Project Report: PubMed Article Summarizer Web Application

# Objective

The objective of this project is to create a web application that summarizes PubMed articles. This project involves data exploration, preparation, and summarization of PubMed articles using various summarization techniques. The application is developed using Flask and employs the facebook/bart-large-cnn model for summarization.

# 1. Data Exploration and Preparation

## 1.1 Dataset Loading

The PubMed Summarization dataset was loaded using the datasets library from Hugging Face.

```python  
from datasets import load\_dataset  
  
ds = load\_dataset("ccdv/pubmed-summarization", "document")  
```

## 1.2 Text Preprocessing

A preprocessing function was implemented to clean the text by removing newline characters.

```python  
def preprocess(text):  
 text = ''.join(text.split("  
"))  
 return text  
```

# 2. Summarization

## 2.1 Summarization Pipeline

The facebook/bart-large-cnn model from the Hugging Face transformers library was used to create a summarization pipeline.

```python  
from transformers import pipeline  
  
summarizer = pipeline("summarization", model="facebook/bart-large-cnn")  
```

## 2.2 Extractive Summarization

An extractive summarization function was implemented. This function splits the text into chunks, summarizes each chunk, and then combines the summarized chunks into a final summary.

```python  
def extractive\_summarize(text, max\_length=150, min\_length=40):  
 max\_chunk = 1024  
 chunks = [text[i:i+max\_chunk] for i in range(0, len(text), max\_chunk)]  
 summary = ""  
 for chunk in chunks:  
 summarized\_chunk = summarizer(chunk, max\_length=max\_length, min\_length=min\_length, do\_sample=False)  
 summary += summarized\_chunk[0]['summary\_text'] + " "  
 return summary.strip()  
```

## 2.3 Example Summarization

The summarization function was tested on the first article in the dataset.

```python  
text = ''.join(ds["train"]["article"][0].split("  
"))  
summary = extractive\_summarize(text)  
```

# 3. Web Application Development

## 3.1 Flask Application Setup

A Flask web application was developed to allow users to input or upload PubMed articles and view the summarized versions.

```python  
from flask import Flask, request, render\_template\_string  
  
app = Flask(\_\_name\_\_)  
  
@app.route('/', methods=['GET', 'POST'])  
def home():  
 if request.method == 'POST':  
 article = request.form['article']  
 summarized\_article = extractive\_summarize(article)  
 return render\_template\_string('''  
 <h2>Original Article</h2>  
 <p>{{ original }}</p>  
 <h2>Summarized Article</h2>  
 <p>{{ summary }}</p>  
 <a href="/">Go Back</a>  
 ''', original=article, summary=summarized\_article)  
 return '''  
 <form method="post">  
 <textarea name="article" rows="10" cols="30"></textarea>  
 <br>  
 <input type="submit" value="Summarize">  
 </form>  
 '''  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)  
```

# Conclusion

The project successfully demonstrates the ability to load, preprocess, and summarize PubMed articles using the facebook/bart-large-cnn model. The web application built using Flask allows users to input articles and receive summaries, showcasing the practical application of the summarization technique.

- Integrating advanced features like interactive visualization, video generation, and image generation.

# Repository

The code and detailed instructions for running the application are available in the GitHub repository.  
GIT: