#### NAME: RUCHA DHODAPKAR

**SMART INTERNZ AI-ML EXTERNSHIP May - June 2021**

PROJECT REPORT

#### PROJECT TITLE: Summarize an article using Adaptive Text Summarizing API with IBM Cloud

##### INTRODUCTION:

#### OVERVIEW:

Today, the amount of content available digitally, is massive. This also includes a large quantity of text based content that is increasing due to its accessibility. With so much textual content also comes the task of reading and analysing long and tiresome articles. This project aims to make this task simpler by helping summarize long, elaborate articles into a short paragraph that includes only the important parts.

#### PURPOSE:

The purpose of this web application is to help shorten very long articles or essays by giving the user a summary of the required article by picking out only the most relevant and important points from the text that also retain the overall idea of the article. Today work is all about speed and simplicity and this application aims to bridge this gap in reading and analysing text, by making use of a text summarizing API that works on advanced natural language processing and make it easier to deal with such large textual content.

##### LITERATURE SURVEY:

#### EXISTING PROBLEM:

The internet has an extensive store of text-based content. This text-based content is consumed very frequently and in large quantities. Long articles, essays, detailed documents, etc. are often time taking and tiresome to read. Most often we require only the key important bits of information that give us a brief, yet precise idea about the text without having to read through the whole document. Most often, the task of summarizing or shortening of long articles or documents is done manually and is still a tedious and mundane task. But, with the onset of technologies like machine learning, Natural Language Processing and such, it is possible to arrive at a solution to this problem.

#### PROPOSED SOLUTION:

The solution to the problem mentioned here is to use Natural Language Processing and Machine Learning to build a model that is capable of analysing long text documents and accurately summarize them by picking out the important points to give the user an accurate and precise summary of the article or document hence saving the time and effort that would have otherwise been spent in reading the long piece of text thoroughly and entirely.

##### THEORETICAL ANALYSIS:

#### BLOCK DIAGRAM:

#### 

#### SOFTWARE DESIGN:

The Text Summarizer Web Application is built on flask (Python Framework). The application consists of 3 web pages. The first is the home page that introduces the application and its purpose briefly. The button at the bottom navigates the user to the second web page where the input is to be given. The input can be direct text in the textbox or the url of the website that has the article to be summarized. Another input is the number of lines into which the user wants the article summarized. Once the input is given, the flask application fetches the text summarization API that summarizes the input into the given number of lines and returns it to the flask app. This output is then displayed to the user on the 'results' webpage.

##### EXPERIMENTAL INVESTIGATIONS:

The experimental investigations made for this project include choosing the right text summarization API by testing available text summarization APIs with various text and url inputs to check their reliability and accuracy; the real life and enterprise applications and future scope of text summarization; using flask and css documentations and tutorials to build a proper, functioning application and referring to tech community websites to help solve issues encountered while building the application. Research was done to gather datasets or data to be used as input to test the application. In order to gain knowledge and understanding on how to use git and github to push project files onto github, the git book and tutorials were referred. Research was also put into finding example code snippets for flask and css in order to build a suitable, easy to use and appealing web application.

##### FLOW CHART:RESULT:

**HOME PAGE**

**SUMMARIZER PAGE**

**TEXT INPUT OR URL**

**TEXT SUMMARIZER API**

**SUMMARIZED ARTICLE**

**RESULTS PAGE**

**(OUTPUT)**

The result of this project is the Adaptive Text Summarizer Web Application. The Application runs on a Flask Framework. The Home Page is the face of the application that introduces the user to our summarizer application. The "Let's Go" button at the bottom directs the user to the page where the application accepts inputs from the user. The inputs are a number, to tell the application how many lines the user wants the article summarized into; then either the article as direct input or the url of the page where the article to be summarized, is located. Once the input is received, the flask application fetches the API to the text summarizer that will then take the inputs and return the summarized article as required. This output is then presented to the user on the Results Page of the application. If the user wants to use the application again to summarize some more text, the "Summarize More Text" button directs the user back to the Summarizer Page to take in more input.

Thus the Adaptive Text Summarizer takes long articles as input from users and gives back a short summarized version as an output. The application is simple yet effective and useful, with a user interface that is easy to use and navigate through, and appealing to the eye.

##### ADVANTAGES AND DISADVANTAGES:

#### ADVANTAGES:

1. Takes input in the form of both text and the url to the text document.
2. User Interface is easy to use and navigate through.
3. Gives the user the option to specify how many lines they want the article to be summarized into.
4. Has no limit for the size or length of the article.
5. The summarized article is often reliable and accurate.
6. Saves the user the time and effort to read through the entire article by summarizing it for them.

#### DISADVANTAGES:

1. Summarizes articles that are typed in english only.
2. Cannot take in an image of the article as input. Needs only either text or the url.
3. Does not provide the option of uploading a text file as input to the user.
4. Only picks the important points from the article as they are. Does not alter or shorten the sentence in any way to summarize it.

##### APPLICATIONS:

1. To summarize news articles online
2. To analyse the large quantity of search results when analysing search queries for Search Engine Optimization.
3. To assist financial analysts in deriving market signals from content by summarizing financial reports.
4. To help analyse legal documents by summarizing them.
5. To help enhance the reach of online bok stores or libraries by providing users with short summaries of books.
6. Making it easier to just skim through long emails by summarizing them.
7. To create an abstract for any academic research paper.
8. To help create content for ppts by summarizing long text into short and important points.

##### CONCLUSION:

The applications of an automatic text summarizer are many. Today, when digital content that is being generated increases manifold by the day, it becomes a need to have a technology that is capable of shortening or summarizing that content for us to make our task of reading and analysing such an extensive store of content, easier and faster. Thus, the Adaptive Text Summarizer is capable of providing users with short and reliable summaries of long, tiring essays and articles at the click of a button. This indeed is an extremely simple yet powerful and effective applicaion with a large scope for application and further research and development, to help make good use of all the digital content today, where the volume of text is an inestimable source of information that needs to be summarized in order for it to be useful.

##### FUTURE SCOPE:

The future scope of automatic text summarization technology is limitless. The scope of text summarization ranges from text classification, to question answering to legal document summarization to headline generation. Albeit all the scope, it is also a challenging task to summarize the document because it requires thorough syntactical, semantical and contextual knowledge of a language and the capability of thoroughly understanding the piece of text before summarizing it. It is difficult for a computer to do this because unlike humans, computers do not possess language capabilities and the ability to understand the language. Nevertheless, with the explosion in the amount of textual data that is available digitally, it is a rising technology with several machine learning models being built so far to do the same. The world runs on data today. And with the endless collection of data we have from various sources today, it will almost become a necessity to develop a model that can give us all the important and relevant data while skipping or eliminating the rest so we can continue making good use of the data at hand without having to spend sleepless nights reading through all of it.

##### BIBLIOGRAPHY:

1. Text Summarization API:

<https://rapidapi.com/marketplace>

2. Bootstrap templates:

<https://getbootstrap.com/docs/4.0/examples/>

3. CSS tutorial and examples:

<https://www.w3schools.com/w3css/defaulT.asp>

4. Flask documentation:

<https://flask.palletsprojects.com/en/2.0.x/>

5. Solving flask and css related issues:

<https://stackoverflow.com/>

6. How to use git and github:

<https://git-scm.com/book/en/v2>

##### APPENDIX:

#### SOURCE CODE:

***1. temp.py:***

#### from flask import Flask, render\_template, request

#### import numpy as np

#### import re

#### import requests

#### import json

#### app = Flask(\_\_name\_\_)

#### app.config['DEBUG'] = True

#### @app.route('/')

#### def home():

#### return render\_template('home.html')

#### @app.route('/summarize',methods=['POST'])

#### def summarize():

#### return render\_template('summarize.html')

#### @app.route('/results', methods=['POST'])

#### def results():

#### lines = request.form["lines"]

#### text = request.form["text"]

#### url\_ip = request.form["url"]

#### if(url\_ip):

#### url = "https://textanalysis-text-summarization.p.rapidapi.com/text-summarizer-url"

#### payload = "url=" +url\_ip+"&sentnum="+lines

#### headers = {

#### 'content-type': "application/x-www-form-urlencoded",

#### 'x-rapidapi-key': "b03ee82bbcmsh3579e3fca7285e5p1ce1a4jsnc7bb6182f40c",

#### 'x-rapidapi-host': "textanalysis-text-summarization.p.rapidapi.com"

#### }

#### response = requests.request("POST", url, data=payload, headers=headers)

#### print(response)

#### op = json.loads(response.text)

#### op\_txt = op["sentences"]

#### output = ""

#### for i in op\_txt:

#### output = output + i

#### elif(text):

#### url = "https://textanalysis-text-summarization.p.rapidapi.com/text-summarizer-text"

#### payload = "text=" +text+ "&sentnum=" + lines

#### headers = {

#### 'content-type': "application/x-www-form-urlencoded",

#### 'x-rapidapi-key': "b03ee82bbcmsh3579e3fca7285e5p1ce1a4jsnc7bb6182f40c",

#### 'x-rapidapi-host': "textanalysis-text-summarization.p.rapidapi.com"

#### }

#### response = requests.request("POST", url, data=payload, headers=headers)

#### op = json.loads(response.text)

#### op\_txt = op["sentences"]

#### output = ""

#### for i in op\_txt:

#### output = output + i

#### return render\_template('results.html',output=output)

#### if \_\_name\_\_ == "\_\_main\_\_":

#### app.run()

#### 

2. home.html:

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>text summarizer</title>

<link rel="preconnect" href="https://fonts.googleapis.com">

<link href="https://fonts.googleapis.com/css2?family=Rubik:wght@600&family=Work+Sans:wght@300&display=swap" rel="stylesheet">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

<link rel="stylesheet" href="{{ url\_for('static', filename='css/styles.css') }}">

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>

</head>

<body style="background-color: #fff;">

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid-nav title">

<a class="navbar-brand font" href="#">ADAPTIVE TEXT SUMMARIZER</a>

</div>

<div class="collapse navbar-collapse" id="navbarText">

<span class="navbar-text side">

No more sleepless nights over long essays.

</span>

</div>

</nav>

<div class="container">

<div class="col-lg-6 image">

<image src="{{url\_for('static',filename = 'homeimg.jpg')}}" class="image" alt="Summarizer">

</div>

<div class="col-lg-6 text">

The <i>Adaptive Text Summarizer</i> is a powerful text summarizing tool built to help summarize large essays into shorter ones to give you only the important bits by freeing them of all unnecessary detail. All you have to do is to put in your essay here and leave the rest to us. <br>

<div class="go">

<form action="{{ url\_for('summarize')}}"method="post">

<button type="submit" class="btn btn-dark">LET'S GO</button>

</form>

</div>

</div>

</div>

</body>

</html>

3. summarize.html:

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>text summarizer</title>

<link rel="preconnect" href="https://fonts.googleapis.com">

<link href="https://fonts.googleapis.com/css2?family=Rubik:wght@600&family=Work+Sans:wght@300&display=swap" rel="stylesheet">

<link href="https://fonts.googleapis.com/css2?family=Ubuntu&display=swap" rel="stylesheet">

<link rel="stylesheet" href="{{ url\_for('static', filename='css/styles.css') }}">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous">

</script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous">

</script>

</head>

<body style = "background-color: #f8ceec; background-image: linear-gradient(315deg, #f8ceec 0%, #a88beb 74%);">

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid-nav">

<a class="navbar-brand font" href="#">ADAPTIVE TEXT SUMMARIZER</a>

</div>

<div class="collapse navbar-collapse" id="navbarText">

<span class="navbar-text side">

No more sleepless nights over long essays.

</span>

</div>

</nav>

<form action="{{ url\_for('results')}}"method="post">

<div class="container contact">

<div class="row">

<div class="col-md-3" style="background-color:#FF9AA2;">

<div class="contact-info">

<img src="{{url\_for('static',filename = 'pen.png')}}" alt="image"/>

<h2>Let's Get Started!</h2>

<h4>What text would you like us to summarize?</h4>

</div>

</div>

<div class="col-md-9">

<div class="contact-form">

<div class="form-group">

<label class="control-label container" for="fname">No. of lines to be summarized into:</label>

<div class="col-sm-10">

<input type="text" class="form-control" id="fname" name="lines" placeholder="Enter a Number" required="required">

</div>

</div>

<div class="form-group">

<label class="control-label container" for="fname">Enter URL for the article to be summarized:</label>

<div class="col-sm-10">

<input type="text" class="form-control" id="fname" name="url" placeholder="Enter URL">

</div>

</div>

<div class="container or">

<h3><span class="label" align="center">OR</span></h3>

</div>

<div class="form-group">

<label class="control-label container" for="comment">Enter text to be summarized:</label>

<div class="col-sm-10">

<textarea class="form-control" rows="8" id="comment" name="text" placeholder="Your text here"></textarea>

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10">

<button type="submit" class="btn btn-default"> Summarize</button>

</div>

</div>

</div>

</div>

</div>

</div>

</form>

</body>

</html>

4. results.html:

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>text summarizer</title>

<link rel="preconnect" href="https://fonts.googleapis.com">

<link href="https://fonts.googleapis.com/css2?family=Rubik:wght@600&family=Work+Sans:wght@300&display=swap" rel="stylesheet">

<link href="https://fonts.googleapis.com/css2?family=Nanum+Gothic&display=swap" rel="stylesheet">

<link rel="stylesheet" href="{{ url\_for('static', filename='css/styles.css') }}">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous">

</script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous">

</script>

</head>

<body style = "background-color: #f8ceec; background-image: linear-gradient(315deg, #f8ceec 0%, #a88beb 74%);">

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid-nav">

<a class="navbar-brand font" href="#">ADAPTIVE TEXT SUMMARIZER</a>

</div>

<div class="collapse navbar-collapse" id="navbarText">

<span class="navbar-text side">

No more sleepless nights over long essays.

</span>

</div>

</nav>

<div class="container contact">

<div class="row">

<div class="col-md-3">

<div class="contact-info">

<img src="{{url\_for('static',filename = 'like.png')}}" alt="image"/>

<h2>Here You Go!</h2>

<h4>All Done!</h4>

</div>

</div>

<div class="col-md-9">

<div class="contact-form">

<div class="form-group">

<label class="control-label container" for="comment"></label>

<div class="col-sm-10">

<p>{{ output }}</p>

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10">

<form action="{{ url\_for('summarize')}}"method="post" class="op">

<button type="submit" class="btn btn-default">Summarize More Text?</button>

</form>

</div>

</div>

</div>

</div>

</div>

</div>

</body>

</html>

5. styles.css:

.font {

font-family: 'Rubik', sans-serif;

}

.side {

font-family: 'Work Sans', sans-serif;

font-size: 15px;

}

.image {

height: 600px;

width: 600px;

float:left;

text-align: left;

}

.text {

font-size: 20px;

padding-left: 100px;

padding-bottom: 30px;

float: right;

text-align: left;

margin-top: 150px;

}

.go {

padding-top: 30px;

}

.contact{

height: 70%;

width: 70%;

padding: 3% 15%;

}

.col-md-3{

background: #FF9AA2;

padding: 4%;

border-top-left-radius: 0.5rem;

border-bottom-left-radius: 0.5rem;

}

.contact-info{

margin-top:10%;

}

.contact-info img{

margin-bottom: 15%;

}

.contact-info h2{

margin-bottom: 10%;

}

.col-md-9{

background: #fff;

padding: 3%;

border-top-right-radius: 0.5rem;

border-bottom-right-radius: 0.5rem;

}

.contact-form label{

font-weight:600;

}

.contact-form button{

background: #25274d;

color: #fff;

font-weight: 600;

width: 25%;

}

.contact-form button:focus{

box-shadow:none;

}

.op {

font-family: 'Nanum Gothic', sans-serif;

}

.label{

background: #FF9AA2;

border: 2px solid black;

border-radius: 7px;

}

.or {

align-items: center;

text-align: center;

font-family: 'Ubuntu', sans-serif;

}

#### UI OUTPUT SCREENSHOTS:









