A

MINI PROJECT

ON

TEXT SUMMARIZER

SUBMITED IN PARTIAL FULFILMENT FOR THE COMPLETION OF
BE-V SEMESTER
IN
INFORMATIONTECHNOLOGY
BY

B ARAVIND KUMAR(160117737033)
N ARUN REDDY(160117737035)

UNDER THE GUIDANCE OF

Ms. E. RAMALAKSHMI
ASST. PROFESSOR

DEPARTMENT OF INFORMATION TECHNOLOGY
CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY(A)
(Affiliated to Osmania university, accredited by NBA and NAAC, ISO certified 9001:2015 certified Institution)
GANDIPET, HYDERABAD-500075
website: www.cbit.ac.in
2019-20

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY(A)

DEPARTMENT OF INFORMATION TECHNOLOGY

(Affiliated to Osmania University)

GANDIPET, HYDERABAD - 500075

CERTIFICATE

This is to certify that the project “**TEXT SUMMARIZER**” submitted to **CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY,** in partial fulfilment of the requirements of the requirements for the award of the completion of V semester of B.E in Information Technology, during the academics year 2019-20, is a record of original work done by **B. Aravind Kumar (160117737033), n. Arun Reddy(160117737035)** during the period of study in Department of IT, CBIT, HYDERABAD, under our supervision and guidance.

Project Guide	**Head of Department**
Ms. E. Ramalakshmi	**Dr. Suresh Pabboju**
Asst. Professor, IT Dept.	Professor, Dept. of IT
CBIT, Hyderabad.	CBIT, Hyderabad.

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to Ms. E. Ramalakshmi, our project guide, for her invaluable guidance and constant support, along with her capable instruction and persistent encouragement.

We are grateful to our Head of Department, Dr. Suresh Pabboju, for his steady support and the provision of every resource required for the completion of this project.

We would like to take this opportunity to thank our Principal, Dr. P Ravinder Reddy, as well as the management of the institute, for having designed an excellent learning atmosphere.

Our thanks are due to all members of the staff and our lab assistants for providing us with the help required to carry out the groundwork of this project.

ABSTRACT

Text summarization is the problem of creating a short, accurate, and fluent summary of a longer text document. Automatic text summarization methods are greatly needed to address the ever-growing amount of text data available online to both better help discover relevant information and to consume relevant information faster.There is an enormous amount of textual material, and it is only growing every single day.Think of the internet, comprised of web pages, news articles, status updates, blogs and so much more. The data is unstructured and the best that this can do to navigate it is to use search and skim the results.There is a great need to reduce much of this text data to shorter, focused summaries that capture the salient details, both so this can navigate it more effectively as well as check whether the larger documents contain the information that we are looking for.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

An extractive summarization method consists of selecting important sentences, paragraphs etc. from the original document and concatenating them into shorter form. An Abstractive summarization is an understanding of the main concepts in a document and then express those concepts in clear natural language.

There are two different groups of text summarization: indicative and informative. Inductive summarization only represent the main idea of the text to the user. The typical length of this type of summarization is 5 to 10 percent of the main text. On the other hand, the informative summarization systems gives concise information of the main text .The length of informative summary is 20 to 30 percent of the main text .

1.1 Motivation

Most of the times the user do not read the entire paragraph of any topic. This project helps in reducing the entire paragraph into brief points. This can also reduce the paragraph into required number of points

1.2 Objective of Project

The main objective of this project is to to reduce the time of reading the entire paragraph. The user gives any http link or any text in text area and the user is also allowed to specify the required number of lines to html page. Then the important points of the paragraph is extracted and displayed by using NLP based techniques

1.3 Problem Statement

The goal of text summarization is presenting the source text into a shorter version with semantics. The most important advantage of using a summary is it reduces the reading time. Text summarization methods can be classified into extractive and abstractive summarization

2. EXISTING SYSTEM

A number of Information Extraction Summarization Systems have been developed in specific fields. Even though most of these systems are not currently used in the internet, the potential is great and implementation of such systems in the internet is relatively simple.

Squash

Squash quickly and easily summarizes web pages, news and finance articles, Facebook posts, emails, blog posts and other apps. Squash extracts the major points and provides a 4 or 8 sentence synopsis for easy reading. Plus, in just one click, the user can share your summary via Facebook, Twitter, Google+, text messaging, email and more. Squash is similar to Summly, Trimit, and Wavii. However, it can summarize and share any text, not just news from a few sources. Squash can provide a news summary from any online newspaper or magazine including NBC News, Fox News, the Washington Post, and USA Today.

SemanTer Pro - Text summarizer

This SemanTer Pro takes the text and gives back a summary so the user can get the most valuable information quick and easy. Just copy and past the text the user want summarized into the app, and hit the Summarize button. SemanTer Pro picks out the most important sentences from the text so the user only have to read through what's important.

3. PROPOSED SYSTEM

3.1 Methodology

This project basic idea is to reduce the large chunks of data into important points,so that reader can reduce time of reading entire paragraph. This project uses python and Flask web framework. Text summarizer extracts the major points and provides a required sentence synopsis for easy reading. Text summarizer picks out the most important sentences from the text so the user only have to read through what's important. Here the user give http link or any text data in text area and user also specify the required number of lines to html page. Then the required data is displayed by using NLP based techniques

3.2 Architecture of Proposed System

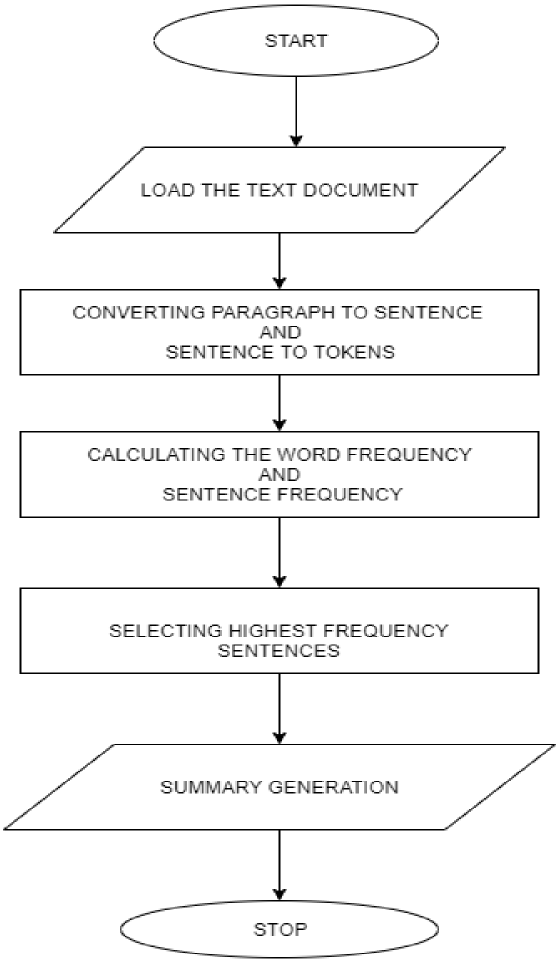

START
LOAD THE TEXT DOCUMENT
CONVERTING PARAGRAPH TO SENTENCE AND SENTENCE TO TOKENS
CALCULATING THE WORD FREQUENCY AND SENTENCE FREQUENCY
SELECTING HIGHEST FREQUENCY SENTENCES
SUMMARY GENERATION
STOP

Fig. 3.1 Flow Chart of proposed system

As shown in Fig. 3.1 it explains the flow of execution of project.Front page loads the text data or web link into browser and it converts paragraphs into sentences , sentences into tokens.Now it calculates the each word frequency. With the help of word frequency sentence frequency is calculated.Then the sentences will be highest frequency count will be considered and summary will be generated

4. SOFTWARE REQUIREMENTS

4.1 Natural Language Processing

Simply and in short, natural language processing (NLP) is about developing applications and services that are able to understand human languages.

We are talking here about practical examples of natural language processing (NLP) like speech recognition, speech translation, understanding complete sentences, understanding synonyms of matching words, and writing complete grammatically correct sentences and paragraphs.

4.2 Flask

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks.

Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

4.3 Text Summarization

Text summarization is a subdomain of Natural Language Processing (NLP) that deals with extracting summaries from huge chunks of texts. There are two main types of techniques used for text summarization: NLP-based techniques and deep learning-based techniques. we will see a simple NLP-based technique for text summarization. We will not use any machine learning library in this article. Rather we will simply use Python's NLTK library for summarizing Wikipedia articles.

4.4 Syntax and semantics

The syntax of the Python programming language is the set of rules that defines how a Python program will be written and interpreted (by both the runtime system and by human readers).The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages. Python was designed to be a highly readable language. It has a relatively uncluttered visual layout and uses English keywords frequently where other languages use punctuation.

4.5 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

A list of most important features of Python language is given below.

- Easy to Learn and Use. Python is easy to learn and use.
- Expressive Language. Python language is more expressive means that it is more understandable and readable.
- Interpreted Language.
- Cross-platform Language.
- Free and Open Source.
- Object-Oriented Language.
- Extensible.
- Large Standard Library.

5. IMPLEMENTATION

5.1 Pre processing

The working process can be divided into two steps: Pre Processing step and Processing step. Pre Processing is structured representation of the original text. It usually includes: a) Sentences boundary identification. In English, sentence boundary is identified with presence of dot at the end of sentence. b) Stop-Word Elimination—Common words with no semantics c) Stemming—The purpose of stemming is to obtain the stem or radix of each word, which emphasize its semantics.

In Processing step, features influencing the relevance of sentences are decided and calculated and the weights are assigned to these features using weight learning method. Final score of each sentence is determined using Feature-weight equation. Top ranked sentences are selected for final summary. Summary evaluation is a very important aspect for text summarization. Generally, summaries can be evaluated using intrinsic or extrinsic measures. While intrinsic methods attempt to measure summary quality using human evaluation and extrinsic methods measure the same through a task-based performance measure such the information retrieval-oriented task.

Accessing web pages

In the summer.py the user has to first import the important libraries required for scraping the data from the web. User then use the urlopen function from the urllib.request utility to scrape the data. Next, user need to call read function on the object returned by urlopen function in order to read the data. To parse the data, user can use BeautifulSoup object and pass it the scraped data object i.e. article and the lxml parser.

In Wikipedia articles, all the text for the article is enclosed inside the <p> tags. To retrieve the text the user need to call find_all function on the object returned by the BeautifulSoup. The tag name is passed as a parameter to the function. The find_all function returns all the paragraphs in the article in the form of a list. All the paragraphs have been combined to recreate the article.

In app.py file, import packages such as Flask,request and jsonify. Flask (web framework) Flask is a micro web framework written in Python. It is classified as a microframework 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. Requests is a Python module that you can use to send all kinds of HTTP requests. jsonify is a syntax for storing and exchanging data

Text Summarization Steps

The following is a paragraph from one of the famous speeches by Denzel Washington at the 48th NAACP Image Awards:

So, keep working. Keep striving. Never give up. Fall down seven times, get up eight. Ease is a greater threat to progress than hardship. Ease is a greater threat to progress than hardship. So, keep moving, keep growing, keep learning. See you at work.

The user can see from the paragraph above that writer is basically motivating others to work hard and never give up. To summarize the above paragraph using NLP-based techniques user need to follow a set of steps, which will be described in the following sections.

Regular Expressions

A regular expression in a programming language is a special text string used for describing a search pattern. It is extremely useful for extracting information from text such as code, files, log, spreadsheets or even documents.

While using the regular expression the first thing is to recognize is that everything is essentially a character, and writing patterns to match a specific sequence of characters also referred as string. Ascii or latin letters are those that are on the keyboards and Unicode is used to match the foreign text. It includes digits and punctuation and all special characters like $#@!%, etc.

Convert Paragraphs to Sentences

First the user need to convert the whole paragraph into sentences. The most common way of converting paragraphs to sentences is to split the paragraph whenever a period is encountered. So if the user split the paragraph under discussion into sentences, we get the following sentences:

- So, keep working
- Keep striving
- Never give up
- Fall down seven times, get up eight
- Ease is a greater threat to progress than hardship
- Ease is a greater threat to progress than hardship
- So, keep moving, keep growing, keep learning
- See you at work

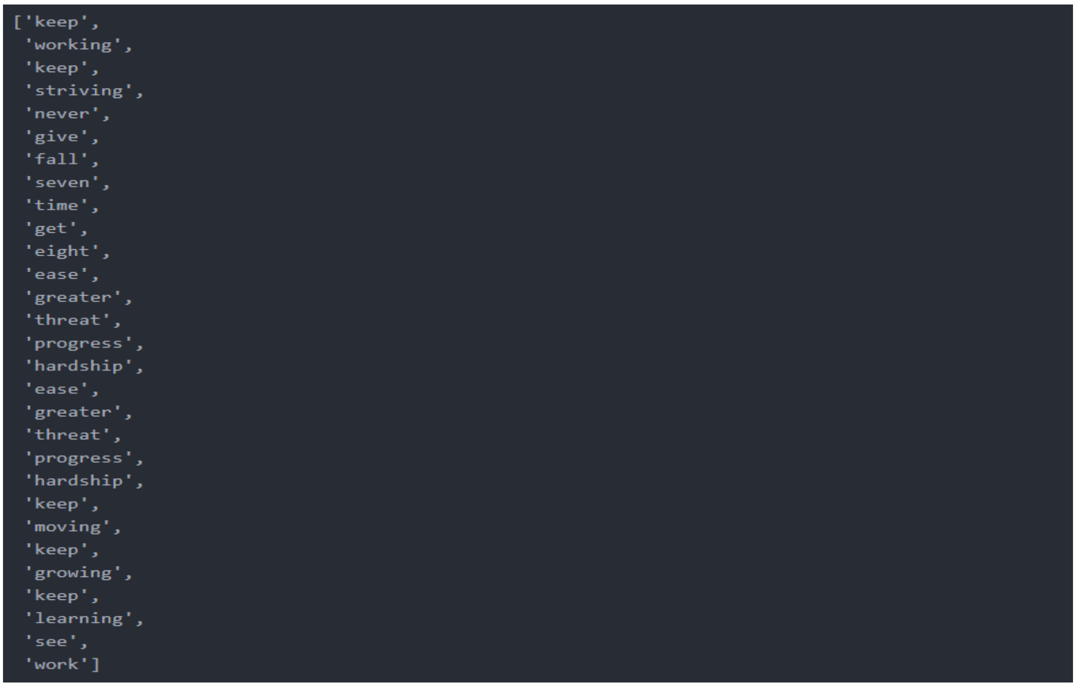
Text Preprocessing

After converting paragraph to sentences, user need to remove all the special characters, stop words and numbers from all the sentences. After preprocessing, user get the following sentences:

- keep working
- keep striving
- never give
- fall seven time get eight
- ease greater threat progress hardship
- ease greater threat progress hardship
- keep moving keep growing keep learning
- see work

Tokenizing the Sentences

The user need to tokenize all the sentences to get all the words that exist in the sentences. After tokenizing the sentences, user gets list of following words:



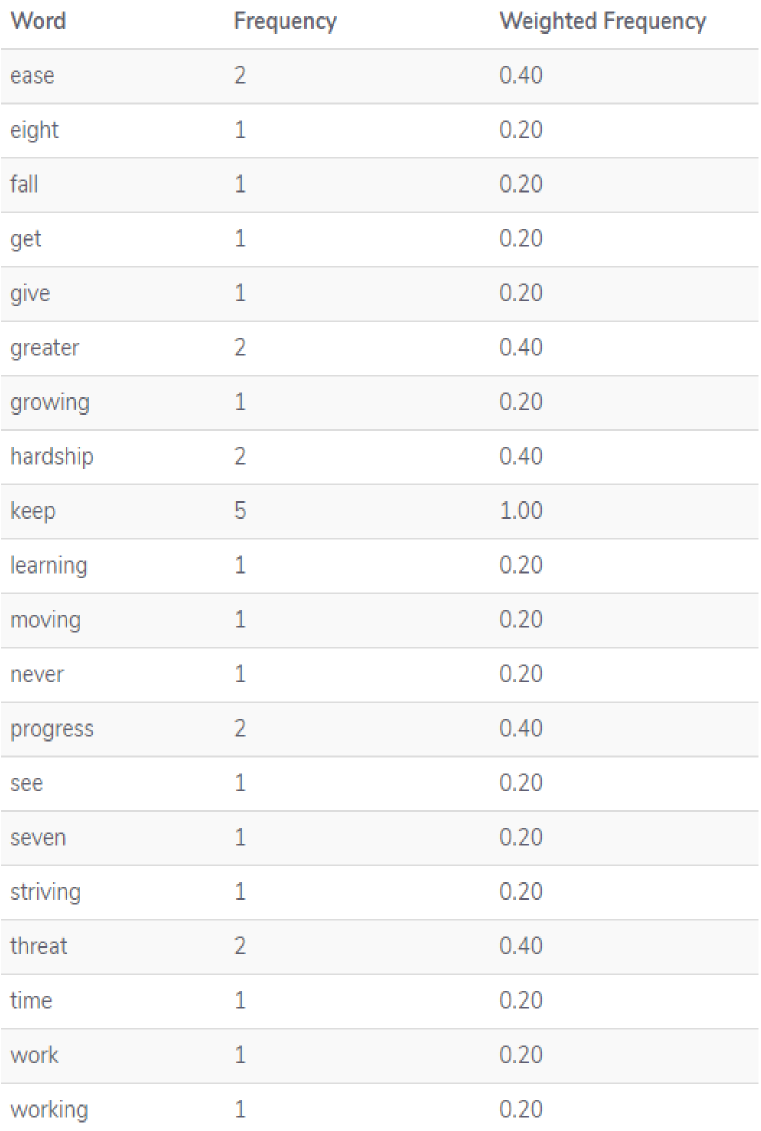
```
['keep',
 'working',
 'keep',
 'striving',
 'never',
 'give',
 'fall',
 'seven',
 'time',
 'get',
 'eight',
 'ease',
 'greater',
 'threat',
 'progress',
 'hardship',
 'ease',
 'greater',
 'threat',
 'progress',
 'hardship',
 'keep',
 'moving',
 'keep',
 'growing',
 'keep',
 'learning',
 'see',
 'work']
```

Fig. 5.1 Tokenizing the sentence into words

As shown in Fig 5.1 in each sentence words are divided into tokens

Find Weighted Frequency of Occurrence

Next user need to find the weighted frequency of occurrences of all the words. User can find the weighted frequency of each word by dividing its frequency by the frequency of the most occurring word. The following table contains the weighted frequencies for each word:



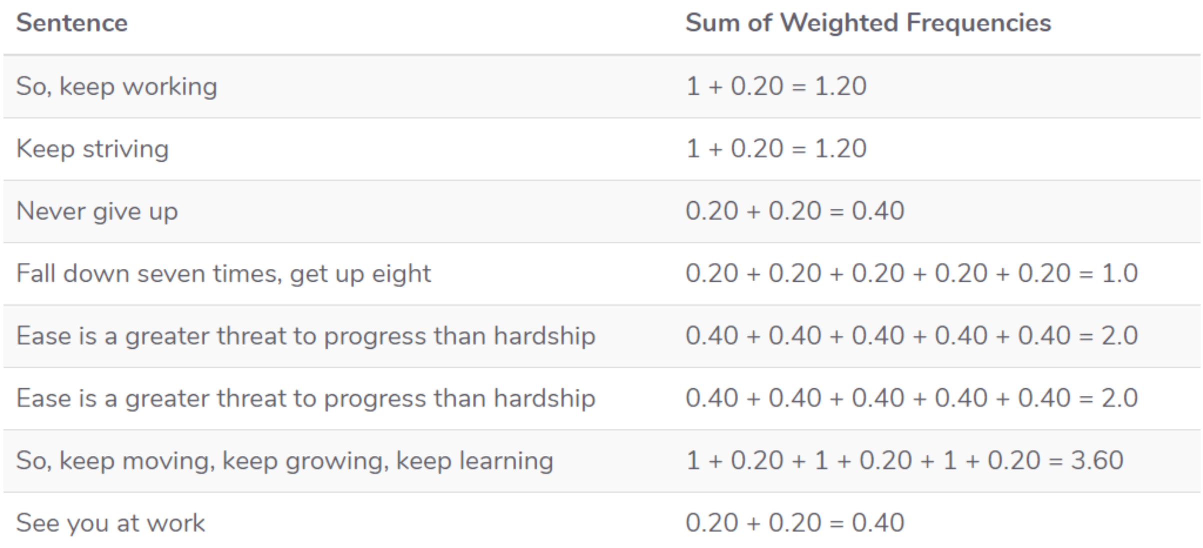
Word	Frequency	Weighted Frequency
ease	2	0.40
eight	1	0.20
fall	1	0.20
get	1	0.20
give	1	0.20
greater	2	0.40
growing	1	0.20
hardship	2	0.40
keep	5	1.00
learning	1	0.20
moving	1	0.20
never	1	0.20
progress	2	0.40
see	1	0.20
seven	1	0.20
striving	1	0.20
threat	2	0.40
time	1	0.20
work	1	0.20
working	1	0.20

Fig. 5.2 Word frequency table

As shown in Fig. 5.2 frequency of each word is calculated in frequency table Since the word "keep" has the highest frequency of 5, therefore the weighted frequency of all the words have been calculated by dividing their number of occurances by 5.

Replace Words by Weighted Frequency in Original Sentences

The final step is to plug the weighted frequency in place of the corresponding words in original sentences and finding their sum. It is important to mention that weighted frequency for the words removed during preprocessing (stop words, punctuation, digits etc.) will be zero and therefore is not required to be added, as mentioned below:



Sentence	Sum of Weighted Frequencies
So, keep working	1 + 0.20 = 1.20
Keep striving	1 + 0.20 = 1.20
Never give up	0.20 + 0.20 = 0.40
Fall down seven times, get up eight	0.20 + 0.20 + 0.20 + 0.20 + 0.20 = 1.0
Ease is a greater threat to progress than hardship	0.40 + 0.40 + 0.40 + 0.40 + 0.40 = 2.0
Ease is a greater threat to progress than hardship	0.40 + 0.40 + 0.40 + 0.40 + 0.40 = 2.0
So, keep moving, keep growing, keep learning	1 + 0.20 + 1 + 0.20 + 1 + 0.20 = 3.60
See you at work	0.20 + 0.20 = 0.40

Fig. 5.3 Sentence frequency table

As shown in Fig. 5.3 frequency of each sentence is calculated and selects the highest frequency count sentence

Sort Sentences in Descending Order of Sum

The final step is to sort the sentences in inverse order of their sum. The sentences with highest frequencies summarize the text. For instance, look at the sentence with the highest sum of weighted frequencies:

So, keep moving, keep growing, keep learning

The user can easily judge that what the paragraph is all about. Similarly, user can add the sentence with the second highest sum of weighted frequencies to have a more informative summary. Take a look at the following sentences:

So, keep moving, keep growing, keep learning. Ease is a greater threat to progress than hardship.

These two sentences give a pretty good summarization of what was said in the paragraph.

6. RESULTS

To execute the project an IDE is required. In this project visual studio code has been used. First load the project into the IDE. Now all the packages which are required should be installed. Clean and build the project and run the whole project. The results can be seen in http://127.0.0.1:5000 address.

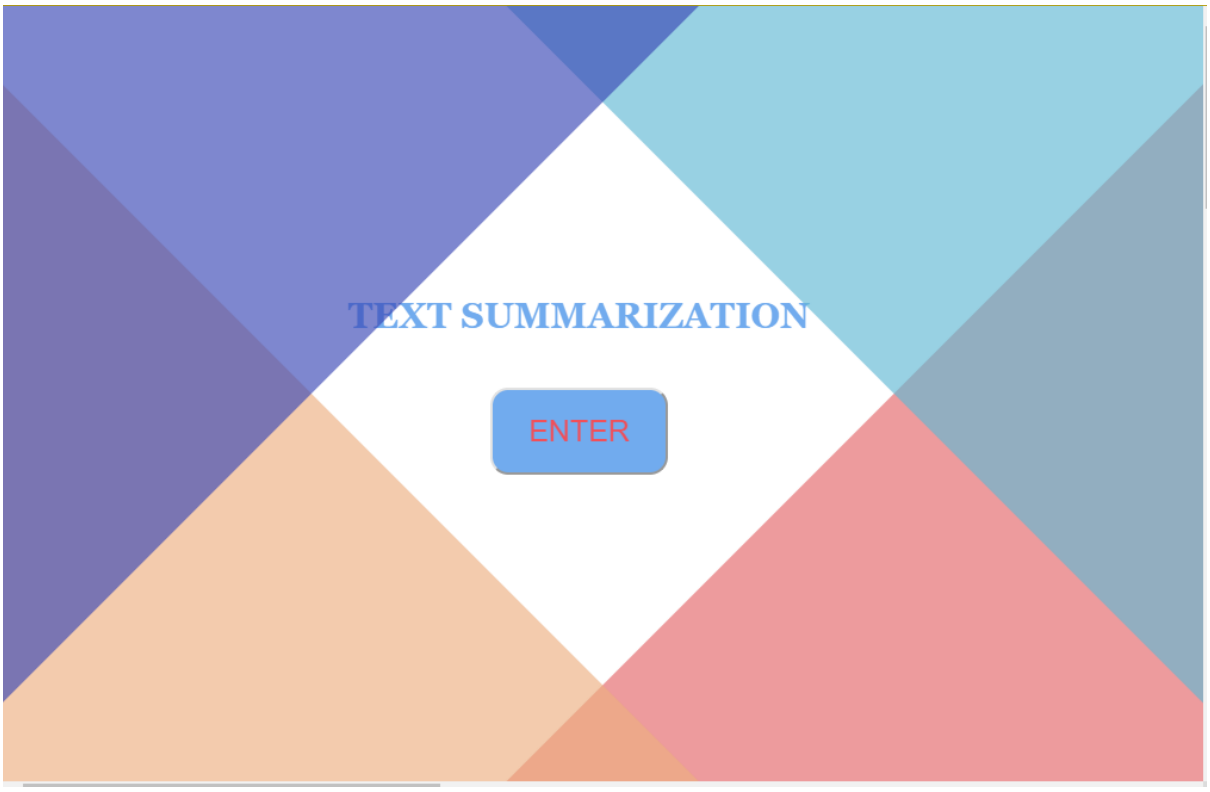
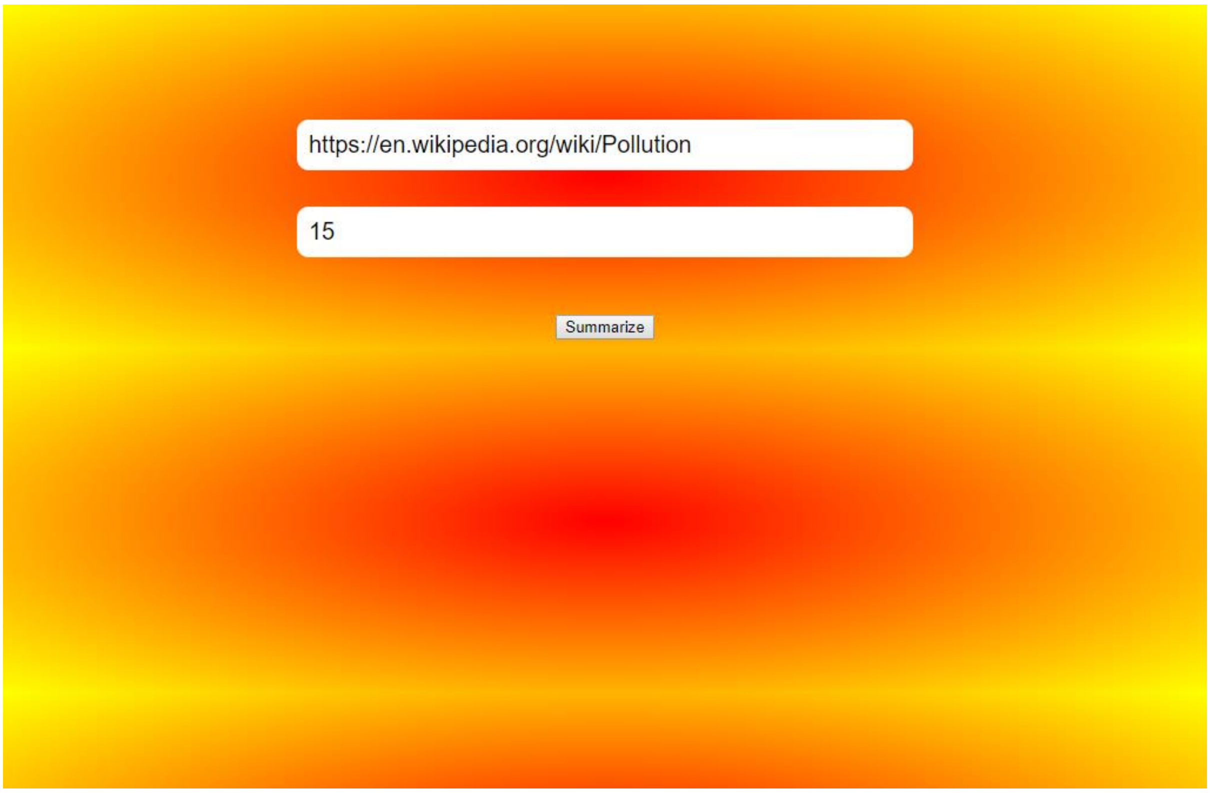


Fig. 5.4 Starting page

As shown in Fig. 5.4 after execution of project this is the first web page of our project after user click ENTER it redirects to next web page

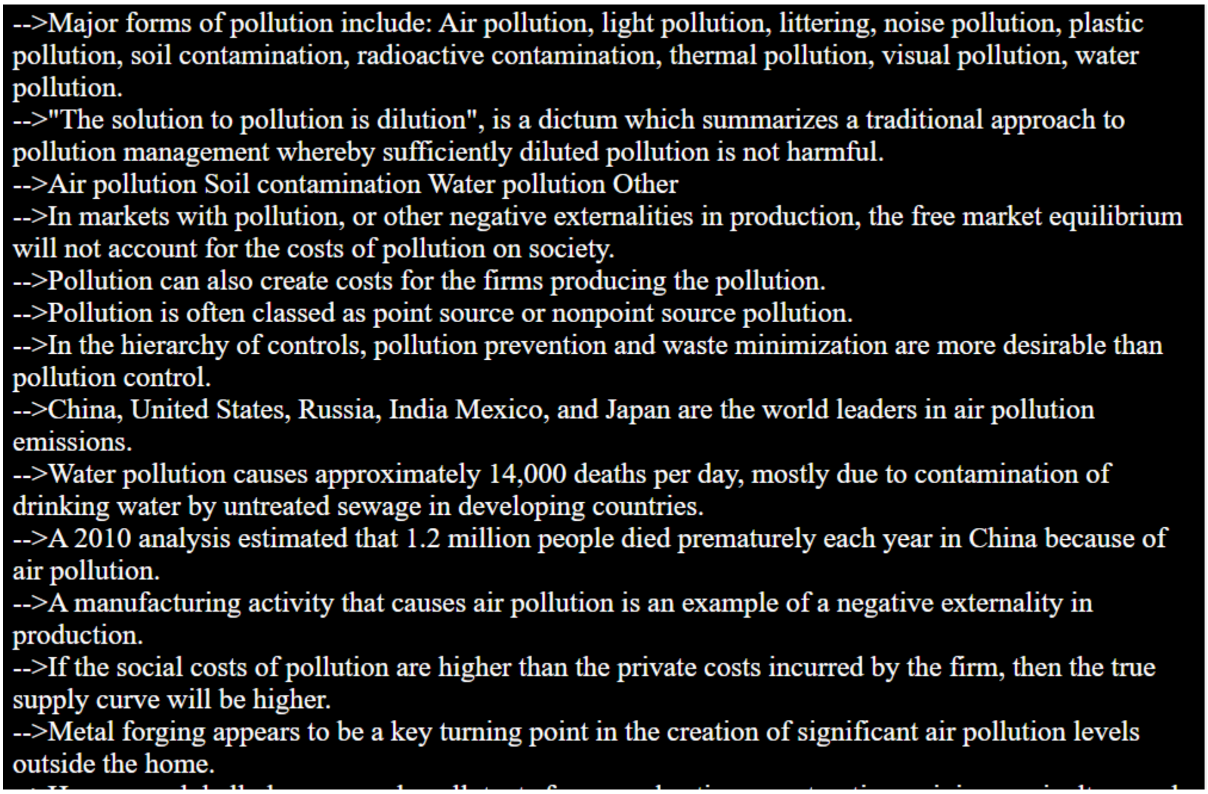


https://en.wikipedia.org/wiki/Pollution

15

Fig. 5.5 Input page

As shown in Fig. 5.5 here the user give any paragraph or http link in the first text field and user specify the number of lines in the next text filed and we click summerize button.



-->Major forms of pollution include: Air pollution, light pollution, littering, noise pollution, plastic pollution, soil contamination, radioactive contamination, thermal pollution, visual pollution, water pollution.

-->"The solution to pollution is dilution", is a dictum which summarizes a traditional approach to pollution management whereby sufficiently diluted pollution is not harmful.

-->Air pollution Soil contamination Water pollution Other

-->In markets with pollution, or other negative externalities in production, the free market equilibrium will not account for the costs of pollution on society.

-->Pollution can also create costs for the firms producing the pollution.

-->Pollution is often classed as point source or nonpoint source pollution.

-->In the hierarchy of controls, pollution prevention and waste minimization are more desirable than pollution control.

-->China, United States, Russia, India Mexico, and Japan are the world leaders in air pollution emissions.

-->Water pollution causes approximately 14,000 deaths per day, mostly due to contamination of drinking water by untreated sewage in developing countries.

-->A 2010 analysis estimated that 1.2 million people died prematurely each year in China because of air pollution.

-->A manufacturing activity that causes air pollution is an example of a negative externality in production.

-->If the social costs of pollution are higher than the private costs incurred by the firm, then the true supply curve will be higher.

-->Metal forging appears to be a key turning point in the creation of significant air pollution levels outside the home.

Fig 5.6 Output page

As shown in Fig. 5.6 after clicking summerize button the user get the above web page by summarizing the paragraph into specified number of lines based on nlp techniques

6. CONCLUSION AND FUTURE SCOPE

This project is designed such a way that the user is allowed to summerize any http link or text data. This project mainly helps to reduce the large chunks of data into brief summary.It selects the highest frequency words in the paragraph and selects the sentence score and highest sentence score will be selected and generates required summary. At present it just takes the frequency of the words in text and returns the output.

In future it can be developed using advanced machine learning techniques to give its exact meaning of the paragraph .This project can also deployed into android app with many features like saving in the device, google drive. To this project login page can be added and previous summarised data can be retrieved using login credentials. User can add the notes to the end of each page after summarisation

BIBILIOGRAPHY

[1] https://glowingpython.blogspot.com/2014/09/text-summarization-with-nltk.html

[2] https://stackabuse.com/text-summarization-with-nltk-in-python/

[3] https://ai.intelligentonlinetools.com/ml/text-summarization/

[4] https://www.udemy.com/course/natural-language-processing-with-python-and-nltk/

[5] https://www.analyticsvidhya.com/blog/2018/11/summarization-textrank-python/

[6] https://nptel.ac.in/courses/106105158/

[7] https://swayam.gov.in/nd1_noc19_cs56

[8] https://machinelearningmastery.com/natural-language-processing/

[9] https://www.coursera.org/learn/language-processing

[10] https://www.nltk.org/book/ch03.html