ISSN: 3139-0862

An AI-Powered Meeting Intelligence System with Persistent Memory Cross-Meeting

Dhanajay Raut[#1], Aman Pandey[#2], Juhi Tawade[*3], Saniya Shaikh[#4], Soniya Pawar[#5]

[#1]*Professor,* [#2,3,4,5]*Students of Watumull Institute of Engineering & Technology, Ulhasnagar-421003*

[1]tawadejuhi2211@gmail.com, [2]pandeyaman995@gmail.com , [3]saniyaashaikh2005@gmail.com

Abstract—
The meetings play an important role in colleges, offices. The meeting recording contains various useful information such as decisions, deadlines, owners responsible for the deadlines. The Transcript or summary of the meeting does not include all essential tasks and deadlines. The proposed system is a web-based meeting intelligence system that converts recordings into timestamped transcripts, summaries and structured records of tasks, deadlines and decision. The system maintains the persistent memory across meeting recordings and display relationship between the multiple meetings. The proposed paper display methodology and proposed system architecture including data model, candidate technology stacks, cross meeting matching.

Keywords ---- Meeting Intelligence, Task Tracking, semantic retrieval, speech recognition, Speaker diarization.

I. INTRODUCTION

Meeting contains various ideas, decision, assign tasks, and deadlines. However important information is often shared through normal conversations, making it difficult to remember multiple tasks, responsibilities and monitors task completions. Studying and reviewing long meeting recording can be difficult also meeting summaries fails to capture all tasks, deadlines, and changes. The proposed system aims to convert meetings into structured and useful information. It uses Speech recognition to converts audio into text and NLP to identify tasks, decision, responsible person, deadlines, task status, and also reasons.

A key feature of the system is Cross Meeting Task Tracking. Taks may be discussed and modified across multiple meetings can be stored in form of knowledge graph. The system maintains task records and uses semantic similarity, contextual information, task names, and related entities to connect the discussion. If the similarities are found than the existing records are updated. The system combines several technologies, including speech recognition, speaker diarization, NLP, semantic similarity, database management and information retrieval.

Unlike basic transcription and summarization tools which only provides summary of the meeting recording, the proposed system focuses on maintaining the continuity throughout multiple meetings. Its performance can evaluate using transcription accuracy, task extraction precision and recall, deadline extraction accuracy, cross-meeting matching performance.

A. PROBLEM STATEMENT

The traditional meeting management system mainly depends on manual notes-taking, written summaries, and separate task tracking tools. But these methods do not maintain a clear connection between information discussed in different meetings. For example, a task assigned during one meeting, discussed again in different meeting and updated in a later meeting. If the information is stored separately, users may find it difficult to identify the tasks status and understand the changes made to the original plan.

Another challenge is identifying tasks and the responsible person. The commitment made for completion of the task. A system that relies on meeting summaries and keywords matching fails to connect related discussion correctly. The problem is therefore to develop a system that can extract information, link related tasks across multiple meetings and maintain their history.

B. NEED FOR THE PROPOSED SYSTEM

There is a need for a system that can reduce the manual effort required to review meeting recordings and track tasks. A system should automatically identify important information including tasks, decisions, assigned persons, deadlines and progress updates. The system should preserve earlier task information while updating the current status when new evidence is available. It can help users to identify pending task, approaching deadlines, possible delay and unresolved issues.

C. OVERVIEW OF PROPOSED SYSTEM

The proposed system is a persistent cross meeting intelligence system that processes meetings recordings and converts conversation into structured and searchable information. It combines various technologies such as NLP, speech recognitions, semantic similarity, database management and information retrieval techniques to support meeting understanding and task tracking.

The Process of proposed system is to accept the meeting recording. The audio is preprocessed and the noise, unwanted sound are removes. The extracted audio is used for further processing. Speech recognition technique is used to convert

ISSN: 3139-0862

speech to text and speaker diarization is used for identifying the speaker. The transcript is analyzed to exact important task, deadlines, responsible person.

After extracting useful and essential information the system compares newly identified task with previously stored task records. It uses semantic similarity and contextual details to determine whether a new task is related to existing task or represent new task.

D. OBJECTIVES

- To convert meeting audio and videos recording into text using speech recognition technique.
- To connect related task mention across multiple meetings using semantic and contextual; information.
- To identify speaker, deadline, person responsible for task completions.
- To identify pending task, overdue activities, using task status and deadline information.

II. PROPOSED SYSTEM ARCHITECTURE

A. PROPOSED ARCHITECTURE

The proposed system architecture is designed to convert meeting recording into structured information and maintain the task continuity across multiple meetings. The system identifies the speaker, important deadlines, tasks and the person responsible for those tasks. The architecture consists of several connected module, including input modules, audio preprocessing, speech recognitions, speaker diarization, database modules.

Each module performs a specific function and passes its output to next step. The design is modular making it easy to debug, test, maintain and improve. It also allows individual components to be evaluated separately before assessing the performance.

B. PROPOSED SYSTEM FLOWCHART

The proposed architecture follows a modular, multi-stage processing approach. It can be divided into five main layers:

1. Input And Preprocessing Layer:
   The input layer accepts meetings recordings. These recordings may contain discussions involving multiple participants, different topics, and varying audio quality.
2. Speech And Language Processing Layer:
   This layer convert spoken conversation into text and extract useful information. The speech recognition module generates textual representation of the meeting. A speaker diarization module can identify the segments associated with different speakers.
3. Cross-Meeting Intelligence Layer:
   The cross-meeting intelligence layer is the central component of the proposed architecture. It connects information extracted from the current meetings with task records created from earlier meetings.
4. Storage And Task Management Layer:
   The storage layer maintains meeting information, task records, supporting evidences and historical changes. This make it possible to retrieve the latest task information without losing the original information.
5. Application And Information Retrieval Layer:
   The application layer allows users to interact with the processed meeting information. Users may search for task, view their current status, inspect previous changes, identify responsible person.

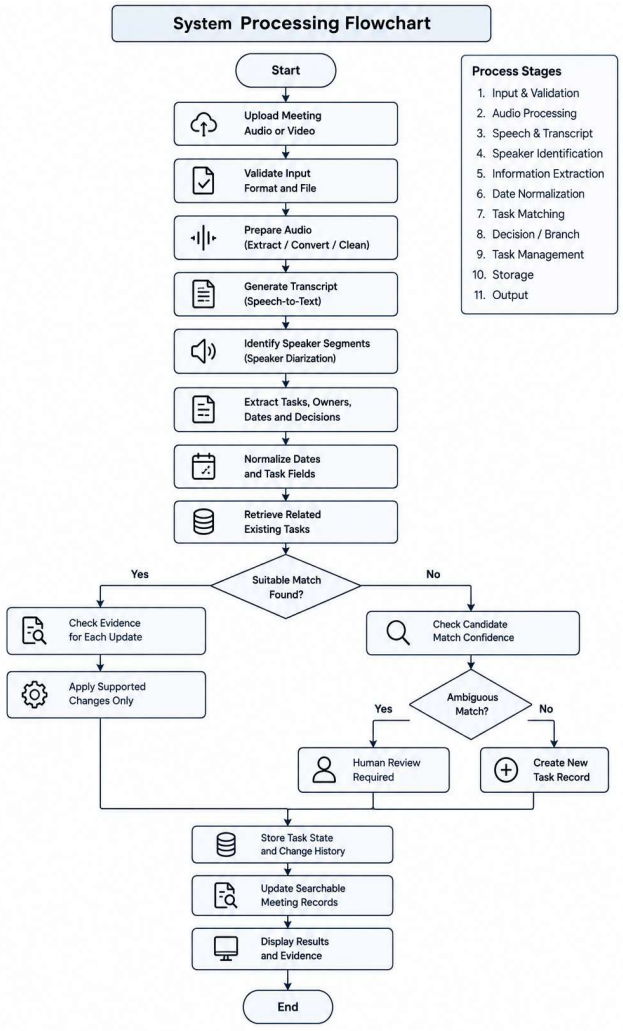


Figure 2.1 : System flowchart

C. PROPOSED TECHNOLOGY STACK

Step	Process	Algorithm / Technology	Output
1	Upload meeting recording	File Upload	Audio/Video
2	Extract audio	FFmpeg , VDA	Audio file
3	Convert speech to text	Whisper (ASR)	Transcript + timestamps
4	Identify speakers	Speaker Diarization (pyannote.audio)	Speaker-labelled transcript

ISSN: 3139-0862

5	Extract important information	NLP + NER + LLM	People, topics, tasks, decisions
6	Generate meeting summary	Transformer/LLM	Short summary
7	Extract deadlines	NER + Date Parser	Task + deadline
8	Generate embeddings	Sentence Transformer	Vector representation
9	Find previous related information	Vector Similarity Search	Related meetings/tasks
10	Match new and old information	Cosine Similarity + Entity Matching	Relationship score
11	Maintain meeting memory	Knowledge Graph	Connected meeting history
12	Detect task dependencies	Graph Traversal	Dependency relationships
13	Detect potential delays	Rule-based / Risk Scoring	Risk status
14	Monitor deadlines	Scheduler	Upcoming/over due tasks
15	Send notification	Event/Rule-based notification	Reminder/alert
16	Answer questions	RAG + Vector Search + LLM	Context-aware answer

Table 2.1: Technology stack

III. EXPECTED OUTCOME

The proposed system is expected to provide an organized approach for processing meeting recordings and managing task information across multiple meetings. By combining multiple modules such as NLP, speech recognitions, semantic similarity the system aims to convert unstructured meeting conversations into structured and searchable information.

1. Automated Meeting Transcription: The system is expected to convert meeting audio into text, making discussions easier to review without listening to the complete recording repeatedly.
2. Task And Deadlines Extractions: The systems aim to identify important tasks, deadlines, responsible person across different meetings.
3. Cross-Meeting Task Identification: The system identifies the similarities between multiple meetings using semantics similarity.
4. Persistent Task Management: The system maintains task records of different meetings and update their status when related new information is detected.
5. Task History Preservation: The system connects extracted information with relevant transcript segments so that user can verify the source of recorded task and decision.

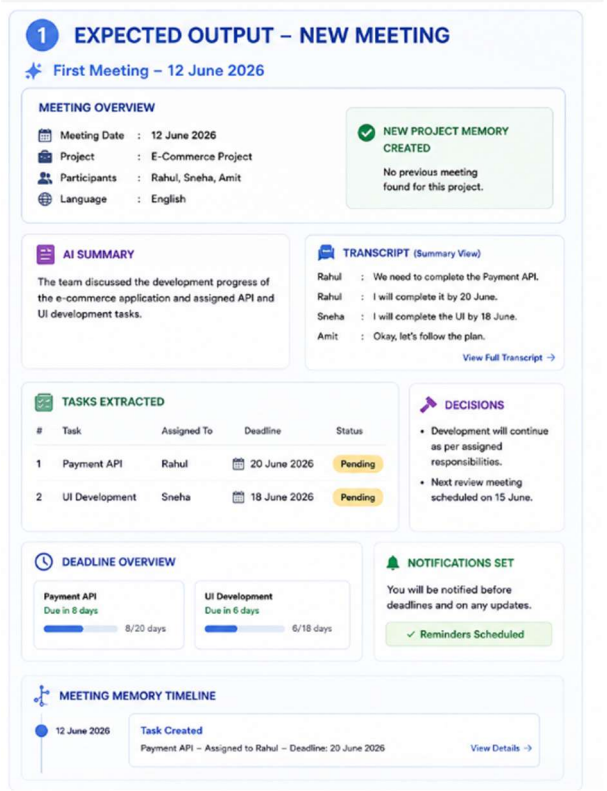


Figure 3.1: Expected output 1

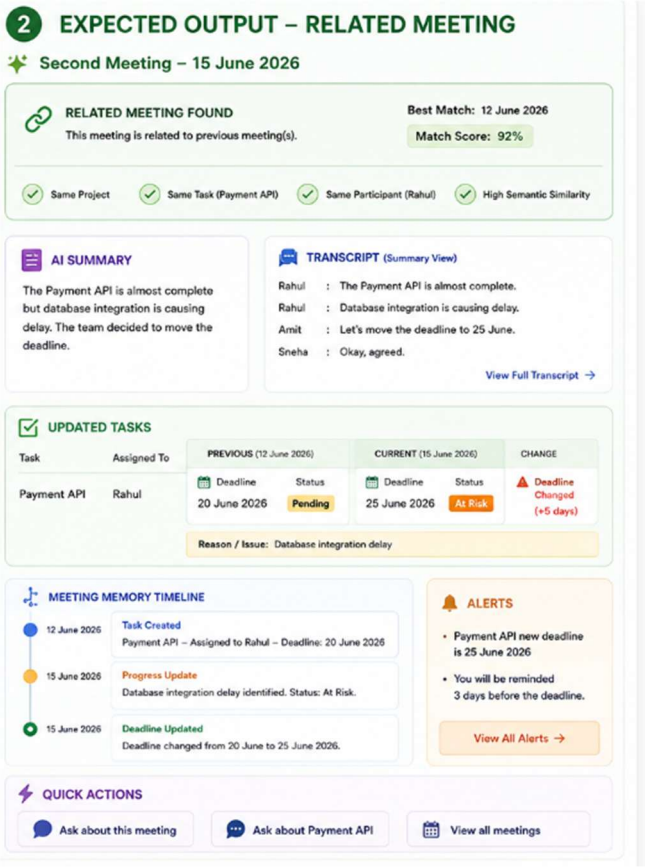


Figure 3.2: Expected output 2

ISSN: 3139-0862

IV. FUTURE SCOPE

1. Improved Speech Recognition:

The future work can focus on improving the accuracy of transcription in meetings with background noises, overlapping speaker conversations, diverse accents, and technical terminologies. The domain-specific vocabulary and suitable speech recognition models can help reduce transcription errors and improve meeting information extraction quality.

2. Improved Task Matching:

The future work can focus on enhancing the cross-meeting task matching by considering additional factors such as semantic similarity, task attributes, participant details, project dependencies, and historical updates. The task-matching thresholds can be optimized using the annotated meeting data. The human feedback can help the model learn to differentiate between existing and new tasks.

3. Advanced Task Dependency Management

The future scope can involve developing the system to manage the task's dependencies, prerequisites, and blockers. For instance, if the API development task depends upon the database development task, the system can highlight this dependency and inform the users about the delayed dependent tasks. The task dependency graph can help visualize and understand the project's status.

4. Intelligent Deadline and Risk Prediction

The future work can utilize the task's historical information, deadline modifications, unresolved blockers, and past due patterns to identify the at-risk tasks. The risk prediction models can be trained and evaluated to estimate the probability of meeting deadlines. This probability can be used to notify the users about the delayed tasks instead of presenting it as a fact.

5. Scalability and Performance Optimization

The system can be fine-tuned to handle hundreds of thousands of meeting recordings and serve multiple companies at once. The asynchronous data processing mechanism can be employed to reduce the long-processing times. The optimization can focus on database queries, candidate retrieval, and model inference acceleration using the appropriate cloud infrastructure. The system's scalability can be evaluated using processing time, memory consumption, retrieval latency, and concurrency tests.

V. CONCLUSION

The proposed system aims to address the issue of recording, organizing, and maintaining the information regarding meetings. Current practices require a significant effort to be spent on recording, reviewing, and tracking decisions, persons responsible for tasks, deadlines, status updates, and other meeting-related activities. This can be achieved by implementing the proposed solution that utilizes speech recognition, Natural Language Processing, semantic similarity, and persistent task management.

The system would allow extracting the relevant information from the audio recordings of the meetings and structuring it in the form of tasks, decisions, persons responsible, deadlines, and status updates. It would then attempt to match the extracted information with the previously recorded tasks to determine if any actions are continued from the previous meetings. Once a match is found with a sufficient level of confidence, the information can be used to update the relevant task while retaining the prior information and the history of changes. Furthermore, the proposed system would support evidence-based management of meeting-related information by associating it with the corresponding transcripts. This would allow verifying the information contained within the system against the audio recordings of the meetings, as well as distinguishing between confidently stated and possibly uncertain information. For instance, when a task is reported as delayed, but no new deadline was specified, the existing deadline should be retained.

Finally, the proposed architecture would allow searching through the past meetings for relevant information regarding tasks, deadlines, decisions, and responsible persons, as well as providing updates on the status of particular tasks. The modularity of the design would facilitate the development and testing of individual components, although the efficacy of the entire system would be dependent on the combined performance of its elements. Nevertheless, the efficacy of such an approach would have to be demonstrated through actual implementation and evaluation using the appropriate performance metrics.

REFERENCES

[1] M. Zhong, D. Yin, T. Yu, A. Zaidi, M. Mutuma, R. Jha, A. H. Awadallah, A. Celikyilmaz, Y. Liu, X. Qiu, and D. Radev, "QMSum: A new benchmark for query-based multi-domain meeting summarization," in *Proc. 2021 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 5905–5921, doi: 10.18653/v1/2021.naacl-main.472.

[2] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. 40th Int. Conf. Machine Learning*, vol. 202, 2023, pp. 28492–28518.

[3] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, M. Lavechin, D. Fustes, H. Titeux, W. Bouaziz, and M. Gill, "pyannote.audio: Neural building blocks for speaker diarization," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7124–7128.

[4] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. 2019 Conf. Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3982–3992, doi: 10.18653/v1/D19-1410.

[5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-T. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459–9474.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.