JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

TEXT SUMMARIZATION AND DETECTION BASED WEBAPPLICATION

[1]Santhosh Voruganti, [2]U.Sairam

[1]Assistant Professor, [2]Assistant Professor
[1,2]Information Technology,
[1,2]CBIT, Hyderabad, India

Abstract : Text summarization refers to the technique of shortening long pieces of text. The intention is to create a coherent and fluent summary having only the main points outlined in the document. As theamount of information present on the internet is increasing, the need for summarization of long pieces of text is more important than ever, as reading long pieces of text may be time consuming.

We aim to deploy the extractive summarization approach to achieve automated summarization. The extractive approach involves picking up the most important phrases and lines from the documents. It then combines all the important lines to create the summary. So, in this case, everyline and word of the summary actually belongs to the original document which is summarized. Sentences are ranked by assigning weights and they are ranked based on their weights. Highly ranked sentences are extracted from the input document so it extracts important sentences which directs to a high-quality summary of the input document and store summary as adocument.

Briefly explaining, we aim to build a web application using flask which will be able to recognize and detect text from various sources, summarize it using Natural Language Processing (NLP) algorithms and libraries such as Natural Language Toolkit (NLTK), gensim, spacy, sumy and thensave the summary in a document.

***IndexTerms* - Component, formatting, style, styling, insert.**

I. Introduction

Text summarization is the method 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 andto consume relevant information faster.

With the present explosion of data circulating the digital space, which is mostly non-structured textual data, there is a need to develop automatic text summarization tools that allow people to getinsights from them easily. Currently, we enjoy quick access to enormous amounts of information.However, most of this information is redundant, insignificant, and may not convey the intended meaning. For example, if you are looking for specific information from an online news article, youmay have to dig through its content and spend a lot of time weeding out the unnecessary stuff before getting the information you want. Implementing summarization can enhance the readabilityof documents, reduce the time spent in researching for information, and allow for more informationto be fitted in a particular area.

Automatic summarization is the process of shortening a set of data computationally, to create a subset that represents the most important or relevant information within the original content. In addition to text, images and videos can also be summarized. Extractive text summarization involves the selection of phrases and sentences from the source document to make up the new summary. Techniques involve ranking the relevance of phrases in order to choose only those mostrelevant to the meaning of the source.

A.PROBLEM DEFINITION

With such a big amount of data circulating in the digital space, there is need to develop machine learning algorithms that can automatically shorten longer texts and deliver accurate summaries thatcan fluently pass the intended messages. Automatic text summarization is a common problem in machine learning and natural language processing (NLP). Machine learning models are usually trained to understand documents and distill the useful information before outputting the required summarized texts.

The primary objective of the paper is to build a web application using flask which will be able torecognize text from various sources such as image, URL or raw format and summarize it using various Natural Language Processing (NLP) algorithms and libraries such as Natural Language Toolkit (NLTK), gensim, spacy, sumy, etc and then save the summary in a document.

II.SYSTEM DESIGN

The developed system generates an efficient and fluent summary for the text data given as input which can be collected from various sources such as URL, image or even raw text. The summarization is performed using Natural Language ToolKit (NLTK) library which enables various language processing functions such as summarization, tokenization, etc.

The design of how the application works is as explained below:

- ***Data extraction from various sources; URL, image or raw:***
  The data which is to be summarized is extracted from various sources as per the convenience of the user. Various necessary libraries and frameworks are deployed to extract the informationaccording to the source of the user.

- ***Clean-up of the data:***
  Any unnecessary information from the extracted data is removed, thus removing any noisy data and giving clean information.

- ***Tokenization:***
  Word tokenization is the process of splitting a large sample of text into words. This is a requirement in natural language processing tasks where each word needs to be captured and subjected to further analysis like classifying and counting them for a particular sentiment etc.

- ***Calculation of word frequency for each word:***
  Calculates the occurrence of each word in a sentence. It is basically done to rank the importanceof words.

- ***Calculation of weighted frequency for each sentence***:
  After ranking the words, sentence frequency is generated in order to predict the rank of each sentence. This is done using the algorithms in the NLTK library.

  - ***Generation of the summary***:
    Finally, the summary will be displayed to the user after choosing 30% of the top weightedsentences.

The figure 2.1 below clearly illustrates how the text summarization application is integrated to carry out its flow of work from both the front-end and back-end perspective.

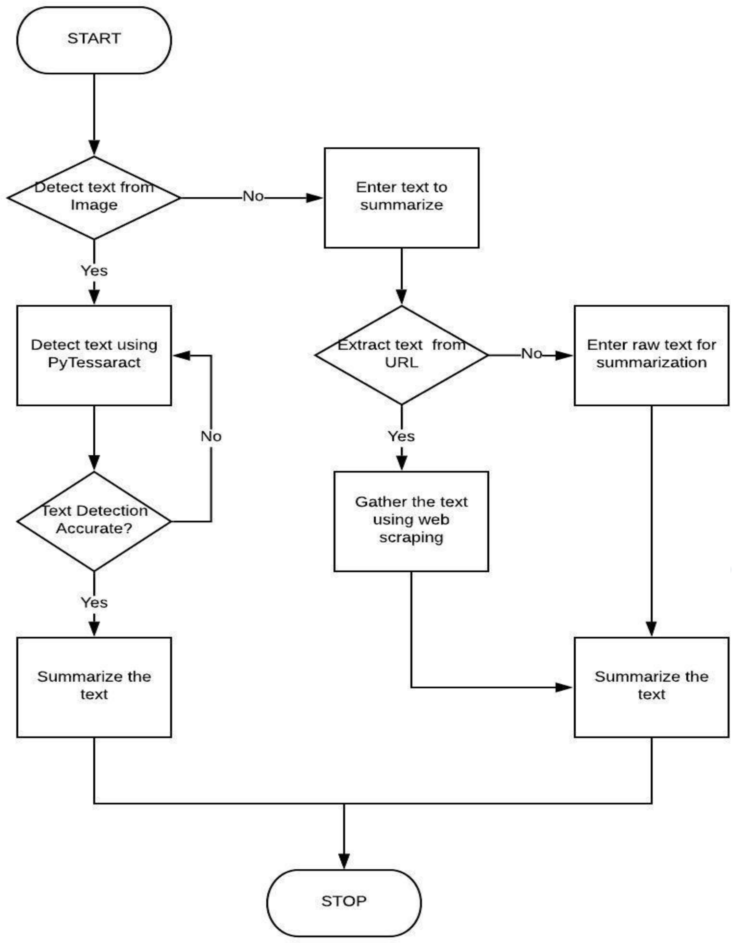


Figure:2.1 Flow chart of the summarization application

The use case diagram of the summarization application is displayed in figure 2.2. The diagram gives clear depiction of the various user functionalities and also the tasks that the summarization tool can perform.

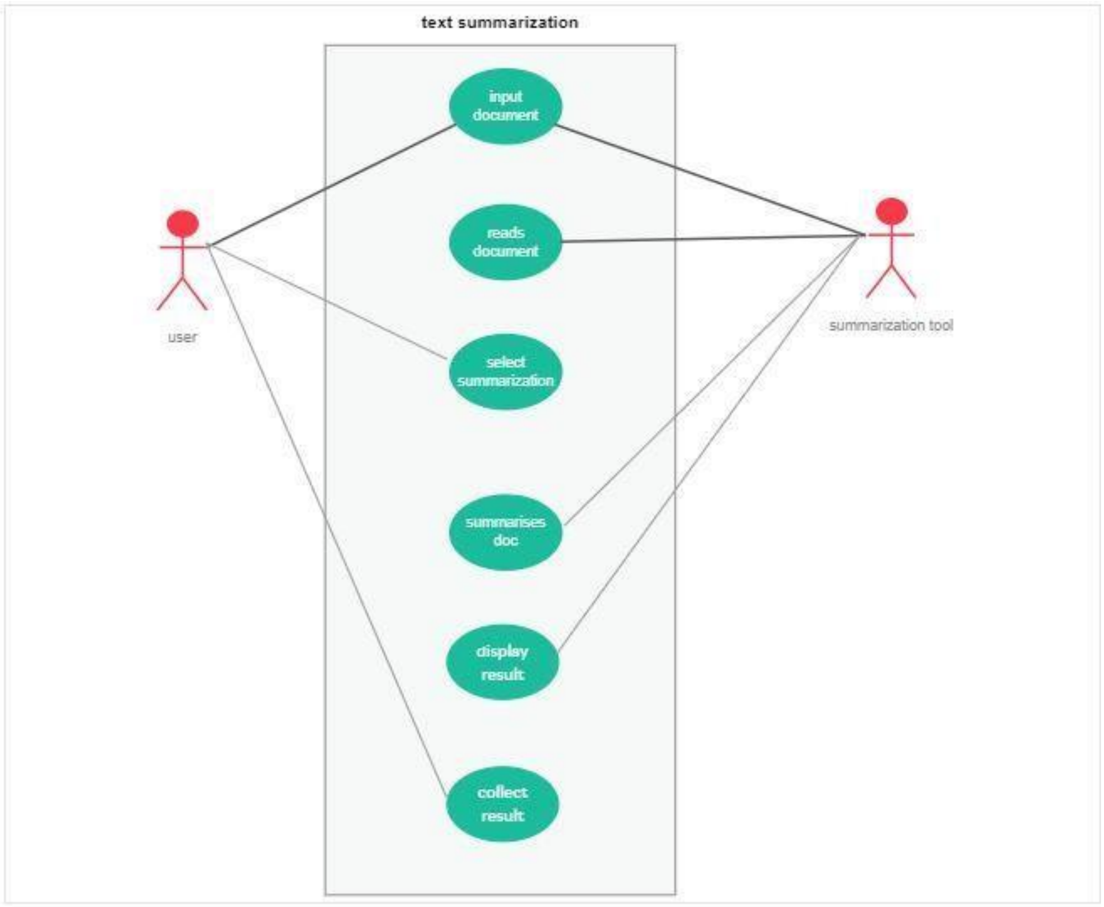


Figure 2.2: Use case diagram for the summarization application

III.IMPLEMENTATION

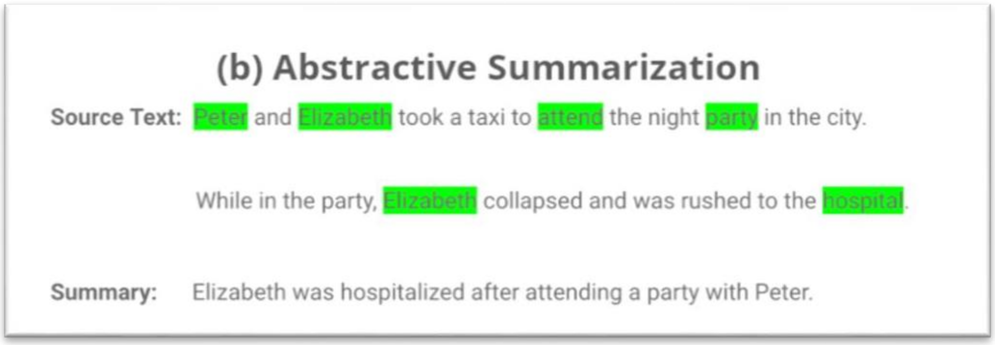
B.ABSTRACTIVE SUMMARIZATION

Abstractive methods select words based on semantic understanding, even those words did not appear in the source documents. It aims at producing important material in a new way. They interpret and examine the text using advanced natural language techniques in order to generate anew shorter text that conveys the most critical information from the original text.

It can be correlated to the way human reads a text article or blog post and then summarizes in their own word.

Input document → understand context → semantics → create own summary.

The figure 3.1 below, contains an example of abstractive summarization. We can clearly notice how new sentences are formed after analyzing the words in the given sentences.


(b) Abstractive Summarization

Source Text: Peter and Elizabeth took a taxi to attend the night party in the city.

While in the party, Elizabeth collapsed and was rushed to the hospital.

Summary: Elizabeth was hospitalized after attending a party with Peter.

Figure 3.1 : Abstractive summarization; An example

C.EXTRACTIVE SUMMARIZATION

Extractive methods attempt to summarize articles by selecting a subset of words that retain the most important points. This approach weights the important part of sentences and uses the same to form the summary. Different algorithm and techniques are used to define weights for the sentences and further rank them based on importance and similarity among each other.

Input document → sentences similarity → weight sentences → select sentences with higher rank.

The figure 3.2 below, clearly explains how extractive summarization technique is used to form the summary. Here, it can be notices how the sentences are analyzed, and the words from withinthe same sentences are used to form the summary.

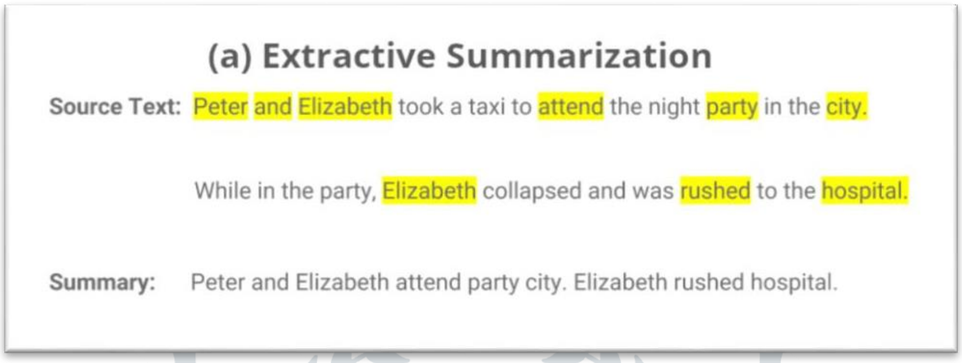

(a) Extractive Summarization
Source Text: Peter and Elizabeth took a taxi to attend the night party in the city.
While in the party, Elizabeth collapsed and was rushed to the hospital.
Summary: Peter and Elizabeth attend party city. Elizabeth rushed hospital.

Figure 3.2: Extractive summarization

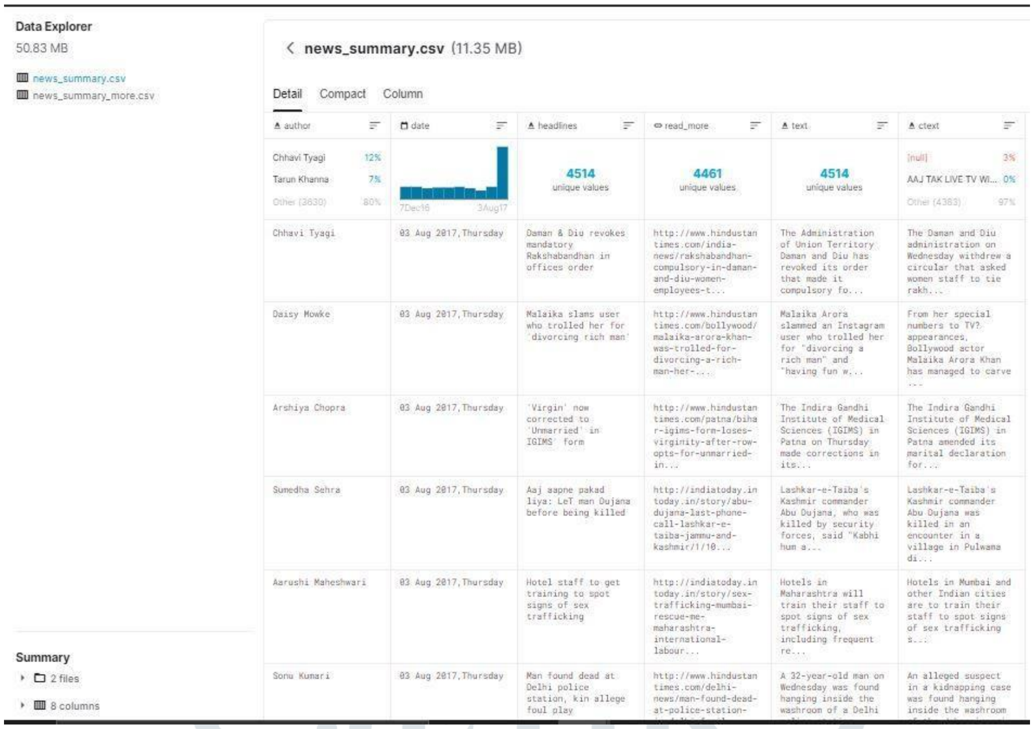
The limited study is available for abstractive summarization as it requires a deeper understanding of the text as compared to the extractive approach.

Purely extractive summaries often times give better results compared to automatic abstractive summaries. This is because of the fact that abstractive summarization methods cope with problems such as semantic representation, inference and natural language generation which is relatively harder than data-driven approaches such as sentence extraction.

D.DATASET

The dataset used for training the summarization model is titled news summary dataset. The source for the dataset is kaggle. The dataset consists of 4515 examples and contains Author_name, Headlines, Url of Article, Short text, Complete Article. The summarized news from In shorts and only scraped the news articles from Hindu, Indian times and Guardian. Timeperiod ranges from February to August 2017.

The figures below contain a small sample of the dataset. The images help in understanding theschema of the dataset.



Data Explorer
50.83 MB

news_summary.csv
news_summary_more.csv

Summary
2 files
8 columns

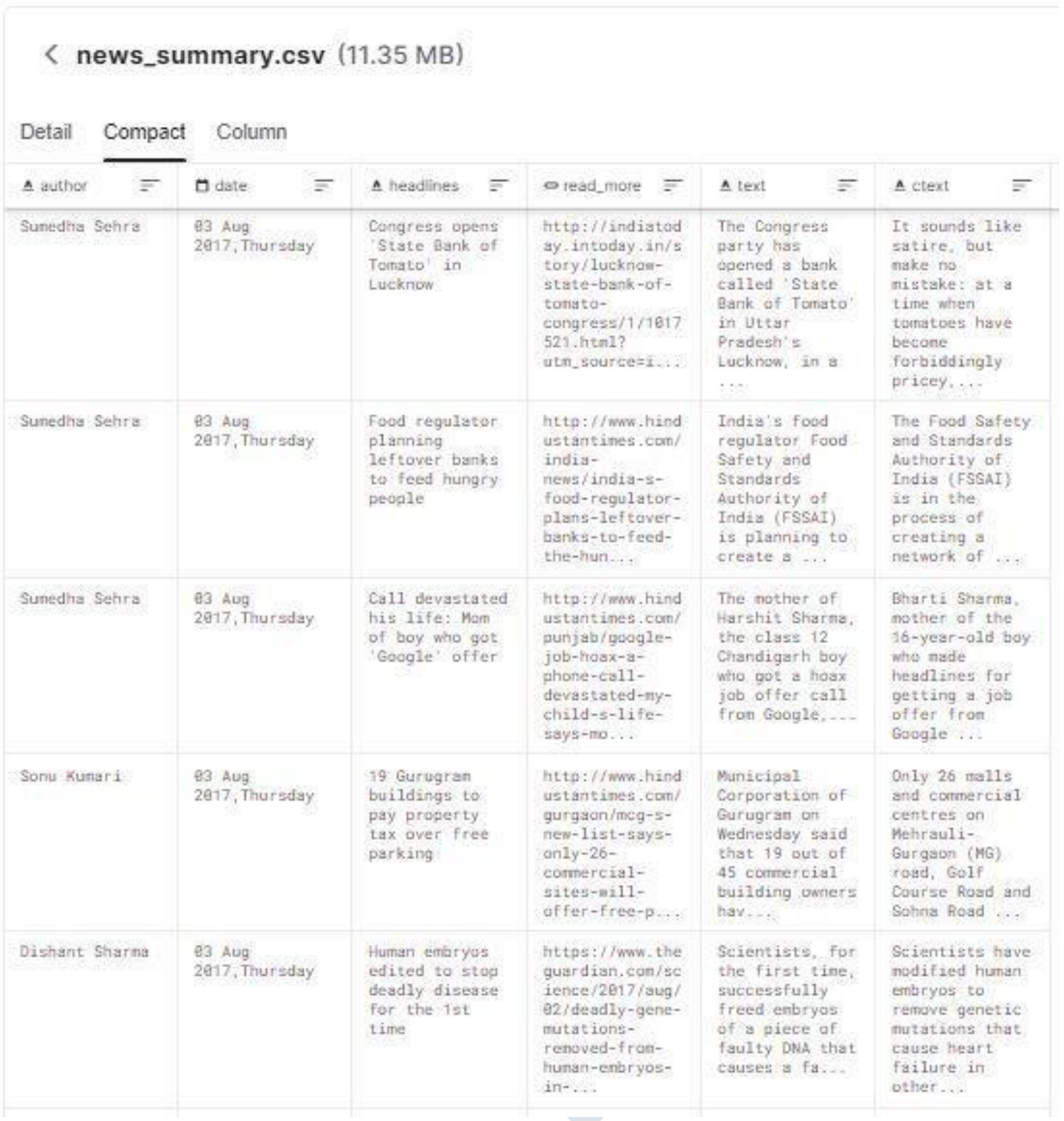
news_summary.csv (11.35 MB)

Detail Compact Column

author	date	headlines	read_more	text	ctext
Chhavi Tyagi 12% Tarun Khanna 7% Other (3630) 80%	7Dec16 3Aug17	4514 unique values	4461 unique values	4514 unique values	[null] 3% AAJ TAK LIVE TV WI... 0% Other (4383) 97%
Chhavi Tyagi	03 Aug 2017, Thursday	Daman & Diu revokes mandatory Rakshabandhan in offices order	http://www.hindustantimes.com/india-news/rakshabandhan-compulsory-in-daman-and-diu-women-employees-t...	The Administration of Union Territory Daman and Diu has revoked its order that made it compulsory fo...	The Daman and Diu administration on Wednesday withdrew a circular that asked women staff to tie rakh...
Daisy Mowke	03 Aug 2017, Thursday	Malaika slams user who trolled her for 'divorcing rich man'	http://www.hindustantimes.com/bollywood/malaika-arora-khan-was-trolled-for-divorcing-a-rich-man-her-...	Malaika Arora slammed an Instagram user who trolled her for "divorcing a rich man" and "having fun w...	From her special numbers to TV? appearances, Bollywood actor Malaika Arora Khan has managed to carve ...
Arshiya Chopra	03 Aug 2017, Thursday	'Virgin' now corrected to 'Unmarried' in IGIMS' form	http://www.hindustantimes.com/patna/bihar-igims-form-loses-virginity-after-row-opts-for-unmarried-in...	The Indira Gandhi Institute of Medical Sciences (IGIMS) in Patna on Thursday made corrections in its...	The Indira Gandhi Institute of Medical Sciences (IGIMS) in Patna amended its marital declaration for...
Sumedha Sehra	03 Aug 2017, Thursday	Aaj aapne pakad liya: LeT man Dujana before being killed	http://indiatoday.intoday.in/story/abu-dujana-last-phone-call-lashkar-e-taiba-jammu-and-kashmir/1/10...	Lashkar-e-Taiba's Kashmir commander Abu Dujana, who was killed by security forces, said "Kabhi hum a...	Lashkar-e-Taiba's Kashmir commander Abu Dujana was killed in an encounter in a village in Pulwama di...
Aarushi Maheshwari	03 Aug 2017, Thursday	Hotel staff to get training to spot signs of sex trafficking	http://indiatoday.intoday.in/story/sex-trafficking-mumbai-rescue-me-maharashtra-international-labour...	Hotels in Maharashtra will train their staff to spot signs of sex trafficking, including frequent re...	Hotels in Mumbai and other Indian cities are to train their staff to spot signs of sex trafficking s...
Sonu Kumari	03 Aug 2017, Thursday	Man found dead at Delhi police station, kin allege foul play	http://www.hindustantimes.com/delhi-news/man-found-dead-at-police-station-	A 32-year-old man on Wednesday was found hanging inside the washroom of a Delhi	An alleged suspect in a kidnapping case was found hanging inside the washroom

Figure 3.3: A subset of the news summarization dataset from kaggle

The figure 3.3 clearly shows the count of the samples from the dataset and also contains some of the samples of the dataset.



news_summary.csv (11.35 MB)

Detail Compact Column

author	date	headlines	read_more	text	ctext
Sumedha Sehra	03 Aug 2017,Thursday	Congress opens 'State Bank of Tomato' in Lucknow	http://indiatoday.intoday.in/story/lucknow-state-bank-of-tomato-congress/1/1017521.html?utm_source=i...	The Congress party has opened a bank called 'State Bank of Tomato' in Uttar Pradesh's Lucknow, in a ...	It sounds like satire, but make no mistake: at a time when tomatoes have become forbiddingly pricey,...
Sumedha Sehra	03 Aug 2017,Thursday	Food regulator planning leftover banks to feed hungry people	http://www.hindustantimes.com/india-news/india-s-food-regulator-plans-leftover-banks-to-feed-the-hun...	India's food regulator Food Safety and Standards Authority of India (FSSAI) is planning to create a ...	The Food Safety and Standards Authority of India (FSSAI) is in the process of creating a network of ...
Sumedha Sehra	03 Aug 2017,Thursday	Call devastated his life: Mom of boy who got 'Google' offer	http://www.hindustantimes.com/punjab/google-job-hoax-a-phone-call-devastated-my-child-s-life-says-mo...	The mother of Harshit Sharma, the class 12 Chandigarh boy who got a hoax job offer call from Google,...	Bharti Sharma, mother of the 16-year-old boy who made headlines for getting a job offer from Google ...
Sonu Kumari	03 Aug 2017,Thursday	19 Gurugram buildings to pay property tax over free parking	http://www.hindustantimes.com/gurgaon/mcg-s-new-list-says-only-26-commercial-sites-will-offer-free-p...	Municipal Corporation of Gurugram on Wednesday said that 19 out of 45 commercial building owners hav...	Only 26 malls and commercial centres on Mehrauli-Gurgaon (MG) road, Golf Course Road and Sohna Road ...
Dishant Sharma	03 Aug 2017,Thursday	Human embryos edited to stop deadly disease for the 1st time	https://www.theguardian.com/science/2017/aug/02/deadly-gene-mutations-removed-from-human-embryos-in-...	Scientists, for the first time, successfully freed embryos of a piece of faulty DNA that causes a fa...	Scientists have modified human embryos to remove genetic mutations that cause heart failure in other...

Figure 3.4: Samples from the dataset

The figure 3.4 above, contains some more samples from the dataset which was used for training the summarization model.

E.SUMMARIZATION USING RAW TEXT

The working methodology of the text summarization process when raw text is given as input isclearly explained with the help of a flowchart as shown in figure 3.5 below:

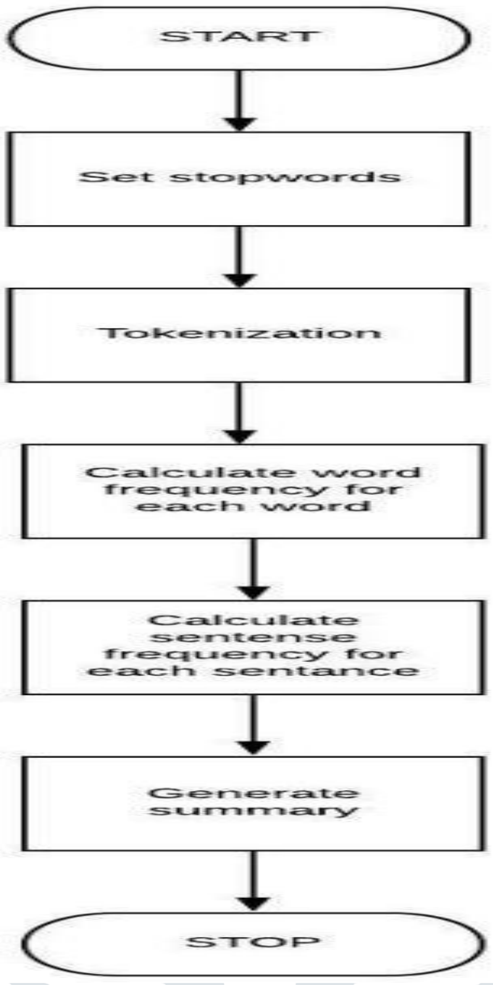
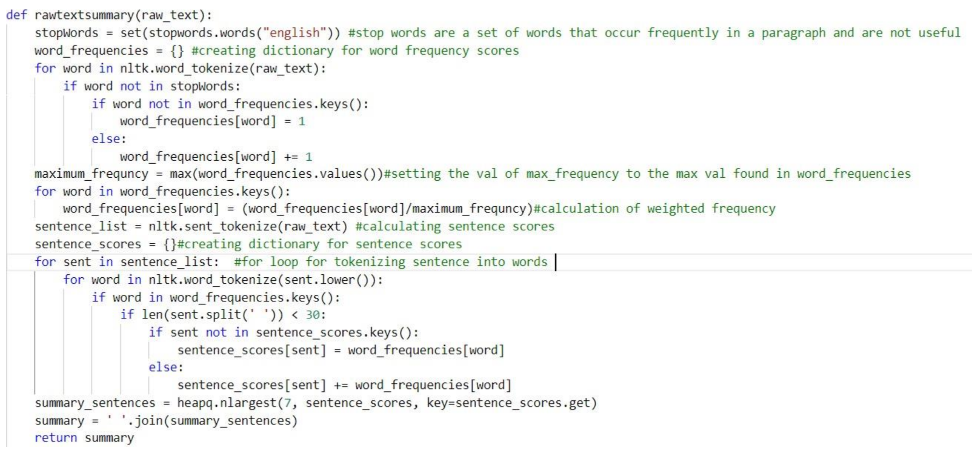

START
Set stopwords
Tokenization
Calculate word frequency for each word
Calculate sentense frequency for each sentance
Generate summary
STOP

Figure 3.5: Summarization process using raw text

The figure 3.6 below contains the code snippet for summarization using raw text, with a briefexplanation of the code.



```
def rawtextsummary(raw_text):
    stopWords = set(stopwords.words("english")) #stop words are a set of words that occur frequently in a paragraph and are not useful
    word_frequencies = {} #creating dictionary for word frequency scores
    for word in nltk.word_tokenize(raw_text):
        if word not in stopWords:
            if word not in word_frequencies.keys():
                word_frequencies[word] = 1
            else:
                word_frequencies[word] += 1
    maximum_frequncy = max(word_frequencies.values())#setting the val of max_frequency to the max val found in word_frequencies
    for word in word_frequencies.keys():
        word_frequencies[word] = (word_frequencies[word]/maximum_frequncy)#calculation of weighted frequency
    sentence_list = nltk.sent_tokenize(raw_text) #calculating sentence scores
    sentence_scores = {}#creating dictionary for sentence scores
    for sent in sentence_list:  #for loop for tokenizing sentence into words
        for word in nltk.word_tokenize(sent.lower()):
            if word in word_frequencies.keys():
                if len(sent.split(' ')) < 30:
                    if sent not in sentence_scores.keys():
                        sentence_scores[sent] = word_frequencies[word]
                    else:
                        sentence_scores[sent] += word_frequencies[word]
    summary_sentences = heapq.nlargest(7, sentence_scores, key=sentence_scores.get)
    summary = ' '.join(summary_sentences)
    return summary
```

Figure 3.6: Code snippet for raw text summarization

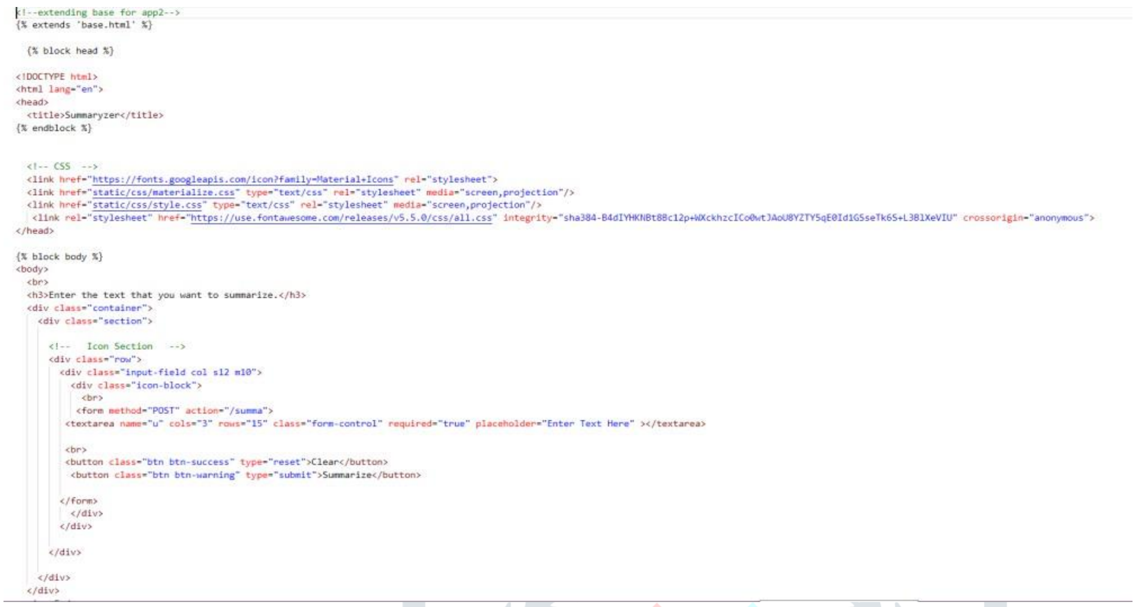
The general working methodology of the code is as explained below:

The data when collected in a raw format, is first cleaned to removed punctuations, stopwords, numeric values, special characters, etc.

This sentence in then tokenized into words and the word frequency is generated.

Then, the weighted frequencies of the sentences are calculated. Finally, the summary is generated

The figure 3.7 contains the HTML code for the development of the initial page of raw textsummarization



```html
<!--extending base for app2-->
{% extends 'base.html' %}

  {% block head %}

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Summaryzer</title>
{% endblock %}


  <!-- CSS  -->
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
  <link href="static/css/materialize.css" type="text/css" rel="stylesheet" media="screen,projection"/>
  <link href="static/css/style.css" type="text/css" rel="stylesheet" media="screen,projection"/>
   <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.5.0/css/all.css" integrity="sha384-B4dIYHKNBt8Bc12p+WXckhzcICo0wtJAoU8YZTY5qE0Id1GSseTk6S+L3BlXeVIU" crossorigin="anonymous">
</head>

{% block body %}
<body>
  <br>
  <h3>Enter the text that you want to summarize.</h3>
  <div class="container">
    <div class="section">

      <!--   Icon Section   -->
      <div class="row">
        <div class="input-field col s12 m10">
          <div class="icon-block">
            <br>
           <form method="POST" action="/summa">
         <textarea name="u" cols="3" rows="15" class="form-control" required="true" placeholder="Enter Text Here" ></textarea>

         <br>
         <button class="btn btn-success" type="reset">Clear</button>
          <button class="btn btn-warning" type="submit">Summarize</button>

        </form>
          </div>
        </div>

      </div>

    </div>
  </div>
```

Figure 3.7: HTML input code snippet for raw text summarization

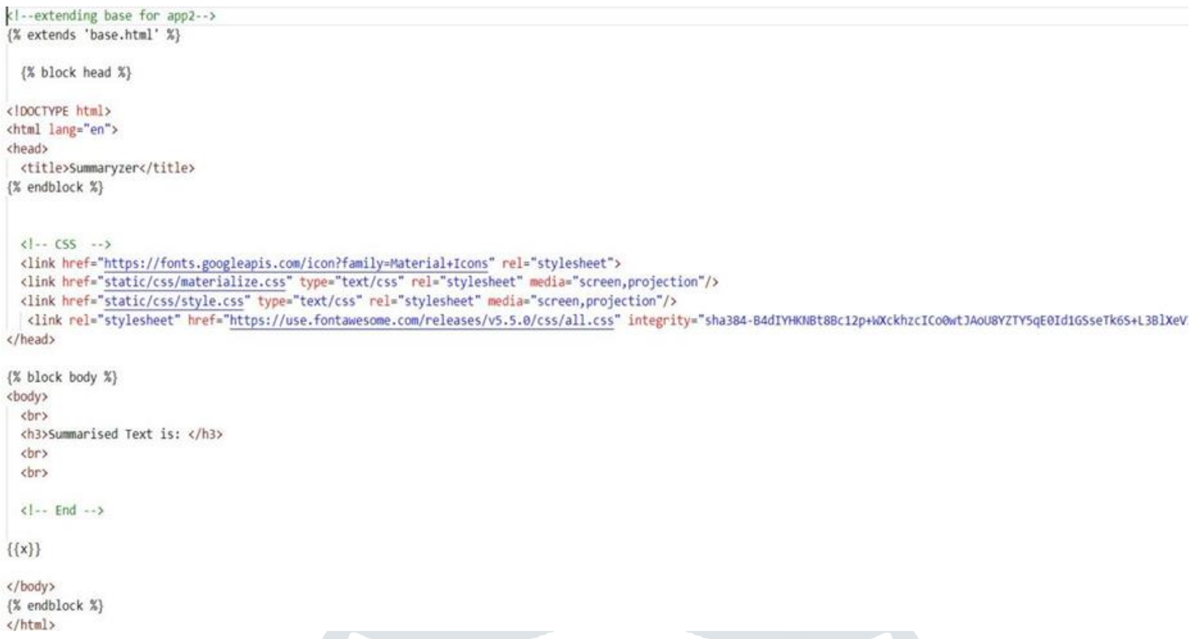
The general working methodology of the code is as explained below:

First, the text area is assigned for the user to be able to input the raw text that has to summarized.

Then, the form is created with the method of POST in order to be accessible to takeinput from the user.

Finally, the text given by the user is sent to the raw text summarization function wherethe summarization operation takes place.

The figure 3.8 contains the HTML code for displaying the summarized text produced from rawtext summarization:



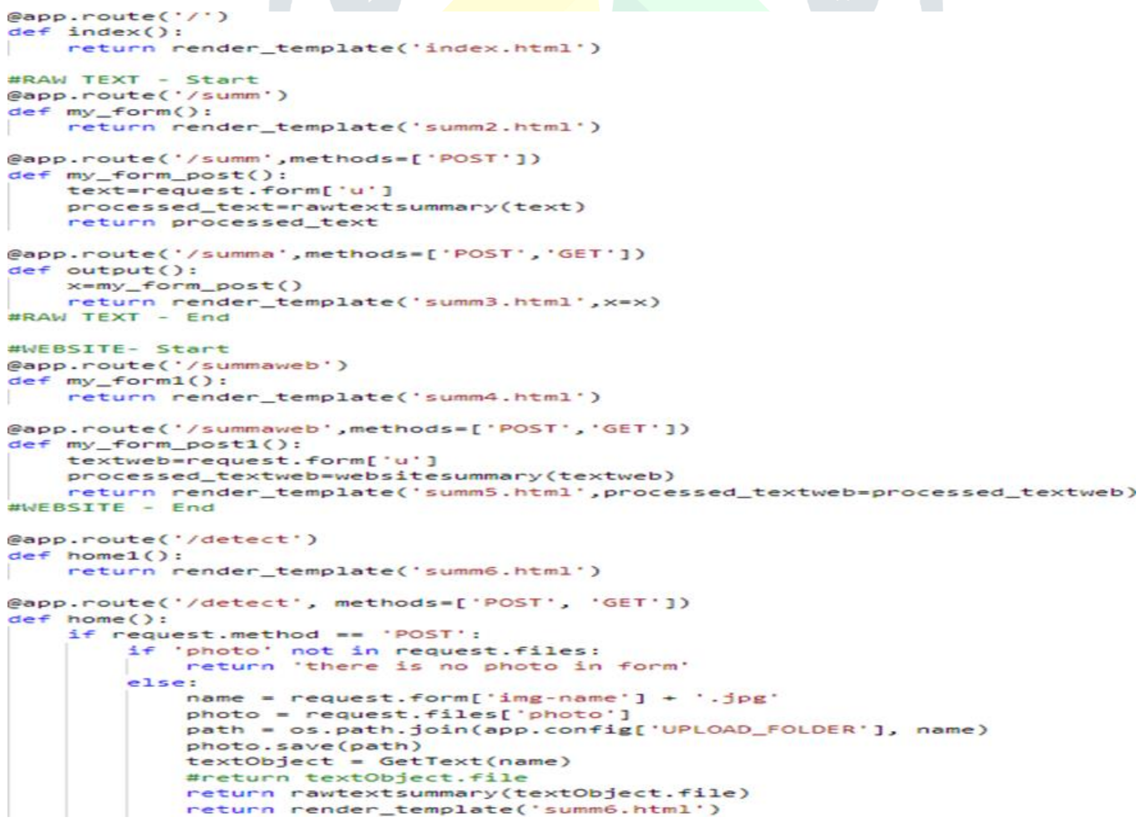
```
<!--extending base for app2-->
{% extends 'base.html' %}

  {% block head %}

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Summaryzer</title>
{% endblock %}


  <!-- CSS  -->
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
  <link href="static/css/materialize.css" type="text/css" rel="stylesheet" media="screen,projection"/>
  <link href="static/css/style.css" type="text/css" rel="stylesheet" media="screen,projection"/>
   <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.5.0/css/all.css" integrity="sha384-B4dIYHKNBt8Bc12p+WXckhzcICo0wtJAoU8YZTY5qE0Id1GSseTk6S+L3BlXeV
</head>

{% block body %}
<body>
  <br>
  <h3>Summarised Text is: </h3>
  <br>
  <br>

  <!-- End -->

{{x}}

</body>
{% endblock %}
</html>
```

The general working methodology of the code is as explained below:

- First, the text area is assigned where the output will be generated.
- Then, the html document is linked with method of GET in the flask web app folder inorder to return the output generated from the raw text summarization function.
- Finally, the text is summarized.

F.FLASK APPLICATION DEPLOYMENT

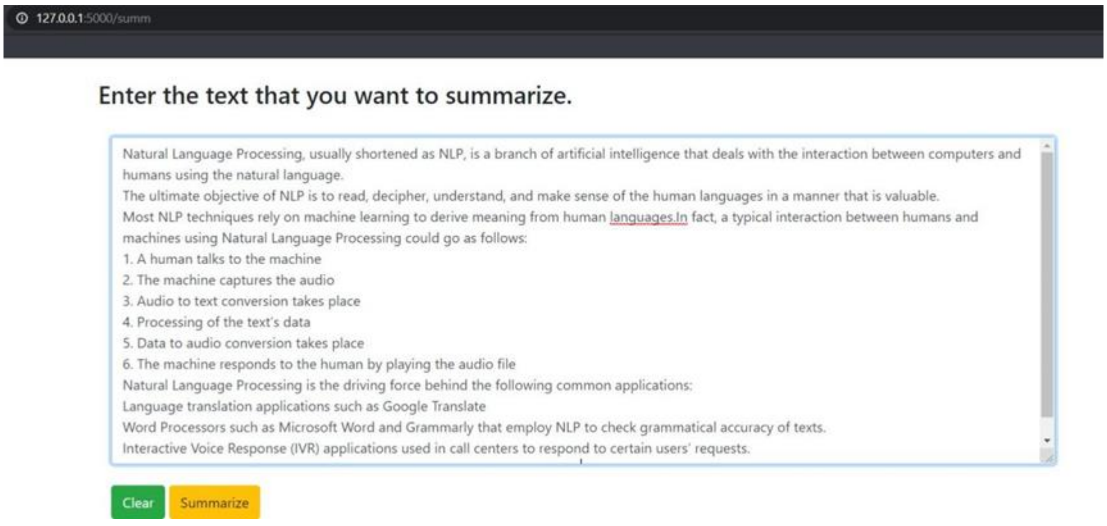
The functions that are responsible for the deployment of the Flask web applications are aspresented in figure 3.9



```
@app.route('/')
def index():
    return render_template('index.html')

#RAW TEXT - Start
@app.route('/summ')
def my_form():
    return render_template('summ2.html')

@app.route('/summ',methods=['POST'])
def my_form_post():
    text=request.form['u']
    processed_text=rawtextsummary(text)
    return processed_text

@app.route('/summa',methods=['POST','GET'])
def output():
    x=my_form_post()
    return render_template('summ3.html',x=x)
#RAW TEXT - End

#WEBSITE- Start
@app.route('/summaweb')
def my_form1():
    return render_template('summ4.html')

@app.route('/summaweb',methods=['POST','GET'])
def my_form_post1():
    textweb=request.form['u']
    processed_textweb=websitesummary(textweb)
    return render_template('summ5.html',processed_textweb=processed_textweb)
#WEBSITE - End

@app.route('/detect')
def home1():
    return render_template('summ6.html')

@app.route('/detect', methods=['POST', 'GET'])
def home():
    if request.method == 'POST':
        if 'photo' not in request.files:
            return 'there is no photo in form'
        else:
            name = request.form['img-name'] + '.jpg'
            photo = request.files['photo']
            path = os.path.join(app.config['UPLOAD_FOLDER'], name)
            photo.save(path)
            textObject = GetText(name)
            #return textObject.file
            return rawtextsummary(textObject.file)
            return render_template('summ6.html')
```

Figure 3.9: The code snippet of the flask application deployment

IV.RESULTS

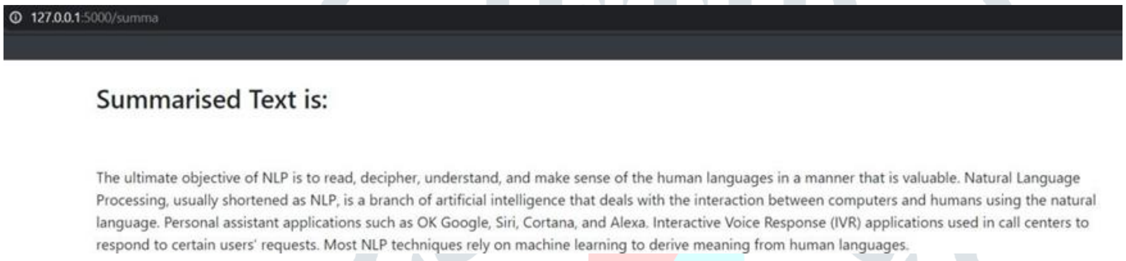


127.0.0.1:5000/summ

Enter the text that you want to summarize.

Natural Language Processing, usually shortened as NLP, is a branch of artificial intelligence that deals with the interaction between computers and humans using the natural language.
The ultimate objective of NLP is to read, decipher, understand, and make sense of the human languages in a manner that is valuable.
Most NLP techniques rely on machine learning to derive meaning from human languages.In fact, a typical interaction between humans and machines using Natural Language Processing could go as follows:
1. A human talks to the machine
2. The machine captures the audio
3. Audio to text conversion takes place
4. Processing of the text's data
5. Data to audio conversion takes place
6. The machine responds to the human by playing the audio file
Natural Language Processing is the driving force behind the following common applications:
Language translation applications such as Google Translate
Word Processors such as Microsoft Word and Grammarly that employ NLP to check grammatical accuracy of texts.
Interactive Voice Response (IVR) applications used in call centers to respond to certain users' requests.

Figure 4.1 : Summarizing an article using raw text summarizer



127.0.0.1:5000/summa

Summarised Text is:

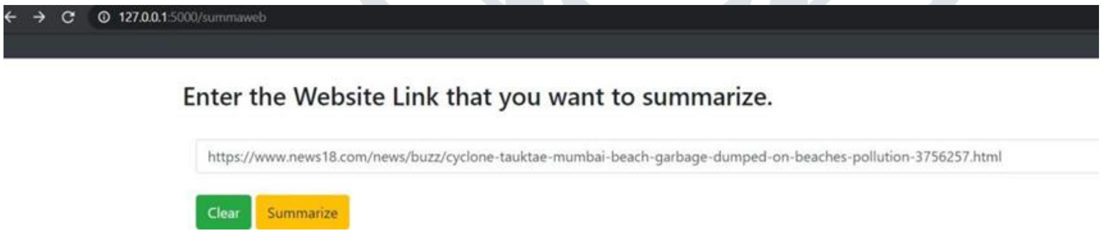
The ultimate objective of NLP is to read, decipher, understand, and make sense of the human languages in a manner that is valuable. Natural Language Processing, usually shortened as NLP, is a branch of artificial intelligence that deals with the interaction between computers and humans using the natural language. Personal assistant applications such as OK Google, Siri, Cortana, and Alexa. Interactive Voice Response (IVR) applications used in call centers to respond to certain users' requests. Most NLP techniques rely on machine learning to derive meaning from human languages.

Figure 4.2: The summarized result

G.WEBSITE SUMMARIZER

The website URL extracts text from any URL and gives it to the summarizer for summarizing the text. The user can extract the URL from any webpage of his choice and give it to the summarizerfor summarizing its content.

The figure 4.3 and 4.4 below, displays the testing results when a URL is to the website summarization feature.

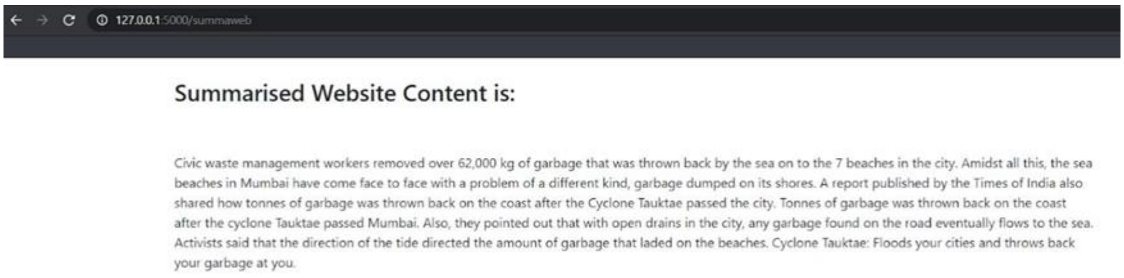


127.0.0.1:5000/summaweb

Enter the Website Link that you want to summarize.

https://www.news18.com/news/buzz/cyclone-tauktae-mumbai-beach-garbage-dumped-on-beaches-pollution-3756257.html

Figure 4.3: Giving input to the webpage summarizer



127.0.0.1:5000/summaweb

Summarised Website Content is:

Civic waste management workers removed over 62,000 kg of garbage that was thrown back by the sea on to the 7 beaches in the city. Amidst all this, the sea beaches in Mumbai have come face to face with a problem of a different kind, garbage dumped on its shores. A report published by the Times of India also shared how tonnes of garbage was thrown back on the coast after the Cyclone Tauktae passed the city. Tonnes of garbage was thrown back on the coast after the cyclone Tauktae passed Mumbai. Also, they pointed out that with open drains in the city, any garbage found on the road eventually flows to the sea. Activists said that the direction of the tide directed the amount of garbage that laded on the beaches. Cyclone Tauktae: Floods your cities and throws back your garbage at you.

Figure 4.4: The summarized output

V.CONCLUSION AND FUTURE SCOPE

Summarization can be defined as a task of producing a concise and fluent summary while preserving key information and overall meaning. 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. The data is unstructured and the best that we can do to navigate it is to use search and skim the results. There is a great need toreduce much of this text data to shorter, focused summaries that capture the salient details, both sowe can navigate it more effectively as well as check whether the larger documents contain the information that we are looking for. Extractive text summarization involves the selection of phrases and sentences from the source document to make up the new summary. Techniques involve ranking the relevance of phrases in order to choose only those most relevant to the meaningof the source. Coming to the deployed system, the developed web application was implemented by using the flask framework to deploy the NLP model. Text summarization was achieved using natural Language Processing using the NLTK libraries. The detection of text from an image was successfully performed using the PyTesseract API. All the modules implemented in the project have been trained and tested to give valid results and the same have been embodied in this report.

In the future, we aim to work towards improving the accuracy of the summarizer by deploying deep learning methods. We wish to deploy the approach of abstractive summarization. To generateplausible outputs, abstraction-based summarization approaches must address a wide variety of NLP problems, such as natural language generation, semantic representation, and inference permutation. We could also add in a feature related to text translation wherein summarization of various languages can be carried out within a single integrated web application. A feature related to voice detection can also be added to the web application where it can directly convert the summarized text to voice so the user can save time on even reading the text. Furthermore, we can work on making the application platform independent by making it compatible with mobile phones, tabs, etc.

VI.REFERENCES

[1] Narendra Andhale; L.A Bewoor: "An overview of text summarization techniques.", vol. 68, pp. 355–367, IEEE 2017

[2] Nikhil S. Shirwandkar; Samidha Kulkarni: "Extractive text summarization using deep learning.", vol. 78, pp. 255–350, IEEE 2019

[3] Pei-ying Zhang; Cun-he Li : "Automatic text summarization based on sentenceclustering and extraction.", vol. 23, no. 9, pp. 1188–1192, IEEE 2009

[4] Ravali Boorugu; G. Ramesh; "A survey on NLP based text summarization for summarizing product reviews.", vol. 7, no. 1, pp. 37–41, IEEE 2020.

[5] Tom Young, Devamanyu Hazarika, Erik Cambria: "Recent trends in Deep Learning Based Natural Language Processing", vol. 5, no. 2, pp. 57–66, IEEE 2018.

[6] V. Kunta, C. Tuniki and U. Sairam, "Multi-Functional Blind Stick for Visually Impaired People," 2020 5th International Conference on Communication and Electronics Systems (ICCES), 2020, pp. 895-899, doi:10.1109/ICCES48766.2020.9137870.

[7] Y. Adepu, V. R. Boga and S. U, "Interviewee Performance Analyzer Using Facial Emotion Recognition and Speech Fluency Recognition," 2020 IEEE International Conference for Innovation in Technology (INOCON), 2020, pp. 1-5, doi: 10.1109/INOCON50539.2020.9298427.

[8]. U. Sairam, D. K. Reddy Gowra and S. C. Kopparapu, "Virtual Mouse using Machine Learning and GUI Automation," 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), 2022, pp. 1112-1117 doi: 10.1109/ICACCS54159.2022.9784972. https://ieeexplore.ieee.org/document/9784972.

[9]."FAKE ONLINE REVIEWS DETECTION USING MACHINE LEARNING", International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN:2349-5162, Vol.9, Issue 6, page no.a247-a257, June 2022,Available :http://www.jetir.org/papers/JETIR2206028.pdf

[10]. **"ENHANCED RATING PREDICTION BASED ON LOCATION AND FRIEND SET"**, International Journal ofEmerging Technologies and Innovative Research (www.jetir.org), ISSN:2349-5162, Vol.6, Issue 5, page no.384-388, May-2019, Available :http://www.jetir.org/papers/JETIR1905D55.pdf

[11].Santhosh Voruganti. Survey on Data-intensive Applications, Tools and Techniques for Mining Unstructured Data. *International Journal of Computer Applications* 146(12):23-27, July 2016.

[12].U.Sairam,Santhosh Voruganti, **"HEART D,ISEASE PREDICTION USING DATA MINING"**, International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.10, Issue 5, pp.a886-a895, May 2022, Available at :http://www.ijcrt.org/papers/IJCRT2205103.pdf

[13].Santhosh Voruganti and Karnati Ramyakrishna, "Local Security Enhancement and Intrusion Prevention in Android Devices", *IRJET*, vol. 07, no. 01, Jan 2020.

[14].Santhosh Voruganti, "Map Reduce a Programming Model for Cloud Computing Based on Hadoop Ecosystem", *InternationalJournal of Computer Science and Information Technologies*, vol. 5, no. 3, pp. 3794-3799, 2014.

[15] Santhosh Voruganti, Karnati Ramyakrishna, Srilok Bodla, E. Umakanth, "Comparative Analysis of Dimensionality Reduction Techniques for Machine Learning", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 4 Issue 8, pp. 364-369, May-June 2018. Journal URL : https://ijsrst.com/IJSRST184871

[16]. Santhosh Voruganti and Karnati Ramyakrishna, "*Effective IoT Techniques to Monitorthe Levels of Garbage in Smart Dustbins", IRJET* ,vol. 07, no. 06, June 2020.

[17].Voruganti, Santhosh and U, Sairam and S, Meghana and M, Sravanthi, Visual Question Answering with External Knowledge(May 25, 2021). Proceedings of the International Conference on Smart Data Intelligence (ICSMDI 2021), Available at SSRN: https://ssrn.com/abstract=3853031 or http://dx.doi.org/10.2139/ssrn.3853031