**Report: Meet Mate – AI Agent for Smarter Online Meetings**

**Team Details:**

**Team Name: Deep Minders**

**Member 1:**J.Divya (231FA04162), CSE, VFSTR

**Member 2**:A.Meghana (231FA04D58), CSE ,VFSTR

**Member 3:** L.Saritha(231FA04D70), CSE, VFSTR

**Member 4:** B.Ahalya(231FA04540), CSE, VFSTR

### 1.Abstract

In the evolving landscape of remote and hybrid work environments, efficient management of online meetings is critical to maintain productivity and collaboration. **MeetMate** is an intelligent AI-powered virtual meeting assistant designed to automate and enhance online meetings by providing real-time transcription, summarization, translation, and interactive question-answering capabilities. Leveraging state-of-the-art natural language processing (NLP) and speech recognition technologies, MeetMate generates concise meeting summaries, extracts action items, translates content into multiple languages, and facilitates follow-up communication through automated emails and calendar integrations. Additional features include sentiment analysis, topic clustering, and voice-controlled interactions, aiming to make meetings more inclusive, engaging, and productive. This AI agent significantly reduces manual note-taking effort, improves accessibility, and ensures key decisions and tasks are clearly communicated, ideal for remote teams, multilingual workplaces, and educational settings.

**2. Introduction**

MeetMate is an innovative AI-powered virtual meeting assistant designed to enhance workplace productivity by automating key meeting tasks. The system leverages state-of-the-art AI and NLP technologies to transcribe, summarize, translate, analyze, and interact with online meeting content, thereby reducing manual effort, improving clarity, and fostering better communication among participants.

**3.Literature Review**

The development of an intelligent meeting assistant like MeetMate involves an interdisciplinary blend of speech processing, natural language understanding, and human-computer interaction. This review explores the foundational and recent advancements in the areas of speech recognition, text summarization, question answering, translation, action extraction, and sentiment analysis.

**3.1 Automatic Speech Recognition (ASR)**

Automatic Speech Recognition converts spoken language into text, forming the foundation for further NLP tasks in meeting assistants. Widely-used ASR models include:

* **Whisper by OpenAI (2022):** A multilingual, multitask ASR system trained on 680,000 hours of diverse audio. It shows robustness across accents and noisy environments [1].
* **Google Speech-to-Text API:** Offers real-time transcription with high accuracy, leveraging Google’s extensive data and models [2].
* **CMU Sphinx & Kaldi:** Earlier toolkits used in academic research, though less popular for modern deep-learning-based applications.

The accuracy of transcription directly impacts the quality of summarization and Q&A, making ASR a critical component.

**3.2Text Summarization**

Text summarization in meeting contexts aims to extract key points from long transcripts:

* **Abstractive Summarization Models:**
  + **BART (Lewis et al., 2020)** and **T5 (Raffel et al., 2020)** are transformer-based models pre-trained on text infilling and translation tasks, achieving state-of-the-art performance on summarization benchmarks like CNN/DailyMail [3][4].
  + **PEGASUS (Zhang et al., 2020)** is specially designed for abstractive summarization, using gap-sentence generation for pretraining.

These models are fine-tuned on large summarization corpora, enabling them to produce fluent, concise summaries of complex dialogues.

**3.3 Question Answering (Q&A)**

Contextual question answering allows users to extract specific information from a transcript:

* **BERT-QA (Devlin et al., 2018)**: Fine-tuned on the SQuAD dataset, BERT can answer questions by identifying relevant spans within a document. It has been foundational in building chatbots and QA systems over static texts [5].
* **DistilBERT and RoBERTa**: Lighter and faster alternatives to BERT, offering nearly equivalent accuracy for real-time applications.

These models allow MeetMate to function as an interactive assistant capable of on-demand querying of past meeting content.

**3.4 Translation and Accessibility**

Making summaries multilingual and accessible is vital for global and inclusive communication.

* **Google Translate API**: One of the most widely used APIs for real-time, multi-language translation, supporting over 100 languages [6].
* **gTTS (Google Text-to-Speech)**: Converts textual summaries into spoken audio, aiding visually impaired users and enhancing accessibility.

Recent work in multilingual NLP (mBART, mT5) has improved translation accuracy for low-resource languages like Tamil and Hindi.

**3.5 Action Item Extraction**

Identifying and assigning tasks from meetings is an active research area in dialogue analysis:

* **Named Entity Recognition (NER)** and **Dependency Parsing** can identify people, dates, and verbs to extract actionable tasks.
* **Task-oriented Dialogue Systems** (Chen et al., 2019) show how intent recognition and slot filling can be applied to detect decisions and commitments from dialogues [7].

Combining rule-based and model-based methods can increase the precision of task extraction.

**3.6 Sentiment Analysis**

Sentiment detection in meetings can uncover emotions and engagement levels:

* **VADER (Hutto & Gilbert, 2014)**: A lexicon-based sentiment analysis tool fine-tuned for social media text but applicable to short dialogues.
* **BERT-based Sentiment Models**: Offer contextual sentiment understanding, detecting subtle emotional tones in speech.

Understanding sentiment can help team leaders assess the atmosphere of a meeting or detect frustration/conflict.

**3.7 Clustering and Topic Modeling**

To segment and organize meeting conversations, topic modeling and clustering are used:

* **KMeans & Agglomerative Clustering**: Applied to TF-IDF or sentence embeddings (e.g., using BERT or SBERT) to group similar sentences or topics.
* **LDA (Latent Dirichlet Allocation)**: A generative probabilistic model used for unsupervised topic extraction from documents [8].

**4.Methodology**

The development of MeetMate involves multiple AI and NLP components working together in a pipeline. The methodology can be broken down into the following sequential steps:

**4.1 Audio Capture and Speech Recognition**

* **Objective:** Convert meeting speech into accurate, structured text.
* **Method:**
  + The system either connects to live meeting streams or processes uploaded audio files.
  + Speech is transcribed using deep learning-based Automatic Speech Recognition (ASR) models.

**4.2 Transcript Preprocessing**

* **Objective:** Clean and prepare the transcript for analysis.
* **Method:**
  + Remove filler words and background noise.
  + Segment speech by speakers (speaker diarization) for accurate attribution.
  + Tokenize text for downstream NLP tasks.

**4.3 Summarization Module**

* **Objective:** Generate concise summaries of lengthy meeting transcripts.
* **Method:**
  + Use transformer-based **abstractive summarization** (e.g., BART, T5, or Pegasus).
  + Divide large transcripts into manageable chunks.
  + Merge generated summaries into a final cohesive output.
* **Output:** Bullet-point summary, highlights, and decisions.

**4.4 Question Answering (Q&A) Module**

* **Objective:** Allow users to query specific information from the meeting.
* **Method:**
  + Use pretrained Q&A models (like BERT fine-tuned on SQuAD) to extract context-aware answers from the transcript.
  + Accept both voice and text input for user queries.

**4.5 Action Item Extraction**

* **Objective:** Identify and assign tasks or decisions made in the meeting.
* **Method:**
  + Apply rule-based and machine learning methods (like NER and dependency parsing).
  + Detect named entities (people, dates, tasks) and extract commitments (e.g., “John will finalize the report”)

**4.6 Translation and Accessibility**

* **Objective:** Translate summaries for multilingual teams and improve accessibility.
* **Method:**
  + Use Google Translate API or translate library to translate summaries.
  + Generate audio summaries using Google Text-to-Speech (gTTS) for visually impaired users.

**4.7 Sentiment Analysis**

* **Objective:** Analyze participant mood and emotional tone of meetings.
* **Method:**
  + Apply BERT or lexicon-based models (e.g., VADER) to detect sentiment polarity and shifts during the meeting.
  + Optionally visualize mood over time.

**4**.**8 Topic Clustering and Visualization**

* **Objective:** Identify major themes and agenda items discussed.
* **Method:**
  + Apply clustering algorithms (KMeans, Agglomerative Clustering) on sentence embeddings.
  + Group sentences into coherent topics or agenda items.

**4.9 Follow-Up Communication**

* **Objective:** Share meeting results and plan next steps.
* **Method:**
  + Automatically send meeting summaries and action items via email.
  + Schedule next meetings using calendar APIs (Google Calendar, Microsoft Outlook).
  + Provide reminders and notifications.

**4.10 User Interface and Interaction**

* **Objective:** Provide an intuitive way for users to interact with MeetMate.
* **Method:**
  + **Web interface** using Flask/React for uploading files, displaying summaries, and interaction.
  + **Voice commands** supported via Dialogflow or Gemini for hands-free control.

**5.Block Diagram**

**Online Meeting Audio**

**(Zoom, Teams, Google Meet, etc.)**

**Speech Recognition**

**(speech\_recognition, Whisper)**

**Converts audio → text transcript in real-time**

**Transcript Preprocessing**

**- Clean transcript**

**- Speaker diarization (optional)**

**Summarization Model Q&A Model**

**(BART, T5, Pegasus) (BERT fine-tuned SQuAD)**

**Generates meeting Answers participant**

**summary + action questions based on**

**items + highlights transcript**

**Action Item Extraction Module (NLP-based)**

**Detects to-do's and assigns responsible participants**

**Keyword & Topic Clustering + Sentiment Analysis**

**- KMeans clustering groups conversation topics**

**- Sentiment analysis tracks participant mood & shifts**

**Summary Translation Module**

**- Translate summaries to multiple languages**

**- Generate audio summary with gTTS**

**Auto-Mail Summary + Follow-Up Communication**

**- Email summary & action items to participants**

**- Schedule follow-ups & calendar invites**

**User Interaction Module**

**- Voice-controlled queries/commands (Gemini/Dialogflow)**

**- Text-based Q&A and commands**

****

**6.Complete Code**

!pip install flask transformers openai-whisper googletrans==4.0.0-rc1

!pip install torch --index-url https://download.pytorch.org/whl/cpu

!apt install ffmpeg -y

import os

import threading

from flask import Flask, request, render\_template\_string

import whisper

from transformers import pipeline

from googletrans import Translator

from google.colab import output

# Load models

whisper\_model = whisper.load\_model("base")

summarizer = pipeline("summarization", model="facebook/bart-large-cnn")

qa\_pipeline = pipeline("question-answering", model="bert-large-uncased-whole-word-masking-finetuned-squad")

translator = Translator()

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

stored\_transcript = ""

stored\_summary = ""

# Styled HTML Template

HTML = """

<!doctype html>

<html>

<head>

  <title>MeetMate: Smart Meeting Assistant</title>

  <style>

    body {

      font-family: 'Segoe UI', sans-serif;

      background: #f4f7f9;

      color: #333;

      padding: 30px;

      max-width: 900px;

      margin: auto;

    }

    h2, h3 {

      color: #2c3e50;

    }

    form, .output-box {

      background: #fff;

      padding: 20px;

      border-radius: 8px;

      box-shadow: 0 2px 4px rgba(0,0,0,0.1);

      margin-bottom: 30px;

    }

    input[type="file"], input[type="text"], select {

      padding: 10px;

      margin-top: 10px;

      width: 100%;

      box-sizing: border-box;

    }

    input[type="submit"] {

      background-color: #3498db;

      color: white;

      padding: 10px 15px;

      border: none;

      border-radius: 4px;

      margin-top: 10px;

      cursor: pointer;

    }

    input[type="submit"]:hover {

      background-color: #2980b9;

    }

    pre {

      white-space: pre-wrap;

      word-wrap: break-word;

    }

  </style>

</head>

<body>

  <h2>📋 MeetMate: Smart Meeting Assistant</h2>

  <form method=post enctype=multipart/form-data>

    <label>🎧 Upload a Meeting Audio File (.mp3/.wav):</label><br>

    <input type="file" name="audio" accept="audio/\*" required><br>

    <input type="submit" value="Transcribe & Summarize">

  </form>

  {% if transcript %}

  <div class="output-box">

    <h3>📝 Transcript:</h3>

    <pre>{{ transcript }}</pre>

    <h3>📌 Summary:</h3>

    <p>{{ summary }}</p>

  </div>

  <form method="post">

    <h3>🔍 Ask a Question:</h3>

    <input type="text" name="question" placeholder="What was discussed about the budget?" required>

    <input type="submit" value="Get Answer">

  </form>

  {% if answer %}

    <div class="output-box">

      <h3>🤖 Answer:</h3>

      <p>{{ answer }}</p>

    </div>

  {% endif %}

  <form method="post">

    <h3>🌐 Translate Summary:</h3>

    <select name="language">

      <option value="hi">Hindi</option>

      <option value="ta">Tamil</option>

      <option value="fr">French</option>

      <option value="es">Spanish</option>

    </select>

    <input type="submit" value="Translate">

  </form>

  {% if translated %}

    <div class="output-box">

      <h3>🌎 Translated Summary:</h3>

      <p>{{ translated }}</p>

    </div>

  {% endif %}

  {% endif %}

</body>

</html>

"""

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

def index():

    global stored\_transcript, stored\_summary

    transcript = summary = answer = translated = None

    # Audio upload

    if 'audio' in request.files:

        audio = request.files['audio']

        ext = audio.filename.rsplit('.', 1)[-1]

        audio\_path = f"temp\_audio.{ext}"

        audio.save(audio\_path)

        result = whisper\_model.transcribe(audio\_path)

        transcript = result["text"]

        stored\_transcript = transcript

        summary\_output = summarizer(transcript, max\_length=150, min\_length=40, do\_sample=False)

        summary = summary\_output[0]['summary\_text']

        stored\_summary = summary

        os.remove(audio\_path)

    # Q&A

    if 'question' in request.form:

        question = request.form['question']

        if stored\_transcript:

            try:

                result = qa\_pipeline(question=question, context=stored\_transcript)

                answer = result['answer']

            except:

                answer = "Sorry, I couldn't find an answer."

    # Translation

    if 'language' in request.form:

        lang = request.form['language']

        if stored\_summary:

            try:

                translated = translator.translate(stored\_summary, dest=lang).text

            except:

                translated = "Translation failed."

    return render\_template\_string(HTML,

                                  transcript=stored\_transcript if stored\_transcript else None,

                                  summary=stored\_summary if stored\_summary else None,

                                  answer=answer,

                                  translated=translated)

# Start Flask in background

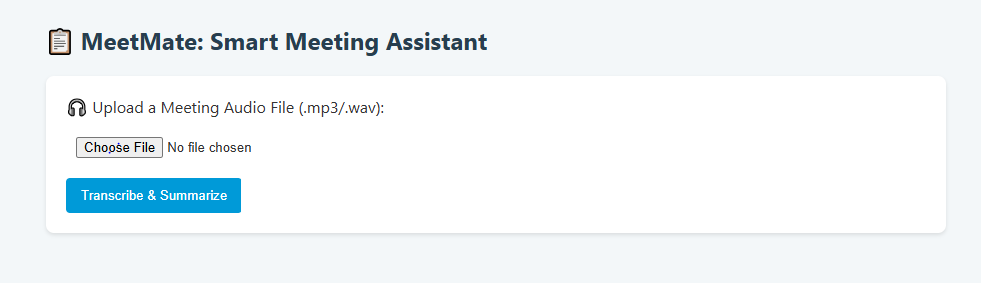
def run\_flask():

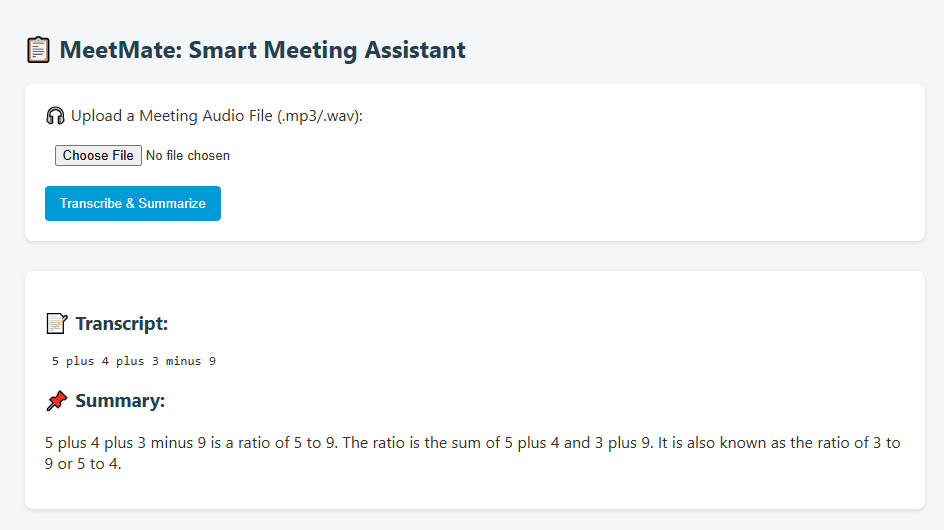
    app.run(host='0.0.0.0', port=5000)

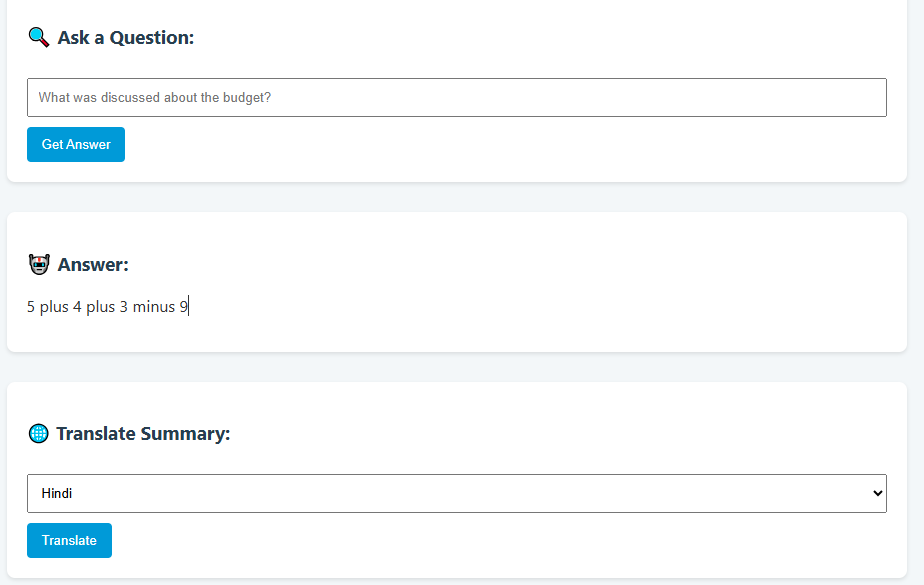
threading.Thread(target=run\_flask).start()

output.serve\_kernel\_port\_as\_window(5000)

**7.Result and Discussion**

****

****

****

**8. Conclusion and Future scope**

**MeetMate** demonstrates the potential of integrating advanced AI technologies to enhance the productivity and inclusivity of online meetings. By combining speech recognition, natural language processing, and multilingual support, MeetMate automates critical tasks such as real-time transcription, summarization, question answering, and follow-up communication. These capabilities not only save time and reduce manual effort but also ensure that important information is captured, clearly communicated, and accessible to all participants including those in multilingual or remote environments.The prototype highlights how transformer models (like BART and BERT) can be effectively leveraged for meeting summarization and contextual understanding. Additional features like sentiment analysis, translation, text-to-speech, and action item extraction enrich the overall meeting experience, turning passive communication into an interactive, intelligent process.MeetMate contributes significantly to modern workplace productivity by ensuring no key point is missed, no task goes unassigned, and every participant, regardless of language or accessibility needs, remains informed.

**Future Scope**

The future scope of MeetMate is vast and promising, given the rapid advancements in AI, natural language processing, and remote collaboration tools. One of the key enhancements lies in integrating MeetMate directly with popular video conferencing platforms like Zoom, Microsoft Teams, and Google Meet to enable real-time audio processing and live summarization during ongoing meetings. Speaker diarization can also be incorporated to attribute spoken content accurately to individual participants, thereby enhancing clarity in summaries and action assignments. Additionally, MeetMate can evolve into a comprehensive task management assistant by introducing features such as smart dashboards for tracking action items, automatic scheduling of follow-ups using calendar APIs, and proactive reminders. A mobile application version would further improve accessibility and on-the-go usage. Moreover, advanced sentiment and emotion detection could be introduced to analyze participant engagement and emotional tone throughout the meeting. Voice-controlled interaction using natural language understanding would allow users to command the assistant hands-free, making it even more interactive and efficient. With further customization capabilities and enterprise-level API integrations, MeetMate can become an indispensable productivity tool for businesses, educational institutions, and multilingual teams across the globe.

**9.References and bibilography**