



National University Of Computer and Emerging Sciences

Project Proposal

Presented to

Sir, Owais Idrees

In partial fulfillment

of the requirements for the course of

Natural Language Processing

By:

Umer Farooq (22I-1007)

Muhammad Rayyan(22I-1022)

Zayyam Hassan (22I-1247)

Hafsa Imtiaz (22I-0959)

Areeba Riaz (22I-1244)

Abeer Jawad (22I-1041)

Project Proposal

Meeting Insight Generator

In today's remote and hybrid work environment, meetings have become the primary mode of collaboration. However, the value extracted from these meetings often falls short of the time invested. Our project aims to bridge this gap by building an intelligent system that automatically transforms meeting recordings into structured, actionable insights.

Problem Statement

Organizations and teams face significant challenges in managing meeting outcomes effectively:

- **Information Loss:** Critical decisions and action items discussed during meetings are often forgotten or poorly documented. Team members rely on fragmented notes or memory, leading to miscommunication and dropped tasks.
- **Time Wastage:** After each meeting, someone must spend 15-30 minutes per hour of meeting time creating summaries, extracting action items, and distributing information to stakeholders. This manual process is tedious and error-prone.
- **Limited Accessibility:** Team members who cannot attend meetings struggle to catch up. Reading through lengthy transcripts or recordings is time-consuming and doesn't highlight what matters most.
- **Lack of Context:** When teams need to recall decisions from previous meetings or track recurring issues, they must manually search through notes and transcripts, which is inefficient and frustrating.

Target Users

Our primary users are project managers, team leads, and professionals in remote-first organizations who conduct multiple meetings daily. Secondary users include consultants, sales teams, and executive assistants who need to maintain comprehensive meeting records for clients and stakeholders.

Proposed Solution

We propose building a **Meeting Insight Generator** - an end-to-end audio-to-insight pipeline that automatically processes meeting recordings and delivers structured intelligence. The system will go beyond simple transcription to provide meaningful analysis and actionable outputs.

- **Accurate Transcription:** Convert audio recordings into text with speaker identification, providing a reliable foundation for further analysis.
- **Intelligent Topic Segmentation:** Automatically identify when the conversation shifts from one topic to another, creating an organized structure that mirrors how the meeting actually flowed.
- **Decision Extraction:** Identify and highlight key decisions made during the meeting, including what was decided and who was involved, so nothing important gets lost.
- **Action Item Generation:** Automatically detect tasks that need to be completed, identify who is responsible, and note any mentioned deadlines.
- **Sentiment Analysis:** Analyze the emotional tone of discussions to understand team engagement, identify points of agreement or conflict, and gauge overall meeting effectiveness.
- **Historical Search:** Allow users to search across past meetings to find when specific topics were discussed or decisions were made, providing valuable context for ongoing work.

Why Does This Project Matter?

Current solutions like Otter.ai or Fireflies provide basic transcription services, but they don't deliver the depth of analysis that teams actually need. They answer "what was said" but not "what does it mean" or "what should happen next."

Our system fills this gap by providing structured intelligence rather than raw data. It saves significant time, ensures accountability through clear action item tracking, and makes meeting content searchable and useful long after the meeting ends. This transforms meetings from time sinks into valuable, documented decision-making sessions.

Technical Approach

Our solution uses a multi-agent architecture where specialized AI agents work together to analyze different aspects of the meeting:

- **Transcription Stage:** We'll use OpenAI's Whisper model to convert audio to text with high accuracy. This provides speaker identification and timestamps, creating a structured foundation for analysis.
- **Agent-Based Processing:** Multiple specialized agents will process the transcript in parallel:
 - A topic segmentation agent identifies natural conversation boundaries
 - A decision extraction agent finds and documents key decisions
 - An action item agent identifies tasks and responsibilities
 - A sentiment analysis agent evaluates emotional tone and engagement
 - A summary agent synthesizes everything into coherent reports
- **Knowledge Retrieval System:** We'll implement Retrieval Augmented Generation (RAG) using a vector database to store and search historical meeting data. This allows users to query past meetings and find relevant information quickly.

Technology Stack

We'll build this using modern, production-ready frameworks:

- **Agent Framework:** LangChain for orchestrating multiple AI agents
- **Language Models:** Mistral-7 for analysis tasks, with LLaMA 3.1 as an alternative
- **Speech Recognition:** Whisper for accurate transcription
- **Vector Database:** FAISS for storing and searching meeting embeddings
- **Backend:** FastAPI for building a robust REST API
- **Storage:** PostgreSQL for meeting metadata and structured data

The entire system will be containerized using Docker, making it easy to deploy and scale.

Project Scope

1. What We Will Build

- **Core Pipeline:** A complete working system that takes audio input and produces structured insights, including transcripts, topic segments, decisions, action items, and sentiment analysis.
- **REST API:** A well-documented API that allows users to upload recordings, check processing status, and retrieve insights in various formats.
- **RAG Implementation:** A vector database system that enables semantic search across historical meetings.
- **Multi-Agent Workflow:** A coordinated system where specialized agents handle different analysis tasks, demonstrating advanced AI orchestration.
- **Deployment Package:** A fully containerized application with Docker Compose that can be deployed locally, including all necessary configuration and documentation.
- **Testing Suite:** Comprehensive unit and integration tests ensuring reliability and catching errors before they affect users.

2. What We Will Not Build

We will not create a full web interface or mobile app - our focus is on the backend intelligence and API. We'll provide a simple demo interface for testing, but not a polished end-user application.

We will not implement real-time transcription during live meetings - our system processes recordings after meetings conclude.

We will not build custom speech recognition models - we'll use existing state-of-the-art tools like Whisper.

What will success look like?

We'll measure our project's success through:

- **Transcription Quality:** Achieving over 90% accuracy on meeting transcripts.
- **Extraction Precision:** Correctly identifying at least 85% of action items and decisions.
- **Processing Efficiency:** Completing analysis in less than twice the meeting duration.

- **System Reliability:** Handling errors gracefully and providing clear feedback.

Feasibility & Risk Assessment

1. Why This is Achievable

All the core technologies we need are mature and well-documented. Whisper is proven for transcription, LangChain provides robust agent frameworks, and vector databases are standard tools. Our team has the necessary Python and NLP knowledge to implement these components.

2. Potential Challenges

- **Processing Speed:** Running Whisper locally might be slow. We can mitigate this by using cloud APIs like Deepgram or, if needed, optimized models.
- **Cost Management:** LLM API calls can become expensive. We'll implement caching and consider using open-source models like LLaMA for some tasks.
- **Accuracy Limitations:** Speaker diarization and action item extraction won't be perfect. We'll be transparent about limitations and focus on making the system useful even if not flawless.

Work Distribution

Areeba Riaz	Transcription pipeline (Whisper), audio preprocessing, speaker diarization
Hafsa Imtiaz	Action-item extraction module (tasks, owners, deadlines)
Umer Farooq	Decision extraction module (LLM-based)
Abeer Jawad	RAG system (FAISS), embedding generation, historical search
Zayyam Hassan	Sentiment analysis + overall meeting summary generation
Muhammad Rayyan	Topic segmentation module + conversation structuring

Conclusion

The Meeting Insight Generator addresses a real problem faced by modern teams: extracting value from the hours spent in meetings. By combining speech recognition, multi-agent AI systems, and retrieval augmented generation, we'll create a production-quality tool that demonstrates both technical sophistication and practical utility. This project showcases the power of modern NLP and agentic AI while delivering something genuinely useful for real-world applications.