

Text\_Summarization\_Project\_Report

Haripriya MC

Contents

[Aim: 3](#_Toc178513341)

[Introduction: 3](#_Toc178513342)

[About Hugging Face: 3](#_Toc178513343)

[Modules Used: 3](#_Toc178513344)

[Pegasus: 3](#_Toc178513345)

[Analysis: 3](#_Toc178513346)

[Code of app.py: 3](#_Toc178513347)

[Code of index.html: 4](#_Toc178513348)

[Code of output.html: 5](#_Toc178513349)

[Result: 6](#_Toc178513350)

[Conclusion: 6](#_Toc178513351)

**Text\_Summarization\_Project\_Report**

# Aim:

The aim of this project is to develop a web application for text summarization using the Pegasus model from Hugging Face Transformers. The application allows users to input text and receive a concise summary as output.

# Introduction:

In today's information-driven world, the volume of textual data is overwhelming. Text summarization is a technique used to extract the most important information from a large body of text, making it easier for users to grasp key points quickly. This project utilizes the Pegasus model, a state-of-the-art transformer model designed for abstractive text summarization, to achieve this goal.

## About Hugging Face:

Hugging Face is an AI company known for its contributions to natural language processing (NLP). They have created the Transformers library, which provides pre-trained models for various NLP tasks, including text summarization, translation, and sentiment analysis.

## Modules Used:

1. Flask: A micro web framework for Python, used to create web applications.  
2. Transformers: A library by Hugging Face that provides pre-trained models and tokenizers for NLP tasks.  
3. Torch: A deep learning framework that supports tensor computations and GPU acceleration.

## Pegasus:

Pegasus is a transformer-based model designed specifically for text summarization tasks. It employs a novel training objective that mimics human summarization, enabling it to generate high-quality, coherent summaries. The model can process long documents and condense them into shorter versions while retaining the essential meaning.

# Analysis:

## Code of app.py:

```python  
from flask import Flask, render\_template, request  
  
from transformers import PegasusForConditionalGeneration, PegasusTokenizer  
import torch  
  
app = Flask(\_\_name\_\_)  
  
model\_name = 'google/pegasus-xsum'  
tokenizer = PegasusTokenizer.from\_pretrained(model\_name)  
  
device = 'cuda' if torch.cuda.is\_available() else 'cpu'  
model = PegasusForConditionalGeneration.from\_pretrained(model\_name).to(device)  
  
@app.route('/')  
def home():  
 return render\_template('index.html')  
  
@app.route('/text-summarization', methods=['POST'])  
def summarize():  
  
 if request.method == 'POST':  
  
 inputtext = request.form['inputtext\_']  
  
 input\_text = 'summarize: ' + inputtext  
  
 tokenized\_text = tokenizer.encode(input\_text, return\_tensors='pt', max\_length=512).to(device)  
 summary\_ = model.generate(tokenized\_text, min\_length=30, max\_length=300)  
 summary = tokenizer.decode(summary\_[0], skip\_special\_tokens=True)  
  
 return render\_template('output.html', data={'summary': summary})  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run()  
```

## Code of index.html:

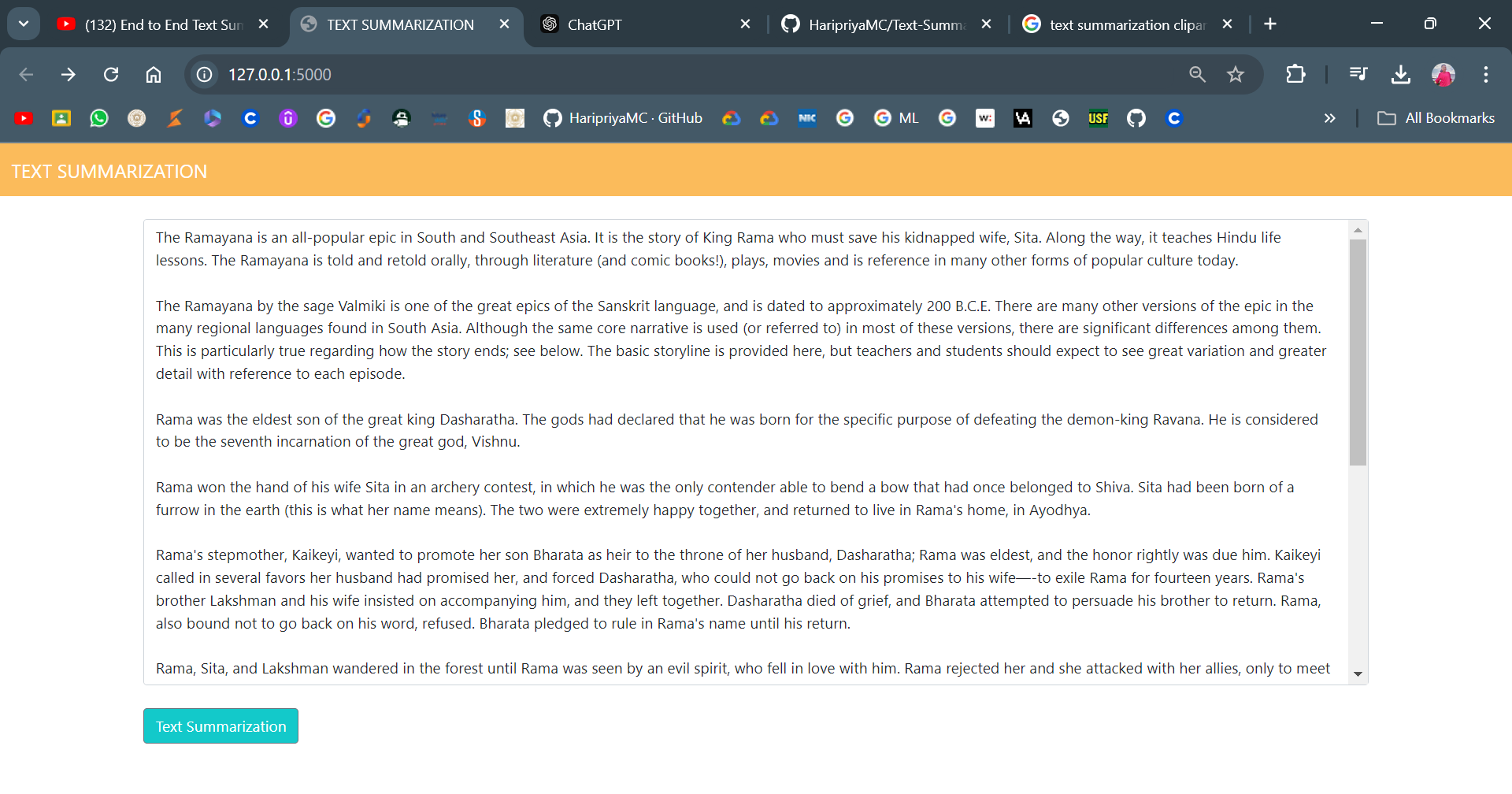
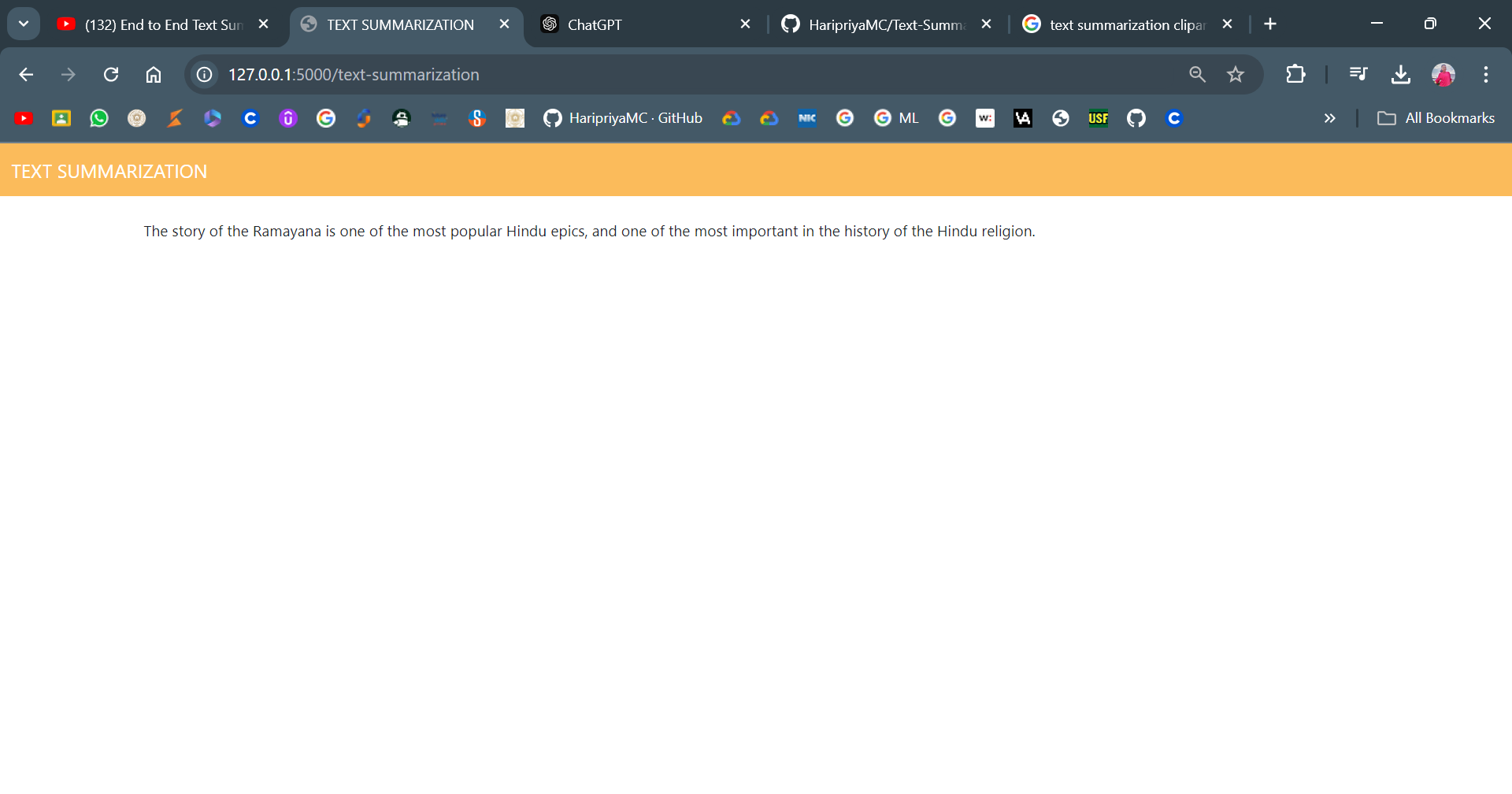
```html  
<!DOCTYPE html>  
<html lang='en'>  
  
<head>  
 <meta charset='utf-8' />  
 <meta name='viewport' content='width=device-width, initial-scale=1' />  
 <link href='https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css' rel='stylesheet' />  
 <title>TEXT SUMMARIZATION</title>  
</head>  
  
<body>  
 <nav class='navbar navbar-expand-lg navbar-dark bg-dark'>  
 <div class='container-fluid'>  
 <a class='navbar-brand' href='#'>TEXT SUMMARIZATION</a>  
 </div>  
 </nav>  
  
 <div class='container w-100'>  
 <div class='row justify-content-md-center'>  
 <div class='col'>  
 <br>  
 <form method='post' action='{{ url\_for('summarize') }}'>  
 <div class='form-group'>  
 <textarea class='form-control' name='inputtext\_' id='inputtext\_' rows='20' style='resize: none;'></textarea>  
 </div>  
 <br>  
 <button type='submit' class='btn btn-secondary float-right'>Text Summarization</button>  
 </form>  
 </div>  
 </div>  
 </div>  
  
 <script src='https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js'></script>  
</body>  
</html>  
```

## Code of output.html:

```html  
<!DOCTYPE html>  
<html lang='en'>  
  
<head>  
 <meta charset='utf-8' />  
 <meta name='viewport' content='width=device-width, initial-scale=1' />  
 <link href='https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css' rel='stylesheet' />  
 <title>TEXT SUMMARIZATION</title>  
</head>  
  
<body>  
 <nav class='navbar navbar-expand-lg navbar-dark bg-dark'>  
 <div class='container-fluid'>  
 <a class='navbar-brand' href='#'>TEXT SUMMARIZATION</a>  
 </div>  
 </nav>  
  
 <div class='container w-100'>  
 <div class='row justify-content-md-center'>  
 <div class='col'>  
 <br>  
 <p>{{ data['summary'] }}</p>  
 </div>  
 </div>  
 </div>  
  
 <script src='https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js'></script>  
</body>  
</html>  
```

# Result:

The web application successfully summarizes the input text using the Pegasus model. Users can input lengthy text, and the model generates a concise summary while maintaining the original context. The application is user-friendly and provides an efficient way to condense information.

# Conclusion:

In conclusion, this project demonstrates the capability of the Pegasus model for text summarization. The application developed using Flask and Hugging Face Transformers allows users to simplify large amounts of text into concise summaries. Future improvements could include enhancing the user interface and expanding the model's capabilities to handle various languages and styles of summarization.