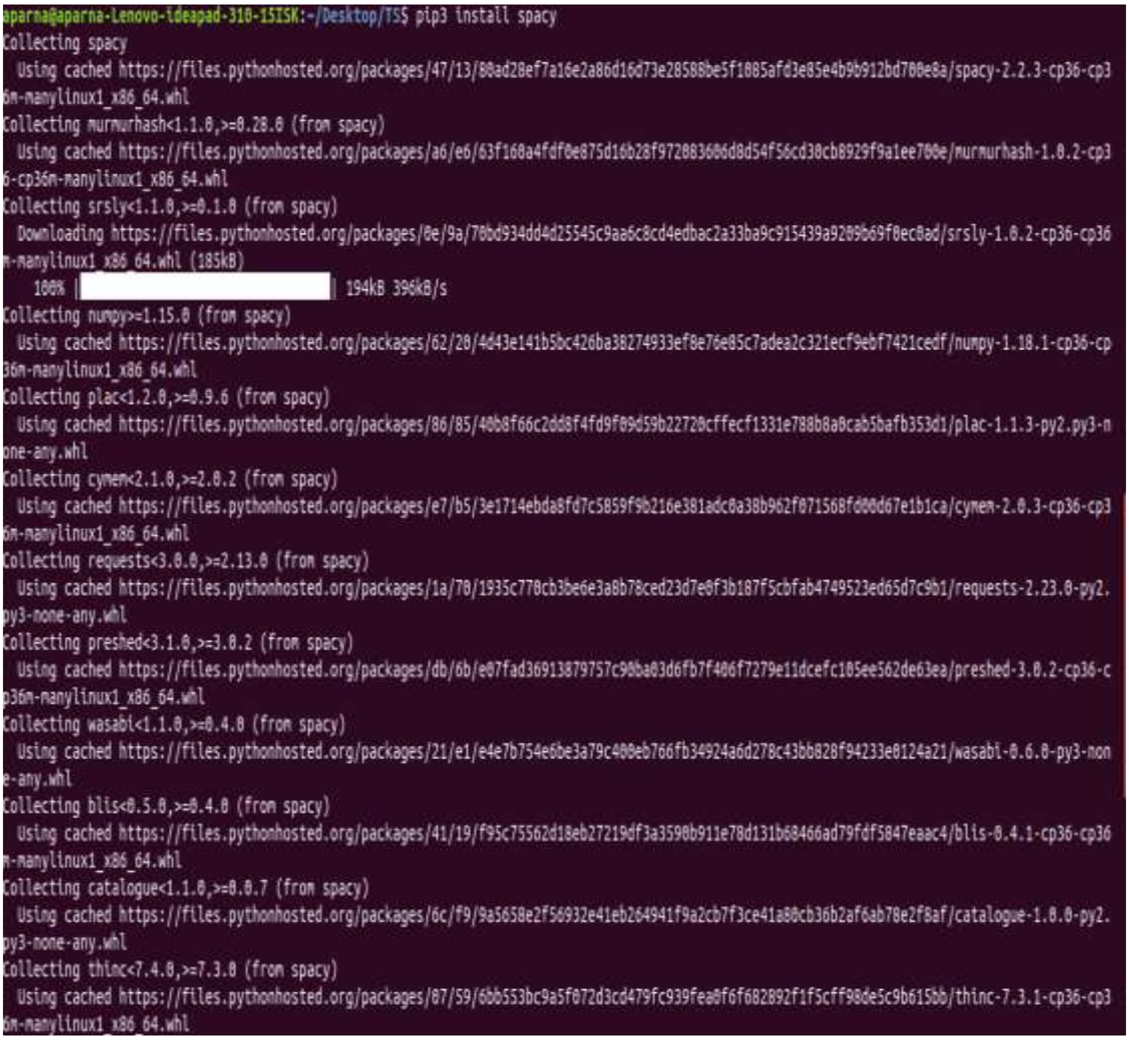


```
aparna@aparna-Lenovo-ideapad-310-15ISK:~/Desktop/TS$ pip3 install flask
Collecting flask
  Using cached https://files.pythonhosted.org/packages/9b/93/628509b8d5dc749656a9641f4caf13540e2cdec85276964ff8f43bbb1d3b/Flask-1.1.1-py2.py3-
none-any.whl
Collecting itsdangerous>=0.24 (from flask)
  Using cached https://files.pythonhosted.org/packages/76/ae/44b03b253d6fade317f32c24d100b3b35c2239807046a4c953c7b89fa49e/itsdangerous-1.1.0-p
y2.py3-none-any.whl
Collecting click>=5.1 (from flask)
  Downloading https://files.pythonhosted.org/packages/dd/c0/4d8f43a9b16e289f36478422031b8a63b54b6ac3b1ba605d602f10dd54d6/click-7.1.1-py2.py3-n
one-any.whl (82kB)
    100% |████████████████████████████████| 92kB 108kB/s
Collecting Werkzeug>=0.15 (from flask)
  Using cached https://files.pythonhosted.org/packages/ba/a5/d6f8a6e71f15364d35678a4ec8a0186f980b3bd2545f40ad51dd26a87fb1/Werkzeug-1.0.0-py2.p
y3-none-any.whl
Collecting Jinja2>=2.10.1 (from flask)
  Using cached https://files.pythonhosted.org/packages/27/24/4f35961e5c669e96f6559760042a55b9bcfcdb82b9bdb3c8753dbe042e35/Jinja2-2.11.1-py2.py
3-none-any.whl
Collecting MarkupSafe>=0.23 (from Jinja2>=2.10.1->flask)
  Using cached https://files.pythonhosted.org/packages/b2/5f/23e0023be6bb885d00ffbefad2942bc51a620328ee910f64abe5a8d18dd1/MarkupSafe-1.1.1-cp3
6-cp36m-manylinux1_x86_64.whl
Installing collected packages: itsdangerous, click, Werkzeug, MarkupSafe, Jinja2, flask
Successfully installed Jinja2-2.11.1 MarkupSafe-1.1.1 Werkzeug-1.0.0 click-7.1.1 flask-1.1.1 itsdangerous-1.1.0
aparna@aparna-Lenovo-ideapad-310-15ISK:~/Desktop/TS$ 
```

Fig 4.1 installation of flask

Installing spaCy:

1. Open command prompt.
2. Give command pip3 install spacy.
3. Click enter.

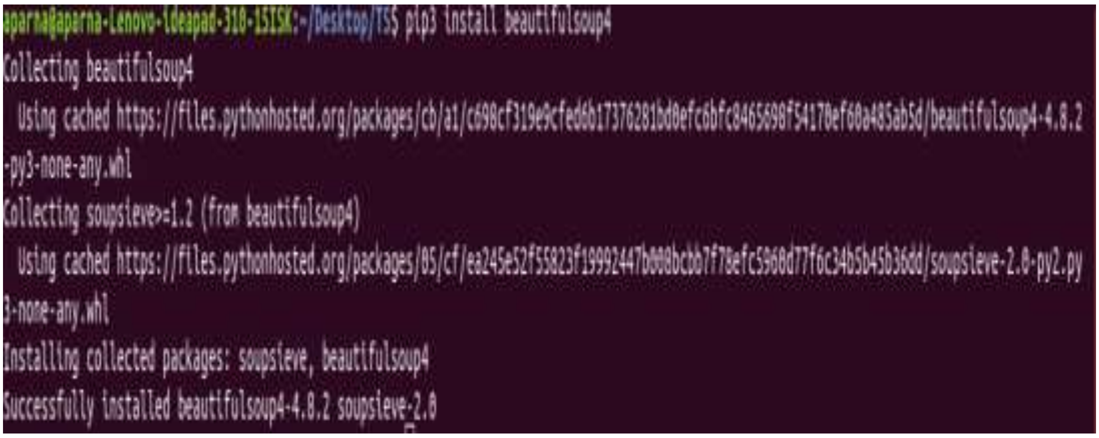


```
aparna@aparna-Lenovo-ideapad-310-15ISK:~/Desktop/TS$ pip3 install spacy
Collecting spacy
  Using cached https://files.pythonhosted.org/packages/47/13/80ad28ef7a16e2a86d16d73e28588be5f1085afd3e85e4b9b912bd700e8a/spacy-2.2.3-cp36-cp3
6m-manylinux1_x86_64.whl
Collecting murmurhash<1.1.0,>=0.28.0 (from spacy)
  Using cached https://files.pythonhosted.org/packages/a6/e6/63f160a4fdf0e875d16b28f972083606d8d54f56cd30cb8929f9a1ee700e/murmurhash-1.0.2-cp3
6-cp36m-manylinux1_x86_64.whl
Collecting srsly<1.1.0,>=0.1.0 (from spacy)
  Downloading https://files.pythonhosted.org/packages/0e/9a/70bd934dd4d25545c9aa6c8cd4edbac2a33ba9c915439a9209b69f0ec0ad/srsly-1.0.2-cp36-cp36
m-manylinux1_x86_64.whl (185kB)
    100% |████████████████████████████████| 194kB 396kB/s
Collecting numpy>=1.15.0 (from spacy)
  Using cached https://files.pythonhosted.org/packages/62/20/4d43e141b5bc426ba38274933ef8e76e85c7adea2c321ecf9ebf7421cedf/numpy-1.18.1-cp36-cp
36m-manylinux1_x86_64.whl
Collecting plac<1.2.0,>=0.9.6 (from spacy)
  Using cached https://files.pythonhosted.org/packages/86/85/40b8f66c2dd8f4fd9f09d59b22720cffecf1331e788b8a0cab5bafb353d1/plac-1.1.3-py2.py3-n
one-any.whl
Collecting cymem<2.1.0,>=2.0.2 (from spacy)
  Using cached https://files.pythonhosted.org/packages/e7/b5/3e1714ebda8fd7c5859f9b216e381adc0a38b962f071568fd00d67e1b1ca/cymem-2.0.3-cp36-cp3
6m-manylinux1_x86_64.whl
Collecting requests<3.0.0,>=2.13.0 (from spacy)
  Using cached https://files.pythonhosted.org/packages/1a/70/1935c770cb3be6e3a8b78ced23d7e0f3b187f5cbfab4749523ed65d7c9b1/requests-2.23.0-py2.
py3-none-any.whl
Collecting preshed<3.1.0,>=3.0.2 (from spacy)
  Using cached https://files.pythonhosted.org/packages/db/6b/e07fad36913879757c90ba03d6fb7f406f7279e11dcefc105ee562de63ea/preshed-3.0.2-cp36-c
p36m-manylinux1_x86_64.whl
Collecting wasabi<1.1.0,>=0.4.0 (from spacy)
  Using cached https://files.pythonhosted.org/packages/21/e1/e4e7b754e6be3a79c400eb766fb34924a6d278c43bb828f94233e0124a21/wasabi-0.6.0-py3-non
e-any.whl
Collecting blis<0.5.0,>=0.4.0 (from spacy)
  Using cached https://files.pythonhosted.org/packages/41/19/f95c75562d18eb27219df3a3590b911e78d131b68466ad79fdf5847eaac4/blis-0.4.1-cp36-cp36
m-manylinux1_x86_64.whl
Collecting catalogue<1.1.0,>=0.0.7 (from spacy)
  Using cached https://files.pythonhosted.org/packages/6c/f9/9a5658e2f56932e41eb264941f9a2cb7f3ce41a80cb36b2af6ab78e2f8af/catalogue-1.0.0-py2.
py3-none-any.whl
Collecting thinc<7.4.0,>=7.3.0 (from spacy)
  Using cached https://files.pythonhosted.org/packages/07/59/6bb553bc9a5f072d3cd479fc939fea0f6f682892f1f5cff98de5c9b615bb/thinc-7.3.1-cp36-cp3
6m-manylinux1_x86_64.whl
```

Fig 4.2 installation of spaCy

Installing beautifulsoup:

1. Open command prompt.
2. Give command pip3 install beautifulsoup.
3. Click enter.

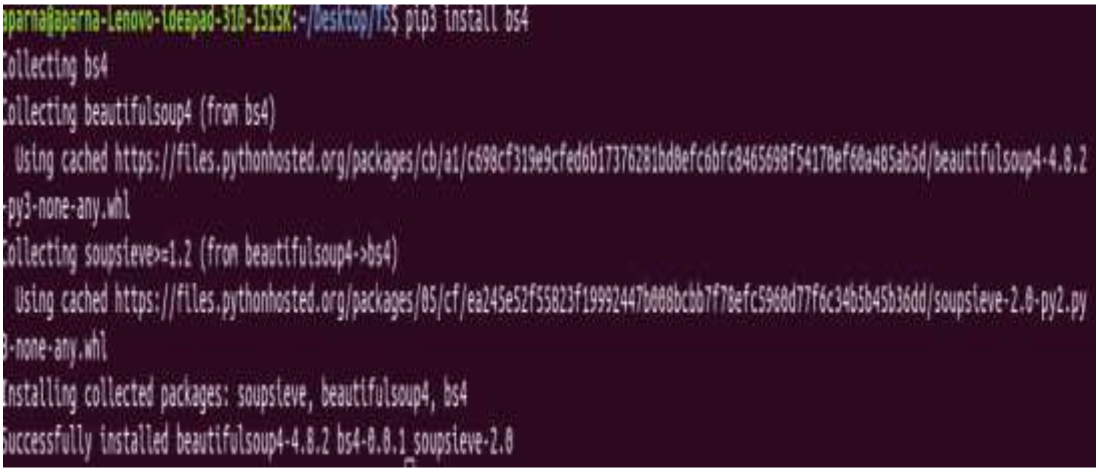


```
aparna@aparna-Lenovo-ideapad-310-15ISK:~/Desktop/TS$ pip3 install beautifulsoup4
Collecting beautifulsoup4
  Using cached https://files.pythonhosted.org/packages/cb/a1/c698cf319e9cfed6b17376281bd0efc6bfc8465698f54170ef60a485ab5d/beautifulsoup4-4.8.2
-py3-none-any.whl
Collecting soupsieve>=1.2 (from beautifulsoup4)
  Using cached https://files.pythonhosted.org/packages/05/cf/ea245e52f55823f19992447b008bcbb7f78efc5968d77f6c34b5b45b36dd/soupsieve-2.0-py2.py
3-none-any.whl
Installing collected packages: soupsieve, beautifulsoup4
Successfully installed beautifulsoup4-4.8.2 soupsieve-2.0
```

Fig 4.3 installation of beautifulsoup

Installing bs4:

1. Open command prompt.
2. Give command pip3 install bs4.
3. Click enter.

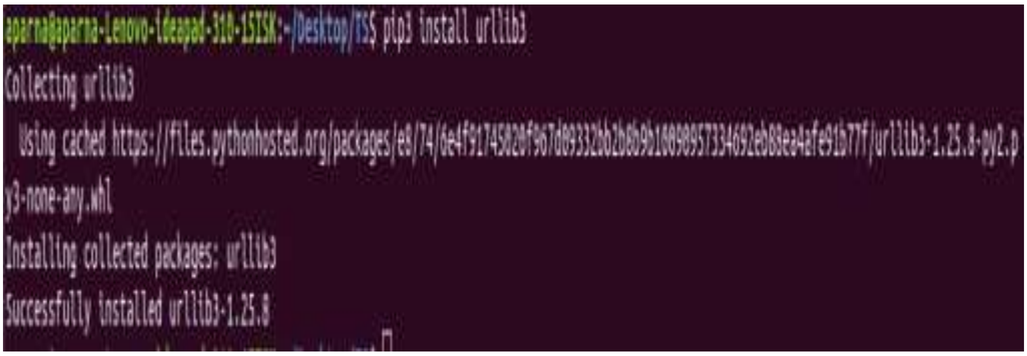


```
parna@aparna-Lenovo-ideapad-310-15ISK:~/Desktop/TS$ pip3 install bs4
ollecting bs4
ollecting beautifulsoup4 (from bs4)
  Using cached https://files.pythonhosted.org/packages/cb/a1/c698cf319e9cfed6b17376281bd0efc6bfc8465698f54170ef60a485ab5d/beautifulsoup4-4.8.2
-py3-none-any.whl
ollecting soupsieve>=1.2 (from beautifulsoup4->bs4)
  Using cached https://files.pythonhosted.org/packages/05/cf/ea245e52f55823f19992447b008bcbb7f78efc5968d77f6c34b5b45b36dd/soupsieve-2.0-py2.py
3-none-any.whl
nstalling collected packages: soupsieve, beautifulsoup4, bs4
uccessfully installed beautifulsoup4-4.8.2 bs4-0.0.1 soupsieve-2.0
```

Fig 4.4 installation of bs4

Installing urllib:

1. Open command prompt.
2. Give command pip3 install urllib3.
3. Click enter.



```
aparna@aparna-Lenovo-ideapad-310-15ISK:~/Desktop/TS$ pip3 install urllib3
Collecting urllib3
  Using cached https://files.pythonhosted.org/packages/e8/74/6e4f91745020f967d09332bb2b8b9b10090957334692eb88ea4afe91b77f/urllib3-1.25.8-py2.p
y3-none-any.whl
Installing collected packages: urllib3
Successfully installed urllib3-1.25.8
```

Fig 4.5 installation of urllib

5. Testing

The development of software involves a series of production activities were opportunities for injection of human fallibilities are enormous. Error may begin to occur at very inspection of the process where the objective may be enormously or imperfectly specified as well as in lateral design and development stage. Because of human inability to perform and communicate with perfection, software development quality assurance activities.

Software testing is a crucial element of software quality assurances and represents ultimate review of specification, design and coding.

5.1 White box testing

It focuses on the program control structure. Here all statement in the project have been executed at least once during testing and all logical condition have been exercised.

5.2 Black box testing

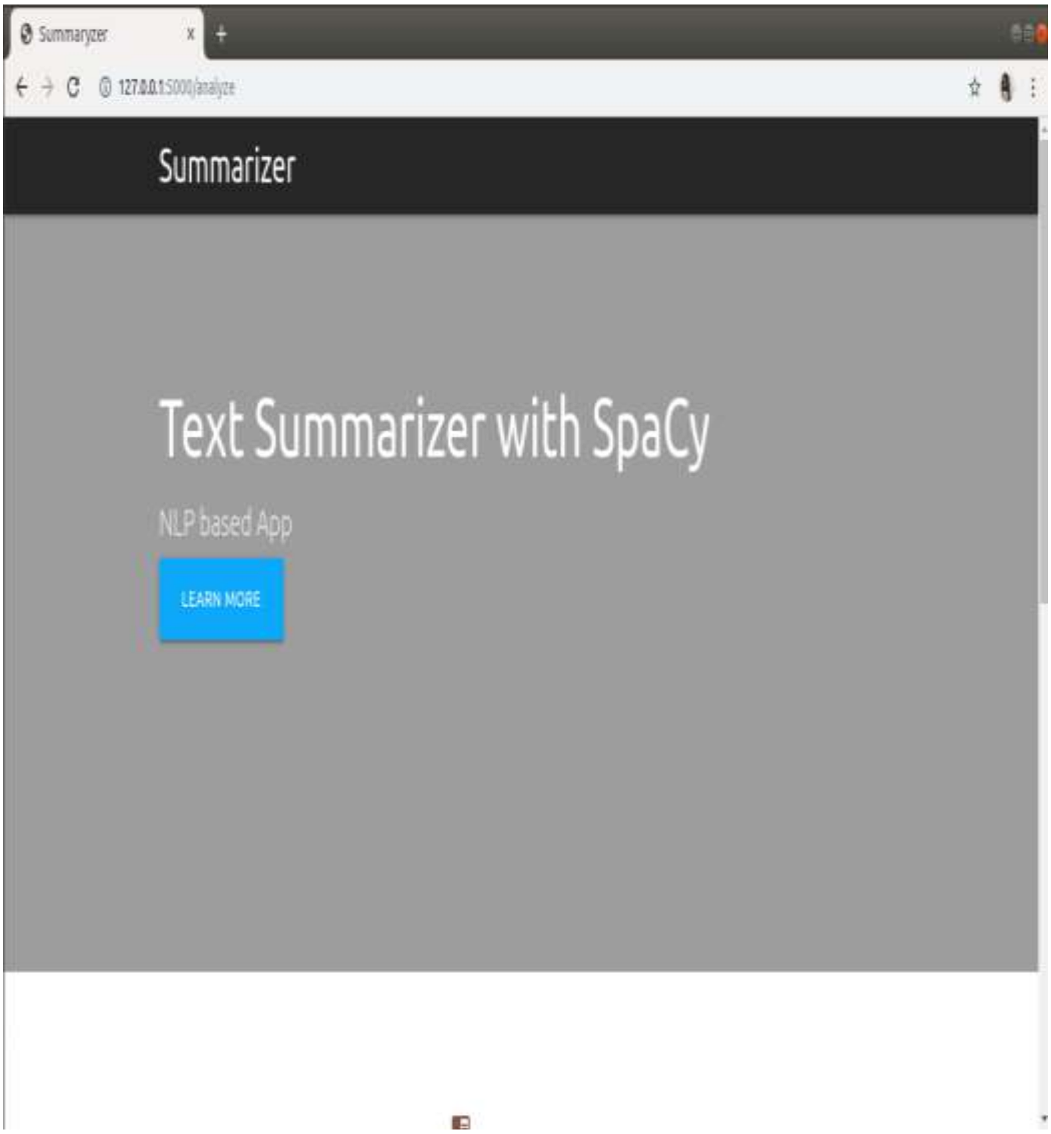
This is designed to uncover the error in functional requirements without regard to the internal working of the project. This testing focuses on the information domain of the project, deriving test case by partitioning the input and output domain of programming – A manner that provides through test coverage.

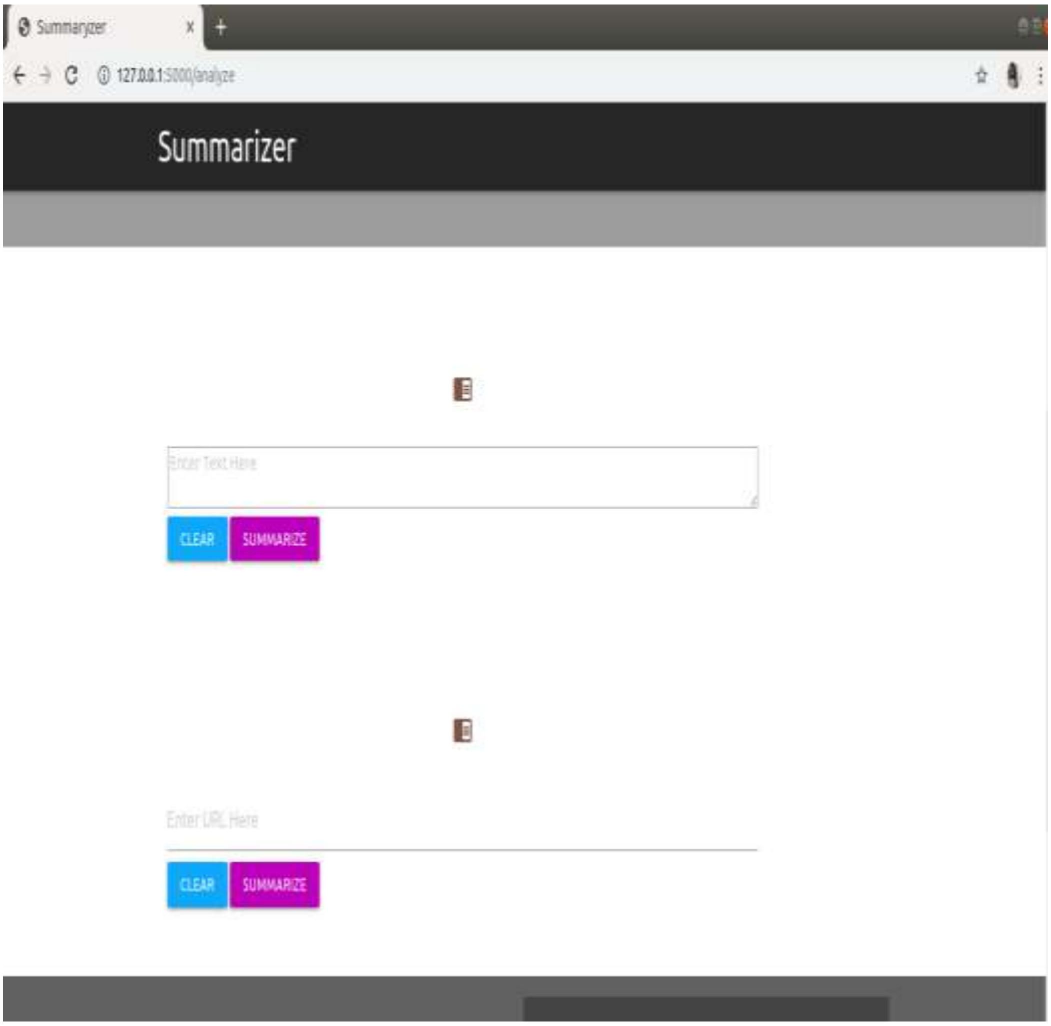
5.3 Test cases

Test case Id	Input	Description	Expected Result
KEP_TC01	Valid text less than seven sentences	A text less than seven sentences given by user	Same text given is displayed as output
KEP_TC02	Valid text less than seven sentences	A text more than seven sentences given by use	Summary is given as output
KEP_TC03	Blank text	User gives blank text and clicks summarize	Displays please fill in this field
KEP_TC04	Valid URL	Give URL and click summarize	Summary is given as output
KEP_TC05	Blank URL	Give blank URL and click summarize	Displays please fill in this field
KEP_TC06	Invalid URL	A invalid URL is given by user	Outputs Forbidden

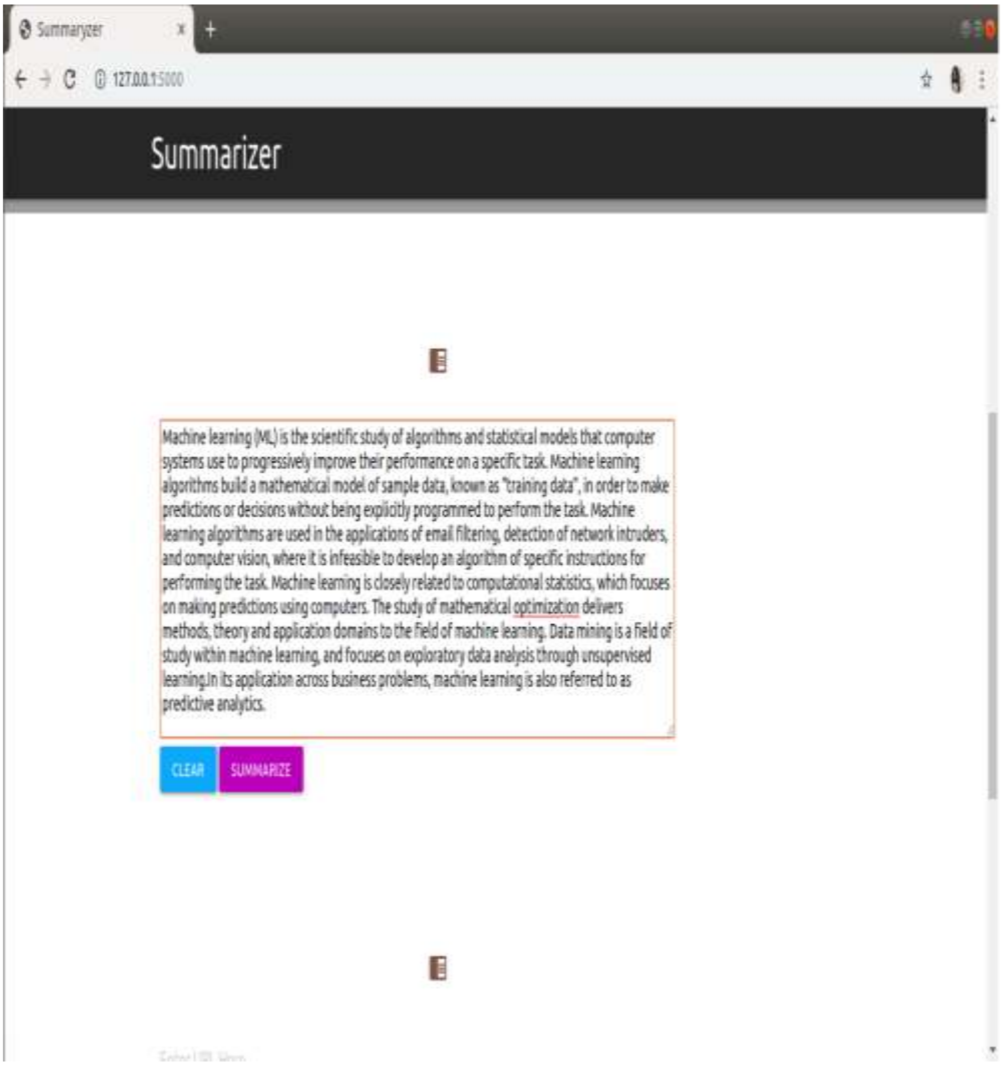
Table 5.1 Home page

6. Screens and output

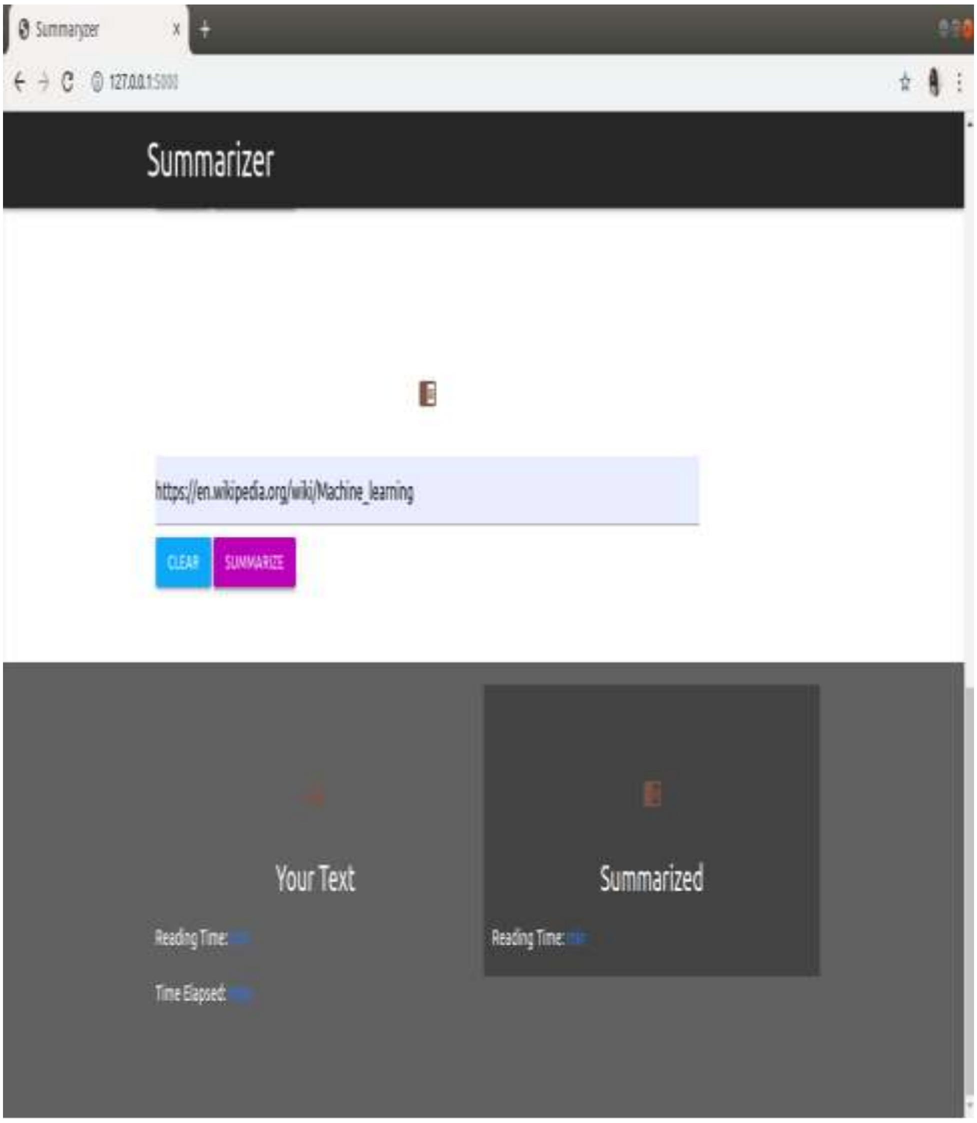




- This is the page which will be shown to us at the beginning when the website is opened.
- It has the header section with name of the app and summarize option is shown with fields to enter text and URL.

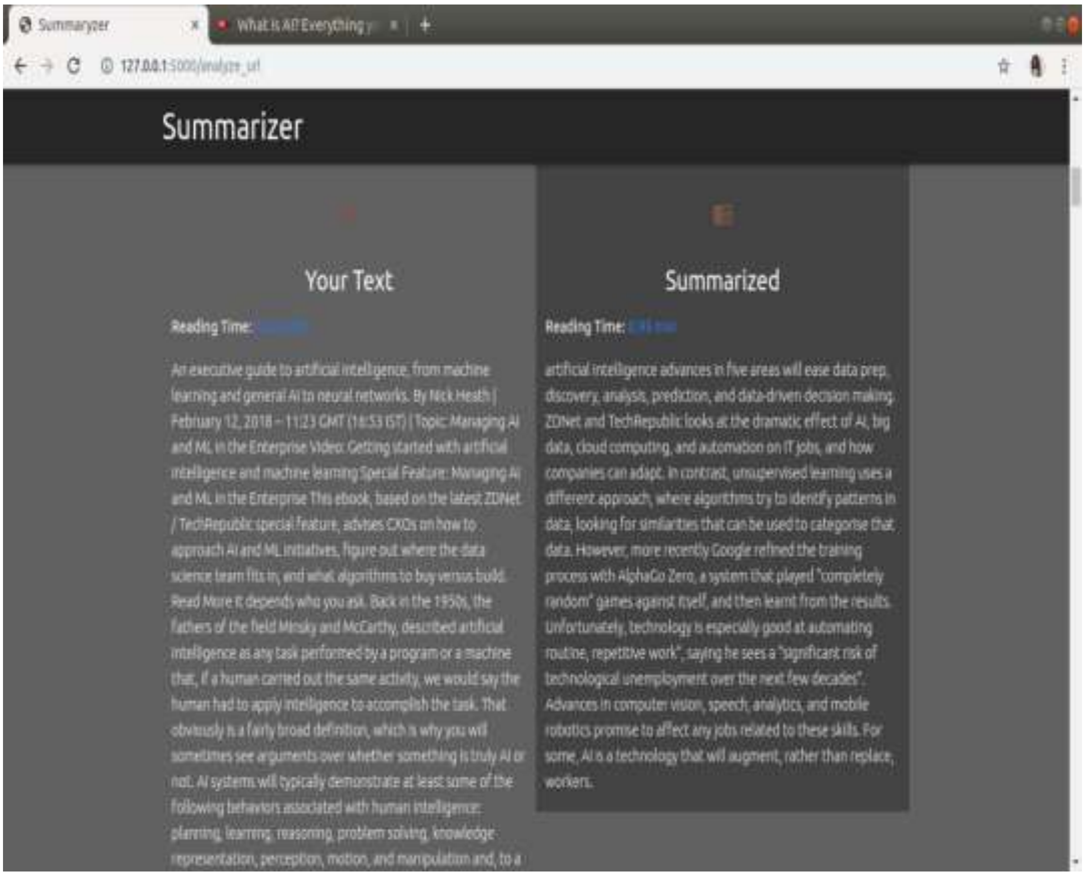


Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to progressively improve their performance on a specific task. Machine learning algorithms build a mathematical model of sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in the applications of email filtering, detection of network intruders, and computer vision, where it is infeasible to develop an algorithm of specific instructions for performing the task. Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through unsupervised learning.In its application across business problems, machine learning is also referred to as predictive analytics.



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

- This is the page asks for input like text or URL.
- Required values are given and summarize is clicked.



Summarizer

Your Text

Reading Time: [illegible]

An executive guide to artificial intelligence, from machine learning and general AI to neural networks. By Nick Heath | February 12, 2018 -- 11:23 GMT (16:53 IST) | Topic: Managing AI and ML in the Enterprise Video: Getting started with artificial intelligence and machine learning Special Feature: Managing AI and ML in the Enterprise This ebook, based on the latest ZDNet / TechRepublic special feature, advises CXOs on how to approach AI and ML initiatives, figure out where the data science team fits in, and what algorithms to buy versus build. Read More It depends who you ask. Back in the 1950s, the fathers of the field Minsky and McCarthy, described artificial intelligence as any task performed by a program or a machine that, if a human carried out the same activity, we would say the human had to apply intelligence to accomplish the task. That obviously is a fairly broad definition, which is why you will sometimes see arguments over whether something is truly AI or not. AI systems will typically demonstrate at least some of the following behaviors associated with human intelligence: planning, learning, reasoning, problem solving, knowledge representation, perception, motion, and manipulation and, to a

Summarized

Reading Time: [illegible]

artificial intelligence advances in five areas will ease data prep, discovery, analysis, prediction, and data-driven decision making. ZDNet and TechRepublic looks at the dramatic effect of AI, big data, cloud computing, and automation on IT jobs, and how companies can adapt. In contrast, unsupervised learning uses a different approach, where algorithms try to identify patterns in data, looking for similarities that can be used to categorise that data. However, more recently Google refined the training process with AlphaGo Zero, a system that played "completely random" games against itself, and then learnt from the results. Unfortunately, technology is especially good at automating routine, repetitive work", saying he sees a "significant risk of technological unemployment over the next few decades". Advances in computer vision, speech, analytics, and mobile robotics promise to affect any jobs related to these skills. For some, AI is a technology that will augment, rather than replace, workers.

- This is the output screen when summarize button is clicked in the previous page.
- The output of the given text or text in URL is displayed along with reading and elapsed times.

6.2 Reports

Testcase Id	Input	Description	Expectation	Result
KEP_TC01	Valid text less than seven sentences	A text less than seven sentences given by user	Same text given is displayed as output	Pass
KEP_TC02	Valid text less than seven sentences	A text more than seven sentences given by user	Summary is given as output	Pass
KEP_TC03	Blank text	User gives blank text and clicks summarize	Displays please fill in this field	Pass
KEP_TC04	Valid URL	Give URL and click summarize	Summary is given as output	Pass
KEP_TC05	Blank URL	Give blank URL and click summarize	Displays please fill in this field	Pass
KEP_TC06	Invalid URL	A invalid URL is given by user	Outputs Forbidden	Pass

7. CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

It is impractical to go through all the relevant documents that are accessible as boundless information is present on the World Wide Web. It is also a laborious activity to find which among the documents are nonessential. There is always a possibility of different summaries of the same primary text when it is done manually by two different experts. The immense need for automatic summarization has produced various methods and proposals. This paper provides an effective way of extractive summarization using web scraping, machine learning, and natural language processing.

This method extends a coherent and productive way of encapsulating data by conserving time. Providing a Uniform Resource Locator (URL) alone is adequate for receiving an abstract overview of the entire information present on that website. Natural language processing ensures to keep the sentences that are needed to be assimilated in the summary, semantically unerring. Hence, this paper provides a methodological approach to procure a brief paragraph that contains the gist of the massive data from the online platform.

7.2 Future Scope

Topic focused summarization: One of the future plans may be to apply the topic-focused summarization framework to news articles or blogs and to extend the work in the machine leaning approaches. Topic focused summaries of news articles would be lot more accurate and valuable to users. It would be more interesting to work on topic modelling and summarization in the domain of social media in future.

Multi lingual summarization:

The rate at which the information is growing is tremendous. Hence it is very important to build a multilingual summarization system and this research could be a stepping stone towards achieving that goal provided there is availability of online lexical databases in other languages. The work presented by the thesis can also be applicable to multi document summarization by using minimal extensions.

BIBLIOGRAPHY

1. Selvani Deepthi Kavila and Dr.Radhika Y, “Extractive Text Summarization Using Modified Weighing and Sentence Symmetric Feature”, I.J. Modern Education and Computer Science, October 2015.
2. Karel Jezek and Josef Steinberger, “Automatic Text summarization”, Vaclav Snasel (Ed.): Znalosti 2008, Ustav Informatiky a softveroveho inzinierstva, pp.1-12, 2008.
3. S.Mohamed Saleem, R.Krithiga, S.K.Rani, S.Celin Sindhya, “Study on text summarization using extractive methods”, International Journal of Science, Engineering and Technology Research (IJSETR), Volume 4, Issue 5, May 2015.

Websites referred:

https://www.quora.com

https://www.codepen.io

https://stackoverflow.com