1. Title: AI Powered Meeting Summarizer & Action Item Generator .

2. Domain: Artificial Intelligence / Natural Language Processing (NLP)

3. Application

- Develop an AI-based application that processes meeting transcripts or text discussions.
- Automatically generate clear and concise summaries of meetings.
- Extract key action items and responsibilities from the discussion.
- Improve productivity for students, professionals, and organizations by reducing manual note-taking and saving time.

4. Hardware & Software Required

Hardware Requirements

- Processor: Intel i3 or higher
- RAM: 4 GB minimum
- Storage: 100 GB or more
- Internet connection

Software Requirements

- Operating System: Windows / Linux
- Programming Language: Python
- AI/NLP Libraries: TensorFlow or PyTorch
- Tool/IDE: VS Code or Jupyter Notebook

5. Input Required for Project

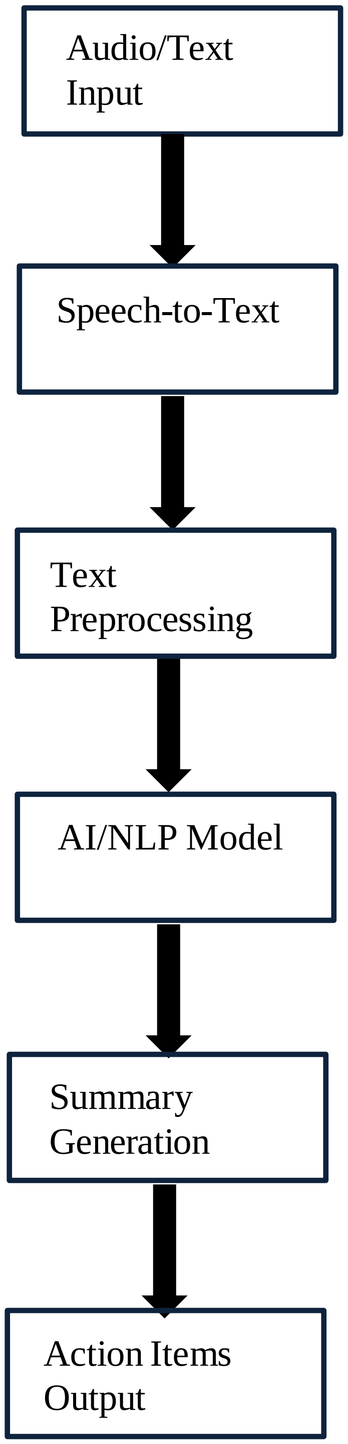
- Text input in the form of :
    - Meeting transcripts
    - Chat logs
    - Discussion notes

    (Audio processing is not included in Stage-I)

6. Objectives of Project

- To develop a system that automatically summarizes meeting transcripts.
- To use a Large Language Model (Ollama) for generating concise summaries.
- To extract important points and key decisions from meeting discussions.
- To identify action items assigned during meetings.

7. Block Diagram of System



8. Expected Algorithm / Processing

1. Accept meeting text as input
2. Perform text cleaning (removal of stopwords, punctuation)
3. Apply tokenization and sentence segmentation
4. Use NLP techniques to identify important sentences
5. Generate a summarized version of the meeting
6. Detect task-related sentences as action items
7. Display the generated summary and action items

9. Expected Output from Project

- A concise summary of the meeting discussion
- A list of extracted action items
- Output displayed in readable text format

10. Abstract

In today's fast-paced professional and academic environments, meetings play a vital role in collaboration, decision-making, and project management. Organizations conduct frequent online and offline meetings to discuss plans, assign tasks, and evaluate progress. However, these meetings often generate large volumes of unstructured information, making it difficult for participants to manually record important discussion points, decisions, and action items. Traditional methods of writing meeting minutes are time-consuming, prone to human error, and often result in incomplete or inconsistent documentation.

To address these challenges, this project presents an AI-powered Meeting Summarization and Action Item Extraction System using Natural Language Processing (NLP). The primary objective of this system is to automatically convert meeting transcripts into concise summaries while identifying key tasks, responsibilities, and important decisions. By automating this process, the system significantly reduces manual effort and improves productivity.

The proposed system operates in multiple stages. In Stage-I, the focus is on processing text-based meeting transcripts. Raw meeting data is first passed through preprocessing steps such as tokenization, stop-word removal, lemmatization, and normalization to ensure clean and structured input. After preprocessing, extractive summarization techniques are applied to identify the most relevant sentences based on keyword frequency, sentence position, and semantic importance. These selected sentences are combined to generate a clear and meaningful meeting summary.

Along with summarization, the system also extracts action items using rule-based methods and pattern recognition. Sentences containing task-oriented keywords such as "assign," "complete," "submit," or "follow up" are analyzed to detect actionable information. This enables the system to automatically list tasks discussed during meetings, helping participants track responsibilities more effectively.

The project emphasizes understanding core NLP concepts including text preprocessing, feature extraction, and summary generation. A modular system architecture is designed to allow easy enhancement in future phases. The current implementation focuses on extractive summarization, but the framework supports future integration of machine learning and deep learning models for more accurate and context-aware summaries.

This system offers several advantages over traditional manual note-taking approaches. It ensures consistency in meeting documentation, minimizes information loss, and saves valuable time for professionals and students. By providing structured summaries and action lists, the system improves clarity and accountability within teams. It also supports better decision-making by allowing users to quickly review meeting outcomes.

The application of this project is wide-ranging, including corporate environments, educational institutions, online classrooms, project management teams, and remote collaboration platforms. It is especially beneficial for organizations conducting frequent meetings, where maintaining accurate records becomes challenging.

From a technical perspective, the system is implemented using Python and NLP libraries for text processing and analysis. The workflow includes data input, preprocessing, summarization, action item extraction, and output generation. The design prioritizes simplicity, scalability, and ease of integration with future components such as speech-to-text modules and task management dashboards.

Future enhancements of this project include incorporating real-time speech recognition, abstractive summarization using deep learning models, speaker identification, deadline detection, and automatic task assignment. Additional features such as meeting analytics, sentiment analysis, and visualization dashboards can further enhance the system's capabilities. These improvements will transform the application into a complete intelligent meeting assistant.

In conclusion, this project demonstrates how artificial intelligence can streamline traditional meeting workflows by automating summary generation and action item extraction. By leveraging NLP techniques, the proposed system reduces manual effort, improves documentation quality, and enhances overall productivity. The project serves as a strong foundation for building advanced AI-based meeting management solutions and highlights the growing importance of intelligent automation in modern digital environments.

✅ Sample Input 1 — Project Planning Meeting

‹/› Code

```
Good morning everyone. Today we discussed the AI Meeting Summarizer project.
Harshal explained the system architecture and overall workflow.
Diksha will collect meeting transcript datasets by Wednesday.
Tushar is assigned frontend development and will complete UI design by Friday.
We also decided to use Python and NLP libraries for implementation.
Next meeting is scheduled on Monday to review progress.
```

📌 Extracted Action Items

Person	Task	Deadline
Diksha	Collect meeting transcript datasets	Wednesday
Tushar	Complete UI design	Friday

Literature Survey :

1)

International Journal of Technology Management & Humanities (IJTMH)
e-ISSN: 2454 – 566X, Volume 7, Issue 4, (December 2021), www.ijtmh.com

AI-Driven Real-Time Summarization and Action Item Extraction in Video Conferencing Platforms

(Authors Details)
Sravan Komar Reddy Pullamma
Affiliation: PMP, USA.
email: psravanreddy@gmail.com
(Co Author)
Dr. Sujeevan Kumar Agir
CEO, Aarya Digiverse Private Limited
Jayaprakash Narayan College of Engineering, India.
Email.: sujeevankumaragir@gmail.com

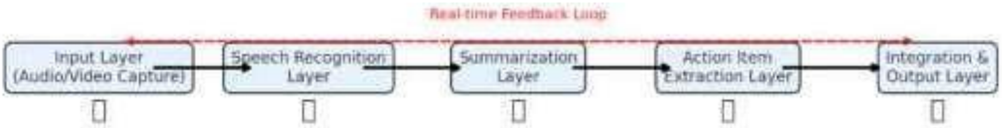


Fig 1: The flow diagram showing the process from Input → Output with arrows, feedback loops, and icons for clarity.

Methodology

The methodology for developing the AI-driven real-time summarization and action item extraction framework is structured into five core phases: data collection, preprocessing, model design, system integration, and evaluation. Each phase is designed to ensure the framework can operate efficiently within live video conferencing environments while maintaining accuracy and scalability.

Data Collection

Meeting transcripts and audio datasets were collected from publicly available meeting corpora, open-domain conversational datasets, and simulated enterprise video conference sessions. The datasets were chosen to capture multi-speaker interactions, conversational overlap, and domain diversity.

Preprocessing

The raw audio streams were processed using speech-to-text pipelines to generate transcripts. Preprocessing steps included speaker diarization, removal of filler words, normalization of timestamps, and segmentation of conversational turns. Stop words were retained selectively to preserve contextual meaning in action item detection.

Model Design

The framework consists of two primary modules:

- **Summarization Module**: Implements a hybrid of extractive and abstractive approaches. Extractive summarization identifies key sentences based on importance scores, while abstractive summarization employs transformer-based architectures to generate human-like summaries.
- **Action Item Extraction Module**: Uses deep learning classifiers with attention mechanisms to identify commitments, tasks, and decisions. Contextual embeddings capture dependencies across multi-speaker dialogues, ensuring accurate task assignment.

System Integration

The trained models were integrated into a real-time pipeline. The pipeline continuously processes incoming audio streams, transcribes them, applies summarization models, and extracts action items dynamically. An API interface allows seamless integration with popular video conferencing platforms such as Zoom and Microsoft Teams.

System Integration

The trained models were integrated into a real-time pipeline. The pipeline continuously processes incoming audio streams, transcribes them, applies summarization models, and extracts action items dynamically. An API interface allows seamless integration with popular video conferencing platforms such as Zoom and Microsoft Teams.

Evaluation Strategy

The performance of the system was evaluated using benchmark metrics for both summarization and action item extraction. Summarization quality was assessed using ROUGE and BLEU scores, while action item extraction was measured with precision, recall, and F1-scores. Latency and throughput were also evaluated to ensure feasibility in real-time deployment.

Table 1: Methodological Framework for AI-Driven Summarization and Action Item Extraction

Phase	Description	Techniques/Tools Used	Output Generated
Data Collection	Gathered meeting transcripts and audio datasets from multiple sources	Public corpora, enterprise simulations	Raw audio and text data

2)

Meeting Summarizer and Plan of Action Generator using Natural Language Processing

Gajender Kumar[a] , Sahej Kumar Seth[b], Manu Kant Sharma[c], Shivay Garg[d], Vaibhav Sharma[e]

gk201719nit@gmail.com[a] , sahej.seth@gmail.com[b], manukant2002@gmail.com[c], shivaygarg65@gmail.com[d], vaibhavsh1500@gmail.com[e],

[a] *Department of Computer Science & Engineering, SRM Institute of Science and Technology, Modinagar,* Ghaziabad, Uttar Pradesh, India

[b]*Department of Computer Science & Engineering, SRM Institute of Science and Technology, Modinagar,* Ghaziabad, Uttar Pradesh, India

[c]*Department of Computer Science & Engineering, SRM Institute of Science and Technology, Modinagar,* Ghaziabad, Uttar Pradesh, India

[d]*Department of Computer Science & Engineering, SRM Institute of Science and Technology, Modinagar,* Ghaziabad, Uttar Pradesh, India

[e]*Department of Computer Science & Engineering, SRM Institute of Science and Technology, Modinagar,* Ghaziabad, Uttar Pradesh, India

Abstract:

There is a growing need for meeting summaries in the virtual environment. Transcripts of meeting, such as those from Microsoft-Teams, Google-Meet, and Jio-meet etc are readily available for virtual meetings. However, lengthy transcripts can be challenging to read, making the creation of summary transcripts essential for obtaining a to the point documents. To highlight the key conclusions of a meeting there is a need of a summary. The primary aim of our research is to generate a summary from the available transcript during meeting. We achieved through the systematic integration of various natural language processing (NLP) techniques, including meeting summarization, which have been developed separately and tested offline with standard datasets. This research focuses on Microsoft Teams transcripts. We tested this research with Term-Frequency-Inverse and Document Frequency (TFI-DF) method, also, the Page-Rank algorithm utilized as NLP tools in this research. We also automate our meetings on a specific date and time with the help of scheduler.

Keywords— Natural Language Processing, Summarization, Transcript, Term-Frequency-Inverse-Document-Frequency, Page-Rank

I. Introduction

Meetings are a fundamental way for employees to engage and communicate. They provide an opportunity for idea exchange and in-depth to communicate. Summaries of meeting are vital as professionals or academicians distill the essential information from discussions into a brief format. Transcripts from meetings are easily accessible through platforms like Microsoft Teams and Google Meet. Generally, reading and understanding the entire transcript can be time-consuming. Therefore, summaries are crucial since readers typically seek only the relevant context of discussions. Consequently, a summarizer of meeting is proposed to assist in summarizing available transcripts.

Summarization of document differs from extraction and topic modeling of key phrases in that it emphasizes the document's content. In this context, the final output remains a document, albeit a much shorter one, depending on the desired length of the summary. This is similar to an abstract of a research paper. Here, the aim of automatically scanning documents is to extract them without human intervention, unless it is solely for computer code operations. By analyzing the text content and background, mathematical and computational models work to facilitate automation in text summarization tasks. By analyzing the content and status of documents, mathematical and computer models help automate powerful systems and document collections. In this context, there are two main follow-ups to the text: residuals and residual formulas. Residual reviews use mathematical and computer techniques such as SVD, which extracts a significant portion of the essential content of the original document that contains the most important information and influences the format of the document.

This material may contain words, phrases, or entire sentences. The outcome of this method is a concise summary of actions based on the text selected from the original document, and hence it is referred to as extractive-based. Abstractive methods are always more complex and advanced. They utilize representational techniques and natural language generation (NLG), allowing a computer to independently generate text and summaries using the foundational infrastructure of information and social representations that are akin to human capabilities. These strategies can be effectively implemented using deep learning, but they require a significant amount of data and computational resources. Its study lesson focuses on compiling the lesson and holding meetings, particularly considering the publications of Microsoft groups and implementing appropriate procedural methods such as text line and TF-IDF.

The paper is organized into five sections: Section I, Introduction; Section II, Interpretation; Section III, Literature Study; Section IV discusses projected methods; Section VI discusses the applied conditions. The sixth part summarizes the research, and in the final part, we discuss the general conclusions.

II. Motivation

With the growth resilience of digital communication, online meetings have become more popular among academicians and professionals, due to shortage of times and the convenience of virtual connectivity. There are lots of online platform available today, such as Microsoft Team and Google Meet to meet virtually and discuss the important things instead of physical mode interaction. When we meet online, there are lots of challenges that a common person or academicians faced during conversation of important points, so, when we conduct a meeting or attend a meeting lots of things missed during that. Our proposed system stores all the information and maintains the document whatever discussed during the meeting in text form and summarize in an organized way. After that the system sends back those documents or all the participants, after the meeting.

A. *Problem Statement*

Meetings are a significant aspect of any organization's activities, whether they are online or in person. On the other hand, organizing meetings and creating their summary often becomes cumbersome as it requires significant effort from individuals. A summary typically gives us a brief insight into the meeting rather than having to go through lengthy records. Additionally, individuals in managerial roles across all organizations need to report the meeting summary to their colleagues. When one has to manually summarize all the meeting proceedings, it becomes a challenge. Therefore, there is a need for an automated system to organize the meeting scorecard and summary. This saves effort and time.

III. Literature Servey

Virtual meeting is a tool that allows users to automatically record, transcribe, synthesize, and organize meeting content in person. ASR transcription, improving transcription, and meeting synthesis are three processes. ASR transcription refers to converting sound recordings made by each participant with their device into text, which includes hybrid ASR models. Improving transcription is used for separation of streams and identification of speakers based on the temperament of the speaker's voice for partitioning and classification. WS neuro-synthesis uses uncontrolled synthesis models. The position of words between NLTK models is mentioned, a comparison of T5 models is presented, resulting in a concise synthesis. Extraction synthesis for NLTK models uses sentence ordering algorithms; the word positional model uses pre-trained GloVe embeddings.
In [3], A system has been created that integrates the selection of article titles and words in the template with the return of quotation marks, thus providing better readable comments. It has 10 parts: blowing up the text channel, monitoring the presentation of the template, preparing various documents, designing the title, clarifying the subject, selecting keywords, creating comments, returning comments, noting comments, and analyzing comments. The authors of the article introduced three techniques: TF-IDF, TextRank, and Latent Dirichlet Allocation (LDA). The author presented a comparison of three techniques: TF-IDF, TextRank, and Latent Dirichlet Allocation (LDA). It is reasonable to argue that if an error is inadvertently applied in the text and aligns with the characteristics of each adopted method, TF-IDF and LDA will be surpassed.
In [5], the Maximum Marginal Relevance (MMR) technique was used for identifying significant phrases. In [6], an NLP system has been developed that utilizes powerful AI algorithms to extract metadata from transcripts. Everyday discussions are being clarified, necessary ideas are being outlined, extracting facts, eliminating noises, generally explaining the steps, determining the levels of steps, and labeling documents repeatedly are part of the technique. The research topic for the method of extracting key terms is the texts of English news [7]. TF-IDF and text ranking algorithms have been linked to each other. A word graph model, word frequency, and document frequency have been constructed. The performance of the algorithms has been measured at the level of recovery rate, accuracy rate, and average macro value.
The performance parameters and conclusions regarding the impact and results related to TF-IDF and text ranking have been combined with standard techniques. [8] It is related to BERT, which stands for Bidirectional Encoder Representations from Transformers. It consists of four stages: pre-training and fine-tuning. In this study, a neural maintenance model emerging from Jharkhand has been proposed, which is based on an underlying model. Then, a model for the combination of sentence structures is described. The underlying model utilizes this, and ultimately, the underlying model is proposed. A modified approach has been developed to extract the summary of the content text. When calculating similarity, a weighted cosine compromise is used instead of standard TF-IDF, which yields interesting results during news testing. After careful examination of some journals and research papers, the combination of TF-IDF and text ranking produces an effective summary.

IV. PROPOSED SYSTEM

A. *Proposed Architecture*

"Figure 1." NLP is used to facilitate meetings that focus on proposed architectural power. The system accepts documents in transcript form and provides a summary document. Microsoft Teams transcripts are considered for this project. The timestamps of the transcripts are not included, and each sentence is adjusted for each case. In addition, the text is segmented into sentences, which are then used for other applications. The pre-processing standardizes the text and removes stop words.
The matrix of word features is created from the document - a word repetition matrix - using reverse document duplication (TF-IDF). The frequency of each word will be calculated: word repetitions show how many times a word appears in that particular document. The repetition in opposite documents indicates how common or rare the word is. This can be estimated by dividing the total number of documents by the number of documents containing that word. The TF-IDF score is obtained by multiplying TF and IDF. The higher the score of the word, the more important it is in the document. The similarity matrix of documents is created by multiplying and inverting the word feature matrix. A graph of document similarity is created. These sentences are used as peak points, and the similarities between each pair of documents are viewed as weights or degree coefficients and entered into the ranking page algorithm.
Each sentence's score is calculated. The top sentences are provided as output. "Figure 2" shows the flow chart of the proposed model. The flow chart starts with the input transcript document. Sentence extraction and tokenization are performed for further use of the sentences. Then, TF-IDF is used to create the document similarity matrix. This is fed into the TextRank algorithm which provides a ranking for each sentence. The top-ranked sentences are then used to prepare the output summary document.

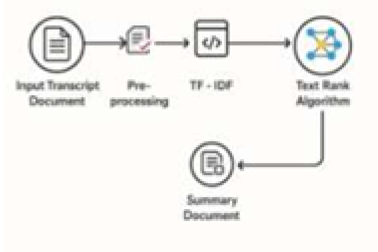


Figure 1. Architecture of the written system.

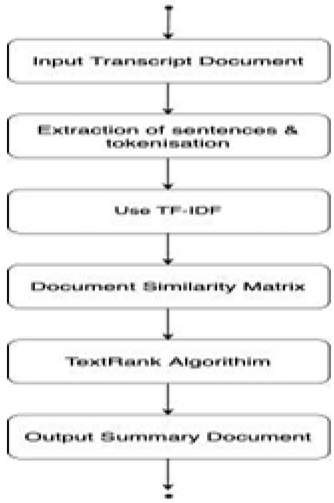


Figure 2. Diagram showing the proposed model.

The text is marked using the nltk method. Stop words will be removed, and stemming will be applied to the words in the document. This is standard. Thus, general sentences can be searched. The tf-idf vectorizer from the sklearn module is included. Through this, the document encoding is prepared. The DT matrix is obtained by transposing the inverse document frequency. The total number of sentences is prepared for the best based on the number of inner sentences after pre-processing. If the number of sentences exceeds 30, the maximum number of sentences in the best case is 20% of the total sentences in the input text. Otherwise, it is considered to be 30%. The similarity matrix can be viewed considering the matrix determinant and its reverse. The PageRank algorithm is derived from the similarity graph. It ranks each sentence based on its importance. Based on the ranking, the top N sentences will be assigned as output.

B. *Set of Steps for Text Rank*

TextRank is a graph-based ranking algorithm that has been successfully used in citation analysis, similar to Google's PageRank algorithm. It can also be utilized in text processing, such as finding the most relevant sentences in text and extracting keywords. Keyword extraction, automatic text summarization, and sentence ranking are all common uses of text ranking. To find the most relevant sentences in the text, a graph is constructed, where each sentence is a milestone in the document, and edges connect sentences based on overlapping content, i.e., the number of words shared between two sentences is measured. Sentences are fed into the PageRank algorithm, which identifies the most important sentences based on the network of sentences. When creating a summary, only the most important sentences can be extracted. The TextRank algorithm builds a network of words to find related keywords. This research is conducted through the study of relationships between words. If two words frequently occur near each other in the text, there is a relation between them. If two words are similar, that relation continues to grow. A page rank is used to determine the meaning of each word. The third most important part of the process is the preservation and maintenance of properties. Similarly, the number of words in a sentence is determined by the count of words in that sentence. Figure 3 shows the image.

C. *Term Frequency-Inverse-Document-Frequency (TF-IDF)*

The TF-IDF technique has been developed for finding and processing documents. TF-IDF is a statistical parameter that evaluates relevant words for a document within a collection of documents. The number of times a word appears in a document increases proportionally. The count of words appearing in the document is used to determine the total as its specified known form. The frequency of a word or its irregularity is determined by an inverse measure of the incorporation of this word in a group of documents. The term TF*IDF contributes with the multiplication of the TF and IDF ratings. The larger the weight, the rarer the word will be in the document.

D. *Set of Steps for Meeting Summarization Using NLP*

Step 1: Remove the timestamp from the speaker selection button text and split the sentence. Divide the text into separate sentences, linking each sentence to its corresponding speaker using the contextual indicator or speaker ID. Step 2: Prepare the text, apply the text normalization method which includes converting it to lowercase and removing punctuation. Normalize or stem word forms. Eliminate stop words and any unnecessary symbols. Step 3: Develop a term frequency-inverse document frequency (TF-IDF) matrix to represent the text data at the sentence level to numerically extract features via TF-IDF. Step 4: Calculate document similarity, build a sentence similarity matrix by computing the cosine similarity between the TF-IDF vectors. Step 5: Create the similarity graph, which generates an undirected weighted graph where the nodes represent the sentences and indicate similarity scores derived from the adjacency matrix. Step 6: Apply the TextRank algorithm. Execute the TextRank algorithm in the graph to determine the importance of each sentence based on their similarity. Step 7: Evaluate and select sentences. Sort all sentences according to their TextRank results. Select the highest-ranking sentences to create the meeting conclusion. Step 8: Prepare a summary. Arrange the selected sentences in a logical sequence to maintain context and improve readability.

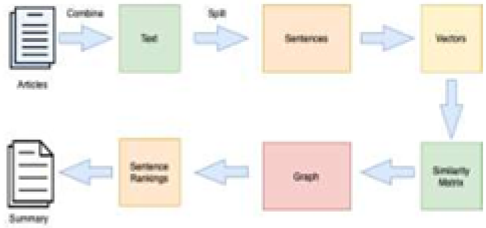


Fig. 3. Text Rank Algorithm for summarization.

V. *Implementation of the work*

Customers are prepared to retrieve important documents while they demand evidence, and the journey begins for drainage, sewage, and sewerage. Consider two losses in .txt and .doc. The .txt format will be used to transfer paper files.

A. *Manual Evaluation*

The proposed methods have been manually tested by various users of the education and software industry and tested for content and readability. The goal was to see if the summary is generally considered valuable for cases of personal use. The evaluation was subjective, and the process was informal. User (customer) feedback was relied upon and they were given full control over their level of evaluation and satisfaction. Answers for queries that are both generic and particular to the use cases are primarily required during user assessments. Some instances of the two types are given below.

General: Does the summary help users better understand the content of the copy compared to the main copy?

1. Does the summary include any additional issues (themes) besides the main topic? If yes, how many?
2. If a summary is available, how does our summary compare with the manual summary?

The method and results are improved by including the opinions of each user.

Table I: Appendix A: Generated Transcript using Meeting Summarizer Model

Time	Speaker	Dialogue
00:01:04.220 → 00:01:08.220	Rohit	Hi Mahesh, you look bit down. What's the matter?
00:01:09.240 → 00:01:10.220	Mahesh	Nothing much.
00:01:10.220 → 00:01:13.220	Rohit	Looks like something isn't right.
00:01:02.600 → 00:01:09.770	Babu Mahesh	Ya. It's at the job front. You know that the telecom industry is going through a rough patch because of falling prices and shrinking margins. These factors along with consolidation in the industry is threatening the stability of our jobs. And even if the job remains, career growth isn't exciting.
00:01:13.880 → 00:01:24.260	Rohit	I know. I've been reading about some of those issues about your industry in the newspapers. So have you thought of any plan?
00:01:24.900 → 00:01:39.300	Babu Mahesh	I've been thinking about it for a while, but haven't concretized anything so far.
00:01:40.490 → 00:01:41.550	Rohit	What have you been thinking, if you can share?
00:01:41.800 → 00:01:52.000	Mahesh	Well, I've been thinking of switching to an industry that has at least few decades of growth left.

Drawbacks :

- Many existing systems rely heavily on extractive summarization, which only selects sentences and does not generate meaningful human-like summaries.
- Traditional approaches such as TF-IDF and TextRank fail to capture deep contextual understanding of conversations.
- Existing models struggle with multi-speaker dialogue understanding and speaker role identification.
- Long meeting transcripts cause performance degradation due to memory and token limitations in transformer-based models.
- Some systems require large annotated datasets, which are expensive and time-consuming to create.

What's New :

- The system uses a **locally deployed Large Language Model (Ollama)** instead of cloud-based APIs, ensuring privacy and offline functionality.
- Unlike traditional extractive methods, the system generates **context-aware abstractive summaries**.
- The project integrates **conversation memory** to maintain context across long discussions.
- The system supports **action item and key decision extraction** from meeting transcripts.

Team Members:
Harshal Mahesh Patil (TY AIML-Roll No. 37)
Diksha Sudhakar Patil (TY AIML-Roll No. 25)
Tushar Rajendra pardeshi (TY AIML-Roll No.35)
Harshal Dyaneshwar Patil (TY AIML-Roll No.54)