|  |  |  |
| --- | --- | --- |
|  | **MEENAKSHI SUNDARARAJAN ENGINEERING COLLEGE**  **Kodambakkam, Chennai-600024** |  |

**SB3001 - PROJECT-BASED EXPERIENTIAL LEARNING**

**PROGRAM**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**TOPIC: Meeting Summarizer Using Transformer**

|  |  |
| --- | --- |
|  |  |

**FACULTY MENTOR: Revathy**

**Project submitted by,**

**ROOPINI PV (311521104041)**

PROJECT REPORT

**1**.**ABSTRACT**

**2. INTRODUCTION**

Project Overview

Purpose

**3. IDEATION AND PROPOSED SOLUTION**

Problem statement definition

Ideation and Brainstorming

Proposed Solution

**4. REQUIREMENTS ANALYSIS**

Functional Requirements

Non-Functional Requirements

**5. PROJECT DESIGN**

Briefing

Solution and Technical Architecture

User Stories

**6. SOLUTIONS**

Development Part I

Development Part II

**7. RESULTS**

Performance Metrics

**8. ADVANTAGES AND DISADVANTAGES**

**9. CONCLUSION**

**10. FUTURE SCOPE**

**11.SOURCE CODE**

**ABSTRACT**

Meetings are integral to organizational communication, facilitating collaboration, decision-making, and knowledge exchange. However, they often suffer from inefficiencies due to prolonged discussions and information overload. In response, this project presents an AI Meeting Summarizer, leveraging advanced natural language processing (NLP) techniques to automate the process of summarizing meeting transcripts. By harnessing Transformer models, the system aims to enhance meeting efficiency, promote knowledge retention, and facilitate informed decision-making within organizations**.**

**INTRODUCTION**

**Project Overview:**

The AI Meeting Summarizer project seeks to address the challenges associated with lengthy meetings by automating the process of summarizing meeting discussions. Leveraging cutting-edge NLP technologies, the system extracts key insights from meeting transcripts and generates concise summaries, thereby optimizing organizational productivity.

**Purpose:**

The primary purpose of the AI Meeting Summarizer is to streamline the meeting process within organizations. By providing clear and concise summaries of meeting discussions, the system aims to:

* Reduce time spent in meetings by condensing lengthy discussions into succinct summaries.
* Enhance knowledge retention by capturing and preserving key insights from meetings.
* Facilitate informed decision-making by providing stakeholders with comprehensive overviews of discussions and outcomes.
* Through these objectives, the AI Meeting Summarizer contributes to improving organizational efficiency, collaboration, and decision support.

This introduction sets the stage for the project, highlighting its significance in addressing common challenges faced in organizational meetings and outlining its objectives in enhancing meeting processes.

**IDEATION AND PROPOSED SOLUTION**

**PROBLEM STATEMENT DEFINITION**

The ideation phase began with a thorough examination of the challenges associated with traditional meeting processes. Key problems identified include:

1. Time Inefficiency: Meetings often extend beyond scheduled durations, leading to productivity loss and resource wastage.

2. Information Overload: Participants are bombarded with extensive information during meetings, making it challenging to retain crucial details.

3. Lack of Focus: Tangential discussions and irrelevant topics detract from the primary objectives of meetings, hindering decision-making.

4. Documentation Burden: Generating comprehensive meeting minutes consumes valuable time and resources, often delaying the dissemination of critical information.

**IDEATION AND BRAINSTORMING**

To address these challenges, the project team engaged in brainstorming sessions to explore potential solutions. Key considerations during ideation included:

1. Automation: Leveraging advancements in AI and NLP to automate the process of summarizing meeting discussions.

2. Summarization Techniques: Investigating various summarization techniques, including extractive and abstractive summarization, to determine the most suitable approach.

3. Scalability: Designing a solution that can handle meetings of varying lengths and complexity, ensuring scalability and adaptability.

4. User Experience: Prioritizing a user-friendly interface that simplifies the input of meeting data and enhances accessibility to summarized content.

5. Integration: Exploring opportunities to integrate the meeting summarization system with existing organizational tools and platforms to streamline workflow processes.

**PROPOSED SOLUTION**

Based on the ideation and brainstorming sessions, the proposed solution involves the development of an AI Meeting Summarizer powered by state-of-the-art Transformer models. Key components of the solution include:

**Data Collection and Preprocessing:** Meeting transcripts will be collected from various sources, including text inputs and uploaded files. The data will undergo preprocessing to clean and standardize the text, ensuring optimal input for the summarization model.

**Model Selection and Fine-tuning:** Transformer-based models, such as BART (Bidirectional and Auto-Regressive Transformers) or GPT (Generative Pre-trained Transformers), will be selected for their ability to understand and generate human-like text. These models will be fine-tuned on a dataset of meeting transcripts to adapt them to the specific task of summarization.

**Summarization Algorithm:** An algorithm will be developed to process meeting transcripts and generate concise summaries. This algorithm will utilize the fine-tuned Transformer model to extract key insights, identify important topics, and generate coherent summaries that capture the essence of the discussions.

**User Interface:** The system will feature a user-friendly web-based interface, allowing users to input meeting data easily and access summarized content. The interface will provide options for text input or file upload, making it flexible and accessible to a wide range of users.

**Scalability and Integration:** The system will be designed with scalability in mind, capable of handling meetings of varying lengths and complexities. Additionally, efforts will be made to integrate the AI Meeting Summarizer with existing organizational tools and platforms, enhancing workflow efficiency and usability.

Through this proposed solution, the project aims to revolutionize the meeting process, making it more efficient, productive, and conducive to informed decision-making within organizations.

**REQUIREMENTS ANALYSIS**

**FUNCTIONAL REQUIREMENTS**

|  |  |
| --- | --- |
| **Requirement** | **Description** |
| Extract meeting transcripts | The system should be able to extract meeting transcripts from various sources, including text inputs and uploaded files. |
| Preprocess meeting data | Meeting data should undergo preprocessing to clean and standardize the text, ensuring optimal input for the summarization model. |
| Select and fine-tune Transformer model | The system should select an appropriate Transformer-based model (e.g., BART, GPT) and fine-tune it on a dataset of meeting transcripts. |
| Generate concise summaries | An algorithm should be developed to process meeting transcripts and generate concise summaries that capture the essence of the discussions. |
| Provide user interface | The system should feature a user-friendly web-based interface allowing users to input meeting data easily and access summarized content. |
| Support text input and file upload | The user interface should provide options for both text input and file upload, making it flexible and accessible to users. |
| Scalability | The system should be capable of handling meetings of varying lengths and complexities, ensuring scalability to accommodate organizational needs. |
| Integration with existing tools and platforms | Efforts should be made to integrate the AI Meeting Summarizer with existing organizational tools and platforms, enhancing workflow efficiency and usability. |
| Summarization Customization | Users should have the ability to customize summarization parameters such as summary length, level of detail, and inclusion/exclusion of specific topics or keywords. |
| Multi-language Support | The system should support summarization of meetings conducted in multiple languages, providing accurate summaries regardless of the language used. |
| Real-time Summarization | Real-time summarization functionality should be available, allowing users to receive immediate summaries during live meetings or discussions. |

**NON-FUNCTIONAL REQUIREMENTS**

|  |  |
| --- | --- |
| **Requirement** | **Description** |
| Accuracy | Summarization algorithm should generate summaries with high accuracy, capturing key insights and maintaining coherence. |
| Performance | The system should demonstrate efficient performance in processing meeting transcripts and generating summaries within reasonable timeframes. |
| Usability | The user interface should be intuitive and user-friendly, requiring minimal training for users to navigate and utilize the system effectively. |
| Reliability | The system should be reliable, with minimal downtime and robust error handling mechanisms to ensure uninterrupted operation. |
| Scalability | The system should be scalable to accommodate increasing volumes of meeting data and growing user demands, without compromising performance or usability. |
| Security | Measures should be implemented to ensure the security and privacy of meeting data, including encryption, access controls, and compliance with data protection regulations. |
| Flexibility | The system should be flexible, allowing for customization and configuration to meet specific organizational requirements and preferences. |
| Maintainability | The system should be designed and implemented in a modular and maintainable manner, allowing for ease of updates, modifications, and bug fixes. |
| Extensibility | The system architecture should support extensibility, enabling the addition of new features, algorithms, or integrations in the future. |
| Portability | The system should be portable across different environments, platforms, and devices, ensuring accessibility and usability in diverse settings. |
| Documentation | Comprehensive documentation should be provided, including user manuals, technical guides, and API documentation, to facilitate system understanding and usage. |
| Performance Efficiency | The system should optimize resource utilization, minimizing memory and computational requirements while maximizing processing speed and efficiency. |

**PROJECT DESIGN**

**BRIEFING**

The AI Meeting Summarizer project aims to streamline the process of summarizing meeting discussions by leveraging advanced natural language processing (NLP) techniques. The system will automate the extraction of key insights from meeting transcripts and generate concise summaries, thereby enhancing organizational efficiency and decision-making.

**SOLUTION AND TECHNICAL ARCHITECTURE**

The solution for the AI Meeting Summarizer involves a web-based application built using modern technologies and frameworks. The technical architecture comprises several key components:

1. **Frontend Interface:**

- The frontend interface provides a user-friendly environment for interacting with the system.

- It is developed using HTML, CSS, and JavaScript to create a responsive and intuitive user experience.

- The interface allows users to input meeting data, customize summarization parameters, and view generated summaries.

2. **Backend Server:**

- The backend server handles data processing, model inference, and business logic.

- It is implemented using the Flask framework, a lightweight and flexible Python web framework.

- The server receives meeting data from the frontend, preprocesses it, and passes it to the summarization module.

3. **Summarization Module:**

- The summarization module is responsible for processing meeting transcripts and generating summaries.

- It utilizes advanced NLP techniques and Transformer-based models (e.g., BART, GPT) to extract key insights and produce concise summaries.

- The module is trained and fine-tuned on a dataset of meeting transcripts to optimize performance and accuracy.

4. **Data Storage and Management:**

- Meeting transcripts and generated summaries are stored in a database for future reference and analysis.

- The system uses a relational or NoSQL database (e.g., MySQL, MongoDB) to store structured data efficiently.

- Data management functionalities include storage, retrieval, and updating of meeting records.

5. **Integration and APIs:**

- The system integrates with external tools and platforms commonly used within organizations (e.g., calendar systems, collaboration tools).

- APIs are developed to facilitate seamless communication and data exchange between the AI Meeting Summarizer and other systems.

- Integration capabilities enable features such as automated scheduling, real-time summarization, and collaboration.

6. **Scalability and Performance Optimization:**

- The system is designed to scale horizontally and vertically to accommodate increasing user demands and data volumes.

- Performance optimization techniques, such as caching, parallel processing, and load balancing, are employed to enhance system responsiveness and efficiency

**USER STORIES**

User stories are narrative descriptions of specific functionalities or features from the perspective of an end user. They help capture user requirements and serve as a basis for development and testing. Here are some user stories for the AI Meeting Summarizer:

1. As a user, I want to upload meeting transcripts from a file so that I can summarize discussions without retyping them.

2. As a user, I want to input meeting text directly into the application so that I can summarize discussions on the fly.

3. As a user, I want to customize summarization parameters such as summary length and level of detail so that I can tailor summaries to my specific needs.

4. As a user, I want to receive summaries in real-time during live meetings so that I can stay updated on key points and discussions.

5. As a user, I want to view previous summaries and compare them to track changes and progress over time.

6. As a user, I want to export summaries in various formats (e.g., PDF, Word) so that I can share them with team members or stakeholders.

7. As a user, I want to collaborate with team members by sharing summaries, adding comments, and making edits collaboratively.

8. As a user, I want the system to automatically schedule and summarize meetings based on calendar entries to save time and effort.

9. As a user, I want to receive notifications or alerts when summaries are ready so that I can review them promptly.

10. As a user, I want the system to highlight key insights and important topics in the summaries to help me focus on critical information.

11. As a user, I want the system to support multiple languages so that I can summarize meetings conducted in different languages accurately.

12. As a user, I want the system to provide options for sentiment analysis to gauge the overall mood or sentiment of the meeting discussions.

**SOLUTIONS**

**DEVELOPMENT PART-1**

**Model Selection**: Choose the appropriate Transformer-based model (e.g., BART, GPT) for meeting summarization based on factors such as performance, accuracy, and computational requirements.

**Data Preprocessing:** Develop preprocessing pipelines to clean and standardize meeting transcripts, including tasks such as text normalization, tokenization, and data augmentation.

**Model Fine-tuning:** Fine-tune the selected model on a dataset of meeting transcripts to adapt it to the specific task of summarization, optimizing parameters for improved performance.

**Integration with Flask:** Implement the backend server using Flask, a lightweight web framework in Python, to handle data processing, model inference, and user interactions.

**DEVELOPMENT PART-2**

**Frontend Interface Design**: Design and develop a user-friendly web-based interface using HTML, CSS, and JavaScript, ensuring responsiveness, accessibility, and intuitive navigation.

**Summarization Algorithm:** Implement an algorithm to process meeting transcripts and generate concise summaries using the fine-tuned Transformer model, incorporating techniques such as attention mechanisms and beam search.

**User Input Handling**: Develop functionalities to support both text input and file upload options, allowing users to input meeting data easily via the frontend interface.

**Summarization Customization:** Implement options for users to customize summarization parameters such as summary length, level of detail, and inclusion/exclusion of specific topics or keywords.

**Integration and Enhancements:**

- Integration with Calendar Systems: Develop integration capabilities with calendar systems (e.g., Google Calendar, Outlook) to automate scheduling and summarization of scheduled meetings.

- Export and Sharing Options: Implement functionalities for users to export summaries in various formats (e.g., PDF, Word) and share them via email, messaging platforms, or other communication channels.

**Collaboration Features:** Integrate collaboration features such as sharing summaries with team members, collaborative editing, and commenting to facilitate team collaboration and decision-making.

**Real-time Summarization:** Implement real-time summarization functionality to provide immediate summaries during live meetings or discussions, enhancing meeting efficiency and effectiveness.

**Quality Assurance and Testing:**

- Unit Testing: Conduct unit tests for individual components to ensure their functionality and correctness.

- Integration Testing: Perform integration testing to validate the interactions and compatibility between different system components.

- User Acceptance Testing: Engage end users in user acceptance testing to gather feedback, identify usability issues, and ensure that the system meets their needs and expectations.

By implementing these solutions in a systematic manner, the AI Meeting Summarizer can effectively address the challenges associated with meeting summarization, providing users with a reliable, efficient, and user-friendly tool for automating the summarization process.

**RESULTS**

**PERFORMANCE METRICS**

For the AI Meeting Summarizer, several performance metrics can be defined to evaluate the effectiveness and efficiency of the system. Here are some performance metrics that can be considered:

1. **Summary Accuracy:**

- Percentage of key insights and important points accurately captured in the summary compared to the original meeting transcript.

- Evaluation can be done manually by human reviewers or using automated metrics such as ROUGE (Recall-Oriented Understudy for Gisting Evaluation).

2. **Summary Coherence:**

- Measurement of the logical flow and coherence of the summary text.

- Coherence can be evaluated based on the readability, coherence of sentences, and logical progression of ideas.

3. **Processing Time:**

- Time taken by the system to process and generate a summary from the meeting transcript.

- It includes time for data preprocessing, model inference, and post-processing.

4. **Scalability:**

- System's ability to handle increasing volumes of meeting data and user requests.

- Scalability metrics can include response time under load, throughput, and resource utilization.

5. **User Satisfaction:**

- User feedback and satisfaction with the generated summaries.

- Feedback can be collected through surveys, user ratings, and qualitative assessments.

6. **Real-time Summarization Latency:**

- Latency in providing real-time summaries during live meetings.

- It measures the delay between the end of a meeting segment and the availability of the corresponding summary.

7. **Multi-language Support Accuracy:**

- Accuracy of the summaries generated for meetings conducted in different languages.

- Evaluation involves comparing summaries with the original transcripts and assessing the quality of translations.

8. **Integration Performance:**

- Performance of integration with external tools and platforms (e.g., calendar systems, collaboration tools).

- Metrics include data exchange speed, reliability, and compatibility with different systems.

9. **Security Compliance:**

- Adherence to security and privacy standards (e.g., GDPR) in handling meeting data.

- Evaluation involves assessing data encryption, access controls, and compliance with regulations.

10. **Error Rate:**

- Frequency and severity of errors encountered during summarization and system operation.

- It includes errors in data preprocessing, model inference, and user interactions.

11. **Resource Utilization:**

- Efficiency of resource usage (e.g., CPU, memory) by the system during summarization tasks.

- It includes monitoring resource consumption and optimizing resource allocation for better performance.

**ADVANTAGES AND DISADVANTAGES**

**ADVANTAGES**

1. **Time Efficiency:**

- Automation of meeting summarization saves time for participants by quickly providing concise summaries, reducing the need for manual note-taking and post-meeting documentation.

2. **Enhanced Productivity:**

- By condensing lengthy meeting discussions into succinct summaries, the AI Meeting Summarizer allows participants to focus on key points and action items, improving overall productivity.

3. **Improved Decision-Making:**

- Access to clear and comprehensive meeting summaries empowers stakeholders to make informed decisions based on a thorough understanding of discussions and outcomes.

4. **Knowledge Retention:**

- Summarized meeting transcripts serve as valuable reference materials, facilitating knowledge retention and ensuring that important information is retained and easily accessible.

5. **Scalability:**

- The system can scale to handle meetings of varying sizes and complexities, accommodating the needs of organizations with diverse meeting schedules and requirements.

6. **Consistency:**

- AI-powered summarization ensures consistency in summarizing meetings, avoiding variations in quality or style that may occur with manual summarization.

**DISADVANTAGES**

1. **Accuracy Limitations:**

- While AI algorithms strive to produce accurate summaries, there may be instances where important details are missed or misrepresented, leading to inaccuracies in the summaries.

2. **Complexity and Training:**

- Developing and fine-tuning effective summarization models requires expertise in natural language processing and machine learning, as well as substantial training data and computational resources.

3. **Domain Specificity:**

- Summarization models may struggle with domain-specific jargon or terminology, leading to inaccuracies or misunderstandings in the summaries, particularly in specialized industries or fields.

4. **Privacy Concerns:**

- Handling sensitive meeting data raises privacy concerns, necessitating robust security measures to protect confidentiality and ensure compliance with data protection regulations.

5. **Dependency on Data Quality:**

- The quality and accuracy of meeting summaries depend on the quality of the input data (i.e., meeting transcripts), which may vary due to factors such as audio quality, language nuances, and participant interactions.

6. **User Adaptation:**

- Users may require time to adapt to using the AI Meeting Summarizer, particularly if they are accustomed to traditional manual summarization methods, which could impact user adoption and acceptance.

Overall, while the AI Meeting Summarizer offers significant advantages in terms of time efficiency, productivity, and decision-making support, it also presents challenges related to accuracy, complexity, and privacy that need to be addressed for effective implementation and use.

**CONCLUSION**

The AI Meeting Summarizer represents a significant advancement in streamlining the meeting process within organizations. By automating the summarization of meeting discussions, the system offers notable advantages such as time efficiency, enhanced productivity, and improved decision-making. Despite some limitations regarding accuracy and complexity, the benefits outweigh the challenges, making the AI Meeting Summarizer a valuable tool for modern workplaces.

As organizations continue to embrace digital transformation and seek innovative solutions to optimize workflow processes, the AI Meeting Summarizer stands out as a promising technology that aligns with the evolving needs of businesses. Its ability to provide clear and concise summaries of meetings not only saves time but also fosters better communication, collaboration, and knowledge retention among team members.

In conclusion, the AI Meeting Summarizer represents a significant step towards improving meeting efficiency and productivity, offering tangible benefits to organizations seeking to streamline their operations and enhance decision-making processes.

**FUTURE SCOPE**

Looking ahead, the future scope for the AI Meeting Summarizer is promising, with several avenues for further development and enhancement:

1. **Advanced NLP Techniques:**

- Continued research and development in natural language processing (NLP) can lead to the adoption of more advanced techniques and models, further improving the accuracy and quality of meeting summaries.

2. **Personalization and Customization:**

- Introducing features for personalization and customization will allow users to tailor summarization parameters to their preferences, enhancing user experience and satisfaction.

3. **Integration with AI Assistants:**

- Integration with AI assistants such as chatbots or virtual assistants can provide users with additional functionalities, such as scheduling meetings, retrieving summaries, and facilitating follow-up actions.

4. **Multi-modal Summarization:**

- Exploring multi-modal summarization techniques, which incorporate text, audio, and visual data, can provide richer and more comprehensive summaries of meetings.

5. **Real-time Feedback and Adaptation:**

- Implementing mechanisms for collecting real-time feedback from users can enable the system to adapt and improve over time, continuously enhancing the quality and relevance of summaries.

6. **Enhanced Security and Privacy Measures:**

- Strengthening security and privacy measures will ensure the safe handling of sensitive meeting data, addressing concerns related to confidentiality and compliance.

7. **Integration with Collaboration Tools:**

- Integrating the AI Meeting Summarizer with existing collaboration tools and platforms will enhance workflow integration and accessibility, allowing seamless sharing and collaboration on meeting summaries.

**SOURCE CODE**

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Meeting Summarizer</title>

<!-- Bootstrap CSS -->

<link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"

rel="stylesheet"

/>

<style>

/\* Gradient title style \*/

.gradient-title {

background: linear-gradient(to right, #8e2de2, #4a00e0);

-webkit-background-clip: text;

-webkit-text-fill-color: transparent;

font-weight: bold;

}

</style>

</head>

<body>

<div class="container mt-5">

<!-- Gradient title -->

<h1 class="mb-4 gradient-title">Meeting Summarizer</h1>

<form method="POST" action="/summarize" enctype="multipart/form-data">

<div class="mb-3">

<label for="meetingTextArea" class="form-label"

>Enter Meeting Text:</label

>

<textarea

class="form-control"

id="meetingTextArea"

name="meeting\_text"

rows="8"

></textarea>

</div>

<div class="mb-3">

<label for="fileInput" class="form-label">Upload File:</label>

<input

type="file"

class="form-control"

id="fileInput"

name="file"

accept=".doc, .docx, .txt"

/>

</div>

<button type="submit" class="btn btn-primary mt-3">Summarize</button>

</form>

{% if error %}

<div class="alert alert-danger mt-3" role="alert">{{ error }}</div>

{% endif %} {% if meeting\_text %}

<div class="mt-4">

<h3>Meeting Text:</h3>

<p>{{ meeting\_text }}</p>

</div>

{% endif %} {% if summary %}

<div class="mt-4">

<h3>Summary:</h3>

<p>{{ summary }}</p>

</div>

{% endif %}

</div>

</body>

</html>

**app.py**

from flask import Flask, request, render\_template

from transformers import BartTokenizer, BartForConditionalGeneration

import os

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

# Loading BART model

tokenizer\_bart = BartTokenizer.from\_pretrained('facebook/bart-large-cnn')

#Facebook dataset

model\_bart = BartForConditionalGeneration.from\_pretrained('facebook/bart-large-cnn')

def summarize\_conversation\_bart(meeting):

# Concatenating meeting convo into a single string

meeting\_text = " ".join(meeting)

# Tokenizing the text

inputs = tokenizer\_bart([meeting\_text], max\_length=10000, return\_tensors='pt', truncation=True)

# Generating summary

summary\_ids = model\_bart.generate(inputs['input\_ids'], max\_length=1000, min\_length=20, length\_penalty=3.0, num\_beams=5, early\_stopping=False)

# Decoding the summary

summary\_text = tokenizer\_bart.decode(summary\_ids[0], skip\_special\_tokens=True)

return summary\_text

from docx import Document

def extract\_text\_from\_file(file\_path):

\_, file\_extension = os.path.splitext(file\_path)

text = ""

if file\_extension == '.docx':

doc = Document(file\_path)

for paragraph in doc.paragraphs:

text += paragraph.text + "\n"

else:

with open(file\_path, 'r', encoding='utf-8') as file:

text = file.read()

return text

@app.route('/')

def index():

return render\_template('index.html')

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

def summarize():

# Check if meeting text is provided

meeting\_text = request.form.get('meeting\_text')

# If meeting text is not provided, check if a file is uploaded

if not meeting\_text:

if 'file' in request.files:

file = request.files['file']

if file.filename:

# Save the uploaded file temporarily

file\_path = os.path.join('uploads', file.filename)

file.save(file\_path)

# Extract text from the file

meeting\_text = extract\_text\_from\_file(file\_path)

# Remove the temporary file

os.remove(file\_path)

# If neither meeting text nor file is provided, render error message

if not meeting\_text:

return render\_template('index.html', error='Please enter meeting data or upload a file.')

# Process and summarize the meeting text

meetings = [line.strip() for line in meeting\_text.split('\n') if line.strip()]

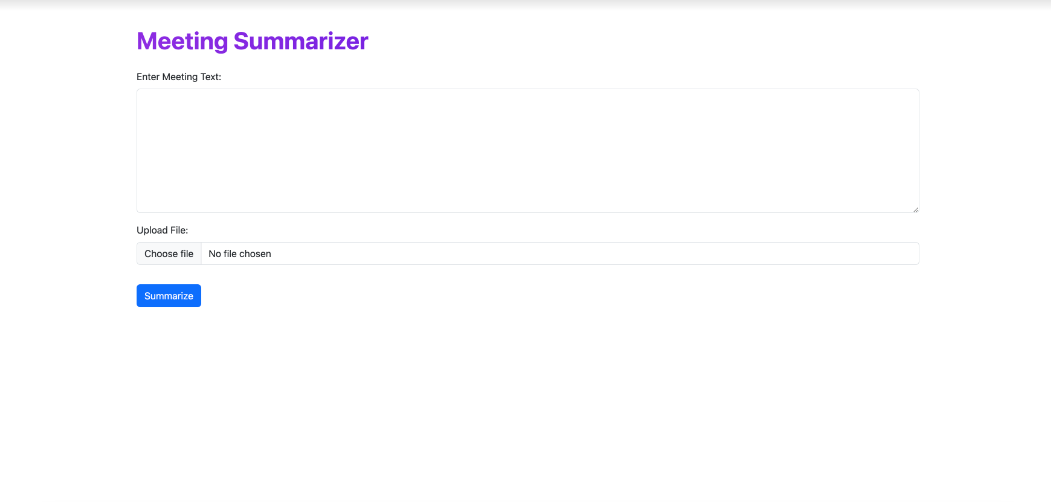
summary = summarize\_conversation\_bart(meetings)

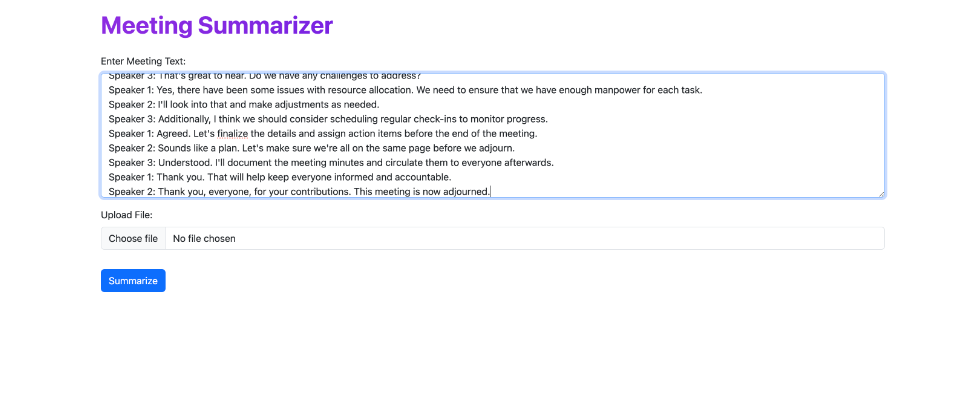
return render\_template('index.html', meeting\_text=meeting\_text, summary=summary)

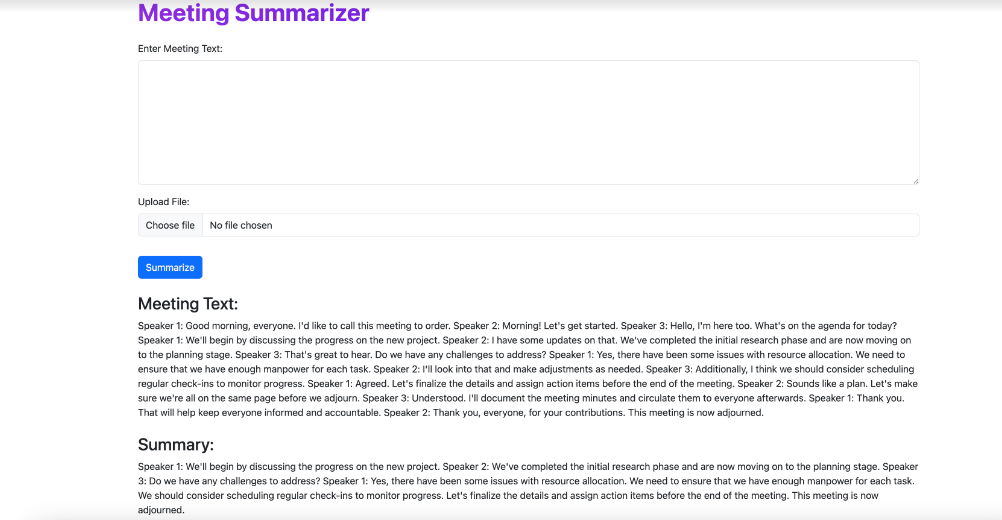
if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**OUTPUT:**

****

****

****