DATA SUMMARIZER using NLP

Anukanksha Aashi - RA2011003010824
Jigyasa Sharma - RA20110033010832
Chinmoyee Gogoi - RA2011003010884

Problem Statement:

To extract the most important and relevant information from a large dataset and presenting it in a concise and coherent manner.

With the increasing availability of big data, organizations and individuals face the challenge of dealing with vast amounts of information, and summarization techniques are critical for extracting insights and making informed decisions.

Objective:

The goal of this problem statement is to develop a data summarization solution that can efficiently and accurately summarize large datasets while preserving the relevant information and maintaining coherence, conciseness, and accuracy.

The solution should be flexible, scalable, and adaptable to different data types, domains, and contexts, and should provide meaningful summaries that can support decision-making, information retrieval, and knowledge discovery tasks.

Technical Depth:

The user inputs text in a textarea element, and when the "Submit" button is clicked, the summarize function is called. This function uses AJAX to call a Python script that performs the summarization and returns the output. The summary is then displayed on the HTML page.

HTML CODE

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width,
initial-scale=1.0" />
<title>Data Summarization</title>
<link

href="https://unpkg.com/tailwindcss@^2/dist/tailwind.min.css"
rel="stylesheet"
/>
<link href="/static/css/jquery-ui.min.css" rel="stylesheet" />
</head>

<body>
<header
style="background-color: #004a56"
class="fixed inset-0 w-full flex flex-col justify-center
justify-items-center content-center h-20 rounded-b-lg"
>
<img
src="/static/SynopserLogo.png"
alt="Logo"
class="object-contain w-16 self-center md:self-left md:w-30"
/>
<div class="self-center text-white">Data
Summarization</div>
</header>
```

```
<section class="flex flex-wrap mt-20 w-full">
<div class="w-full md:w-1/2">
<textarea
id="dataBox"
class="w-11/12 md:h-3/4 m-2 p-2 border-black
rounded-lg border self-center justify-center"
name="data"
id="data"
cols="30"
rows="10"
placeholder="Enter your Data"
required="required"
></textarea>
<div class="flex self-center">
<h3>Summary Length</h3>
<input type="range" class="m-2" min="20"
max="1000" name="maxL" />
</div>
<div class="flex self-center">
<button
class="m-1 bg-green-500 hover:bg-blue-700
text-white font-semi-bold py-2 px-4 rounded-lg"
onclick="Check()"
>
Submit
</button>
```

```
<button
class="m-1 bg-red-500 hover:bg-blue-700 text-white
font-semi-bold py-2 px-4 rounded-lg"
type="reset"
>
Clear
</button>
</div>
</div>
<div class="w-full md:w-1/2">
<div class="mt-2 flex flex-col w-full md:h-screen">
<textarea
id='resultBox'
style="background-color: #edffd2"
class="w-11/12 border-green-600 rounded-lg p-2 border
self-center justify-center md:h-3/4"
rows="10"
cols="30"
name="result"
readonly
placeholder="Your Summary"
>
</textarea
>
<button
class="m-2 bg-blue-400 hover:bg-blue-700 text-white
font-semi-bold py-2 px-4 rounded-lg self-center"
onclick="myFunction()"
>
```

```
Copy text
     </button>
    </div>
   </div>
  </section>

  <script>
   function Check(e) {
    console.log("Check");
    data =
document.getElementById("dataBox").value;
    console.log(data);
    const xhttp = new XMLHttpRequest();
    xhttp.onload = function () {

document.getElementById("resultBox").innerHTML =
this.responseText;
    };
    xhttp.open("POST",
"https://api-inference.huggingface.co/models/faceboo
k/bart-large-cnn");
    xhttp.setRequestHeader("Authorization", "Bearer
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");
    xhttp.send(data)
   }
  </script>
 </body>
</html>
```

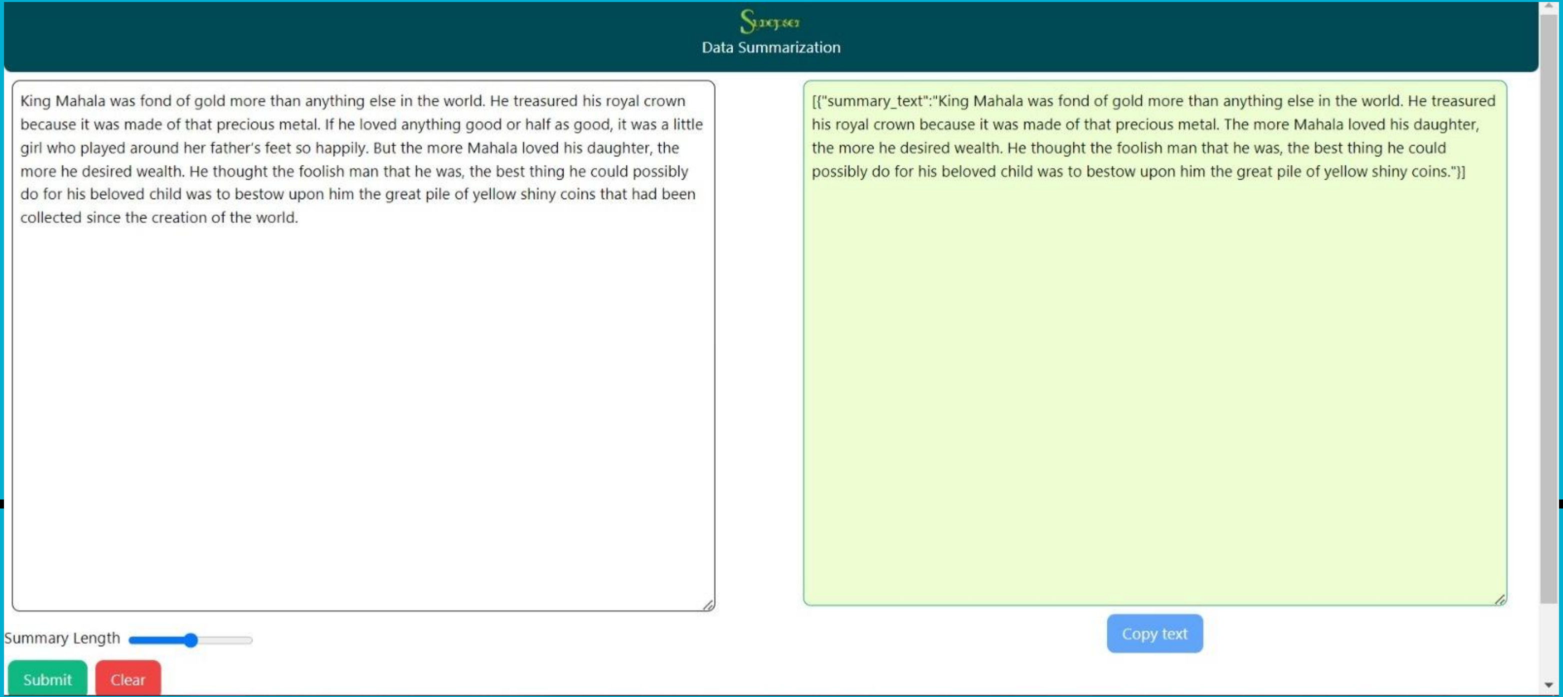
HTML

C O D E

PYTHON CODE

```
import requests
from flask import Flask,render_template,url_for
from flask import request as req


app = Flask(_name_)
@app.route("/",methods=["GET","POST"])
def Index():
return render_template("index.html")

@app.route("/Summarize",methods=["GET","POST"])
def Summarize():
if req.method == "POST":
API_URL = "https://api-inference.huggingface.co/models/facebook/bart-large-cnn"
headers = {"Authorization": "Bearer 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"}

data=req.form["data"]

#maxL=int(req.form["maxL"])
#minL=maxL//4

def query(payload):
response = requests.post(API_URL, headers=headers, json=payload)
return response.json()

output = query({
"inputs":data,
"parameters":{"min_length":minL,"max_length":maxL},
})[0]

return render_template("index.html",result=output["summary_text"])
else:
return render_template("index.html")
if _name=="main_":
app.run(debug=True)
```

Output:

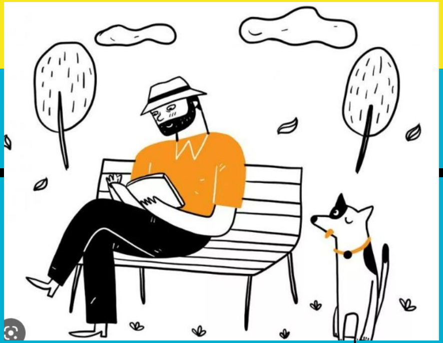


King Mahala was fond of gold more than anything else in the world. He treasured his royal crown because it was made of that precious metal. If he loved anything good or half as good, it was a little girl who played around her father's feet so happily. But the more Mahala loved his daughter, the more he desired wealth. He thought the foolish man that he was, the best thing he could possibly do for his beloved child was to bestow upon him the great pile of yellow shiny coins that had been collected since the creation of the world.

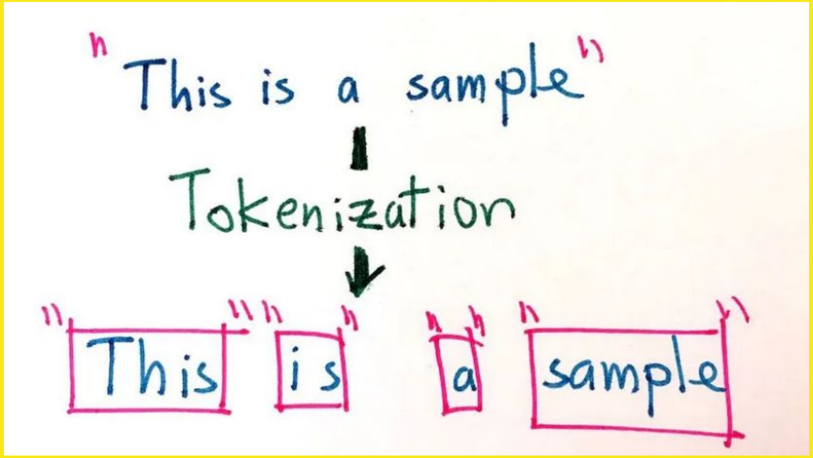
```
[{"summary_text":"King Mahala was fond of gold more than anything else in the world. He treasured his royal crown because it was made of that precious metal. The more Mahala loved his daughter, the more he desired wealth. He thought the foolish man that he was, the best thing he could possibly do for his beloved child was to bestow upon him the great pile of yellow shiny coins."}]
```

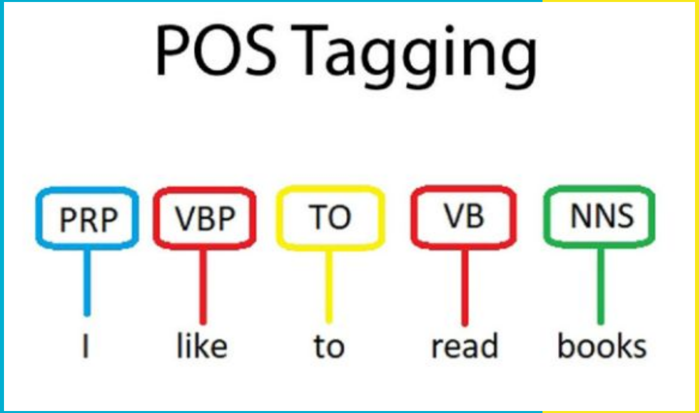
Data summarizers are tools or programs that can extract and summarize important important information from large sets of data. When combined with HTML and NLP, data summarizers can help to create web pages that display summarized data in a structure and organized way, while also using NLP to analyze and understand the text within the data.

NLP can be used in Python to summarize data by identifying and extracting the most important information from a large amount of text data. This is achieved by applying several NLP techniques, such as tokenization, part-of-speech tagging, and text summarization algorithms.



Tokenization involves breaking down a text document into individual words or phrases, known as tokens.





Part-of-speech (POS) tagging involves assigning a grammatical category, such as noun, verb, or adjective, to each token in a text document.

SUMMARY

NLP techniques can be combined with python libraries to create powerful data summarization tools that can help users to quickly and efficiently extract the most important information from large amount of text data.