SUMMARIZE AN ARTICLE USING ADAPTIVE TEXT SUMMARIZATION API

Introduction

The Adaptive Text Summarization API allows you to quickly summarize a piece of text, an article, or a webpage (by URL) by extracting the most important sentences that best capture the meaning of the entire text. You can specify the number of sentences up to which the given content must be summarized. This API can be used API to generate the following types of summaries as per their use-cases:**General-purpose Summary**Given an input text or URL, the API provides an extractive-summary of the text returning Top-N sentences that best represent the entire text or URL.**News Summary**News content is often different from general purpose content via an algorithm to focused on news style content when you use this parameter.**Diversified Summary**Sometimes a text says more than one type of thing. The diversified-summary picks out the all distinct sentences as a summary. The API can take any type of text as input and will perform best on URLs corresponding to news, blog, content, etc.

Overview

Summarize an Article using Adaptive Text Summarization API with IBM Cloud.This webpage helps to summarize an article for the user in a short time.It saves the time of the user by summarize and telling the meaning of big or small article.

Software Designing

1. Jupyter Notebook Environment
2. Spyder
3. Text Analysis-Text Summarization API
4. Python
5. HTML
6. Flask

We developed this summarization web application by using the Python language, which is a high level programming language along with text analysis-text summarization API. For coding we used the Jupyter Notebook of Anaconda distributions and Spyder, an integrated scientific programming in python language. Flask is used as a user interface for the summarization. Hypertext Markup Language (*HTML*) is the standard markup language for documents designed to be displayed in a web browser.

Text Analysis-Text Summarization API

Text Summarization API provides professional text summarizer service which is based on advanced Natural Language Processing and Machine Learning technologies. It can be used to summarize short important text from the URL or document that user provided

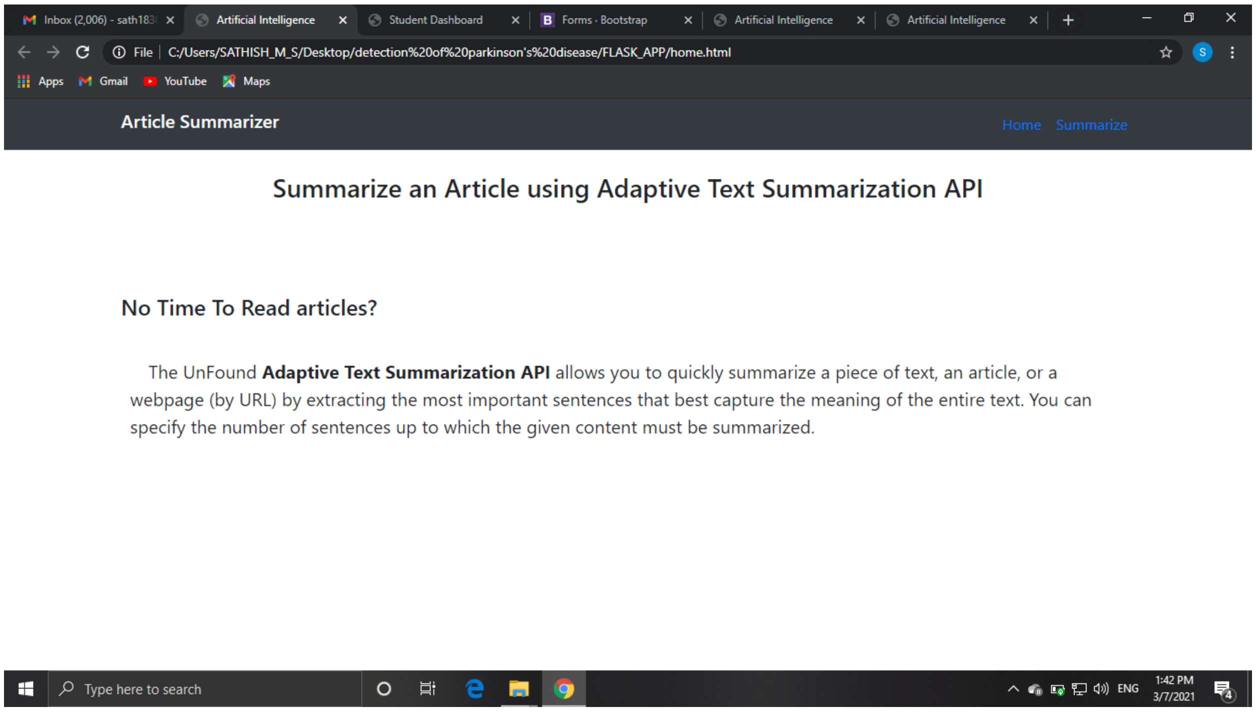
API CODE

```
import requests

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

 payload = "{
   \"url\": \"http://en.wikipedia.org/wiki/Automatic_summarization\",
   \"text\": \"\",
   \"sentnum\": 8
 }"
 headers = {
     'content-type': "application/json",
     'x-rapidapi-key':
"537649ff73msh477f6855911bb13p129cf4jsnd5cb001617b2",
     'x-rapidapi-host': "textanalysis-text-summarization.p.rapidapi.com"
     }

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

 print(response.text)
```

Result

This web application successfully summarize an article using text analysis-text summarization API in quick and effective manner.

Screenshots

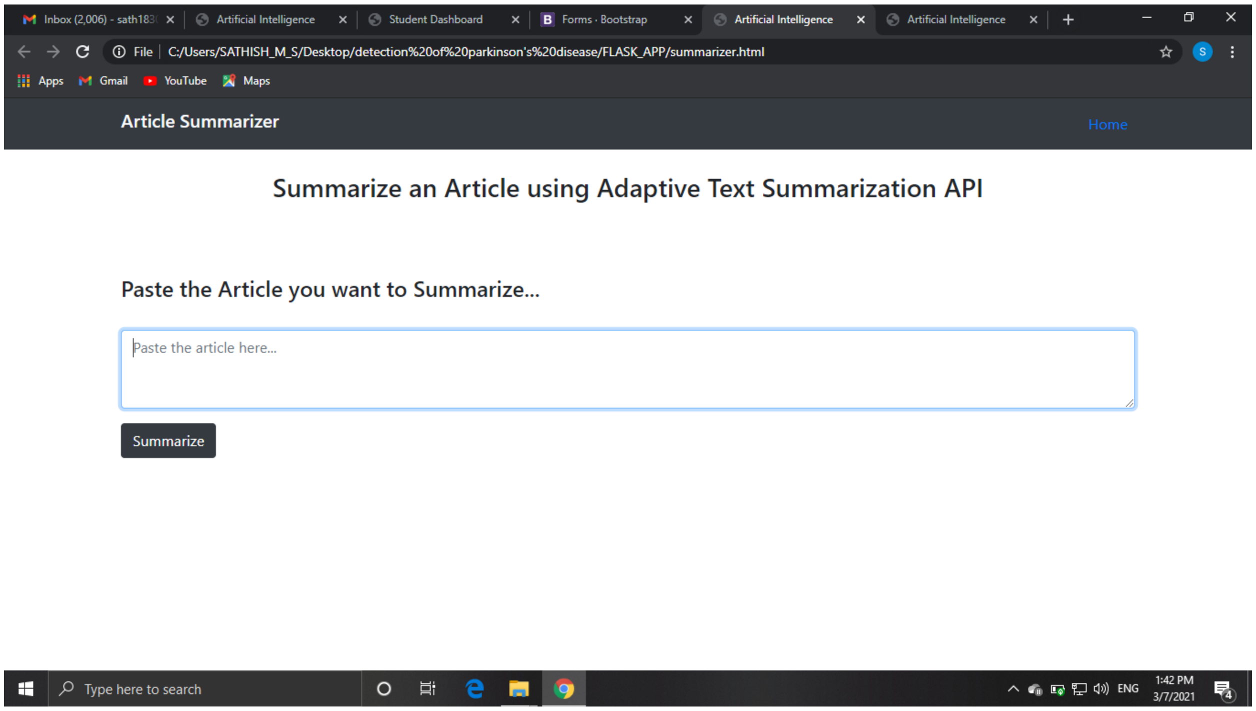


Article Summarizer

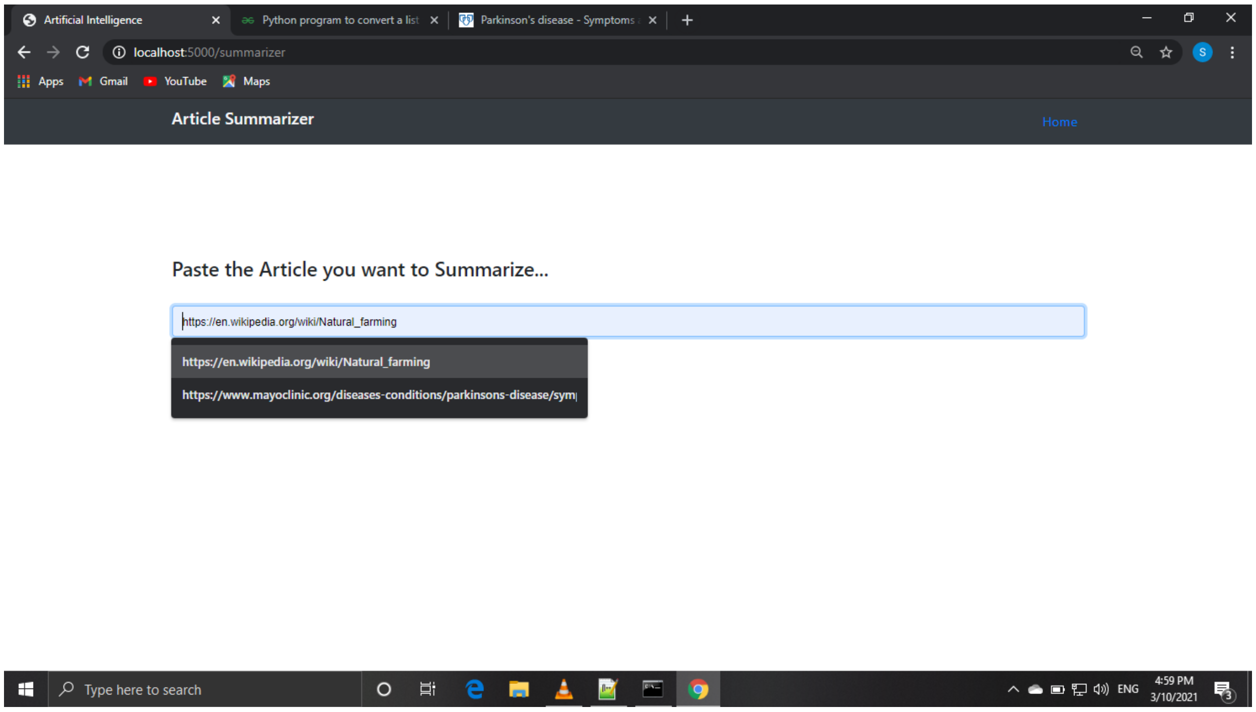
Summarize an Article using Adaptive Text Summarization API

No Time To Read articles?

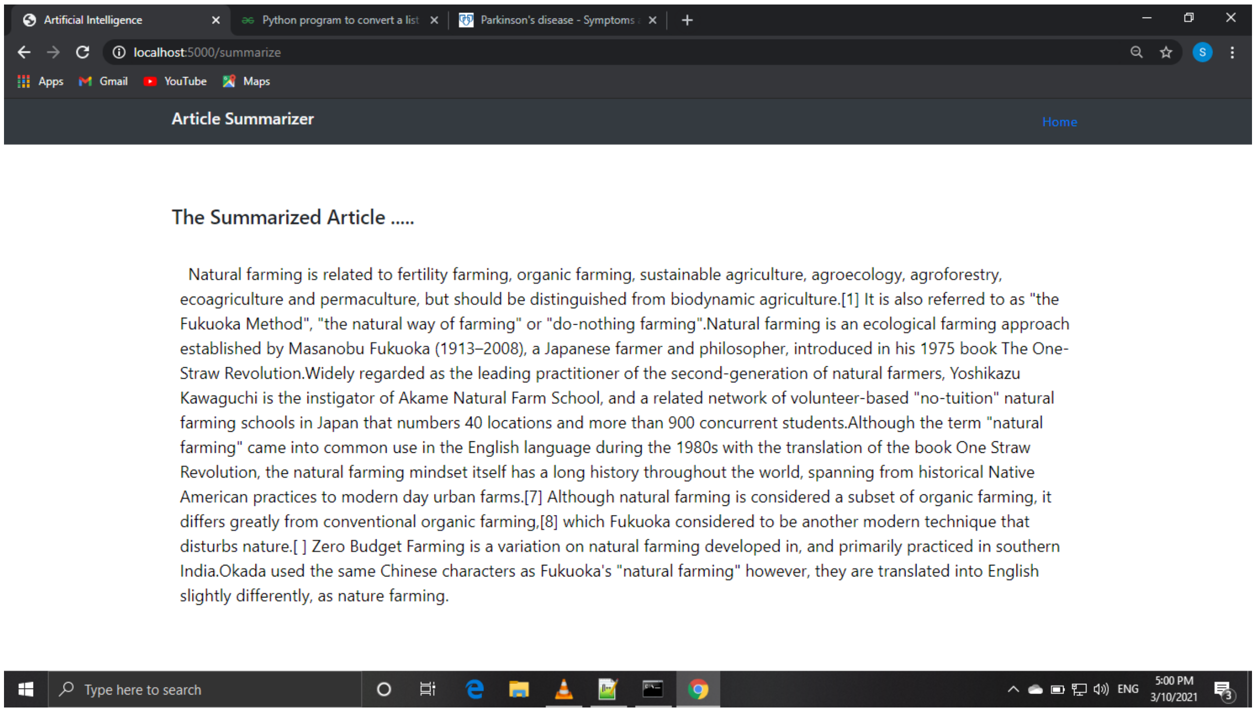
The UnFound **Adaptive Text Summarization API** allows you to quickly summarize a piece of text, an article, or a webpage (by URL) by extracting the most important sentences that best capture the meaning of the entire text. You can specify the number of sentences up to which the given content must be summarized.



Inbox (2,006) - sath183
Artificial Intelligence
Student Dashboard
Forms · Bootstrap
Artificial Intelligence
Artificial Intelligence
File | C:/Users/SATHISH_M_S/Desktop/detection%20of%20parkinson's%20disease/FLASK_APP/summarizer.html
Apps
Gmail
YouTube
Maps
Article Summarizer
Home
Summarize an Article using Adaptive Text Summarization API
Paste the Article you want to Summarize...
Paste the article here...
Summarize
Type here to search
ENG
1:42 PM
3/7/2021



Artificial Intelligence
Python program to convert a list
Parkinson's disease - Symptoms
localhost:5000/summarizer
Apps
Gmail
YouTube
Maps
Article Summarizer
Home
Paste the Article you want to Summarize...
https://en.wikipedia.org/wiki/Natural_farming
https://en.wikipedia.org/wiki/Natural_farming
https://www.mayoclinic.org/diseases-conditions/parkinsons-disease/sym
Type here to search
ENG
4:59 PM
3/10/2021



The Summarized Article .....

Natural farming is related to fertility farming, organic farming, sustainable agriculture, agroecology, agroforestry, ecoagriculture and permaculture, but should be distinguished from biodynamic agriculture.[1] It is also referred to as "the Fukuoka Method", "the natural way of farming" or "do-nothing farming".Natural farming is an ecological farming approach established by Masanobu Fukuoka (1913–2008), a Japanese farmer and philosopher, introduced in his 1975 book The One-Straw Revolution.Widely regarded as the leading practitioner of the second-generation of natural farmers, Yoshikazu Kawaguchi is the instigator of Akame Natural Farm School, and a related network of volunteer-based "no-tuition" natural farming schools in Japan that numbers 40 locations and more than 900 concurrent students.Although the term "natural farming" came into common use in the English language during the 1980s with the translation of the book One Straw Revolution, the natural farming mindset itself has a long history throughout the world, spanning from historical Native American practices to modern day urban farms.[7] Although natural farming is considered a subset of organic farming, it differs greatly from conventional organic farming,[8] which Fukuoka considered to be another modern technique that disturbs nature.[ ] Zero Budget Farming is a variation on natural farming developed in, and primarily practiced in southern India.Okada used the same Chinese characters as Fukuoka's "natural farming" however, they are translated into English slightly differently, as nature farming.

CODE SNIPPET

HTML CODE

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Artificial Intelligence</title>
<link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
   <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
   <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
   <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
                    <style>
                             body{
                                       margin:0;padding:0;
```

```
                        }
                        h3{
                                text-align:center;
                                color:white;
                        }
                        .main{
                                margin-top:80px;
                        }
                        p{
                                color:white;
                                text-indent:10px;
                                margin:10px;
                                font-size:20px;
                        }

                </style>
</head>
<body>
                <div class="bg">
                        <nav class="navbar navbar-expand-sm bg-dark rounded-pill" id="navv">
                          <div class="container">
                          <div class="navbar-header">
                                <h5 class="text-light">Article Summarizer</h5>
                          </div>
                          <ul class="navbar-nav">
                        <li class="nav-item ">
                          <a class="nav-link" href="{{url_for('home')}}">Home</a>
                        </li>
                        </ul>
                          </div>
                        </nav>
                        <br>
                        <center><h3>Summarize an Article using Adaptive Text Summarization
API</h3></center>
                        <div class="container main">
                        <h4>Paste the Article you want to Summarize...</h4>
                        <br>
                        <form action="{{url_for('summarize')}}" method="post">
                                <div class="form-group">
                                        <input type="text" name="article" class="form-control"
placeholder="Paste the article here..." id="exampleFormControlTextarea1" rows="3">
```

```
                                        </div>
                                        <button type="submit" class="btn btn-dark "
id="btn-summarize">Summarize</button>
                              </form>
                              </div>
                     </div>
</body>
</html>
```

Flask APP

```
from flask import Flask, request, render_template
import numpy as np
import re
import requests

app = Flask(_name_)

def check(typ,output):
   url =
"https://textanalysis-text-summarization.p.rapidapi.com/text-summarizer"
   #payload = "{\"url\": \""+output+"\",\"text\": \"\",\"sentnum\": 8}"
   payload='''{"url":"'''+output+'''","text":"","sentnum":8}'''
   print(payload)
   headers = {
   'content-type': "application/json",
   'x-rapidapi-key':
"537649ff73msh477f6855911bb13p129cf4jsnd5cb001617b2",
   'x-rapidapi-host': "textanalysis-text-summarization.p.rapidapi.com"
   }
   response = requests.request("POST", url, data=payload,
headers=headers)
   return response.json()["sentences"]
#home page
```

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

#summarizer page
@app.route('/summarizer')
def summarizer():
   return render_template('summarizer.html')
#Results screen
@app.route('/summarize',  methods=['POST'])
def summarize():
   typ="text"
   output = request.form.get("article")
   print(output)
   essay = ''.join(check(typ,output))
   return render_template('summary.html',essay=essay)
if _name_ == "_main_":
   app.run()
```