VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

(An Autonomous Institute Affiliated to University of Mumbai)

Department of Computer Engineering

Project Report on

CareerLens - Unsloth PEFT Based Multilingual Summarization and Dashboard Integration for Career Counseling Meets

In partial fulfilment of the Fourth Year, Bachelor of Engineering (B.E.) Degree in Computer Engineering at the University of Mumbai

Academic Year 2024 - 25

By

Deven Bhagtani D17B / 05

Piyush Chugeja D17B / 10

Sakshi Kirmathe D17B / 24

Manraj Singh Virdi D17B / 63

Project Mentor

Dr. (Mrs.) Nupur Giri

A.Y. 2024 - 25

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

(An Autonomous Institute Affiliated to University of Mumbai)

Department of Computer Engineering

Certificate

This is to certify that **Deven Bhagtani (D17B - 05), Piyush Chugeja (D17B - 10), Sakshi Kirmathe (D17B - 24), & Manraj Singh Virdi (D17B - 63)** of Fourth Year Computer Engineering studying under the University of Mumbai have satisfactorily completed the project on **"CareerLens - Unsloth PEFT Based Multilingual Summarization and Dashboard Integration for Career Counseling Meets"** as a part of their coursework of Project II for Semester VIII under the guidance of their mentor **Dr. (Mrs.) Nupur Giri** in the year 2024 - 25.

This project report entitled **"CareerLens - Unsloth PEFT Based Multilingual Summarization and Dashboard Integration for Career Counseling Meets"** by Deven Bhagtani, Piyush Chugeja, Sakshi Kirmathe, & Manraj Singh Virdi is approved for the degree of **Bachelor of Engineering** in **Computer Engineering.**

Programme Outcomes	Grade
PO1, PO2, PO3, PO4, PO5, PO6, PO7, PO8, PO9, PO10, PO11, PO12 PSO1 & PSO2	

Date: 28th April, 2025

Project Guide:

Project Report Approval
For
B. E (Computer Engineering)

This project report entitled **"CareerLens - Unsloth PEFT Based Multilingual Summarization and Dashboard Integration for Career Counseling Meets"** by **Deven Bhagtani (D17B - 05), Piyush Chugeja (D17B - 10), Sakshi Kirmathe (D17B - 24), & Manraj Singh Virdi (D17B - 63)** is approved for the degree of **Bachelor of Engineering** in **Computer Engineering.**

Examiners

1. ……
(Internal Examiner name & sign)

2. ……
(External Examiner name & sign)

3. ……
(Head of Department)

4. ……
(Principal)

Date: 28th April, 2025

Place: Chembur, Mumbai

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Deven Bhagtani - D17B / 05)

(Piyush Chugeja - D17B / 24)

(Sakshi Kirmathe - D17B / 24)

(Manraj Singh Virdi - D17B / 63)

Date: 28th April, 2025

Acknowledgement

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to the Head of the Computer Department **Dr. (Mrs.) Nupur Giri** (Project Guide) for her kind help and valuable advice during the development of the entire project, for her guidance and suggestions.

We are deeply indebted to our Principal **Dr. (Mrs.) J.M. Nair**, for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult to finish this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is a great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement several times.

Computer Engineering Department

COURSE OUTCOMES FOR B.E. PROJECT

Learners will be able to,

Course Outcome	Description of the Course Outcome
CO1	Able to apply the relevant engineering concepts, knowledge and skills towards the project.
CO2	Able to identify, formulate and interpret the various relevant research papers and to determine the problem.
CO3	Able to apply the engineering concepts towards designing solutions for the problem.
CO4	Able to interpret the data and datasets to be utilized.
CO5	Able to create, select and apply appropriate technologies, techniques, resources and tools for the project.
CO6	Able to apply ethical, professional policies and principles towards societal, environmental, safety and cultural benefit.
CO7	Able to function effectively as an individual, and as a member of a team, allocating roles with clear lines of responsibility and accountability.
CO8	Able to write effective reports, design documents and make effective presentations.
CO9	Able to apply engineering and management principles to the project as a team member.
CO10	Able to apply the project domain knowledge to sharpen one's competency.
CO11	Able to develop a professional, presentational, balanced and structured approach towards project development.
CO12	Able to adopt skills, languages, environment and platforms for creating innovative solutions for the project.

Index

List Of Figures

List Of Tables

Abstract

In today's technologically advanced learning environment, career counseling has emerged as a crucial component for students navigating their academic and career paths. But typical counseling sessions frequently lack organized records, make it hard to monitor students' development, and lack smart tools to help with follow-up after sessions. In order to overcome these obstacles, our project suggests CareerLens, a clever Google Meet extension created especially to improve and streamline the career counseling process.

CareerLens incorporates a bot into Google Meet to get transcripts in real time as the session progresses, removing the need for manual note-taking or audio recording. An improved large language model built on Meta's LLaMA 3.1 and fine-tuned for counseling contexts is then used to process these transcripts. Context-aware summaries that emphasize significant concepts, practical recommendations, and crucial conversation points are produced by this model. The structured output is an excellent tool for both counselors and students, assuring clarity and consistency in career planning.

Together with summary, CareerLens offers interactive elements like modules for QnA that let users get certain insights from the transcript of the session. For example, depending on the chat, a student can ask questions like "What were the recommended career paths?" or "What steps should I take next?" and get concise, relevant responses. Additionally, the module offers action item extraction, which generates a roadmap for the student's development and recommends tailored next actions.

CareerLens also provides speaker timelines and frequency graphs, which display who talked when and how frequently, helping in visualizing session dynamics. This makes it clear how fair or biased the discussion was, which is helpful for evaluation and progress. The platform is constructed with a ReactJS frontend for an intuitive and responsive user experience and a Flask-based backend for model inference and logic. As it is currently aimed for English, Hindi, and Marathi, it allows multilingual transcripts and summary, making them accessible to students with a variety of language backgrounds.

CareerLens easily integrates automated technology into career advising, redefining student support in education. The technology turns counseling sessions into organized, data-driven experiences by emphasizing accessibility, customisation, and effective information delivery. It provides counselors and students with valuable insights that improve advising, communication, and career decision-making.

Chapter I: Introduction

1.1 Introduction

With the rise of remote work in 2020 - 2021, online meetings have become integral to professional and academic interactions. Platforms like Zoom and Google Meet now report millions of daily participants, highlighting a global shift toward online communication environments [1, 2, 3, 4]. This digital transformation has extended to career counseling as well, where online sessions offer flexible, scalable ways to support students in shaping their academic and professional futures. However, these sessions often lack structured documentation, making it difficult to revisit key insights or action items using conventional methods.

To address this challenge, CareerLens introduces an AI-powered solution that enhances the efficiency and accessibility of career counseling sessions. It automatically transcribes and summarizes conversations by leveraging a fine-tuned LLaMA model customized for educational dialogue. Using QLoRA [19] for parameter-efficient fine-tuning, the system achieves high performance with significantly reduced memory consumption, making it suitable for real-time use. Unlike traditional note taking tools, CareerLens provides context-aware summaries, interactive Q&A modules, and timeline-based visualizations to improve user engagement and clarity. This allows both counselors and students to extract key information effortlessly, focus on actionable insights, and make better-informed decisions regarding career planning and development [7, 13].

1.2 Motivation

The CareerLens project is motivated by the rising demand for efficient and effective communication in career counseling. Since many people find it difficult to express their job goals and obstacles, counselors must offer clear direction. Traditional note-taking techniques may hinder the process, resulting in miscommunication and loss of important information. CareerLens wants to give counselors and clients useful insights by automating the summarizing of counseling sessions. In order to help users make educated choices regarding their professional pathways, the initiative aims to improve counseling overall. In the end, We want to close the communication gap between customers and counselors in order to promote a more productive and encouraging atmosphere for professional growth.

1.3 Drawback of existing systems

Current systems are mostly concerned with broad meeting functions and frequently lack the specific skills needed for effective career advising. Since these systems lack any analysis created especially for career counseling, they are limited in their capacity to offer precise advice and insights for people's professional paths. Additionally, a lot of apps rely on captions or transcriptions from the meeting app rather than live transcription, which reduces speed and accuracy. Furthermore, effective user-application communication suffers by the absence of real-time Q&A capabilities. Due to their inflexible user interfaces and limited analytics, these systems are unable to efficiently organize and retrieve meeting data, leaving them without ways to track users' progress after sessions.

1.4 Problem Definition

In today's fast-paced digital age, career counseling meetings have become necessary to lead individuals onto successful career pathways. However, the approach of writing notes, condensing, and deciding on things from these sessions is tiresome and prone to errors, creating delays in accessing data as well as making decisions. This can lead to a negative impact on the quality of counseling being provided. Further, the absence of advanced technologies to automatically summarize and analyze meeting data constraints counselors' ability to provide real-time and appropriate feedback to clients. It may prove to be challenging to derive meaningful insights due to the challenging process of documenting and reading these meetings, which often leads to inefficiency. Consequently, it is sometimes difficult for the clients and counselors to monitor progress, identify key issues to discuss, and make decisions regarding counseling sessions.

1.5 Relevance of the project

The project's importance lies from its potential to change how career guidance is provided. The project fills in current gaps in career counseling systems, like real-time transcribing and interactive Q&A features, through the use of modern technologies like artificial intelligence (AI) and natural language processing. This innovative method offers consumers personalized insights and helpful recommendations in addition to improving the effectiveness and efficiency of counseling sessions. The project's ultimate goal is to enable people to enhance their professional experiences and make well-informed career decisions.

Chapter II: Literature survey

2.1 Research Papers Review

1) Vaswani et al. (2023), Attention is all you need [5]

- **Abstract:** This foundational paper introduces the Transformer architecture, a novel neural network architecture based entirely on attention mechanisms, dispensing with recurrence and convolutions. It demonstrates superior performance in machine translation tasks compared to previous state-of-the-art models.
- **Inference:** The Transformer architecture has become the backbone of many state-of-the-art NLP models, including those used for abstractive summarization. Its ability to capture long-range dependencies in text is crucial for understanding the context of lengthy meeting transcripts.
- **Relevance to our work:** Our work leverages Transformer-based models (specifically LLaMA 3) for meeting summarization, making this paper fundamental to the underlying architecture we employ.

2) Sadri et al. (2021), MeetSum: Transforming meeting transcript summarization using transformers! [13]

- **Abstract:** This paper explores the application of Transformer models for meeting transcript summarization. It proposes the MeetSum model, which demonstrates the effectiveness of Transformer architectures in capturing the nuances of meeting conversations and generating coherent summaries.
- **Inference:** This work directly shows the potential of Transformer models for the specific task of meeting summarization, providing a baseline and inspiration for further research in this area.
- **Relevance to our work:** MeetSum serves as a direct predecessor and a point of comparison for our work, as we also utilize Transformer-based models for meeting summarization.

3) Bhat et al. (2023), Jotter: An approach to summarize the formal online meeting [14]

- **Abstract:** Jotter is a hybrid approach that combines BERT embeddings (for extractive capabilities) with sequence-to-sequence models (for abstractive generation). It aims to balance accuracy and fluency in meeting summaries.

- **Inference:** Hybrid approaches like Jotter attempt to leverage the strengths of both extractive and abstractive summarization techniques. While achieving good coherence, they can be computationally intensive, which might be a limitation for real-time applications.
- **Relevance to our work:** Jotter presents an alternative approach to pure abstractive summarization. Understanding its strengths and weaknesses helps us position our Transformer-based abstractive approach.

4) Kumar and Kabiri (2022), Meeting Summarization: A Survey of the State of the Art [15]
    - **Abstract:** This paper provides a comprehensive survey of the existing research in meeting summarization. It highlights the various approaches, challenges, and future directions in the field, including the limitations of current datasets and the need for domain-specific models and evaluation metrics.
    - **Inference:** This survey underscores the gaps in the field, such as the lack of focus on domain-specific summarization (like career counseling) and the reliance on generic datasets. It emphasizes the need for tailored solutions and evaluation methods.
    - **Relevance to our work:** This paper directly supports the motivation for our work by identifying the need for domain-specific meeting summarization in areas like career counseling, which our research aims to address.
5) Wyawahare et al. (2024), AI Powered Multilingual Meeting Summarization [16]
    - **Abstract:** This paper discusses AI-powered methods for summarizing multilingual meetings. It explores the use of Latent Semantic Analysis (LSA) and Transformer models, noting that while LSA can oversimplify, Transformer models are more effective at retaining context across languages.
    - **Inference:** This work highlights the importance of handling multilingualism in meeting summarization and confirms the superiority of Transformer models over older techniques like LSA for this task due to their contextual understanding.
    - **Relevance to our work:** If our system aims to handle multilingual career counseling sessions, this paper provides valuable insights into the challenges and effective techniques for multilingual summarization using Transformer architectures.

6) Srivastava et al. (2022), Counseling Summarization Using Mental Health Knowledge Guided Utterance Filtering [17]

- **Abstract:** This paper presents ConSum, a structured summarization method for mental health counseling. It utilizes domain-specific knowledge (PHQ-9 scoring) to filter important speech patterns, demonstrating the benefits of incorporating specialized knowledge into the summarization process.
- **Inference:** This research shows that leveraging domain-specific knowledge can significantly improve the relevance and quality of summaries in specialized fields like counseling.
- **Relevance to our work:** ConSum inspires our approach of tailoring the summarization process to the specific domain of career counseling, potentially by incorporating role information or other relevant contextual cues.

7) Zhang et al. (2023), Benchmarking Large Language Models for News Summarization [18]

- **Abstract:** This paper benchmarks large language models for news summarization and finds that instruction tuning is more effective than traditional fine-tuning. Instruction tuning allows models to perform well even without large domain-specific datasets, relying instead on high-quality prompts and instructions.
- **Inference:** Instruction tuning offers a promising approach for improving the performance of large language models on summarization tasks, especially when domain-specific data is limited.
- **Relevance to our work:** This finding supports our use of fine-tuning LLaMA 3 with a focused career counseling dataset, potentially enhanced with carefully crafted instructions.

8) Dettmers et al. (2023), QLoRA: Efficient Fine Tuning of Quantized LLMs [19]

- **Abstract:** QLoRA is a parameter-efficient fine-tuning (PEFT) method that significantly reduces the memory footprint required to fine-tune large language models by quantizing the pre-trained weights and adding small, learnable layers.
- **Inference:** QLoRA makes it feasible to fine-tune very large language models with limited computational resources, democratizing access to specialized model training.

- **Relevance to our work:** Our work explicitly mentions using QLoRA for fine-tuning LLaMA 3, making this paper directly relevant to the efficiency aspect of our methodology.

9) Singh et al. (2024), HindiSumm: A Hindi Abstractive Summarization Benchmark Dataset [21]

- **Abstract:** This paper introduces the HindiSumm dataset, a benchmark for Hindi abstractive summarization. It also proposes evaluation metrics like redundancy, conciseness, novel n-grams ratio, and abstractivity to assess the quality of generated Hindi summaries.
- **Inference:** This work highlights the growing attention towards summarization in low-resource languages like Hindi and provides valuable resources and evaluation methods for this context.
- **Relevance to our work:** If our system aims to handle Hindi career counseling sessions, this dataset and the proposed evaluation metrics could be valuable for training and assessing the performance of our model in Hindi.

10) Daisy et al. (2023), Abstractive Hindi Text Summarization: A Challenge in a Low-Resource Setting [22]

- **Abstract:** This paper discusses the challenges of abstractive Hindi text summarization in a low-resource setting and proposes the ICE-H metric for evaluating summary quality in such scenarios.
- **Inference:** This research further emphasizes the difficulties and the need for specialized evaluation techniques for summarization in low-resource languages like Hindi.
- **Relevance to our work:** Similar to the previous paper, if our system includes Hindi summarization, the insights and the ICE-H metric could be relevant for our evaluation process.

2.2 Inference Drawn

Based on the literature survey, several key inferences can be drawn regarding the field of meeting summarization and its application to our work in career counseling:

- **Transformer Architectures are Dominant:** Transformer-based models have become the state-of-the-art for abstractive summarization, including meeting summarization, due to their ability to capture long-range dependencies and generate coherent summaries.

- **Domain Specificity is Crucial:** Generic summarization models trained on news or other general datasets may not effectively capture the nuances and important information in domain-specific conversations like career counseling. There's a recognized need for models and evaluation metrics tailored to specific domains.
- **Multilingual Support is a Growing Requirement:** With the increasing prevalence of online and international meetings, the ability to summarize conversations in multiple languages is becoming increasingly important. Transformer models show promise in this area.
- **Efficiency is Key for Real-Time Applications:** Computational cost and latency are significant challenges, especially for real-time meeting summarization. Parameter-efficient fine-tuning techniques like QLoRA offer a way to address these challenges for large language models.
- **Instruction Tuning Enhances Performance:** Instruction tuning can be a more effective fine-tuning strategy than traditional methods, especially when domain-specific data is limited, by guiding the model with clear prompts and instructions.
- **Evaluation Metrics Need to Be Contextualized:** Traditional summarization evaluation metrics like ROUGE and BLEU might not fully capture the quality and relevance of summaries in specialized domains. There's a need for more nuanced, domain-aware evaluation approaches.
- **Hybrid Approaches Offer a Balance:** Combining extractive and abstractive techniques can offer a balance between accuracy and fluency, but might come with computational overhead.
- **Open-Source LLMs are Competitive and Offer Advantages:** Open-source large language models are becoming increasingly powerful and offer benefits in terms of cost and privacy compared to proprietary models.

2.3 Comparison with the Existing System

Feature	Existing Meeting Summarization Systems (General)	Our Proposed System (Career Counseling Focused)	Potential Advantages of Our System
Domain Focus	Primarily trained and evaluated on generic meeting datasets (e.g., AMI, ICSI).	Specifically fine-tuned and evaluated on career counseling conversations.	Improved relevance, accuracy, and extraction of domain-specific insights.
Language Handling	Some systems offer multilingual capabilities, often relying on general MT models.	Aims for robust multilingual summarization tailored to career counseling terminology.	Better understanding and summarization of multilingual career-related discussions.
Real-time Capability	Can be limited by the computational cost of large models.	Focus on efficient fine-tuning (e.g., QLoRA) for potential real-time processing.	Lower latency and feasibility for real-time summarization.
Model Architecture	Often based on standard Transformer models or hybrid approaches.	Leverages the latest open-source LLMs (e.g., LLaMA 3) fine-tuned for the domain.	Potential for better performance and adaptability.
Fine-tuning Approach	May rely on traditional fine-tuning methods.	Employs parameter-efficient fine-tuning (e.g., QLoRA) and potentially instruction tuning.	More efficient use of resources and potentially better generalization with limited domain data.

Feature	**Existing Meeting Summarization Systems (General)**	**Our Proposed System (Career Counseling Focused)**	**Potential Advantages of Our System**
Evaluation Metrics	Primarily uses generic metrics like ROUGE and BLEU.	Aims to incorporate or develop evaluation metrics that assess domain-specific accuracy and actionable insights.	More meaningful assessment of summary quality in the context of career counseling.
Privacy & Cost	May rely on proprietary, API-based models (e.g., GPT-4).	Focuses on open-source models for better privacy and lower costs.	Enhanced data privacy and reduced operational expenses.
Role Awareness	Generally does not explicitly incorporate speaker roles into the summary.	Aims to integrate user roles and speaker information for role-aware insights.	Improved contextual understanding and relevance of summaries in professional settings.

Table 2.1: Comparison of proposed system with existing systems

Chapter III: Requirement Gathering for the Proposed System

3.1 Introduction to requirement gathering

Requirement gathering is the process of identifying, analyzing, and documenting the needs and expectations of stakeholders for a new or existing system. It forms the foundation of any successful project, ensuring that developers and designers clearly understand what the end-users want and what the system is supposed to do. This process involves interactions with clients, users, domain experts, and project managers to collect both functional requirements (what the system should do) and non-functional requirements (how the system should perform). The main purpose of requirement gathering is to avoid misunderstandings, reduce development rework, and deliver a product that meets user needs effectively. In this chapter, we will explore the specific requirements and constraints of our project to ensure clear direction and successful implementation.

3.2 Functional Requirements

Functional requirements specify the specific functionalities and features that the system must provide to meet the needs of stakeholders and users. Functional requirements for our system are:

- Real-time Transcription: The system must capture and transcribe audio from Google Meet sessions in real-time, providing accurate and timely captions for users.
- Summarization of Meetings: The system should generate concise and contextually relevant summaries of the counseling sessions, highlighting key discussion points and action items.
- Interactive Q&A Module: Users should be able to ask questions related to the session's content, and the system must provide accurate answers based on the transcribed text.
- Integration with Existing Tools: The system must be able to integrate with popular video conferencing tools, such as Google Meet, to facilitate seamless operation.
- Multilingual Support: The transcription feature should support multiple languages, allowing users from diverse linguistic backgrounds to access the system.

3.3 Non-Functional Requirements

Non-functional requirements define the quality attributes and constraints that the system

must adhere to. In our project, non-functional requirements includes:

- Performance: The system should provide real-time transcription with minimal latency, ensuring that users receive timely and accurate captions during counseling sessions.
- Scalability: The architecture must support scalability to handle an increasing number of users and concurrent sessions without degrading performance.
- Reliability: The system should maintain a high level of reliability, ensuring that transcriptions and summaries are consistently accurate and available to users.
- Security: The system must implement robust security measures to protect user data, including encryption for stored data and secure authentication for user access.
- Compatibility: The application should be compatible with multiple devices and operating systems, including desktops, tablets, and mobile devices.

3.4 Hardware & Software Requirements

Category	Requirement
Hardware Requirements	
CPU	Intel Core i7 or higher
RAM	16 GB or more
GPU	NVIDIA T4 / NVIDIA P100 – 16 GB or more VRAM
Webcam / Camera	Required for video-based interactions
Microphone	Required for audio-based meetings or recordings
Software Requirements	
Operating Environment	Jupyter Notebook
Code Editor	VS Code or any Python-supported IDE
Python Libraries	`unsloth`, `transformers`, `LoRA` (PEFT), `datasets`, `torch`, `fastapi`, etc.
Visualization Tools	Matplotlib, Seaborn, Plotly (for charts/graphs)
Browser	Any modern browser (e.g., Chrome, Firefox, Edge)

Containerization Tool	Docker 28.0.4 (for scalable deployment and meet bot creation)
Language Model	LLaMA 3.1 (8B), loaded using Unsloth in 4-bit quantized format

Table 3.1: Hardware & Software Requirements

3.5 Constraints

- Data Availability: The effectiveness of the model is dependent on the availability of high-quality training data, which may be limited or require extensive preprocessing.
- Technical Limitations: There may be limitations related to the performance of the Llama model, such as processing speed and memory usage, which can affect real-time transcription capabilities.
- Budget Constraints: Financial limitations may restrict the resources available for cloud services, infrastructure, and technology needed to implement and maintain the system.
- Integration Issues: Challenges may arise in integrating the new system with existing platforms or tools used by counselors and clients, affecting overall usability and functionality.
- Multilingual Support Complexity: Providing effective multilingual support can increase the complexity of the system, requiring additional resources for language models and testing.

3.6 Tools & Techniques utilized till date

- Google Meet API + Attendee Bot: Integrated for creating meetings and collecting real-time transcripts via a Docker-based bot that joins Google Meet sessions.
- Fine-tuned LLaMA LLM: Used for analyzing transcripts, generating abstractive summaries, extracting insights, and enabling interactive Q&A.
- PEFT (Parameter-Efficient Fine-Tuning): Employed for lightweight, high-performance model tuning tailored to career counseling content.
- Multilingual Support: Enabled in English, Hindi, and Marathi for inclusive summarization and interaction.
- Real-time Processing: Achieved through integration with Google Meet and DynamoDB for immediate transcript capture and response.
- Data Visualization: Speaker timelines, frequency charts, and summaries presented via interactive dashboard visuals.

- Interactive Q&A Module: Lets users query the summarized content and receive instant, context-aware responses.
- Feedback Loop: Captures user reviews to continuously improve platform accuracy and experience.
- DynamoDB: Used for secure, scalable storage and retrieval of session transcripts and outputs.

3.7 Development Stack

- Languages: Python (backend, AI), JavaScript (frontend, UI interaction)
- Libraries & Frameworks: Hugging Face Transformers, Flask, React
- Platforms: Google Colab, Jupyter Notebook (for EDA and model prototyping)
- Version Control: Git & GitHub for collaborative development
- Dataset Source: Mentioned in Section 5.3

Chapter IV: Proposed Design

4.1 Block diagram of system

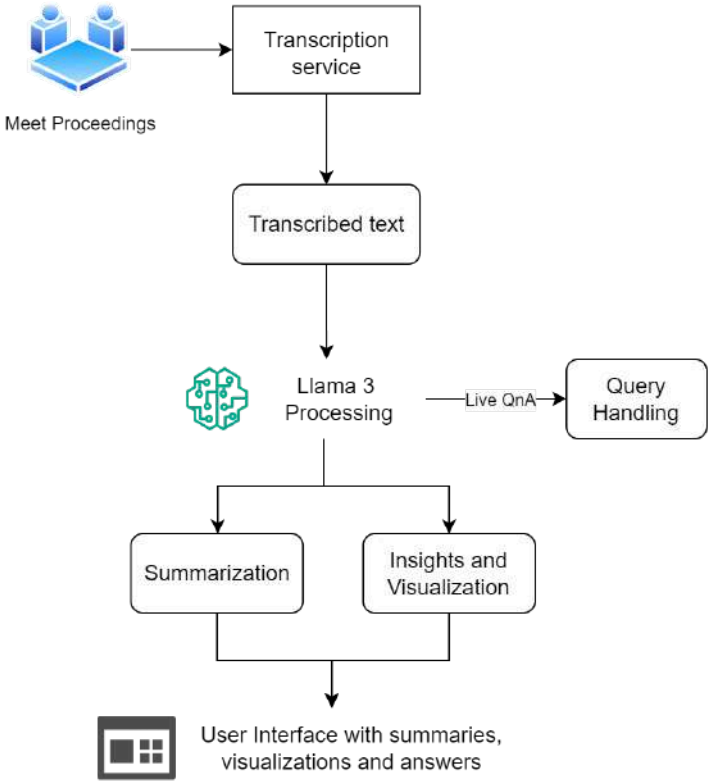


Figure 4.1: Block diagram of the system

The block diagram displays the workflow of the CareerLens system, which combines transcription, language processing, and visualization to provide users with an effortless experience.

1. Meet Proceedings: This is a representation of the direct conversation or meeting that occurs during a career counseling session.
2. Transcription service: A transcription service records and transforms the meeting's spoken information into transcribed text in real time.
3. Processing by Llama 3: The transcribed text is fed into the fine-tuned Llama 3 model for advanced processing, allowing the system to analyze and understand the meeting content.
4. Query Handling: The system supports live Q&A functionality, enabling users to ask questions related to the session topics and receive instant responses.
5. Summarization: The Llama 3 model also automatically summarizes the counseling session by extracting important ideas and conclusions into brief summaries.

6. Visualizations and Insights: The system generates visual summaries and key insights, highlighting important data points and patterns from the meeting.

7. User Interface: A dashboard displays summaries, insights, and visualizations, allowing users to interact easily with the system and get answers to their questions.

4.2 Modular design of the system

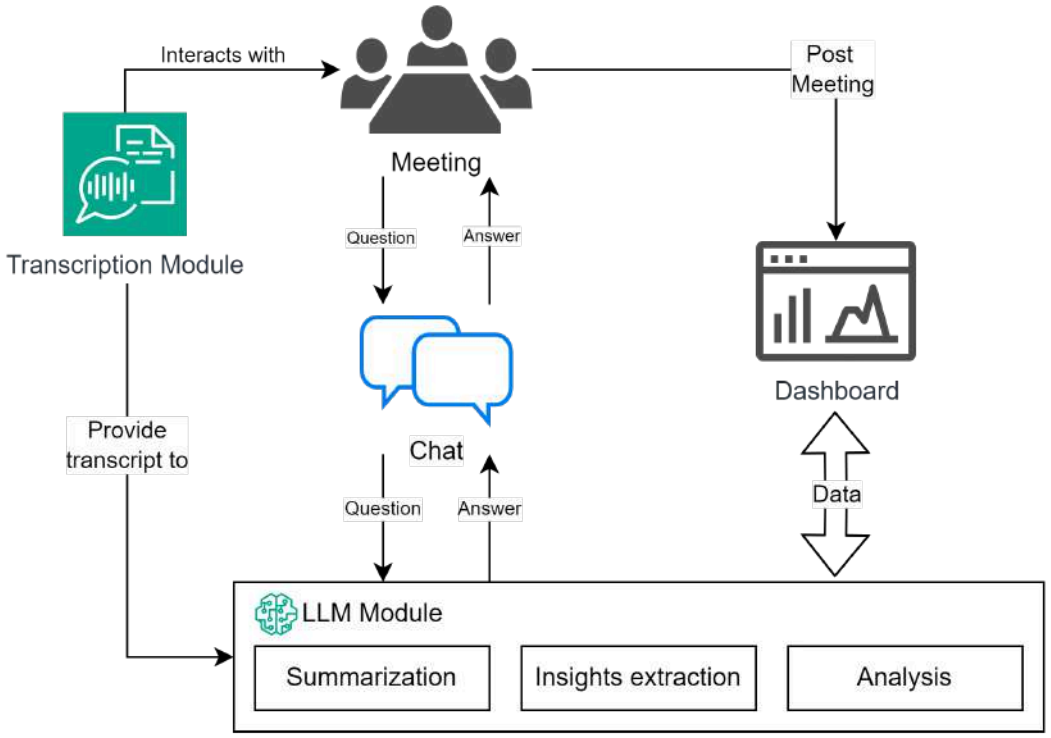


Figure 4.2: Modular diagram of the system

The modular design of our system encompasses several key components, each responsible for specific functionalities to ensure the overall effectiveness and efficiency of the system.

1. Meeting: The system collects data in real-time from career counseling meetings, capturing all discussions among participants. This live data collection enables immediate analysis and interaction, enhancing the overall counseling experience.

2. Transcription Module: This module captures the live conversation during the meeting, converting spoken words into real-time transcripts. These transcripts are essential for the subsequent analysis and are transmitted to the LLM module for deeper insights.

3. LLM Module (Llama 3): As the core AI component, this module analyzes the transcripts to perform various tasks. It extracts key insights, summarizes important points, and provides responses to questions raised during the session, enhancing the understanding of the discussions.

4. Chat Interaction: This feature facilitates real-time question-and-answer sessions, allowing users to engage with the system to receive relevant responses on the topics discussed in the meeting. It fosters an interactive environment and ensures participants' inquiries are addressed promptly.

5. Dashboard: After the meeting concludes, the system displays a comprehensive dashboard featuring detailed summaries, insights, visualizations, and actionable items. This presentation allows users to review and analyze the key takeaways from the session effectively.

4.3 Design of the proposed system

a. Data Flow Diagrams

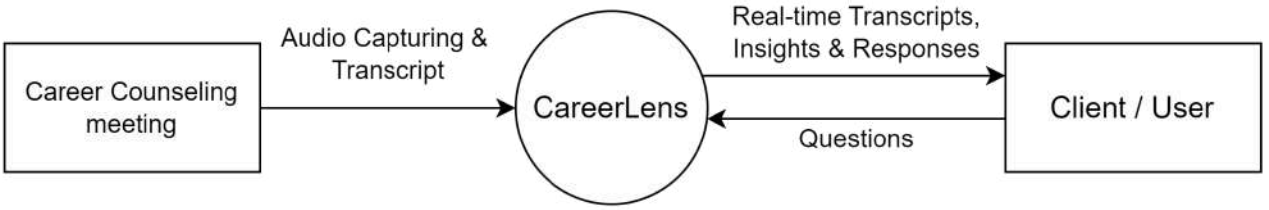


Figure 4.3.a: Level 0 DFD

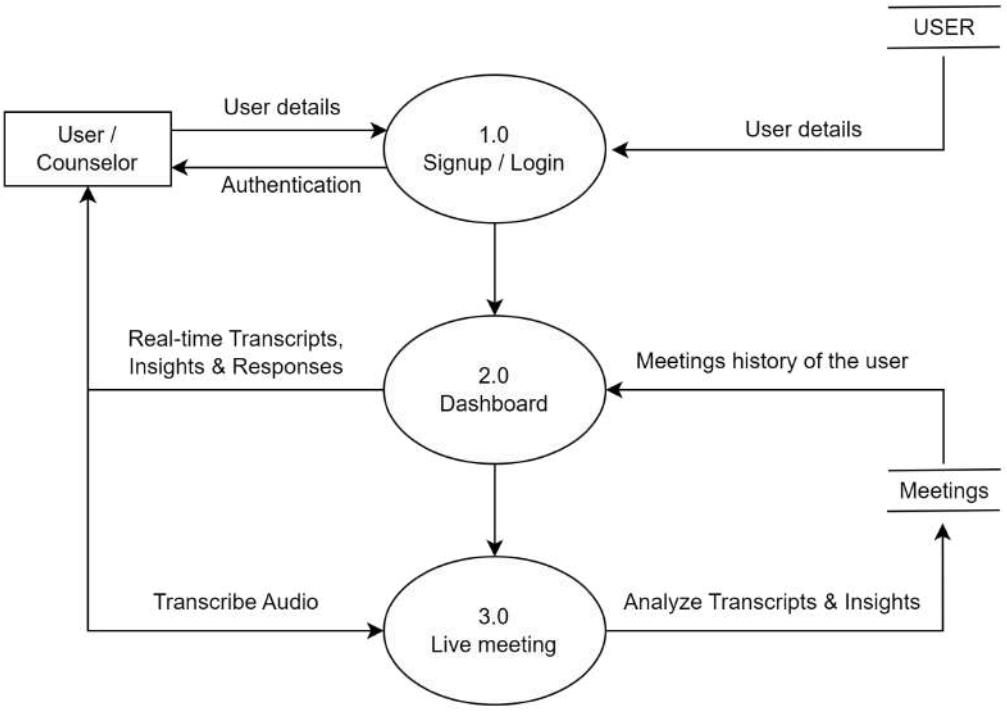


Figure 4.3.b Level 1 DFD

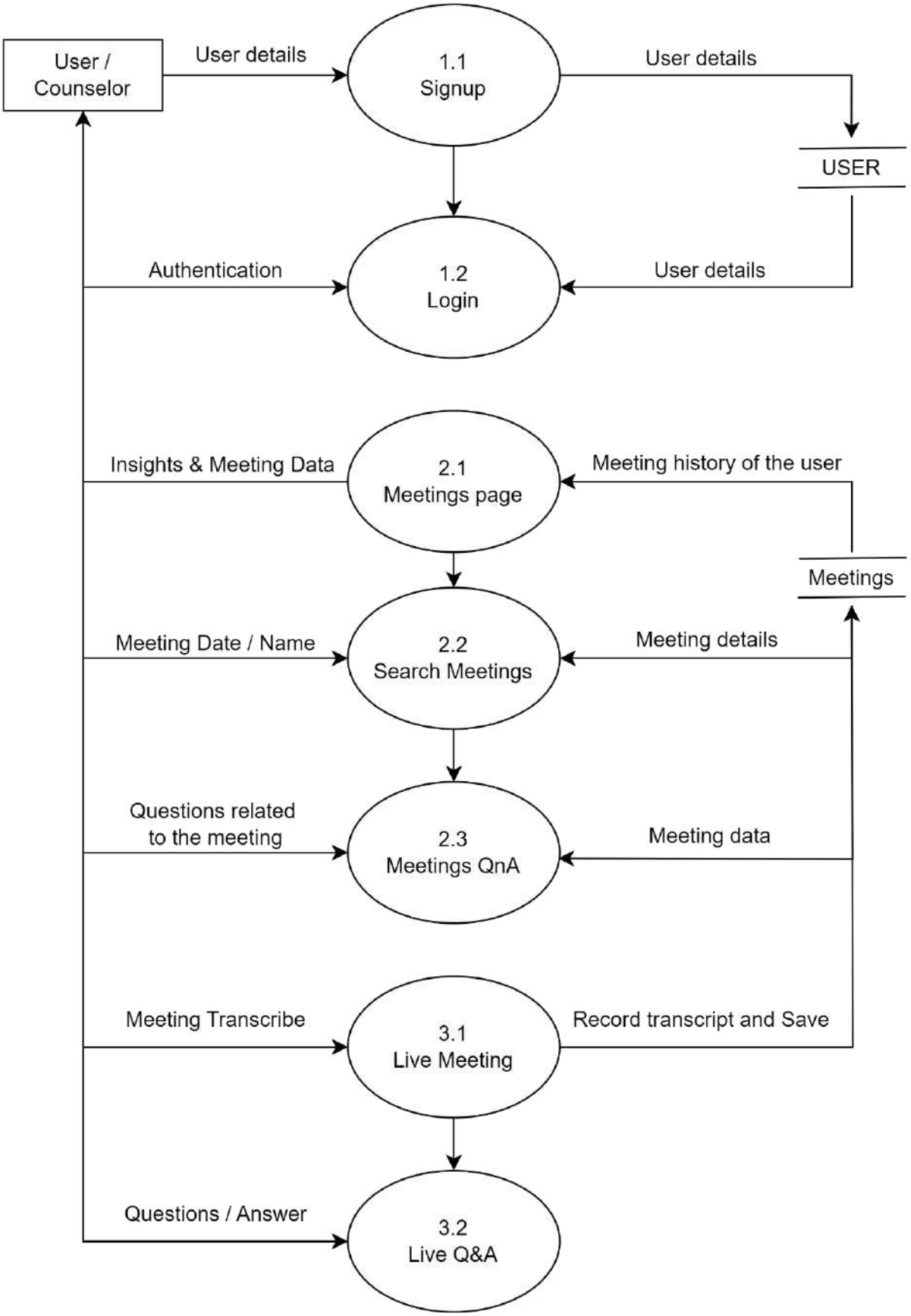


Figure 4.3..c: Level 2 DFD

b. Flowchart for the proposed system

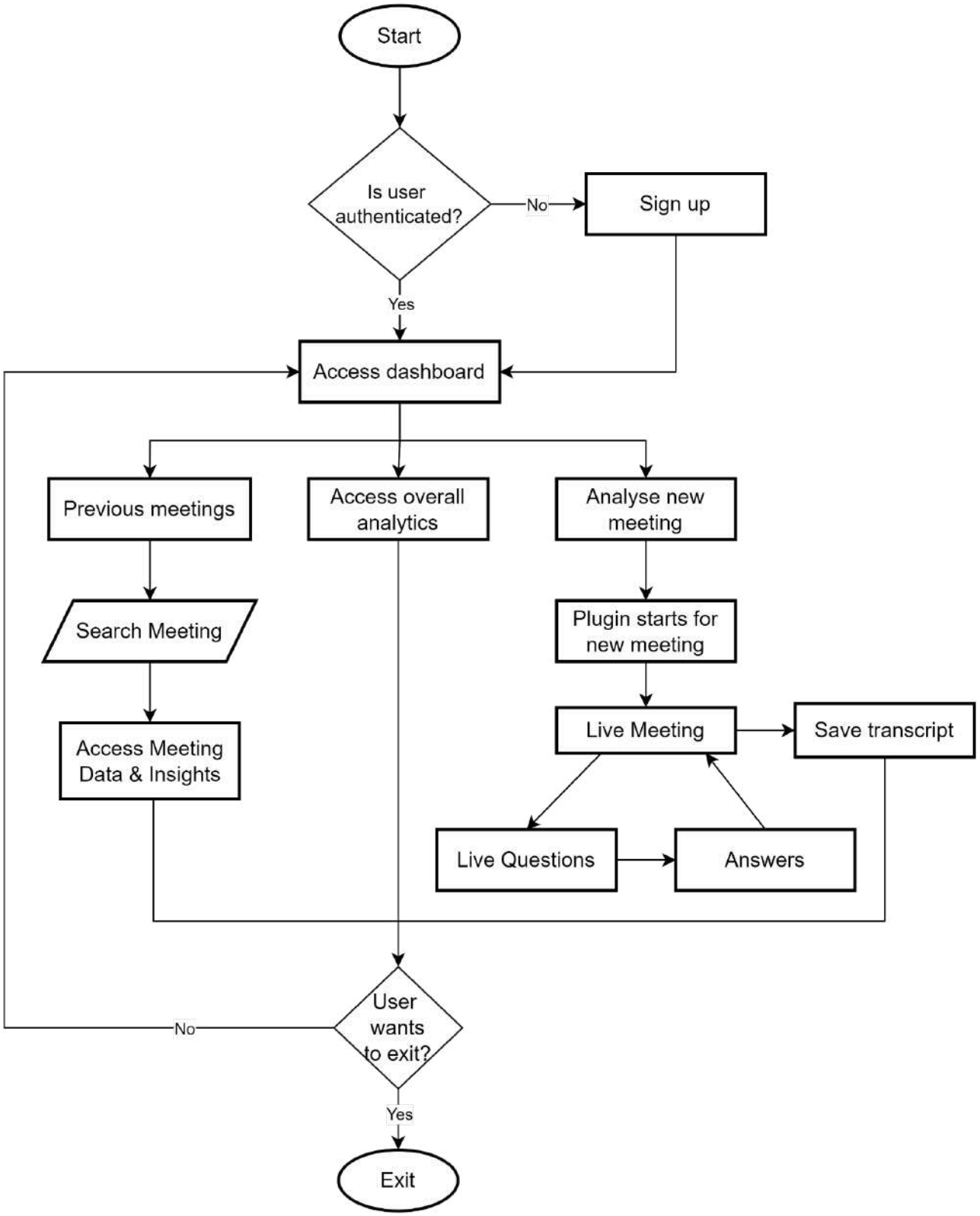


Figure 4.3.d: Flowchart of the system

4.4 Gantt Chart

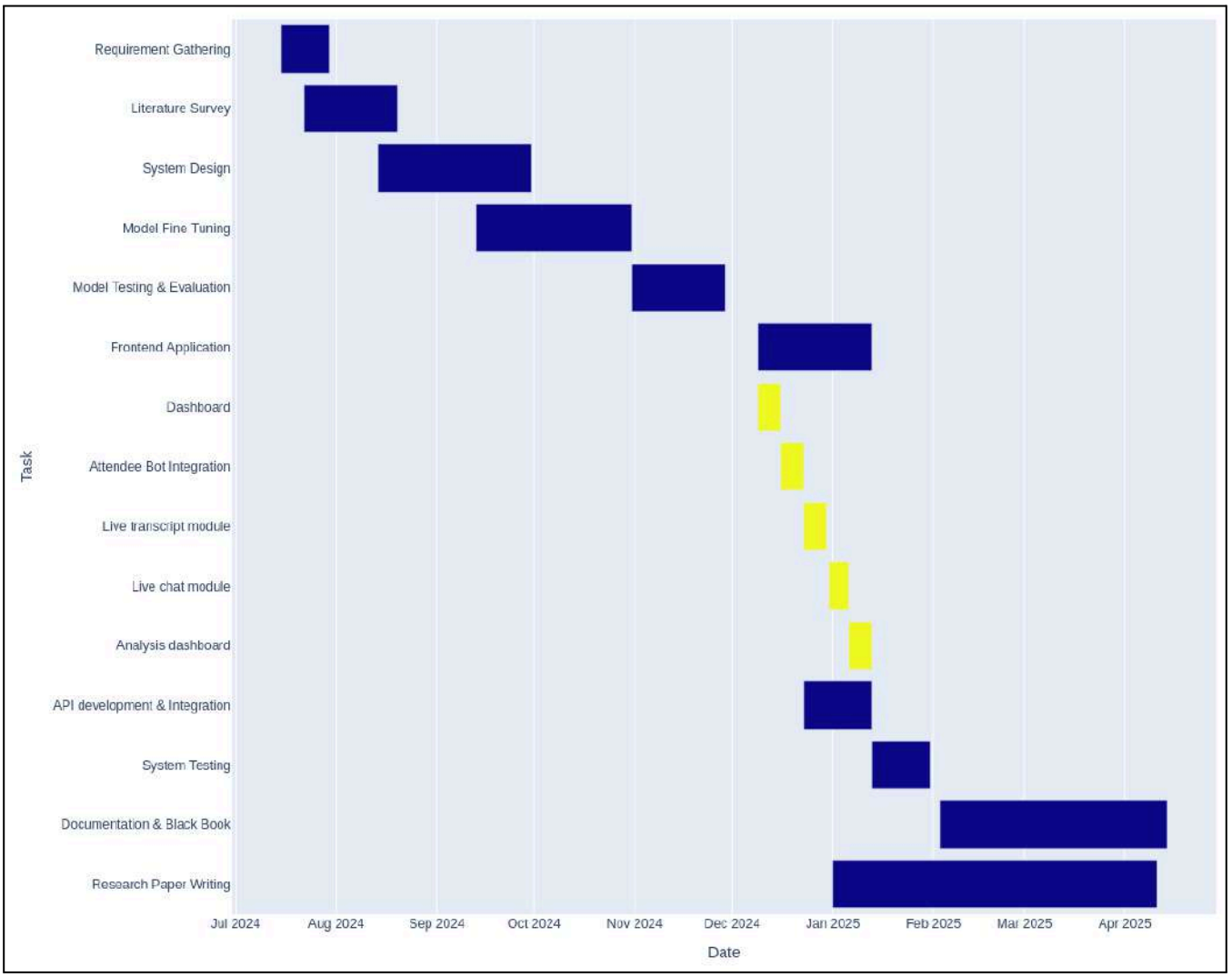


Figure 4.4: Gantt chart - project timeline

Chapter V: Implementation of Proposed System

5.1 Methodology employed for development

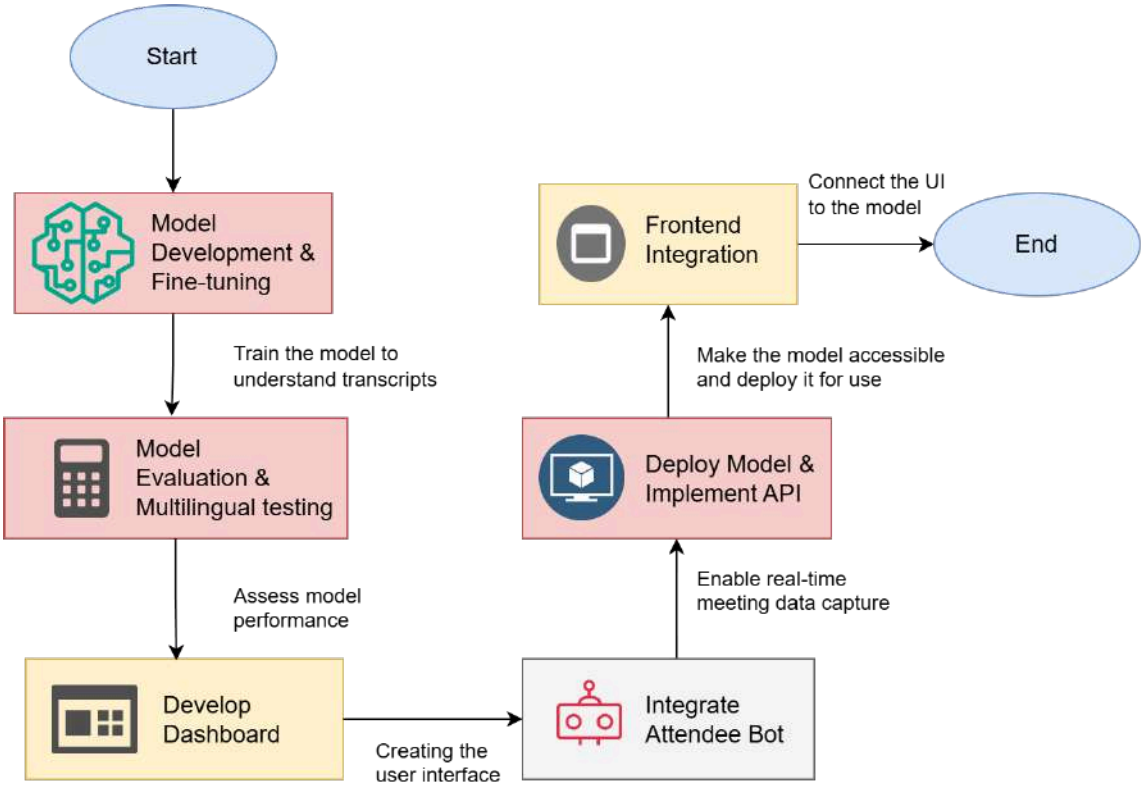


Figure 5.1: Methodology Employed

The development of the CareerLens platform followed a structured and sequential methodology to ensure the integration of all components for a smooth user experience. The process involved the following steps:

1. Model Development and Fine-Tuning

   The project focused on building a robust meeting summarization system tailored to the domain of career counseling. To this end, open-source LLMs from three major families - LLaMA, Mistral, and DeepSeek - were selected for experimentation. The development pipeline began with preprocessing career counseling transcripts and formatting them into instruct-style datasets (transcript-summary pairs) as further explained in Section 5.3. These structured datasets were versioned and fed into Unsloth's 4-bit quantization and PEFT (LoRA) framework for efficient fine-tuning, as described in recent literature [12, 19, 20].

   Fine-tuning followed a Supervised Fine-Tuning (SFT) approach [23], allowing the pre-trained models to adapt to domain-specific summarization tasks. Training was

executed on Kaggle's P100 GPUs using the AdamW optimizer (8-bit variant) and cosine learning rate scheduler.

Key hyperparameters included:

- Learning Rate: 1e-5
- Per-device Batch Size: 2 (with gradient accumulation of 8 steps)
- Max Sequence Length: 4096 tokens

Performance monitoring was conducted via Weights & Biases (W&B). Multiple experiments were run in parallel, and models were iteratively evaluated and stored based on their output quality and inference efficiency.

The overall workflow is illustrated in Figure 5.3, which outlines the flow from raw transcript preprocessing, dataset creation, fine-tuning, and final evaluation before deployment.

2. Evaluation and Multilingual Testing

For systematic evaluation, models were tested using **zero-shot, one-shot, and three-shot settings,** following the methodology described by [18]. These settings helped assess in-context learning performance and determine the optimal model from each family (see Table 7.1 in the report).

The LLaMA 3.1 8B model consistently outperformed others and was selected for fine-tuning and multilingual evaluation. Testing extended across English, Hindi, and Marathi, confirming LLaMA's capability to generalize across languages.

Evaluation metrics included both conventional and multilingual-specific ones as defined by Singh et. al. and Daisy et. al. [21, 22]:

- **ROUGE, BLEU, GLEU, BERT Score** [24, 25, 26, 27]: Standard measures for comparing generated summaries with references.
- **Information Coverage Estimate (ICE):** Cosine similarity between Sentence-BERT embeddings of generated and reference summaries.
- **Redundancy:** Quantifies repeated content in summaries. Lower is better.
- **Abstractivity:** Ratio of novel (non-copied) words to total words—higher indicates better paraphrasing.
- **N-gram Ratio:** Measures lexical diversity via unique n-grams - higher values signify richer language.
- **Conciseness:** Ratio of summary length to original transcript - lower values imply better compression.

These metrics ensured summaries were not only accurate and fluent but also informative, succinct, and linguistically diverse across multiple languages, as shown in Table 7.3.

3. Dashboard Development

The first step in the development was to create the React Dashboard, which acts as the user interface for interacting with the platform. The dashboard provides users with three primary options: Create a Meeting, Join an Existing Meeting, and Upload a Transcript. This dashboard is designed to be user-friendly and provides easy navigation for users to engage with the platform's various features.

4. Working with Attendee Bot

One of the core features of CareerLens is the Attendee Bot, an open-source tool available on GitHub [28]. This bot automates the process of joining virtual meetings (like Google Meet), recording the session, and transcribing the conversation in real time. The bot runs inside a Docker container and is controlled through a Flask API. When a meeting needs to be joined, our system makes an API call to spin up a new bot instance. The bot joins the session, starts recording the audio, and begins generating the live transcript.

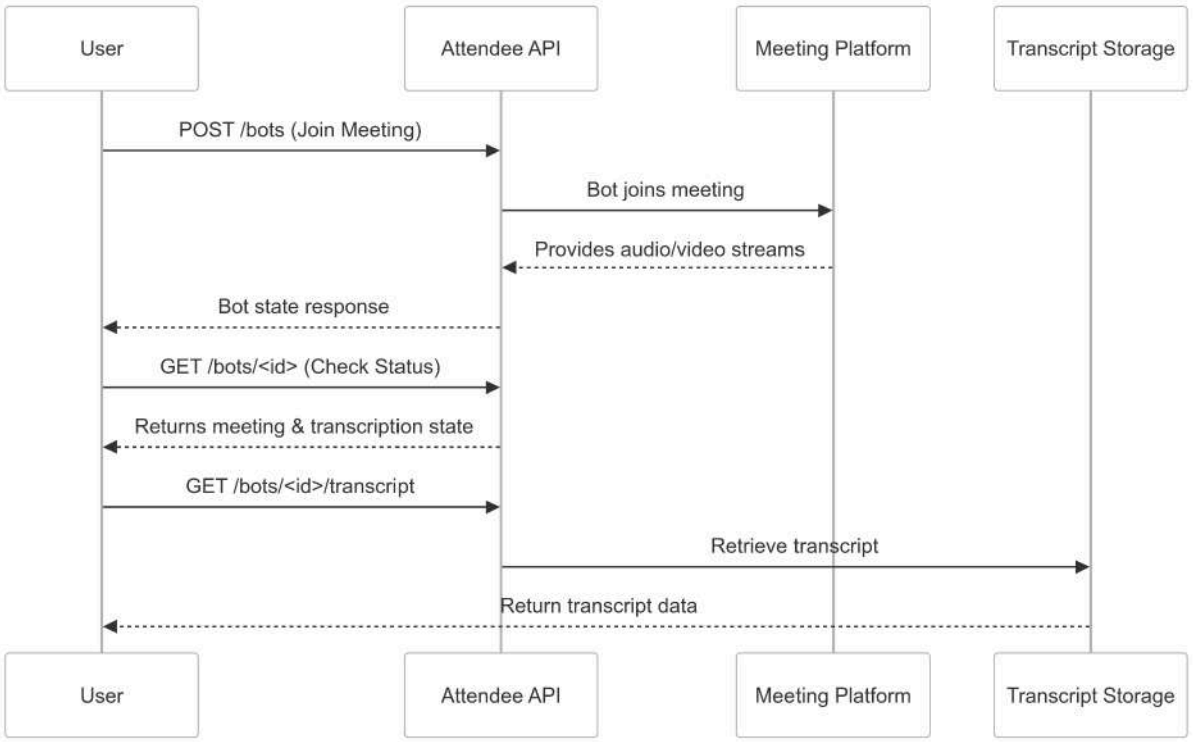


Figure 5.2: Attendee bot workflow

The figure 5.2 represents how the Attendee bot works and functions. The transcript can be fetched via the `/bots/<id>/transcript` API route in real time. The refined transcript is then passed on to the summarization model for analysis. The bot concludes its task by storing the processed summary and transcript in a cloud database, making it accessible to users through the dashboard.

5. Integrating Attendee Bot to Frontend

To integrate this functionality into the CareerLens frontend, we created API endpoints that the React-based dashboard interacts with. This allows users to simply click a "Join Meeting" button, and the bot will automatically join the meeting, transcribe it, and send all the relevant data back to our system for analysis.

Similarly, users can click the "Leave Meeting" button to have the bot exit the session. This seamless integration ensures that the entire meeting experience, from joining to summarization, is handled smoothly without needing any manual setup.

6. Model Deployment and API Implementation

With the LLaMA model fine-tuned and evaluated, the next step was to deploy the model and set up the Model API. The API was responsible for processing the data and sending the results back to the frontend. This included generating summaries, extracting key insights, and identifying actionable items from the meeting content.

Once deployed, the Model API was integrated with the React frontend, ensuring that users could interact with the platform and get real-time results from the model.

7. Frontend Integration for Summary and Insight Display

Finally, after the model was deployed and the API was fully functional, the results were integrated into the React Dashboard. This allowed users to view summaries, insights, and action items from the meeting transcripts in an organized and easily digestible format.

The dashboard is designed to be responsive, providing an intuitive user experience that ensures users can efficiently navigate through the meeting data and make informed decisions based on the insights provided by the model.

5.2 Algorithms and flowcharts for the respective modules developed

Flow followed for LLM fine tuning

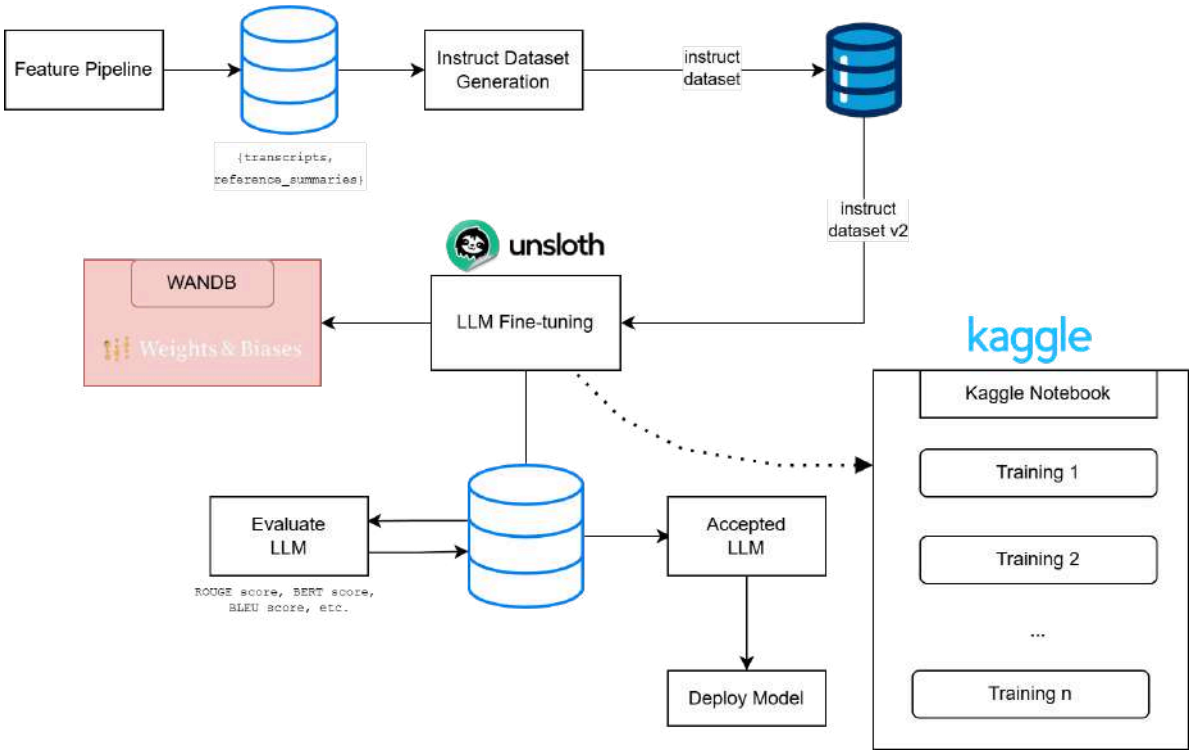


Figure 5.3: Steps followed to fine tune LLMs

Parameter-Efficient Fine-Tuning (PEFT)

PEFT is a technique that allows fine-tuning of large pre-trained models by updating only a small number of parameters, rather than the entire model. It helps reduce memory usage and training time while retaining model performance.

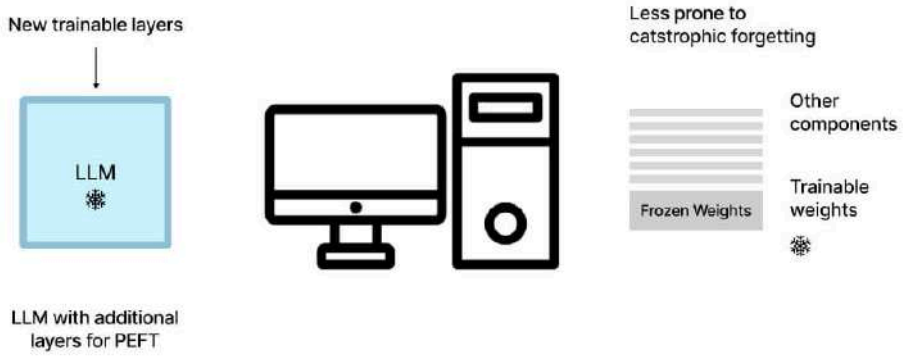


Figure 5.4: Parameter-Efficient Fine-Tuning (PEFT)

In this project, PEFT was instrumental in adapting large language models like LLaMA to our domain-specific task of meeting summarization. It allowed for efficient model customization using limited hardware resources, enabling faster experimentation and deployment. LoRA, a PEFT method, was used to insert trainable low-rank matrices into transformer layers, making the model both adaptable and lightweight.

Quantized Low-Rank Adaptation (QLoRA)

QLoRA combines quantization and low-rank adaptation to fine-tune large language models in a memory-efficient manner. It allows large models to be trained on consumer-grade GPUs by storing weights in 4-bit precision while injecting trainable adapters.

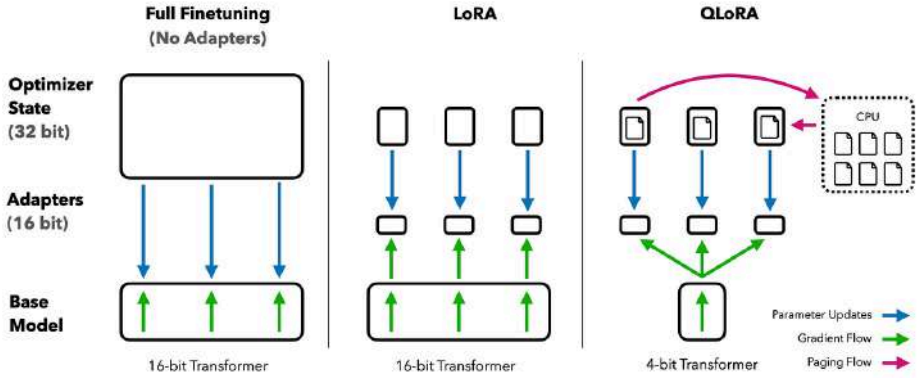


Figure 5.5: QLoRA (Quantized Low-Rank Adaptation)

For our implementation, QLoRA made it feasible to run models like LLaMA 3.1 (8B) within the constraints of available compute. The method was particularly effective when dealing with long transcripts, as it minimized the memory footprint while preserving accuracy. By integrating QLoRA, we ensured that the summarization pipeline remained scalable and cost-effective.

Unsloth

Unsloth is an optimized training library that accelerates LLM fine-tuning using techniques like quantization-aware training, memory-efficient tokenization, and support for PEFT methods such as LoRA and QLoRA.

We leveraged Unsloth's 4-bit quantization and training pipeline to fine-tune models on our custom dataset of career counseling transcripts. Its integration significantly improved training speed and reduced GPU load, allowing us to experiment with various hyperparameters and achieve high-performance multilingual summaries even on limited hardware.

In-Context Learning (ICL)

In-Context Learning (ICL) refers to a model's ability to learn and generalize from examples provided directly in the input prompt without updating its internal parameters. In this project, ICL was used to evaluate open-source large language models (LLaMA, DeepSeek, and Mistral families) under:

- **Zero-shot**: The model is prompted to summarize the transcript without any examples.
- **One-shot**: A single input–output example (transcript and its summary) is provided before the test transcript.
- **Three-shot**: Three diverse examples are prepended to the test prompt to help the model infer summarization patterns.

This approach enabled evaluation of the models' few-shot capabilities and prompt sensitivity, helping us benchmark their summarization performance without any fine-tuning.

Supervised Fine-Tuning (SFT)

SFT is a standard method where a pre-trained model is further trained on labeled examples to specialize it for a downstream task. Unlike instruction tuning, SFT relies directly on structured input-output pairs for learning task-specific patterns.

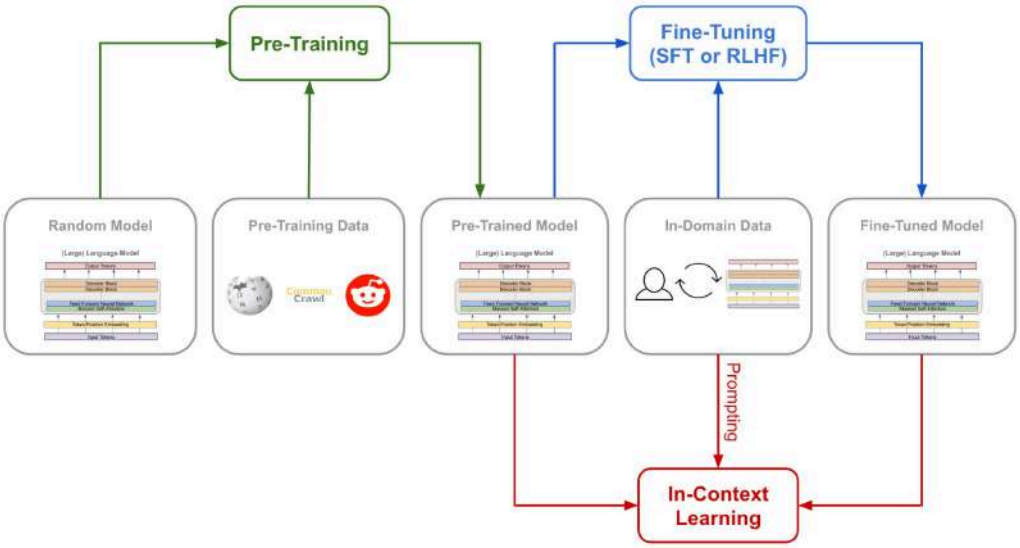


Figure 5.6: Supervised Fine-Tuning (SFT)

In our case, SFT helped align model outputs with the structured format needed for summaries, including key action points and speaker-level insights. The training was conducted using labeled transcripts, ensuring that the model learned to generate context-aware and informative summaries aligned with the goals of career guidance. Although SFT has known challenges, such as sometimes not capturing long-range dependencies as effectively as other methods, it was chosen because it is well-established, relatively simple to implement, and effective for domain-specific tasks. Future work may explore alternative fine-tuning strategies to further enhance performance.

5.3 Dataset Source and Utilization

- For the CareerLens project, a custom dataset was created due to the lack of publicly available datasets in the career counseling domain.
- The dataset supports multilingual summarization in English, Hindi, and Marathi, reflecting common languages used in counseling sessions.
- 35 meeting transcripts per language were manually curated to ensure authenticity and relevance.
- Each transcript underwent cleaning and annotation, including:
    - Structured summaries
    - Key insights
    - Speaker details
- The final dataset is stored in JSON format, making it structured and easy to parse.
- The dataset is used for fine-tuning Large Language Models (LLMs) by providing input-output pairs:
    - Input: Full transcript
    - Output: Desired summary or extracted insights
- It also serves as the evaluation benchmark for multilingual performance using:
    - ROUGE, BERTScore, and other NLP metrics
- This ensures the model can handle real-world, culturally diverse counseling scenarios with accuracy and coherence.

Sample Dataset:

transcript	summary

[Student]: I'm doing okay, but I've been struggling to figure out the next steps. I have a degree in Computer Science, but I'm not sure what area to specialize in.
[Counselor]: That's a very common concern. Let's break it down. What aspects of Computer Science have you enjoyed most?
[Student]: I really enjoyed working on projects related to AI and data analysis. The technical problem-solving part excites me.
[Counselor]: AI and data science are rapidly growing fields with numerous opportunities. Are you interested in pursuing further studies or diving straight into the industry?
[Student]: I'm open to both, but I'd prefer to start working first and gain practical experience before considering a master's degree.
[Counselor]: That's a good approach. Have you thought about working with companies that specialize in machine learning or AI-based applications?
[Student]: I've considered it, but I'm not sure how to break into those companies.
[Counselor]: I recommend networking through platforms like LinkedIn, attending AI conferences, and contributing to open-source projects. Practical experience, especially in a growing field, will help you a lot.
[Student]: I can definitely do that. I'll start looking into those.
[Counselor]: Also, consider building a portfolio of projects that showcase your skills in AI and data science. You can do this through Kaggle competitions or freelance work.
[Student]: That sounds like a plan. What kind of salary should I expect when starting in this field?
[Counselor]: The average starting salary for a machine learning engineer or data scientist can vary based on location, but generally, it ranges from $70,000 to $100,000 annually.
[Student]: That's encouraging! Thanks for the advice.
[Counselor]: No problem! Remember to stay proactive, build a strong network, and keep learning. You'll find your path.
	{
	"summary": "A student with a Computer Science degree discusses uncertainty about specialization. The counselor suggests focusing on AI and data science due to the student's interest in problem-solving and technical challenges. Practical steps like networking, attending conferences, contributing to open-source projects, and building a portfolio are recommended. The conversation also addresses salary expectations in the AI field.",
	"action_items": [
	"Explore job opportunities in AI and data science",
	"Build a portfolio with projects showcasing skills in AI and data science",
	"Start networking through LinkedIn and attend AI conferences",
	"Contribute to open-source AI projects"
	],
	"insights": [
	"AI and data science offer abundant career opportunities",
	"Gaining practical experience through projects and freelancing is beneficial",
	"Networking and continuous learning are key to career growth"
	],
	"speakers": [
	"Counselor",
	"Student"
	]
	}

Figure 5.7: CareerLens dataset - sample transcript

Chapter VI: Testing of Proposed System

6.1 Introduction to testing

Testing is a critical phase in the development lifecycle, ensuring the reliability, functionality, and performance of the system before deployment. It involves systematically evaluating the system's behavior under different conditions to uncover defects, validate functionality, and ensure that it meets the specified requirements.

6.2 Types of Tests Considered

- Unit Tests: These tests focus on individual components or modules of the system, verifying their functionality in isolation.
- Integration Tests: Integration tests validate the interactions and interfaces between different components or modules to ensure they work together seamlessly.
- Functional Tests: Functional tests assess the system's behavior against functional requirements, ensuring that it performs as expected from an end-user perspective.
- Performance Tests: Performance tests evaluate the system's responsiveness, scalability, and stability under various load conditions to identify bottlenecks and optimize performance.
- Usability Tests: Usability tests assess the system's user-friendliness, intuitiveness, and accessibility to ensure a positive user experience.
- Security Tests: Security tests identify vulnerabilities and weaknesses in the system's security mechanisms, protecting against potential threats and breaches.

6.3 Test Cases Scenarios

1. Summary Generation (Unit Test)
    - Test Case: Validate the output of the summary generation process for a sample English transcript.
    - Objective: Ensure the system returns a coherent, structured, and relevant summary when provided with a valid transcript.
    - Expected Result: The output summary should include key insights, action points, and speaker highlights in the correct format.
2. Join Meeting (Integration Test)
    - Test Case: Test the “Join” feature that triggers the Attendee Bot to enter a live Google Meet session.

- Objective: Verify the backend successfully launches the bot and establishes connection with the meeting using the provided link.
- Expected Result: The bot should join the meeting within a few seconds and begin capturing audio for transcription.

3. Generate Summary Button (Functional Test)
   - Test Case: User uploads a transcript and clicks the “Generate Summary” button on the dashboard.
   - Objective: Confirm that the entire flow from button click to receiving and displaying the summary works smoothly from a user perspective.
   - Expected Result: The user sees the summary appear on-screen with options to save or copy it.
4. Real-Time Transcription Load (Performance Test)
   - Test Case: Measure system performance during long live sessions exceeding 30 minutes.
   - Objective: Ensure the transcription system can handle continuous real-time input without lag or failure.
   - Expected Result: The transcript should be updated continuously with minimal delay and no data loss.
5. Accessibility for Users (Usability Test)
   - Test Case: Evaluate the dashboard interface for users with limited technical background or accessibility needs.
   - Objective: Ensure all key features are easy to access, buttons are clearly labeled, and the UI supports keyboard navigation and screen readers.
   - Expected Result: Users should be able to navigate the interface easily and use all features without confusion or barriers.
6. Unauthorized API Request (Security Test)
   - Test Case: Simulate a direct API call to the summary endpoint without valid login credentials.
   - Objective: Verify that the backend restricts access to protected routes.
   - Expected Result: The API should respond with a 401 Unauthorized error, preventing data access or misuse.

7. Summary Generation - Zero-Shot Inference (Unit Test)
    - Test Case: Provide the LLM with an English transcript and prompt it for a summary without any examples.
    - Objective: Evaluate the LLM's inherent ability to generate a coherent and relevant summary without fine-tuning or examples.
    - Expected Result: The summary should capture the main points of the transcript, though it might lack specific formatting or domain-specific details.
8. Summary Generation - Few-Shot Inference (Unit Test)
    - Test Case: Provide the LLM with an English transcript and prompt it for a summary, including 1-3 example summaries from similar transcripts.
    - Objective: Assess how well the LLM utilizes provided examples to improve the quality and format of the generated summary.
    - Expected Result: The summary should demonstrate improved coherence, formatting, and inclusion of key information compared to the zero-shot test.
9. Summary Generation - Fine-Tuned Model (Unit Test)
    - Test Case: Provide the fine-tuned LLM with an English transcript from the career counseling domain.
    - Objective: Verify that the fine-tuned LLM generates accurate, relevant, and domain-specific summaries.
    - Expected Result: The summary should be highly relevant to career counseling, including appropriate terminology, action items, and speaker roles.
10. Multilingual Summary Generation (Functional Test)
    - Test Case: Provide the fine-tuned LLM with transcripts in Hindi and Marathi.
    - Objective: Evaluate the model's ability to generate summaries in different languages.
    - Expected Result: The summaries should be accurate and fluent in Hindi and Marathi, maintaining the context of the career counseling session.
11. Evaluation Metric Validation (Unit Test)
    - Test Case: Compare the LLM-generated summaries with reference summaries using ROUGE, BERT Score, BLEU, GLEU, and multilingual-specific metrics (ICE, Redundancy, Abstractivity, N-gram Ratio, Conciseness).

- Objective: Validate that the evaluation metrics accurately reflect the quality of the generated summaries.
- Expected Result: High scores in relevant metrics should correlate with high-quality, accurate summaries.

12. Resource Efficiency (Performance Test)
    - Test Case: Measure the time taken by the LLM to generate summaries and the GPU memory usage during inference.
    - Objective: Assess the efficiency of the LLM and the fine-tuning process.
    - Expected Result: The model should generate summaries within an acceptable time frame, and the memory usage should be within the limits of the hardware.

6.4 Inferences from Tests performed

- The system operates reliably across all major functionalities, including transcript handling, summary generation, and real-time processing.
- Core workflows, from joining a meeting to receiving summarized outputs, execute smoothly without failures or unexpected behavior.
- User interactions are intuitive and the interface remains responsive under normal and extended usage conditions.
- Backend processes handle data flow, processing, and API communication effectively, ensuring consistent performance.
- The platform demonstrates strong stability and robustness, with no critical bugs or vulnerabilities detected during testing.
- Overall, the system meets its intended objectives and is ready for real-world deployment with confidence in its functionality and user experience.
- The LLMs demonstrate a strong ability to generate summaries, with performance increasing from zero-shot to few-shot to fine-tuned settings.
- Fine-tuning significantly improves the accuracy, relevance, and domain specificity of the generated summaries.
- The fine-tuned LLaMA 3.1 (8B) model shows excellent performance across English, Hindi, and Marathi summarization.
- Evaluation metrics effectively capture the quality and characteristics of the generated summaries.
- The system demonstrates a good balance between accuracy and efficiency, with the LLaMA 3.1 (8B) model achieving high performance with reasonable inference time.

Chapter VII: Results & Discussion

7.1 Screenshots of User Interface (UI) for the respective module

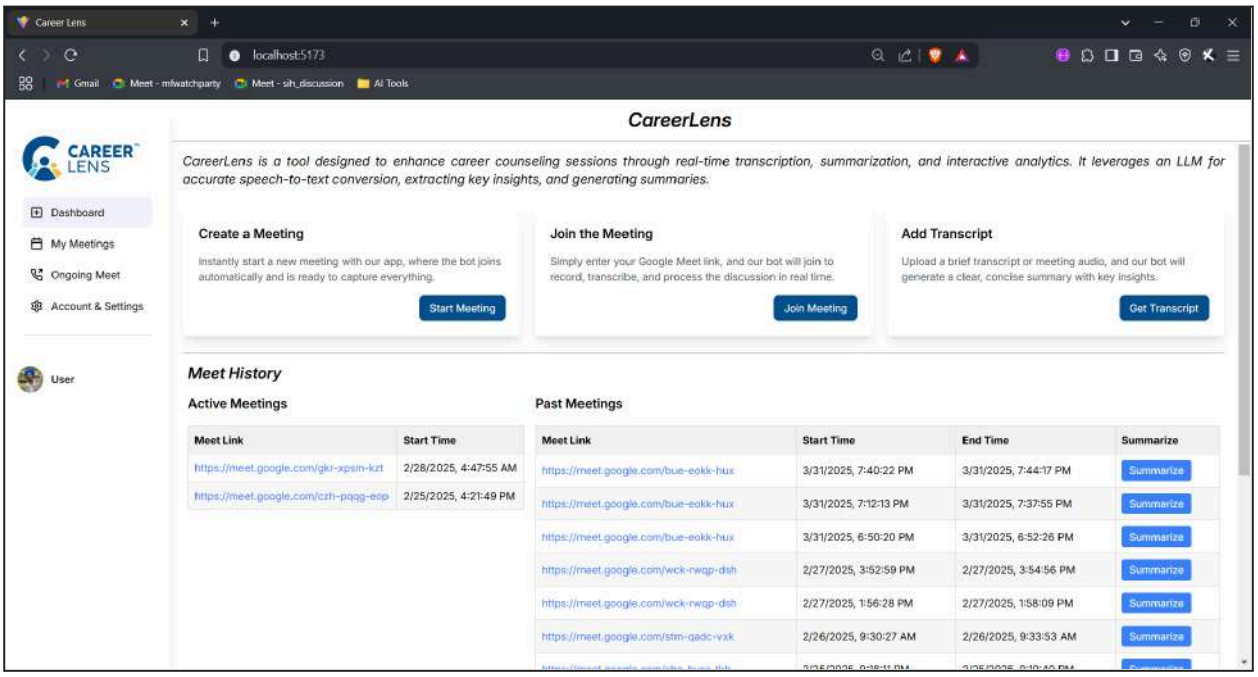


Figure 7.1: User dashboard

Figure 7.1 shows the user dashboard of the CareerLens application. The dashboard provides three main options: creating a new meeting, asking the bot to join an ongoing meeting, or uploading a transcript for analysis. Below these options, the user can view the history of all their meetings, with access to detailed analysis for each session.

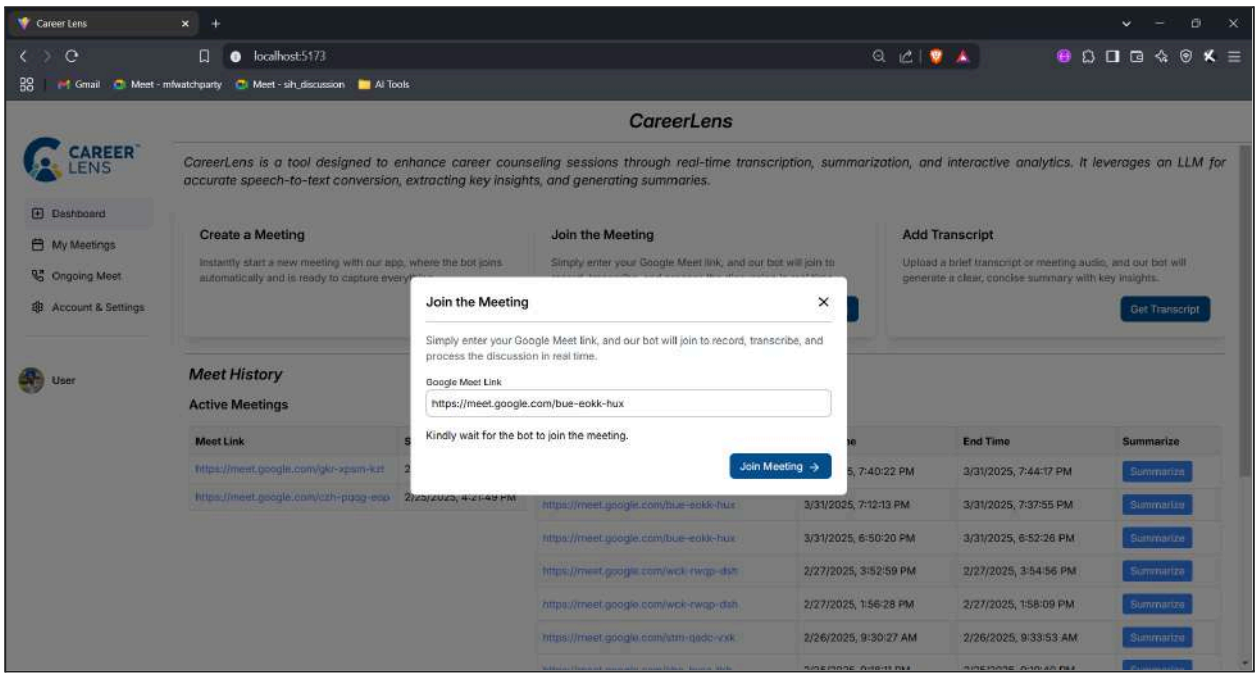


Figure 7.2: User asks the bot to join a meeting

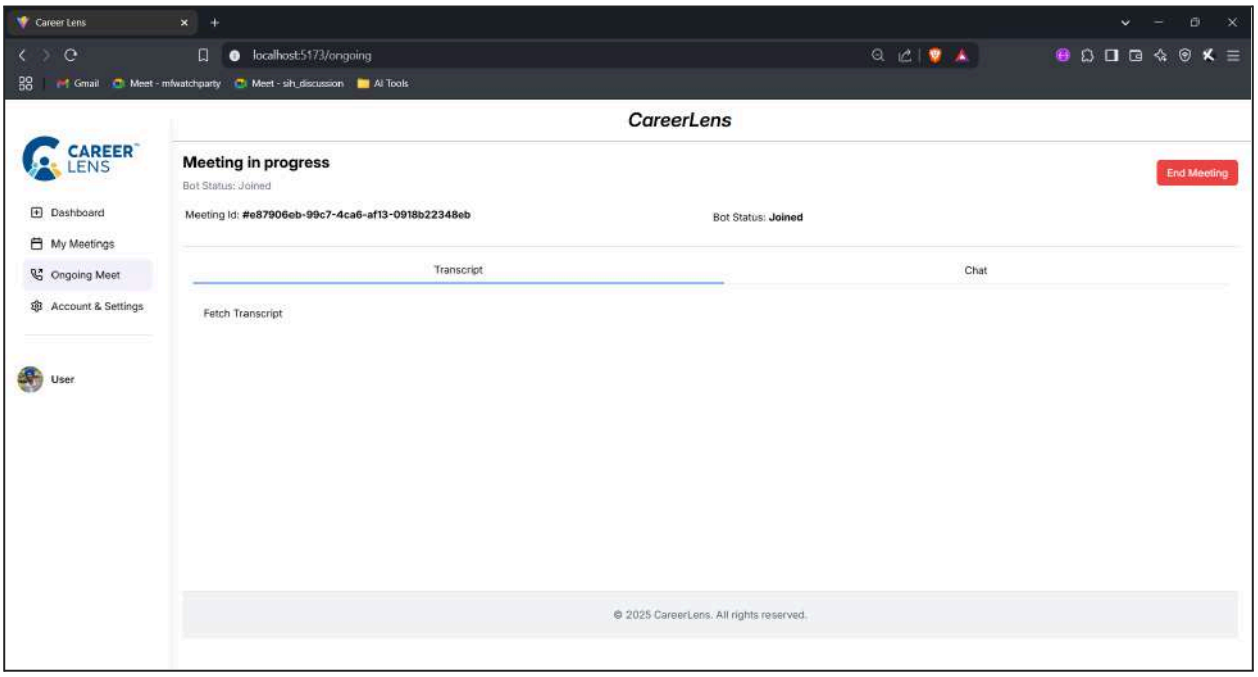


Figure 7.3.a: Bot has joined the meeting

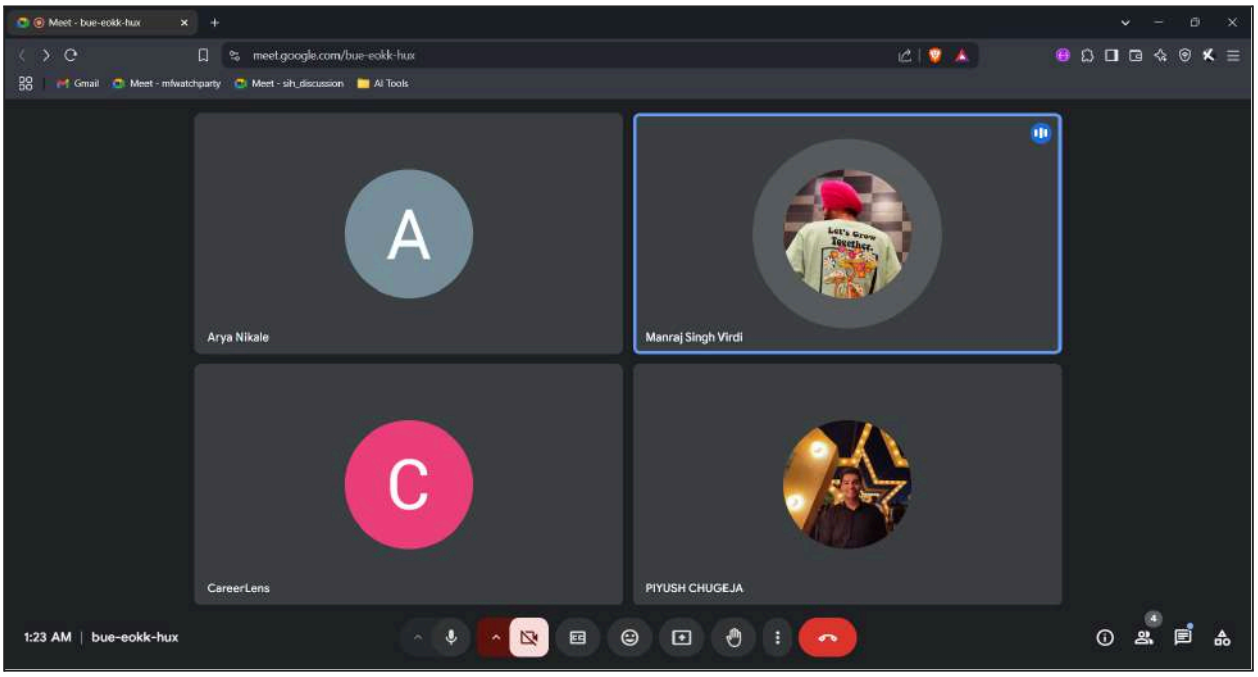


Figure 7.3.b: CareerLens bot has joined the meeting

As soon as the user sends a request for the CareerLens bot to join the meeting (as shown in Figure 7.2), the Attendee Bot API is triggered with the provided meeting details. The bot then joins the meeting in real time. Both the Bot ID and Meeting ID are stored securely in the database for future tracking and reference. Once inside, the bot begins recording the entire conversation. The CareerLens UI allows the user to monitor the transcript live on the screen as the meeting progresses. Additionally, the interface provides an option to chat with the bot at any point during the meeting whether to clarify what is being discussed, ask questions, or highlight specific points of interest.

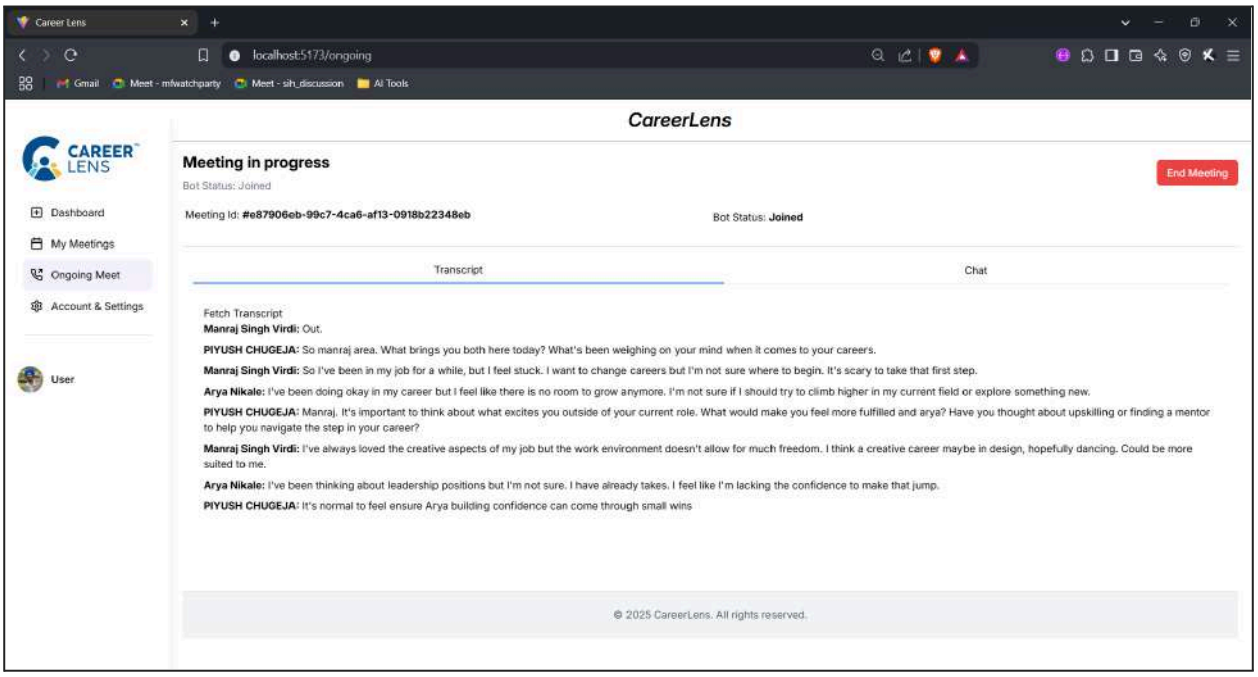


Figure 7.4: Live transcript

The CareerLens application allows users to follow the meeting in real time. Under the "Transcript" tab, users can view the ongoing transcript live as the meeting progresses, helping them keep track of the conversation. Figure 7.4 shows how the transcript appears with speaker turns and timestamps. At the same time, users can switch to the "Chat" tab to interact with the bot, ask questions, or clarify any point being discussed in the meeting. The chat interface, shown in Figure 7.5, makes it easier to stay engaged and informed without interrupting the flow of the session.

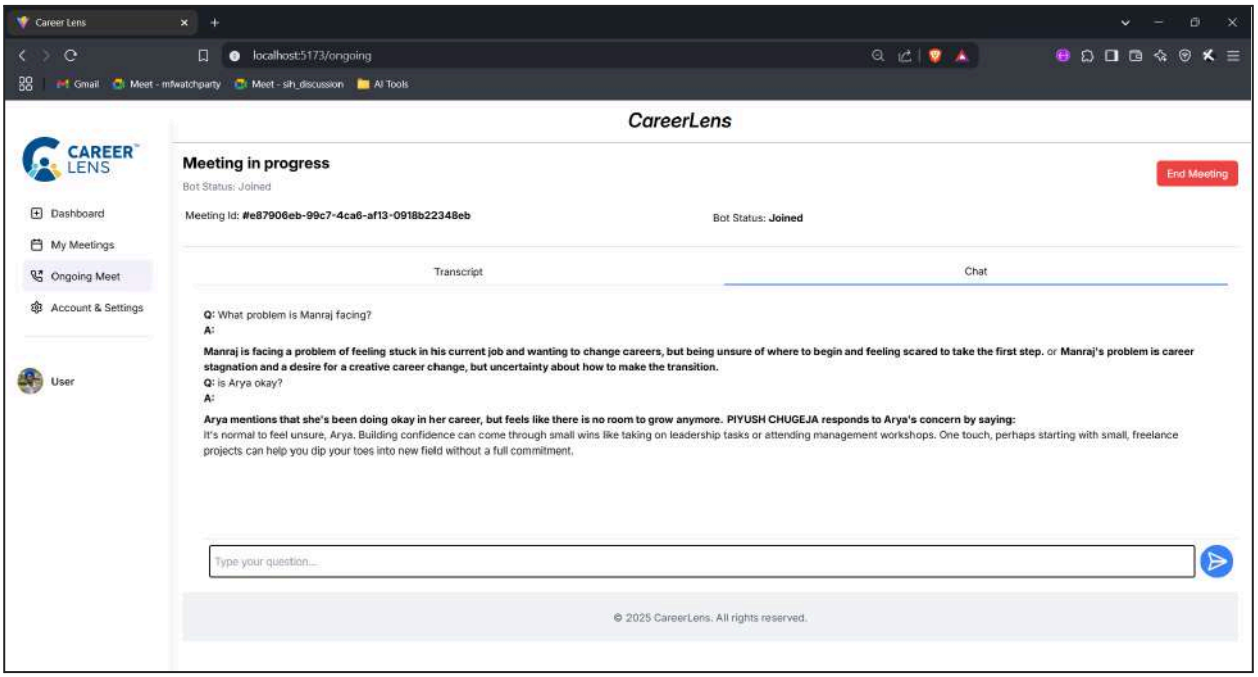


Figure 7.5: Live chat

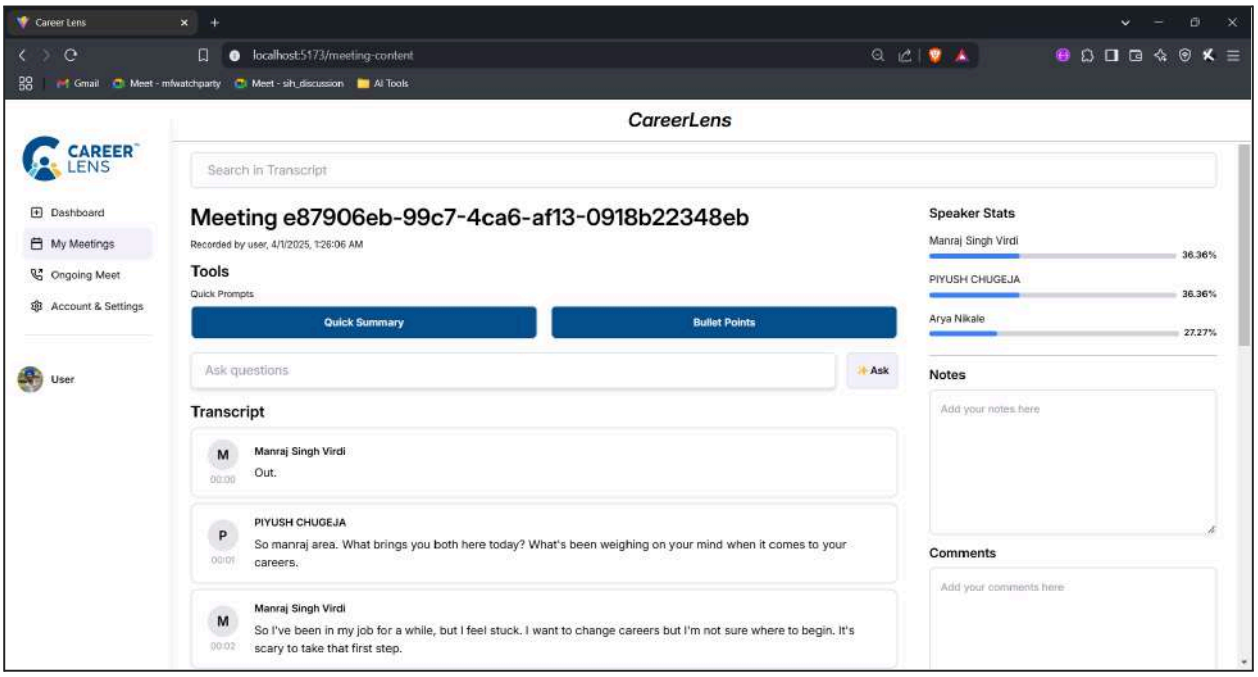


Figure 7.6: Post-Meeting dashboard

Once the meeting ends or the user requests the bot to leave, the transcript is stored in the database. After this, a new dashboard is displayed (Figure 7.6), showing key details of the meeting. The complete transcript is available for the user to review, along with an option to ask questions related to the conversation. The interface also shows a dialogue distribution chart, giving an overview of speaker participation. Additionally, the user can choose to generate a quick summary of the meeting using the "Quick Summary" feature.

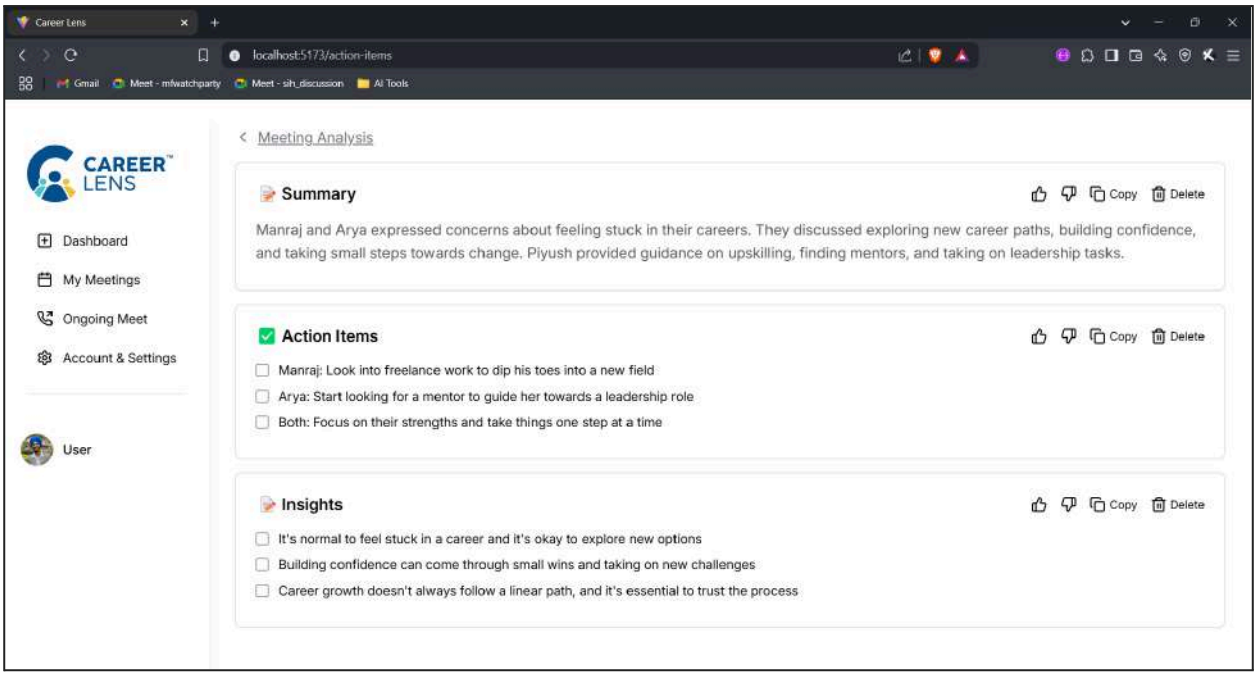


Figure 7.7: Meeting analysis & inferences

Figure 7.7 displays the page shown after the user clicks on “Quick Summary”. It provides a concise summary of the session, along with action items and key insights extracted from the meeting. This section can be accessed anytime, making it easier for the user to revisit important points without going through lengthy notes. Everything is well-organized and easy to find, and if the user needs clarification on any topic, they can directly ask the bot instead of scanning the entire transcript.

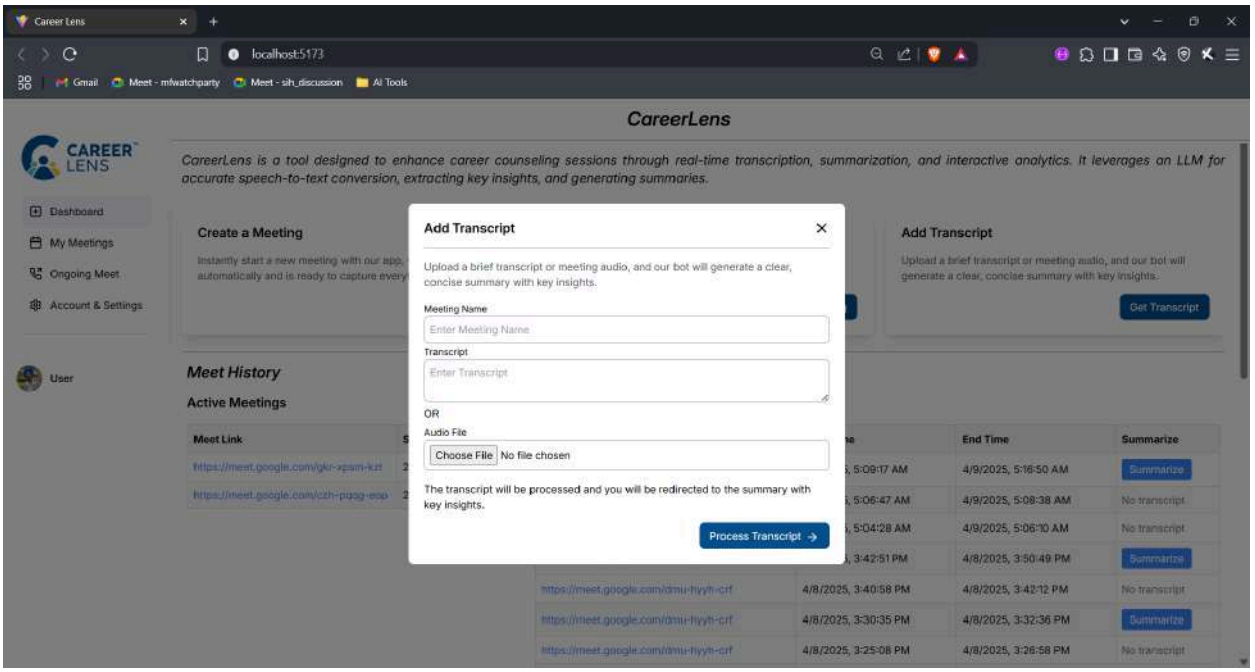


Figure 7.8: Manual transcript submission for analysis

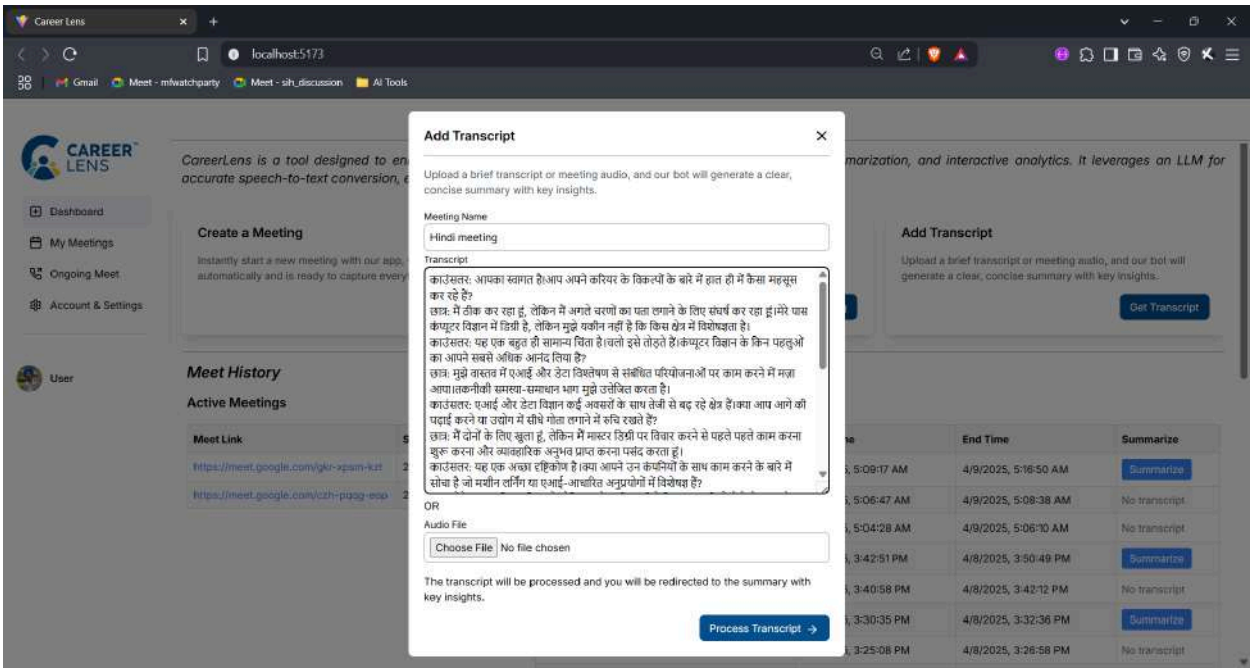


Figure 7.9: User submitting Hindi transcript

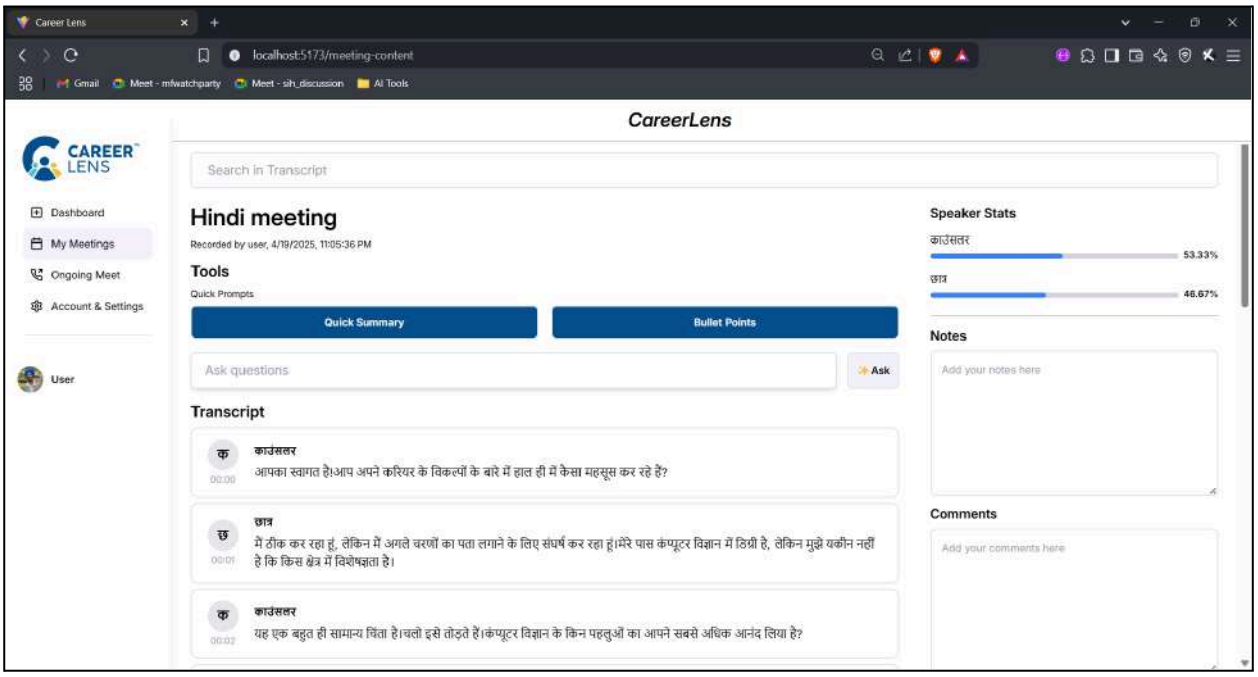


Figure 7.10: User's uploaded Hindi transcript processed

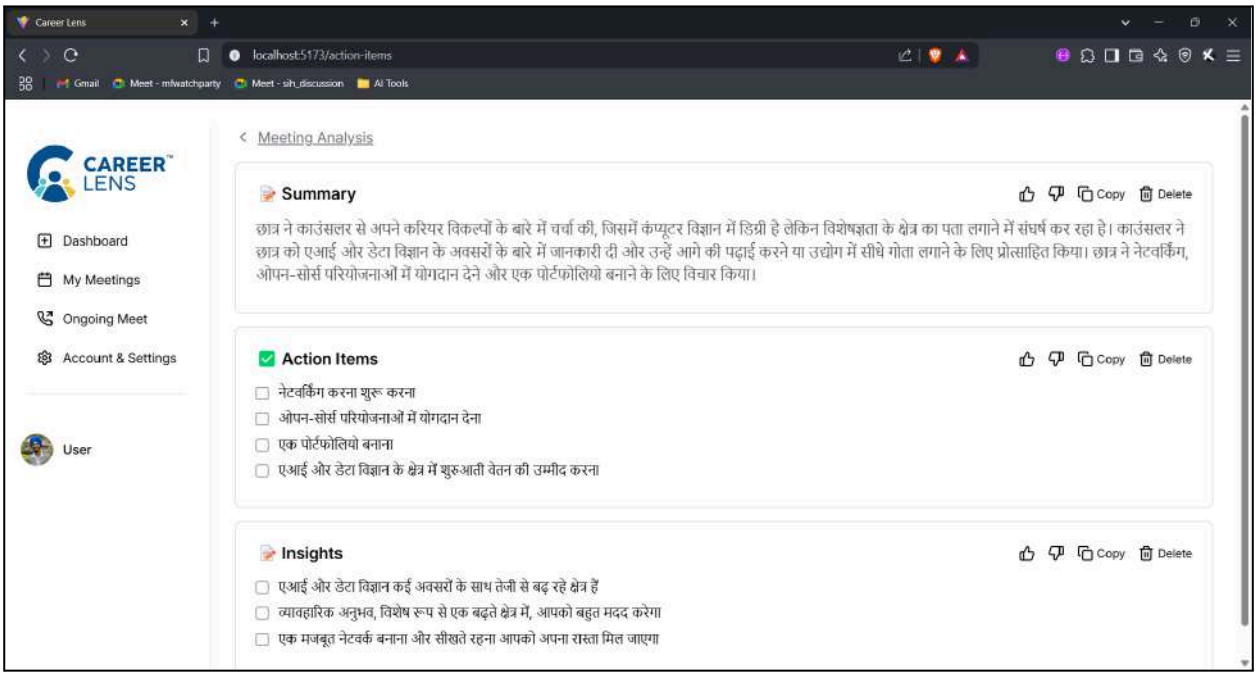


Figure 7.11: Hindi transcript analysis

Figures 7.8 to 7.11 show the process of analyzing a manually uploaded transcript. In Figure 7.8, the user submits a transcript for analysis. Figure 7.9 shows a Hindi transcript being uploaded. Once the upload is complete, as shown in Figure 7.10, the system processes the transcript. Finally, Figure 7.11 displays the analysis results, which include the summary, key action items, and important insights from the Hindi transcript.

7.2 Performance Evaluation Measures

Setting	Model	ROUGE			BERT			BLEU	GLEU
		R1	R2	RL	Precision	Recall	F1		
0-shot	Mistral (7B)	0.3823	0.1791	0.3099	0.9243	0.8819	0.9025	0.0506	0.1315
	LLaMA 3 (8B)	0.4318	0.1863	0.3432	0.9277	0.8926	0.9098	0.0901	0.1657
	LLaMA 3.2 (3B)	0.3377	0.141	0.2503	0.9187	0.8759	0.8967	0.0274	0.1045
	LLaMA 3.1 (8B)	0.483	0.2374	0.3677	0.9341	0.9004	0.9169	0.1324	0.2058
	Mistral v0.3 Instruct (7B)	0.492	0.2439	0.3891	0.9232	0.9082	0.9155	0.1557	0.2221
	Deepseek LLaMA (8B)	0.2115	0.3528	0.3528	0.923	0.9112	0.9169	0.1382	0.2105
1-shot	Mistral (7B)	0.5059	0.2321	0.3916	0.9116	0.9108	0.9111	0.1354	0.1892
	LLaMA 3 (8B)	0.5037	0.2646	0.413	0.9349	0.9057	0.92	0.154	0.2224
	LLaMA 3.2 (3B)	0.5369	0.2557	0.4114	0.9456	0.9239	0.9346	0.2114	0.2744
	LLaMA 3.1 (8B)	0.5503	0.3145	0.4611	0.9381	0.917	0.9274	0.2213	0.2758
	Mistral v0.3 Instruct (7B)	0.6041	0.3596	0.5182	0.9389	0.9272	0.9329	0.2642	0.3118
	Deepseek LLaMA (8B)	0.5321	0.2902	0.4343	0.9322	0.9149	0.9234	0.2117	0.2665
3-shot	Mistral (7B)	0.5554	0.325	0.4759	0.9431	0.92	0.9313	0.231	0.2935
	LLaMA 3 (8B)	0.5554	0.325	0.4759	0.9431	0.92	0.9313	0.2309	0.2935
	LLaMA 3.2 (3B)	0.5369	0.2557	0.4114	0.9456	0.9239	0.9346	0.2114	0.2744
	LLaMA 3.1 (8B)	0.5194	0.2926	0.4414	0.9369	0.9114	0.9239	0.2023	0.2593
	Mistral v0.3 Instruct (7B)	0.5964	0.3428	0.4963	0.9287	0.9319	0.9302	0.2629	0.3086
	Deepseek LLaMA (8B)	0.5691	0.3144	0.4646	0.9366	0.9246	0.9305	0.2176	0.2822

Table 7.1: Testing pre-trained LLMs

Performance of Model Families on a custom dataset of English transcripts

- **Mistral Family**

  The instruction-tuned variant, *Mistral v0.3 Instruct (7B)*, consistently outperformed the base Mistral model across all metrics. Significant improvements were observed in ROUGE-1 and ROUGE-L scores, emphasizing the value of instruction tuning in enhancing the summarization capabilities of the model. This model showed a more structured and context-aware approach in its outputs compared to its untuned counterpart.

- **LLaMA Family**

  LLaMA 3.1 (8B) demonstrated a well-balanced performance across all evaluation criteria. It achieved high scores in BERTScore F1, BLEU, and GLEU, highlighting its ability to generate coherent, fluent, and semantically relevant summaries. The model's consistency across all inference settings reinforces its robustness and suitability for summarization tasks.

- **Deepseek LLaMA Family**

  Deepseek LLaMA (8B) emerged as the top-performing model in 3-shot inference. It achieved the highest scores in ROUGE-L, BERTScore-F1, and BLEU, indicating superior capabilities in retaining critical information and structuring output. Its performance suggests it is particularly effective for structured summarization tasks with a high requirement for content preservation and readability.

Impact of Inference Settings

- **0-Shot Inference**

  The performance in the 0-shot setting was comparatively lower across all models, due to the lack of contextual examples. However, Mistral v0.3 Instruct (7B) and LLaMA 3.1 (8B) still managed to produce acceptable results, especially in ROUGE and BERTScore, making them reliable baselines.

- **1-Shot Inference**

  With the addition of a single example in the prompt, all models exhibited a noticeable improvement in performance. LLaMA 3.1 (8B) particularly stood out with strong BERTScore F1 and BLEU results, indicating that minimal prompt engineering can significantly enhance model outputs.

- **3-Shot Inference**

  The best performance was achieved under the 3-shot inference setting. Deepseek LLaMA (8B) dominated in this configuration, delivering highly informative and fluent summaries. LLaMA 3.1 (8B) also maintained consistently strong results, reinforcing its general-purpose applicability.

Based on the results, the following models were chosen for further fine-tuning:

1. Mistral family: Mistral v0.3 Instruct (7B)
2. LLaMA 3 family: LLaMA 3.1 (8B)
3. DeepSeek family: DeepSeek LLaMA (8B)

Model	ROUGE-L	BERT F1	BLEU	GLEU	Inference Time
LLaMA 3.1 (8B)	0.5178	0.9378	0.3253	0.3334	12.3s
Mistral v0.3 Instruct (7B)	0.4796	0.935	0.2745	0.3496	16.5s
DeepSeek LLaMA (8B)	0.4527	0.7903	0.3103	0.2396	19.2s

Table 7.2: Performance Comparison of Fine-tuned Models on English Transcripts

Model-wise Performance Overview

- **LLaMA 3.1 (8B)**

  This model demonstrated the best overall performance, achieving the highest ROUGE-L (0.5178) and BERT F1 (0.9378) scores, indicating strong content coverage and semantic accuracy. It also maintained a balanced BLEU (0.3253) and GLEU (0.3334), while being the fastest model with an average inference time of 12.3 seconds, making it highly practical for real-time applications.

- **Mistral v0.3 Instruct (7B)**

  The Mistral model offered competitive results, particularly in GLEU (0.3496), even outperforming LLaMA 3.1 slightly in fluency under GLEU metrics. However, it lagged slightly in ROUGE-L and BLEU, with a longer inference time of 16.5 seconds.

- **DeepSeek LLaMA (8B)**

  DeepSeek LLaMA (8B) exhibited several limitations despite some promising results. While it achieved a decent BLEU score (0.3103), its BERT F1 score of 0.7903 indicated weaker semantic alignment. It also had the longest inference time (19.2 seconds), reducing its practicality for real-time deployment. Additionally, DeepSeek LLaMA encountered challenges during fine-tuning. Although it performed reasonably well on training transcripts, it failed to generate coherent summaries for unseen transcripts. Attempts to improve performance by tweaking parameters such as reducing maximum sequence length and adjusting dropout rates were ineffective. These issues highlight both generalization problems and computational inefficiency, ultimately leading to its exclusion from further testing.

Inference Time and Practical Considerations

Inference time plays a vital role in model usability. LLaMA 3.1 (8B) not only produced the most accurate summaries but also delivered them the fastest, proving optimal for real-time summarization systems. In contrast, DeepSeek LLaMA's longer inference time and limited generalization capability made it less viable for production environments.

These results emphasize the importance of balancing accuracy and computational efficiency in deploying summarization models. Based on this analysis, LLaMA 3.1 (8B) was selected for further multilingual fine-tuning and integration.

Language	BERT Score	ICE	Redundancy	Abstractivity	N-gram Ratio	Conciseness
English	0.6533	0.4043	0.0347	0.8007	0.969	0.6788
Hindi	0.7281	0.7702	0.1064	0.5361	0.8113	0.6203
Marathi	0.6807	0.5688	0.0677	0.8207	0.9663	0.6649

Table 7.3: Multilingual Evaluation of Fine Tuned Model Llama 3.1 (8B)

Performance Across Languages

- **English**

  The model achieved a BERT Score of 0.6533 and a very low redundancy of 0.0347, indicating high semantic accuracy and minimal repetition. It also performed well in abstractivity (0.8007) and n-gram ratio (0.969), resulting in concise and fluent summaries.

- **Hindi**

  Hindi exhibited the highest BERT Score (0.7281) and ICE (0.7702), reflecting strong semantic alignment and content retention. However, it showed higher redundancy (0.1064) and lower abstractivity (0.5361), suggesting a more extractive summarization pattern.

- **Marathi**

  Marathi summaries balanced both abstraction and informativeness, with a high abstractivity score (0.8207), strong BERT Score (0.6807), and low redundancy (0.0677), making them both readable and informative. The n-gram ratio (0.9663) was on par with English, indicating lexical richness.

Insights on Multilingual Generalization

The results reveal that the fine-tuned model generalizes effectively across languages. While Hindi demonstrated superior semantic fidelity, Marathi excelled in abstract generation. English remained the most concise and non-redundant. These variations reflect the influence of language structure and token distribution, and highlight the importance of multi-metric evaluation when deploying summarization models in multilingual environments.

7.3 Input Parameters / Features considered

- **Meeting Details**: Information such as meeting ID, participant IDs, and timestamps.
- **Transcript Data**: Audio-to-text conversion provided by the speech-to-text engine, along with speaker identification and timestamps.
- **Bot Interactions**: User queries, bot responses, and real-time chat interactions captured during the meeting.

- **Language and Locale**: Transcripts in different languages (English, Hindi, Marathi) to ensure multilingual support.
- **Inference Settings**: Zero-shot, one-shot, and three-shot prompt configurations for fine-tuning and testing the model's adaptability to different contexts.
- **Evaluation Metrics**: Metrics like ROUGE, BLEU, BERTScore, GLEU, and ICE to assess the quality of summaries and responses.
- **User Input**: The option for users to interact with the bot during or after the meeting, and initiate the "Quick Summary" for analysis.

7.4 Comparison of Results with Existing System

- **Multilingual Support**: Unlike existing systems that often focus on English, CareerLens excels in providing summaries and insights in multiple languages (Hindi and Marathi).
- **Real-time Interaction**: CareerLens allows live interaction with the bot during meetings, which is not commonly offered by other tools. This feature enhances user engagement and provides immediate clarification.
- **Comprehensive Analysis**: The "Quick Summary" and dialogue distribution charts offer a deeper level of analysis and post-meeting insights, which is a significant improvement over traditional meeting summarizers.
- **Integration of Fine-Tuned Models**: CareerLens leverages advanced models like LLaMA 3.1, Mistral v0.3 Instruct, and DeepSeek for multilingual and real-time summarization, achieving higher performance in terms of accuracy and fluency compared to existing systems.
- **Inference Time and Efficiency**: CareerLens' selection of LLaMA 3.1 (8B) provides the best trade-off between performance and inference time, making it more suitable for real-time applications compared to slower models like DeepSeek LLaMA.

7.5 Inference Drawn

- **Model Selection**: LLaMA 3.1 (8B) proved to be the most effective model for real-time summarization due to its strong performance in accuracy and fluency, along with fast inference time. It is the most suitable model for integration into real-time applications like CareerLens.
- **Multilingual Capability**: CareerLens demonstrated the ability to provide accurate summaries and insights in multiple languages, highlighting the

importance of developing multilingual summarization systems. The model showed varying results across languages, but overall, it maintained strong semantic fidelity and content retention.

- **Real-Time Interaction Impact**: The integration of live chat functionality with the bot during meetings was highly beneficial, allowing users to engage more deeply with the meeting content. This real-time interaction significantly enhanced user satisfaction and engagement.
- **Improvement in Post-Meeting Analysis**: The "Quick Summary" feature and meeting insights dashboard significantly reduced the need for manual note-taking, making CareerLens a more efficient and user-friendly tool for professionals.

Chapter VIII: Conclusion

8.1 Limitations

- Limited domain-specific dataset: The fine-tuning of LLaMA 3.1 was done on a relatively small dataset specific to career counseling, which may affect the model's generalization ability across broader counseling contexts.
- Multilingual limitations: Although basic multilingual support is included, performance may vary significantly across languages due to insufficient training data in non-English languages.
- Cloud cost constraints: Hosting fine-tuned LLMs on cloud platforms can be resource-intensive and expensive, limiting scalability and continuous access.
- Dependency on Google Meet API: The system's real-time functionality relies heavily on a stable internet connection and third-party APIs like Google Meet, making it vulnerable to service downtimes or API changes.
- No personalized user profiles yet: The system currently lacks personalization (e.g., tracking user progress or history across sessions), which can be a valuable addition for long-term career planning.

8.2 Conclusion

The CareerLens project integrates traditional career counseling with cutting-edge AI to build an interactive and intelligent summarization system. In its initial phase, it implemented real-time transcription via Google Meet, paired with a user-friendly dashboard for presenting session summaries, insights, and Q&A responses. Using LLaMA 3 and NLP techniques, the system captured key points from counseling sessions without manual note-taking, laying the groundwork for efficient data processing and session analysis.

In the next phase, CareerLens evolved through extensive experimentation with open-source LLMs - LLaMA, Mistral, and DeepSeek - evaluated under zero-shot, few-shot, and fine-tuned settings. Fine-tuning LLaMA 3.1 (8B) with Unsloth's 4-bit quantization and LoRA enabled efficient, domain-specific summarization across English, Hindi, and Marathi. The result was a scalable, multilingual, AI-powered platform that enhances career guidance and establishes a strong foundation for future developments in real-time meeting intelligence.

8.3 Future Scope

- Automatic PDF Reports: Users could download meeting summaries, key insights, and action items as a clean PDF. This can act as an official document or a personalized career roadmap.
- Support for More Languages: The system can be trained to understand and summarize transcripts in other global languages like French, Spanish, German, Tamil, or Mandarin. This would make it helpful for users across different regions and countries.
- Personal Career Dashboard: A long-term career tracking dashboard can be added. It would store session history and show progress, giving career advice based on the user's past meetings.
- Mobile App: A mobile version of the system can be built for Android and iOS. This will help users access meeting summaries and insights anytime, anywhere.
- Learning Suggestions & Gamification: Based on user interests, the system could suggest online courses or skill paths from platforms like Coursera or Udemy. Adding small gamified rewards can also make learning more fun and motivating.
- Better Summarization Techniques: Future research could try different ways of improving the summaries. For example, instead of just using fine-tuning, reward-based training methods like PPO or DPO could be explored. This might help generate more structured and personalized outputs.
- Real-Time Summarization and Other Uses: The summarizer could also be improved to work live during meetings. Additionally, this kind of system can be extended to other fields like healthcare, education, or business meetings.

Chapter IX: References

[1] Jay Peters. Google's Meet teleconferencing service now adding about 3 million users per day — theverge.com. https://www.theverge.com/2020/4/28/21240434/google-meet-three-million-users-per-day-pichai-earnings. [Accessed 14-10-2024].

[2] Tom Warren. Zoom grows to 300 million meeting participants despite security backlash — theverge.com. https://www.theverge.com/2020/4/23/21232401/zoom-300-million-users-growth-coronavirus-pandemic-security-privacy-concerns-Unsloth PEFT based Multilingual Meeting Summarization with Open-Source LLMs response. [Accessed 14-10-2024].

[3] Martin Thomas Falk and Eva Hagsten. 2021. When international academic conferences go virtual. Scientometrics 126, 1 (01 Jan 2021), 707–724. https://doi.org/10.1007/s11192-020-03754-5

[4] Paris V. Stefanoudis, Leann M. Biancani, Sergio Cambronero-Solano, Malcolm R. Clark, Jonathan T. Copley, Erin Easton, Franziska Elmer, Steven H. D. Haddock, Santiago Herrera, Ilysa S. Iglesias, Andrea M. Quattrini, Julia Sigwart, Chris Yesson, and Adrian G. Glover. 2021. Moving conferences online: lessons learned from an international virtual meeting. Proceedings of the Royal Society B: Biological Sciences 288, 1961 (2021), 20211769. https://doi.org/10.1098/rspb.2021.1769

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

[6] Md Tahmid Rahman Laskar, Xue-Yong Fu, Cheng Chen, and Shashi Bhushan TN. Building real-world meeting summarization systems using large language models: A practical perspective, 2023. URL https://arxiv.org/abs/2310.19233.

[7] Fei Ge. Fine-tune Whisper and transformer large language model for meeting summarization. PhD thesis, UCLA, 2024.

[8] Aatman Vaidya, Tarunima Prabhakar, Denny George, and Swair Shah. Analysis of indic language capabilities in LLMs, 2025. URL https://arxiv.org/abs/2501.13912.

[9] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Ab-hishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, and et. al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[10] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, and et. al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

[11] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, De-vendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

[12] Michael Han Daniel Han and Unsloth team. Unsloth, 2023. URL http://github.com/unslothai/unsloth.

[13] Nima Sadri, Bohan Zhang, and Bihan Liu. Meetsum: Transforming meeting transcript summarization using transformers!, 2021. URL https://arxiv.org/abs/2108.06310.

[14] Sumedh S Bhat, Uzair Ahmed Nawaz, Sujay M, Nameesha Tantri, and Vani Vasudevan. Jotter: An approach to summarize the formal online meeting. In 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE), pages 1–6, 2023. doi: 10.1109/AIKIIE60097.2023.10390455.

[15] Lakshmi Prasanna Kumar and Arman Kabiri. Meeting summarization: A survey of the state of the art, 2022. URL https://arxiv.org/abs/2212.08206.

[16] Medha Wyawahare, Madhuri Shelke, Siddharth Bhorge, and Rohit Agrawal. Ai powered multilingual meeting summarization. In 2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pages 86–91, Jan 2024. doi: 10.1109/Confluence60223.2024.10463307.

[17] Aseem Srivastava, Tharun Suresh, Sarah P. Lord, Md Shad Akhtar, and Tan-moy Chakraborty. Counseling summarization using mental health knowledge guided utterance filtering. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '22, page 3920–3930, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539187. URL https://doi.org/10.1145/3534678.3539187.

[18] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, K. McKeown, and Tatsunori Hashimoto. Benchmarking large language models for news summarization. Transactions of the Association for Computational Linguistics, 12:39–57, 2023. doi: 10.1162/tacl_a_00632.

[19] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs, 2023. URL https://arxiv.org/abs/2305.14314.

[20] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

[21] Geetanjali Singh, Namita Mittal, and Satyendra Singh Chouhan. Hindisumm: A hindi abstractive summarization benchmark dataset. ACM Trans. Asian Low-Resour. Lang. Inf. Process., 23(12), November 2024. ISSN 2375-4699. doi: 10.1145/3696207. URL https://doi.org/10.1145/3696207.

[22] Daisy Monika Lal, Paul Rayson, Krishna Pratap Singh, and Uma Shanker Tiwary. Abstractive Hindi text summarization: A challenge in a low-resource setting. In Jyoti D. Pawar and Sobha Lalitha Devi, editors, Proceedings of the 20th Interna-tional Conference on Natural Language Processing (ICON), pages 603–612, Goa University, Goa, India, December 2023. NLP Association of India (NLPAI). URL https://aclanthology.org/2023.icon-1.58/.

[23] Tong Xiao and Jingbo Zhu. Foundations of large language models, 2025. URL https://arxiv.org/abs/2501.09223.

[24] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

[25] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

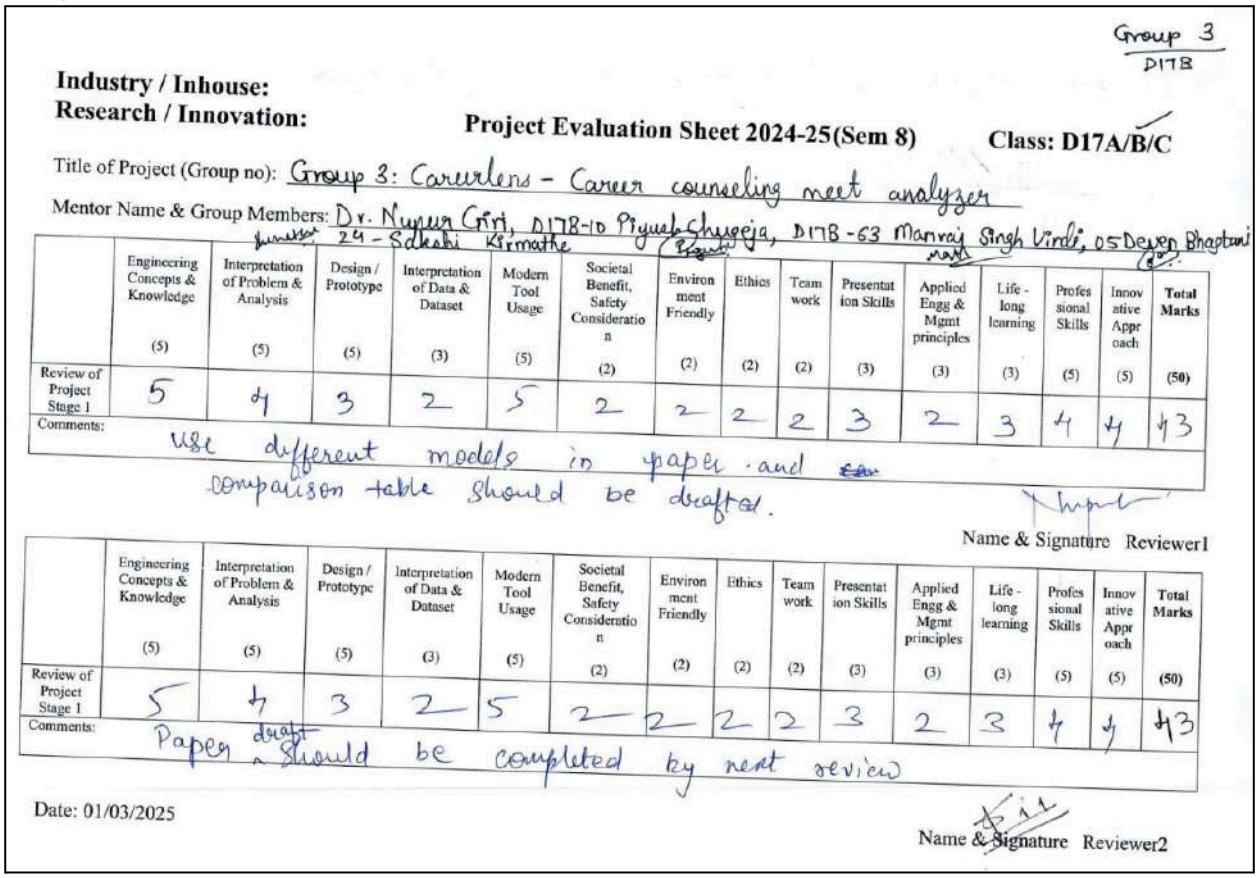
[26] Maxime Peyrard, Teresa Botschen, and Iryna Gurevych. Learning to score system summaries for better content selection evaluation. In Lu Wang, Jackie Chi Kit Cheung, Giuseppe Carenini, and Fei Liu, editors, Proceedings of the Workshop on New Frontiers in Summarization, pages 74–84, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4510. URL https://aclanthology.org/2017.aclweb.org/anthology/W17-4510.

[27] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020. URL https://arxiv.org/abs/1904.09675.

[28] N. Duncan, Attendee. 2024. URL https://github.com/noah-duncan/attendee

Chapter X: Appendix

10.1 Project Review Sheet



Group 3
D17B

Industry / Inhouse:
Research / Innovation:

Project Evaluation Sheet 2024-25(Sem 8) **Class: D17A/B/C**

Title of Project (Group no): Group 3: Careerlens - Career counseling meet analyzer

Mentor Name & Group Members: Dr. Nupur Giri, D17B-10 Piyush Chugeja, D17B-63 Manraj Singh Virdi, 05 Devan Bhagtani, 24 - Sakshi Kirmathe

	Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (3)	Applied Engg & Mgmt principles (3)	Life - long learning (3)	Professional Skills (5)	Innovative Approach (5)	Total Marks (50)
Review of Project Stage 1	5	4	3	2	5	2	2	2	2	3	2	3	4	4	43

Comments: use different models in paper and comparison table should be drafted.

Name & Signature Reviewer1

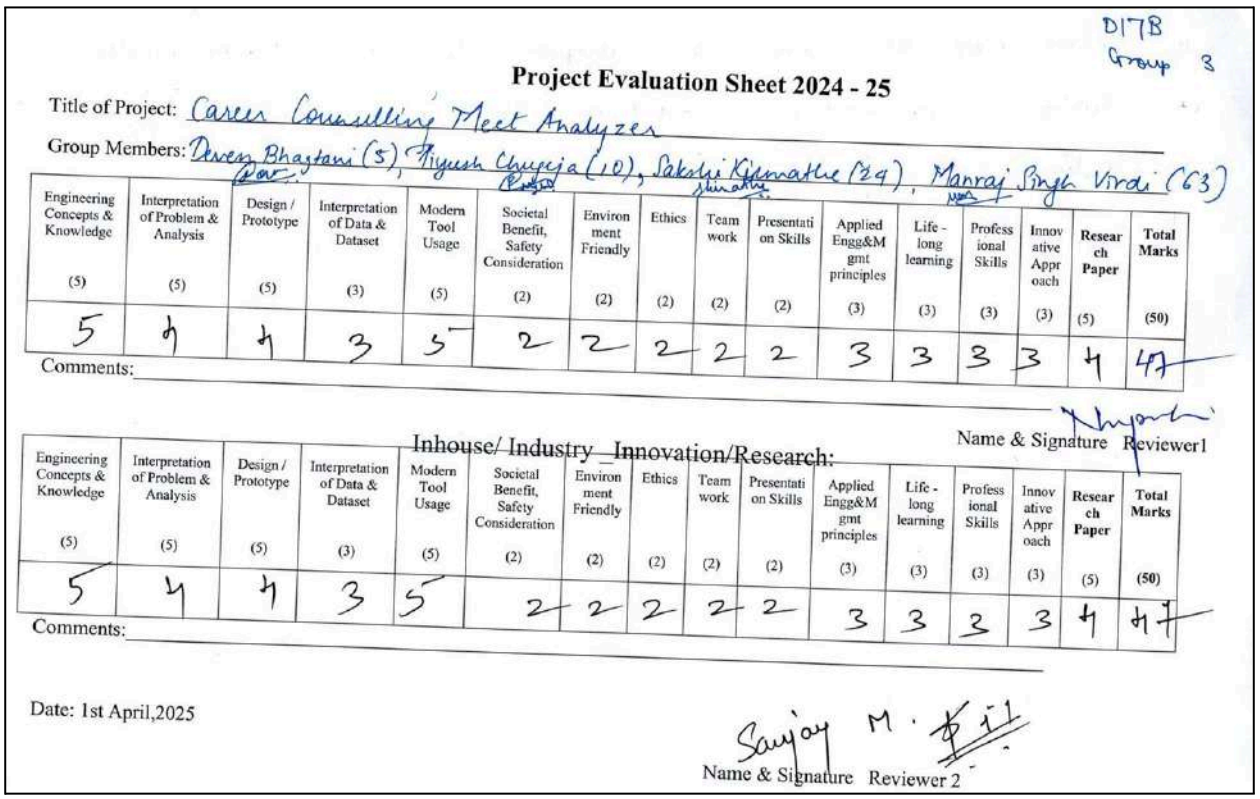
	Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (3)	Applied Engg & Mgmt principles (3)	Life - long learning (3)	Professional Skills (5)	Innovative Approach (5)	Total Marks (50)
Review of Project Stage 1	5	4	3	2	5	2	2	2	2	3	2	3	4	4	43

Comments: Paper draft should be completed by next review

Date: 01/03/2025

Name & Signature Reviewer2

Figure 10.1: Project Review I marks sheet



D17B
Group 3

Project Evaluation Sheet 2024 - 25

Title of Project: Career Counselling Meet Analyzer

Group Members: Devan Bhagtani (5), Piyush Chugeja (10), Sakshi Kirmathe (24), Manraj Singh Virdi (63)

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life - long learning (3)	Professional Skills (3)	Innovative Approach (3)	Research Paper (5)	Total Marks (50)
5	4	4	3	5	2	2	2	2	2	3	3	3	3	4	47

Comments:

Name & Signature Reviewer1

Inhouse/ Industry _Innovation/Research:

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life - long learning (3)	Professional Skills (3)	Innovative Approach (3)	Research Paper (5)	Total Marks (50)
5	4	4	3	5	2	2	2	2	2	3	3	3	3	4	47

Comments:

Date: 1st April,2025

Sanjay M.

Name & Signature Reviewer 2

Figure 10.2: Project Review II marks sheet

Unsloth PEFT based Multilingual Meeting Summarization with Open-Source LLMs

A Comparative Analysis of LLaMA, DeepSeek, and Mistral Models in Zero-Shot and Few-Shot Settings

Dr. Nupur Giri
nupur.giri@ves.ac.in
Vivekanand Education Society's
Institute of Technology
Chembur, Mumbai, India

Manraj Singh Virdi
d2021.manrajsingh.virdi@ves.ac.in
Vivekanand Education Society's
Institute of Technology
Chembur, Mumbai, India

Sakshi Kirmathe
d2021.sakshi.kirmathe@ves.ac.in
Vivekanand Education Society's
Institute of Technology
Chembur, Mumbai, India

Deven Bhagtani
d2021.deven.bhagtani@ves.ac.in
Vivekanand Education Society's
Institute of Technology
Chembur, Mumbai, India

Piyush Chugeja
d2021.piyush.chugeja@ves.ac.in
Vivekanand Education Society's
Institute of Technology
Chembur, Mumbai, India

ABSTRACT

This paper presents a systematic approach to multilingual meeting summarization using open source large language models. Three model families, LLaMA 3, Mistral, and DeepSeek, were evaluated in zero-shot, one-shot, and three-shot settings on a specially prepared dataset of career counseling meeting transcripts. The best models were fine-tuned using Unsloth's 4-bit quantization and Parameter-Efficient Fine-Tuning (PEFT) with Low Rank Adaptation (LoRA) methods. The experimental results showed that the fine-tuned LLaMA 3.1 (8B) model showed greater efficacy in both English and multilingual settings (English, Hindi, and Marathi), generating high-quality summaries, efficiency, and stable cross-lingual generalization. These findings show that using a low learning rate (1×10^{-5}), small batch sizes with gradient accumulation, and a maximum sequence length of 4096 tokens combined with Unsloth's 4-bit quantization and PEFT with LoRA helps the model achieve high accuracy while keeping computational costs low. Evaluation using metrics like ROUGE-L, BERT Score, BLEU, and GLEU, along with fast inference on a GPU P100, confirms that this approach delivers clear and high-quality summaries. This balance of performance and efficiency makes the solution scalable and practical for creating AI-based tools for career counseling.

KEYWORDS

LLM Fine-Tuning, Multilingual Summarization, Open-Source LLMs, Unsloth Fine-Tuning, LLaMA, Mistral, DeepSeek

1 INTRODUCTION

The mass adoption of online meetings has reshaped professional communication, with platforms such as Zoom and Google Meet hosting millions of users daily [1, 2]. Career counseling, which helps people make informed career choices, has also been online, enhancing convenience for both clients and counselors. However, reading lengthy transcripts of these sessions to identify the most important points remains a significant challenge. It is time-consuming and not practical to analyze transcripts manually, so automated summarization is an essential solution.

Large Language Models (LLMs), built on transformer architectures [3], have significantly advanced natural language processing tasks, including summarization. Although LLMs have been widely explored for document summarization, research on summarizing conversations and meetings is relatively limited [4, 5]. Multilingual summarization, particularly for Indic languages like Hindi and Marathi, has received even less attention, despite a large user base that relies on these languages for professional communication. Existing research on Indic language summarization focuses mainly on structured content such as news articles and formal reports [6]. However, career counseling conversations are more dynamic and require summarization techniques that can capture key insights from interactive dialogues. This gap highlights the need for models specifically optimized for multilingual meeting summarization.

In this paper, we evaluate open source LLMs from LLaMA 3 family [7], DeepSeek family [8], and Mistral family [9] to generate a summary of career counseling sessions in English, Hindi and Marathi. We compare their performance in zero-shot, few-shot and fine-tuned settings using multiple evaluation metrics. Fine-tuning was performed using Unsloth [10], leveraging Parameter-Efficient Fine-Tuning (PEFT) to improve adaptability while minimizing computational overhead. In addition, we discuss challenges encountered during fine-tuning, including overfitting issues observed in some models.

2 RELATED WORK

With the rise of online meetings, researchers have been working on ways to summarize them effectively. Different methods exist, such as extractive summarization (which picks key sentences from the text), abstractive summarization (which rewrites the content in a shorter form), and hybrid approaches that combine both. However, challenges remain especially in handling multiple languages, summarizing in real time, and keeping the meaning clear in long, complex discussions.

Transformer-based models [3] have improved abstractive summarization. For example, Pointer Generator Networks [11] help avoid repetition and make summaries easier to read, but they depend on large, general-purpose datasets, making them less useful

Table 1: Open Source Large Language Models (LLMs) chosen for Inference Drawing

Model	Model Creator	#Parameters	Instruction Tuning
LLaMA 3	Meta	8B	✓
LLaMA 3.1		8B	
LLaMA 3.2		3.2B	
Mistral	Mistral	7B	
Mistral v0.3 Instruct		7B	
Deepseek LLaMA	DeepSeek	8B	

for specialized topics. Jotter [12], which combines BERT embeddings with sequence-to-sequence models, balances accuracy and fluency but requires a lot of computing power, making it less practical for real-time applications. Kumar and Kabiri [13] point out that most models use datasets like AMI and ICSI, which, while useful, do not always capture the specific details needed for fields like career counseling. For multilingual meeting summaries, AI-based methods have been developed. One such approach [14] uses Latent Semantic Analysis (LSA) to identify key points, but this method tends to oversimplify discussions. Transformer models perform better because they retain more context, making them more effective for summarizing conversations in different languages. Structured summarization methods have also been useful for specific fields. For example, ConSum [15], designed for mental health counseling, filters important speech patterns using PHQ-9-based scoring, showing that using specialized knowledge can make summaries more relevant.

Another key advancement is instruction tuning, which has been found to be more effective than traditional fine-tuning for text summarization. Zhang et al. [16] studied news summarization and found that instruction tuning helps models perform better without needing large, domain-specific datasets. Unlike fine-tuning, which requires a lot of training data, instruction tuning allows models to improve with high-quality prompts and well-structured instructions. This is particularly useful for summarizing meetings, where clear instructions can help models generate meaningful summaries even with limited training data. The growing popularity of open-source models has also led to new developments in efficient model training. While proprietary models like GPT-4 are strong at summarization without extra training, open-source models like LLaMA, DeepSeek, and Mistral [7–9] are becoming more popular because they offer competitive performance and better privacy. To make these models more efficient, researchers have developed Parameter-Efficient Fine-Tuning (PEFT) methods. One such method, QLoRA [17], helps fine-tune large models with minimal memory usage by adding small, learnable layers instead of retraining the entire model. Similarly, LoRA [18] injects lightweight layers into Transformers, making them more adaptable while keeping computing costs low.

Recent efforts in Hindi summarization have produced helpful assessment tools guiding this paper. Singh et al. [19] presented the HindiSumm dataset together with measures like redundancy, conciseness, novel n-grams ratio, and abstractivity, which help evaluate the quality and diversity of produced summaries. Similarly, Daisy et al. [20] proposed the ICE-H metric to evaluate how well a summary covers key information in low-resource settings.

Despite these advancements, challenges remain, especially in summarizing multilingual meetings, processing summaries in real time, and improving instruction tuning. Addressing these issues will lead to better AI-powered tools for summarizing professional discussions and improving decision-making.

3 OUR WORK

Existing research on meeting summarization often focuses on evaluating single models or relies on domain-agnostic datasets, which limits their effectiveness in more specialized contexts. Our study takes a different approach by evaluating multiple open-source LLM families. Table 1 lists the models used in this research, assessing their performance across zero-shot, one-shot, and three-shot scenarios with meeting transcripts. This comparison provides valuable insights into how different models handle the complexities of structured dialogue-based summarization. In contrast to previous studies that rely on generic benchmark datasets [13], we fine-tune the best-performing models from each family on our own dataset, specifically optimizing for English meeting summarization. This domain-specific fine-tuning improves the contextual coherence of the summaries. After identifying the best performing model, we extended its capabilities to handle multilingual summarization in English, Hindi, and Marathi, addressing a critical gap in non-English meeting summarization research [14].

To improve computational efficiency, we utilize Unsloth's 4-bit quantization [10], which reduces memory usage without compromising performance. This enables us to fine-tune large models with minimal computational overhead, making this approach more scalable for real world applications. This study presents a more adaptable and resource efficient pipeline for meeting summarization by combining structured evaluation, domain-specific fine-tuning and efficient quantization.

4 METHODOLOGY

This section details the steps taken in this study. The methodology is organized into the following subsections:

4.1 Dataset Construction

A custom dataset was created to support career counseling meeting summarization. Since publicly available datasets for this domain are scarce, 35 meeting transcripts per language (English, Hindi, and Marathi) were manually curated. Each transcript underwent careful cleaning and annotation to ensure consistency. The transcripts include structured summaries, key action items, insights, and speaker

details, all formatted in JSON. This structured dataset forms the reference for evaluating model performance.

4.2 Model Evaluation

For model selection, open-source large language models (LLMs) from three families Mistral, LLaMA 3, and DeepSeek as explained in Table 1, were evaluated on a custom dataset of career counseling transcripts. The evaluation was carried out under three inference settings: zero-shot, one-shot, and three-shot.

In the **zero-shot setting**, no examples are provided in the prompt. This tests the model's innate capability to generate a summary without any specific guidance. However, because no task-specific context is given, the generated summary may lack the detailed structure or clarity required for an accurate summarization.

The **one-shot setting** introduces a single example in the prompt. By providing a demonstration, the model is given a clear idea of the desired output format and content. This often improves the coherence of the summary and ensures that key details are better captured, as the model can align its output with the provided example.

In the **three-shot setting**, three examples are provided. With more demonstrations, the model can learn from multiple instances of the expected structure and content, which typically results in more consistent and accurate summaries. The use of multiple shots is particularly helpful when the task is complex or when the domain (in this case, career counseling) requires nuanced understanding.

The selection of these shot settings follows a methodology similar to that described in [16], where the advantages of in-context learning are highlighted. Using different numbers of examples allows for a systematic assessment of the model's performance under varying degrees of guidance, ultimately helping to identify the best-performing model from each family. The results of this evaluation are summarized in Table 2.

4.3 Fine-Tuning Setup

After selecting the best candidates from the initial evaluation, the next step was to fine-tune these models for the specific task of summarizing career counseling transcripts. Fine-tuning was performed on English transcripts using Unsloth's 4-bit quantization framework and Parameter-Efficient Fine-Tuning (PEFT) with LoRA [10, 18].

The fine-tuning process employed a Supervised Fine-Tuning (SFT) approach, where a pre-trained model is further adapted on task-specific data. The SFT method was chosen despite its limitations, such as the potential sensitivity to the amount of labeled data and sometimes requiring careful hyperparameter tuning because it offers a straightforward way to align the model's outputs with the structured requirements of the task. SFT has been widely adopted in recent large language model research (as discussed in [21]) and provides a reliable means to improve model performance for downstream tasks.

Key hyper-parameters for fine-tuning were set as follows:

- **Learning Rate:** 1×10^{-5}, to ensure small and precise updates.
- **Batch Size and Gradient Accumulation:** A per-device batch size of 2 with gradient accumulation over 8 steps, allowing for efficient memory use.
- **Maximum Sequence Length:** 4096 tokens, to accommodate the full context of the transcripts.
- **Optimizer and Scheduler:** The cosine learning rate scheduler and an 8-bit variant of the AdamW optimizer were used to balance performance and computational efficiency.

Training was performed on Kaggle's P100 GPU, providing the necessary computational power for fine-tuning. Although SFT has known challenges, such as sometimes not capturing long-range dependencies as effectively as other methods, it was chosen because it is well-established, relatively simple to implement, and effective for domain-specific tasks. Future work may explore alternative fine-tuning strategies to further enhance performance.

4.4 Training and Evaluation

During training, the model was evaluated using the same metrics as during model selection as explained in 4.2 along with inference time, which was measured in seconds. The training process involved generating structured JSON outputs that captured the summary, key action items, insights, and speaker names. This approach ensured the model not only learned to summarize accurately but also produced outputs that could integrate seamlessly with a dashboard for real-time use. The results of this evaluation are presented in Table 3.

4.5 Multilingual Evaluation Metrics

In addition to conventional metrics such as ROUGE [22], BLEU [23], GLEU [24], and BERT Score [25], the evaluation of summary quality in a multilingual setting (English, Hindi, and Marathi) involved several additional metrics as discussed by Singh et. al. and Daisy et. al. [19, 20]. These metrics were used to capture various aspects of the summaries and to ensure that they are informative, diverse, and succinct.

Information Coverage Estimate (ICE): ICE measures how well the generated summary captures the key information from the original transcript. It is calculated by encoding both the reference and generated summaries using Sentence-BERT and computing the cosine similarity between these embeddings. A higher ICE indicates better retention of important information.

Redundancy: Redundancy quantifies the amount of repeated content within the summary. It is defined as:

$$\text{Redundancy} = 1 - \frac{\text{Number of unique } n\text{-grams}}{\text{Total number of } n\text{-grams}} \tag{1}$$

A lower redundancy score means that the summary is more concise and free from unnecessary repetition.

Abstractivity: Abstractivity evaluates the extent to which the summary is generated using new, rephrased content rather than copying segments of the original text. It is calculated as:

$$\text{Abstractivity} = \frac{\text{Number of novel words}}{\text{Total words in summary}} \tag{2}$$

A higher abstractivity score reflects the model's ability to effectively paraphrase and generate novel expressions.

Table 2: Testing pre-trained LLMs on a custom dataset of English transcripts to identify the best performing model from each model family

Setting	Model	ROUGE			BERT Score				
		R1	R2	RL	Precision	Recall	F1	BLEU	GLEU
0-shot inference	Mistral (7B)	0.3823	0.1791	0.3099	0.9243	0.8819	0.9025	0.0506	0.1315
	LLaMA 3 (8B)	0.4318	0.1863	0.3432	0.9277	0.8926	0.9098	0.0901	0.1657
	LLaMA 3.2 (3B)	0.3377	0.1410	0.2503	0.9187	0.8759	0.8967	0.0274	0.1045
	LLaMA 3.1 (8B)	0.4830	0.2374	0.3677	0.9341	0.9004	0.9169	0.1324	0.2058
	Mistral v0.3 Instruct (7B)	0.4920	0.2439	0.3891	0.9232	0.9082	0.9155	0.1557	0.2221
	Deepseek LLaMA (8B)	0.2115	0.3528	0.3528	0.9230	0.9112	0.9169	0.1382	0.2105
1-shot inference	Mistral (7B)	0.5059	0.2321	0.3916	0.9116	0.9108	0.9111	0.1354	0.1892
	LLaMA 3 (8B)	0.5037	0.2646	0.4130	0.9349	0.9057	0.9200	0.1540	0.2224
	LLaMA 3.2 (3B)	0.5369	0.2557	0.4114	0.9456	0.9239	0.9346	0.2114	0.2744
	LLaMA 3.1 (8B)	0.5503	0.3145	0.4611	0.9381	0.9170	0.9274	0.2213	0.2758
	Mistral v0.3 Instruct (7B)	0.6041	0.3596	0.5182	0.9389	0.9272	0.9329	0.2642	0.3118
	Deepseek LLaMA (8B)	0.5321	0.2902	0.4343	0.9322	0.9149	0.9234	0.2117	0.2665
3-shot inference	Mistral (7B)	0.5554	0.3250	0.4759	0.9431	0.9200	0.9313	0.2310	0.2935
	LLaMA 3 (8B)	0.5554	0.3250	0.4759	0.9431	0.9200	0.9313	0.2309	0.2935
	LLaMA 3.2 (3B)	0.5369	0.2557	0.4114	0.9456	0.9239	0.9346	0.2114	0.2744
	LLaMA 3.1 (8B)	0.5194	0.2926	0.4414	0.9369	0.9114	0.9239	0.2023	0.2593
	Mistral v0.3 Instruct (7B)	0.5964	0.3428	0.4963	0.9287	0.9319	0.9302	0.2629	0.3086
	Deepseek LLaMA (8B)	0.5691	0.3144	0.4646	0.9366	0.9246	0.9305	0.2176	0.2822

N-gram Ratio: The N-gram Ratio measures lexical diversity by comparing the number of novel n-grams in the summary to the total number of n-grams:

$$\text{N-gram Ratio} = \frac{\text{Number of novel } n\text{-grams}}{\text{Total } n\text{-grams in summary}} \tag{3}$$

A higher ratio indicates greater linguistic variety, showing that the model uses a richer vocabulary.

Conciseness: Conciseness is determined by comparing the length of the summary to that of the original transcript:

$$\text{Conciseness} = \frac{\text{Number of words in summary}}{\text{Number of words in original text}} \tag{4}$$

A lower value indicates that the summary is succinct, retaining only the most important content.

These extra measures offer a thorough system for assessing summary quality in a multilingual setting. They make sure that the summaries are varied, clear, and able to convey all vital information across English, Hindi, and Marathi in addition to correctness and fluency. The findings and metric values are presented in Table 4

5 OUTCOMES

The outcomes of this study are discussed in three main parts: the initial model evaluation, the fine-tuning results, and the multilingual performance analysis.

5.1 Performance of Model Families

Table 2 presents the evaluation of pre-trained large language models on English transcripts across zero-shot, one-shot, and three-shot inference settings. The evaluation metrics ROUGE, BERT Score, BLEU, and GLEU are used to determine how well each model captures essential content and maintains fluency in the generated summaries.

- ***Mistral Family*** Within the Mistral family, the base Mistral (7B) model achieves moderate scores in the zero-shot setting, while the Mistral v0.3 Instruct (7B) variant shows noticeable improvements in both one-shot and three-shot scenarios. This suggests that instruction tuning has a positive impact on its summarization capabilities, yielding higher overlap with reference summaries and better semantic alignment.

- ***LLaMA 3 Family*** For the LLaMA 3 family, three variants were tested. LLaMA 3 (8B) and LLaMA 3.2 (3B) deliver competitive results; however, LLaMA 3.1 (8B) consistently stands out. It produces the highest ROUGE scores, indicating superior content retention, and achieves the best BERT Score F1, reflecting strong semantic similarity with the reference summaries. LLaMA 3.1 (8B) shows improved BLEU and GLEU scores, which imply that the summaries are both fluent and well-structured.

- ***DeepSeek LLaMA (8B)*** It reaches competitive ROUGE and BERT Score values in the three-shot setting. Although its performance is notable, its overall scores are slightly lower compared to the top variants from the LLaMA and Mistral families.

Based on the results, the following models were chosen for further fine-tuning:

(1) **Mistral family:** Mistral v0.3 Instruct (7B)
(2) **LLaMA 3 family:** LLaMA 3.1 (8B)
(3) **DeepSeek family:** DeepSeek LLaMA (8B)

Table 3: Performance Comparison of Fine-tuned Models on English Transcripts

Model	ROUGE-L	BERT F1	BLEU	GLEU	Inference Time
LLaMA 3.1 (8B)	0.5178	0.9378	0.3253	0.3334	12.3s
Mistral v0.3 Instruct (7B)	0.4796	0.9350	0.2745	0.3496	16.5s
DeepSeek LLaMA (8B)	0.4527	0.7903	0.3103	0.2396	19.2s

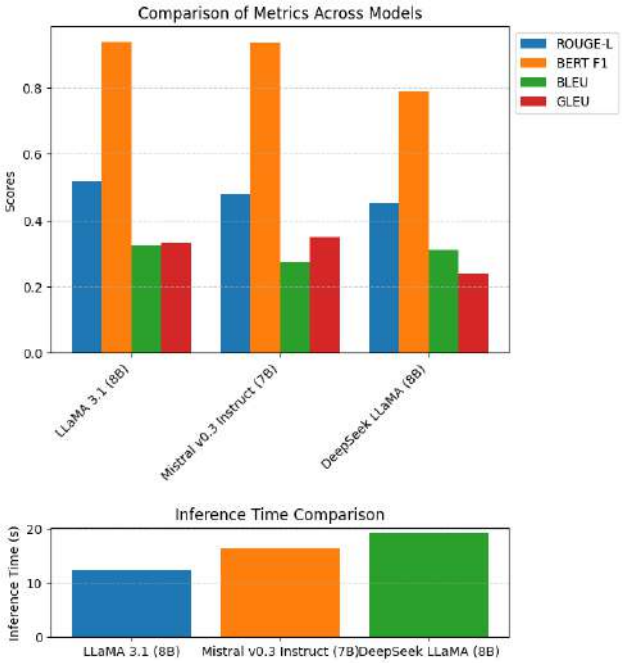


Figure 1: Performance comparison of fine-tuned models on English transcripts.

5.2 Results of Fine-Tuning Process and Comparative Analysis

The fine-tuning was executed as outlined in Section 4.3. Table 3 shows a detailed comparison of the three fine-tuned models, and Figure 1 shows a graphical comparison on English meeting transcripts.

- **LLaMA 3.1 (8B):** This model achieved a ROUGE-L score of 0.5178 and a BERT Score F1 of 0.9378. It also reached a BLEU of 0.3253 and a GLEU of 0.3334. These results reflect an improvement of approximately 7% in ROUGE-L and more than 35% in BERT Score F1 compared to some pre-fine-tuning results.
- **Mistral v0.3 Instruct (7B):** This model recorded a ROUGE-L of 0.4796 and a BERT Score F1 of 0.9350 along with a BLEU of 0.2745 and a GLEU of 0.3496.
- **DeepSeek LLaMA (8B):** While DeepSeek LLaMA (8B) achieved a moderate BLEU score of 0.3103, its overall performance was lower, with a ROUGE-L of 0.4527 and a BERT Score F1 of 0.7903.

5.2.1 Inference Time Trade-offs. Practical uses also depend on inference time, which is as important as correctness. With an average time of 12.3 seconds per transcript, LLaMA 3.1 (8B) not only offered the best accuracy, but also attained the fastest inference. In contrast, DeepSeek LLaMA (8B) was the slowest at 19.2 seconds, while Mistral v0.3 Instruct (7B) needed 16.5 seconds. This trade-off between speed and accuracy is significant; quicker inference allows real-time summarization, which is absolutely vital for interactive systems. LLaMA 3.1 (8B) is the most practical option for deployment given the balance of high performance and low inference time.

DeepSeek LLaMA (8B) encountered challenges during fine-tuning. While it performed reasonably well on transcripts it had seen during training, it struggled to generate meaningful summaries for unseen transcripts. Adjustments to training parameters, such as reducing the maximum sequence length and modifying dropout rates, did not resolve this issue. As a result, DeepSeek was not selected for further testing. This discovery underlines the importance of a model's ability to generalize beyond the training data, a factor that is critical for real-world applications.

The comparison shows that the summary quality improved significantly as a result of fine-tuning. Compared to their pre-fine-tuning outputs presented in Table 2, the models produced more accurate and coherent summaries with faster processing times. The metrics make it evident that accuracy and speed must be traded off; LLaMA 3.1 (8B) provides the fastest inference time and high accuracy (with notable improvements in ROUGE-L and BERT Score F1), achieving the best overall balance. This led to LLaMA 3.1 being chosen for further multilingual fine-tuning and analysis.

5.3 Multilingual Performance Evaluation

After identifying LLaMA 3.1 (8B) as the top model on English transcripts, this model was further fine-tuned on an expanded dataset that includes English, Hindi and Marathi transcripts as explained in Section 4.1. A detailed analysis of the performance of the model in these languages is presented in Table 4 and Figure 2 using the metrics explained in Section 4.5.

The evaluation shows that the model delivers consistent performance in all languages. For example, Hindi transcripts achieved a slightly higher BERT Score (0.7281) than English (0.6533) and Marathi (0.6807), indicating a strong ability to capture semantic meaning. Low Redundancy values confirm that the summaries avoid repetitive content, while high Abstractivity scores demonstrate the model's ability to paraphrase and generate novel expressions while retaining essential information. Furthermore, the elevated N-gram Ratio and solid Conciseness scores attest that the summaries are both varied in vocabulary and succinct.

Table 4: Multilingual Evaluation of the Fine-tuned Model

Language	BERT Score	ICE	Redundancy	Abstractivity	N-gram Ratio	Conciseness
English	0.6533	0.4043	0.0347	0.8007	0.9690	0.6788
Hindi	0.7281	0.7702	0.1064	0.5361	0.8113	0.6203
Marathi	0.6807	0.5688	0.0677	0.8207	0.9663	0.6649

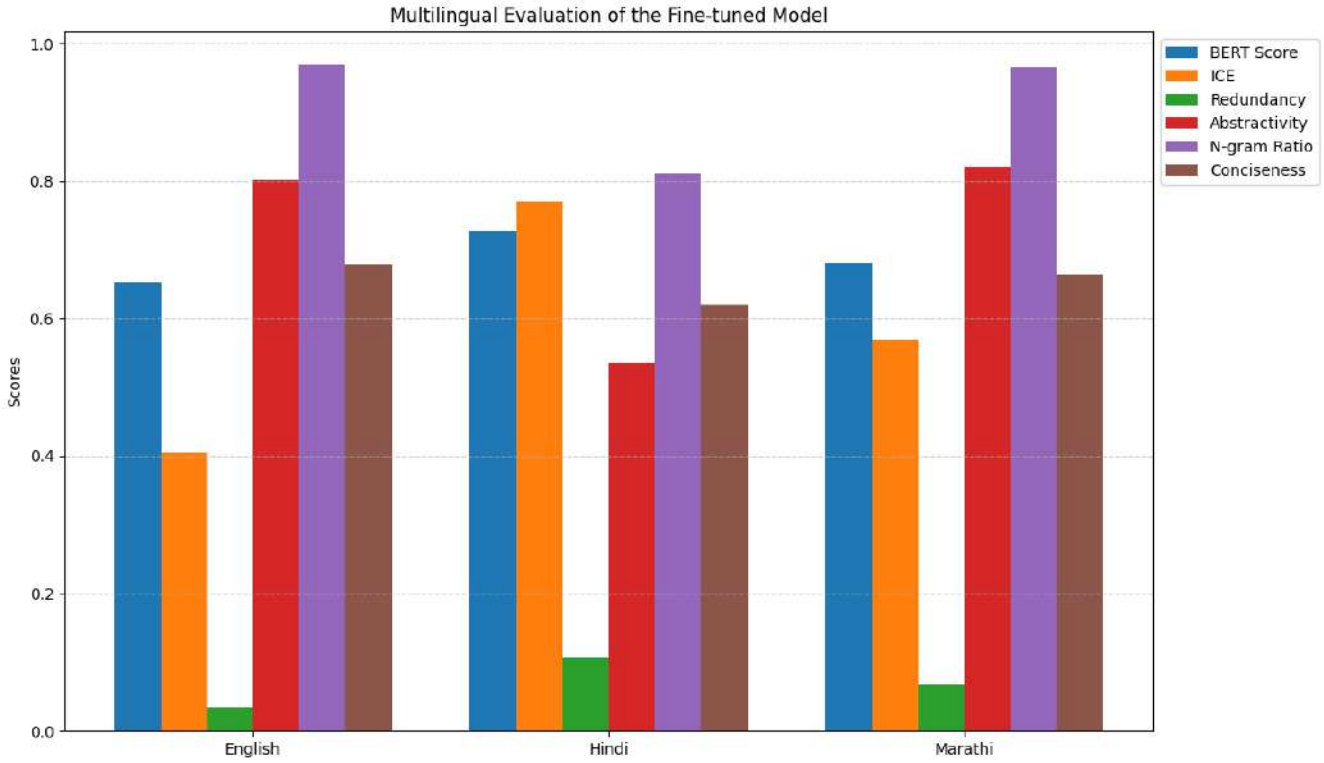


Figure 2: Multilingual Evaluation Metrics for LLaMA 3.1 (8B) across English, Hindi, and Marathi transcripts

6 CONCLUSION

The research in this paper presents a systematic evaluation and optimization procedure for meeting summarization models in a multilingual environment. Three model types were thoroughly tested with zero-shot, one-shot, and three-shot prompting techniques on a specially created dataset of career guidance meeting transcripts. ROUGE-L, BERT Score F1, BLEU, and GLEU were used as metrics to evaluate and determine which model worked best for each type. Fine-tuning experiments with Unsloth's 4-bit quantization architecture and Parameter-Efficient Fine-Tuning (PEFT) with LoRA validated that the LLaMA 3.1 (8B) model not only produced correct summaries, but also performed with exceptional efficiency and pace.

Furthermore, applying the best-performing model to handle multiple languages demonstrated its ability to extract important information in English, Hindi, and Marathi transcripts. The results of the experiment emphasize the requirement of domain-specific fine-tuning for domain-related tasks and indicate the potential of open-source LLMs to develop working and resource-effective AI tools for career guidance.

Future research could investigate comparative tests between supervised fine-tuning and reinforcement learning or reward-based optimization techniques to optimize structured, customized summarization. Future research can also examine trade-offs between instruction tuning and fine-tuning, optimization for live summarization, and more pervasive applications in other professional domains. In general, the study reflects a clear methodology from model choice to effective multilingual summarization, and with it, establishes a solid foundation for next-generation AI-powered communication tools.

REFERENCES

[1] Jay Peters. Google's Meet teleconferencing service now adding about 3 million users per day — theverge.com. https://www.theverge.com/2020/4/28/21240434/google-meet-three-million-users-per-day-pichai-earnings. [Accessed 14-10-2024].

[2] Tom Warren. Zoom grows to 300 million meeting participants despite security backlash — theverge.com. https://www.theverge.com/2020/4/23/21232401/zoom-300-million-users-growth-coronavirus-pandemic-security-privacy-concerns-

response. [Accessed 14-10-2024].

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

[4] Md Tahmid Rahman Laskar, Xue-Yong Fu, Cheng Chen, and Shashi Bhushan TN. Building real-world meeting summarization systems using large language models: A practical perspective, 2023. URL https://arxiv.org/abs/2310.19233.

[5] Fei Ge. *Fine-tune Whisper and transformer large language model for meeting summarization.* PhD thesis, UCLA, 2024.

[6] Aatman Vaidya, Tarunima Prabhakar, Denny George, and Swair Shah. Analysis of indic language capabilities in llms, 2025. URL https://arxiv.org/abs/2501.13912.

[7] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, and et. al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[8] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, and et. al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

[9] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

[10] Michael Han Daniel Han and Unsloth team. Unsloth, 2023. URL http://github.com/unslothai/unsloth.

[11] Nima Sadri, Bohan Zhang, and Bihan Liu. Meetsum: Transforming meeting transcript summarization using transformers!, 2021. URL https://arxiv.org/abs/2108.06310.

[12] Sumedh S Bhat, Uzair Ahmed Nawaz, Sujay M, Nameesha Tantri, and Vani Vasudevan. Jotter: An approach to summarize the formal online meeting. In *2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE)*, pages 1–6, 2023. doi: 10.1109/AIKIIE60097.2023.10390455.

[13] Lakshmi Prasanna Kumar and Arman Kabiri. Meeting summarization: A survey of the state of the art, 2022. URL https://arxiv.org/abs/2212.08206.

[14] Medha Wyawahare, Madhuri Shelke, Siddharth Bhorge, and Rohit Agrawal. Ai powered multilingual meeting summarization. In *2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 86–91, Jan 2024. doi: 10.1109/Confluence60223.2024.10463307.

[15] Aseem Srivastava, Tharun Suresh, Sarah P. Lord, Md Shad Akhtar, and Tanmoy Chakraborty. Counseling summarization using mental health knowledge guided utterance filtering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 3920–3930, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539187. URL https://doi.org/10.1145/3534678.3539187.

[16] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, K. McKeown, and Tatsunori Hashimoto. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57, 2023. doi: 10.1162/tacl_a_00632.

[17] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023. URL https://arxiv.org/abs/2305.14314.

[18] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

[19] Geetanjali Singh, Namita Mittal, and Satyendra Singh Chouhan. Hindisumm: A hindi abstractive summarization benchmark dataset. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 23(12), November 2024. ISSN 2375-4699. doi: 10.1145/3696207. URL https://doi.org/10.1145/3696207.

[20] Daisy Monika Lal, Paul Rayson, Krishna Pratap Singh, and Uma Shanker Tiwary. Abstractive Hindi text summarization: A challenge in a low-resource setting. In Jyoti D. Pawar and Sobha Lalitha Devi, editors, *Proceedings of the 20th International Conference on Natural Language Processing (ICON)*, pages 603–612, Goa University, Goa, India, December 2023. NLP Association of India (NLPAI). URL https://aclanthology.org/2023.icon-1.58/.

[21] Tong Xiao and Jingbo Zhu. Foundations of large language models, 2025. URL https://arxiv.org/abs/2501.09223.

[22] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

[23] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

[24] Maxime Peyrard, Teresa Botschen, and Iryna Gurevych. Learning to score system summaries for better content selection evaluation. In Lu Wang, Jackie Chi Kit Cheung, Giuseppe Carenini, and Fei Liu, editors, *Proceedings of the Workshop on New Frontiers in Summarization*, pages 74–84, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4510. URL https://aclanthology.org/W17-4510/.

[25] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020. URL https://arxiv.org/abs/1904.09675.

Unsloth PEFT based Multilingual Meeting Summarization with Open-Source LLMs

by Piyush Chugeja

Submission date: 24-Apr-2025 03:32PM (UTC+0530)
Submission ID: 2655477828
File name: sed_Multilingual_Meeting_Summarization_with_Open_Source_LLMs.pdf (540.2K)
Word count: 5184
Character count: 29639

Unsloth PEFT based Multilingual Meeting Summarization with Open-Source LLMs

A Comparative Analysis of LLaMA, DeepSeek, and Mistral Models in Zero-Shot and Few-Shot Settings

Dr. Nupur Giri
nupur.giri@ves.ac.in
Vivekanand Education Society's
Institute of Technology
Chembur, Mumbai, India

Manraj Singh Virdi
d2021.manrajsingh.virdi@ves.ac.in
Vivekanand Education Society's
Institute of Technology
Chembur, Mumbai, India

Sakshi Kirmathe
d2021.sakshi.kirmathe@ves.ac.in
Vivekanand Education Society's
Institute of Technology
Chembur, Mumbai, India

Deven Bhagtani
d2021.deven.bhagtani@ves.ac.in
Vivekanand Education Society's
Institute of Technology
Chembur, Mumbai, India

Piyush Chugeja
d2021.piyush.chugeja@ves.ac.in
Vivekanand Education Society's
Institute of Technology
Chembur, Mumbai, India

ABSTRACT

This paper presents a systematic approach to multilingual meeting summarization using open source large language models. Three model families, LLaMA 3, Mistral, and DeepSeek, were evaluated in zero-shot, one-shot, and three-shot settings on a specially prepared dataset of career counseling meeting transcripts. The best models were fine-tuned using Unsloth's 4-bit quantization and Parameter-Efficient Fine-Tuning (PEFT) with Low Rank Adaptation (LoRA) methods. The experimental results showed that the fine-tuned LLaMA 3.1 (8B) model showed greater efficacy in both English and multilingual settings (English, Hindi, and Marathi), generating high-quality summaries, efficiency, and stable cross-lingual generalization. These findings show that using a low learning rate (1×10^{-5}), small batch sizes with gradient accumulation, and a maximum sequence length of 4096 tokens combined with Unsloth's 4-bit quantization and PEFT with LoRA helps the model achieve high accuracy while keeping computational costs low. Evaluation using metrics like ROUGE-L, BERT Score, BLEU, and GLEU, along with fast inference on a GPU P100, confirms that this approach delivers clear and high-quality summaries. This balance of performance and efficiency makes the solution scalable and practical for creating AI-based tools for career counseling.

KEYWORDS

LLM Fine-Tuning, Multilingual Summarization, Open-Source LLMs, Unsloth Fine-Tuning, LLaMA, Mistral, DeepSeek

1 INTRODUCTION

The mass adoption of online meetings has reshaped professional communication, with platforms such as Zoom and Google Meet hosting millions of users daily [1, 2]. Career counseling, which helps people make informed career choices, has also been online, enhancing convenience for both clients and counselors. However, reading lengthy transcripts of these sessions to identify the most important points remains a significant challenge. It is time-consuming and not practical to analyze transcripts manually, so automated summarization is an essential solution.

Large Language Models (LLMs), built on transformer architectures [3], have significantly advanced natural language processing tasks, including summarization. Although LLMs have been widely explored for document summarization, research on summarizing conversations and meetings is relatively limited [4, 5]. Multilingual summarization, particularly for Indic languages like Hindi and Marathi, has received even less attention, despite a large user base that relies on these languages for professional communication. Existing research on Indic language summarization focuses mainly on structured content such as news articles and formal reports [6]. However, career counseling conversations are more dynamic and require summarization techniques that can capture key insights from interactive dialogues. This gap highlights the need for models specifically optimized for multilingual meeting summarization.

In this paper, we evaluate open source LLMs from LLaMA 3 family [7], DeepSeek family [8], and Mistral family [9] to generate a summary of career counseling sessions in English, Hindi and Marathi. We compare their performance in zero-shot, few-shot and fine-tuned settings using multiple evaluation metrics. Fine-tuning was performed using Unsloth [10], leveraging Parameter-Efficient Fine-Tuning (PEFT) to improve adaptability while minimizing computational overhead. In addition, we discuss challenges encountered during fine-tuning, including overfitting issues observed in some models.

2 RELATED WORK

With the rise of online meetings, researchers have been working on ways to summarize them effectively. Different methods exist, such as extractive summarization (which picks key sentences from the text), abstractive summarization (which rewrites the content in a shorter form), and hybrid approaches that combine both. However, challenges remain especially in handling multiple languages, summarizing in real time, and keeping the meaning clear in long, complex discussions.

Transformer-based models [3] have improved abstractive summarization. For example, Pointer Generator Networks [11] help avoid repetition and make summaries easier to read, but they depend on large, general-purpose datasets, making them less useful

Table 1: Open Source Large Language Models (LLMs) chosen for Inference Drawing

Model	Model Creator	#Parameters	Instruction Tuning
LLaMA 3	Meta	8B	✓
LLaMA 3.1		8B	
LLaMA 3.2		3.2B	
Mistral	Mistral	7B	
Mistral v0.3 Instruct		7B	
Deepseek LLaMA	DeepSeek	8B	

for specialized topics. Jotter [12], which combines BERT embeddings with sequence-to-sequence models, balances accuracy and fluency but requires a lot of computing power, making it less practical for real-time applications. Kumar and Kabiri [13] point out that most models use datasets like AMI and ICSI, which, while useful, do not always capture the specific details needed for fields like career counseling. For multilingual meeting summaries, AI-based methods have been developed. One such approach [14] uses Latent Semantic Analysis (LSA) to identify key points, but this method tends to oversimplify discussions. Transformer models perform better because they retain more context, making them more effective for summarizing conversations in different languages. Structured summarization methods have also been useful for specific fields. For example, ConSum [15], designed for mental health counseling, filters important speech patterns using PHQ-9-based scoring, showing that using specialized knowledge can make summaries more relevant.

Another key advancement is instruction tuning, which has been found to be more effective than traditional fine-tuning for text summarization. Zhang et al. [16] studied news summarization and found that instruction tuning helps models perform better without needing large, domain-specific datasets. Unlike fine-tuning, which requires a lot of training data, instruction tuning allows models to improve with high-quality prompts and well-structured instructions. This is particularly useful for summarizing meetings, where clear instructions can help models generate meaningful summaries even with limited training data. The growing popularity of open-source models has also led to new developments in efficient model training. While proprietary models like GPT-4 are strong at summarization without extra training, open-source models like LLaMA, DeepSeek, and Mistral [7–9] are becoming more popular because they offer competitive performance and better privacy. To make these models more efficient, researchers have developed Parameter-Efficient Fine-Tuning (PEFT) methods. One such method, QLoRA [17], helps fine-tune large models with minimal memory usage by adding small, learnable layers instead of retraining the entire model. Similarly, LoRA [18] injects lightweight layers into Transformers, making them more adaptable while keeping computing costs low.

Recent efforts in Hindi summarization have produced helpful assessment tools guiding this paper. Singh et al. [19] presented the HindiSumm dataset together with measures like redundancy, conciseness, novel n-grams ratio, and abstractivity, which help evaluate the quality and diversity of produced summaries. Similarly, Daisy et al. [20] proposed the ICE-H metric to evaluate how well a summary covers key information in low-resource settings.

Despite these advancements, challenges remain, especially in summarizing multilingual meetings, processing summaries in real time, and improving instruction tuning. Addressing these issues will lead to better AI-powered tools for summarizing professional discussions and improving decision-making.

3 OUR WORK

Existing research on meeting summarization often focuses on evaluating single models or relies on domain-agnostic datasets, which limits their effectiveness in more specialized contexts. Our study takes a different approach by evaluating multiple open-source LLM families. Table 1 lists the models used in this research, assessing their performance across zero-shot, one-shot, and three-shot scenarios with meeting transcripts. This comparison provides valuable insights into how different models handle the complexities of structured dialogue-based summarization. In contrast to previous studies that rely on generic benchmark datasets [13], we fine-tune the best-performing models from each family on our own dataset, specifically optimizing for English meeting summarization. This domain-specific fine-tuning improves the contextual coherence of the summaries. After identifying the best performing model, we extended its capabilities to handle multilingual summarization in English, Hindi, and Marathi, addressing a critical gap in non-English meeting summarization research [14].

To improve computational efficiency, we utilize Unsloth's 4-bit quantization [10], which reduces memory usage without compromising performance. This enables us to fine-tune large models with minimal computational overhead, making this approach more scalable for real world applications. This study presents a more adaptable and resource efficient pipeline for meeting summarization by combining structured evaluation, domain-specific fine-tuning and efficient quantization.

4 METHODOLOGY

This section details the steps taken in this study. The methodology is organized into the following subsections:

4.1 Dataset Construction

A custom dataset was created to support career counseling meeting summarization. Since publicly available datasets for this domain are scarce, 35 meeting transcripts per language (English, Hindi, and Marathi) were manually curated. Each transcript underwent careful cleaning and annotation to ensure consistency. The transcripts include structured summaries, key action items, insights, and speaker

details, all formatted in JSON. This structured dataset forms the reference for evaluating model performance.

4.2 Model Evaluation

For model selection, open-source large language models (LLMs) from three families Mistral, LLaMA 3, and DeepSeek as explained in Table 1, were evaluated on a custom dataset of career counseling transcripts. The evaluation was carried out under three inference settings: zero-shot, one-shot, and three-shot.

In the **zero-shot setting**, no examples are provided in the prompt. This tests the model's innate capability to generate a summary without any specific guidance. However, because no task-specific context is given, the generated summary may lack the detailed structure or clarity required for an accurate summarization.

The **one-shot setting** introduces a single example in the prompt. By providing a demonstration, the model is given a clear idea of the desired output format and content. This often improves the coherence of the summary and ensures that key details are better captured, as the model can align its output with the provided example.

In the **three-shot setting**, three examples are provided. With more demonstrations, the model can learn from multiple instances of the expected structure and content, which typically results in more consistent and accurate summaries. The use of multiple shots is particularly helpful when the task is complex or when the domain (in this case, career counseling) requires nuanced understanding.

The selection of these shot settings follows a methodology similar to that described in [16], where the advantages of in-context learning are highlighted. Using different numbers of examples allows for a systematic assessment of the model's performance under varying degrees of guidance, ultimately helping to identify the best-performing model from each family. The results of this evaluation are summarized in Table 2.

4.3 Fine-Tuning Setup

After selecting the best candidates from the initial evaluation, the next step was to fine-tune these models for the specific task of summarizing career counseling transcripts. Fine-tuning was performed on English transcripts using Unsloth's 4-bit quantization framework and Parameter-Efficient Fine-Tuning (PEFT) with LoRA [10, 18].

The fine-tuning process employed a Supervised Fine-Tuning (SFT) approach, where a pre-trained model is further adapted on task-specific data. The SFT method was chosen despite its limitations, such as the potential sensitivity to the amount of labeled data and sometimes requiring careful hyperparameter tuning because it offers a straightforward way to align the model's outputs with the structured requirements of the task. SFT has been widely adopted in recent large language model research (as discussed in [21]) and provides a reliable means to improve model performance for downstream tasks.

Key hyper-parameters for fine-tuning were set as follows:

- **Learning Rate:** 1×10^{-5}, to ensure small and precise updates.
- **Batch Size and Gradient Accumulation:** A per-device batch size of 2 with gradient accumulation over 8 steps, allowing for efficient memory use.
- **Maximum Sequence Length:** 4096 tokens, to accommodate the full context of the transcripts.
- **Optimizer and Scheduler:** The cosine learning rate scheduler and an 8-bit variant of the AdamW optimizer were used to balance performance and computational efficiency.

Training was performed on Kaggle's P100 GPU, providing the necessary computational power for fine-tuning. Although SFT has known challenges, such as sometimes not capturing long-range dependencies as effectively as other methods, it was chosen because it is well-established, relatively simple to implement, and effective for domain-specific tasks. Future work may explore alternative fine-tuning strategies to further enhance performance.

4.4 Training and Evaluation

During training, the model was evaluated using the same metrics as during model selection as explained in 4.2 along with inference time, which was measured in seconds. The training process involved generating structured JSON outputs that captured the summary, key action items, insights, and speaker names. This approach ensured the model not only learned to summarize accurately but also produced outputs that could integrate seamlessly with a dashboard for real-time use. The results of this evaluation are presented in Table 3.

4.5 Multilingual Evaluation Metrics

In addition to conventional metrics such as ROUGE [22], BLEU [23], GLEU [24], and BERT Score [25], the evaluation of summary quality in a multilingual setting (English, Hindi, and Marathi) involved several additional metrics as discussed by Singh et. al. and Daisy et. al. [19, 20]. These metrics were used to capture various aspects of the summaries and to ensure that they are informative, diverse, and succinct.

Information Coverage Estimate (ICE): ICE measures how well the generated summary captures the key information from the original transcript. It is calculated by encoding both the reference and generated summaries using Sentence-BERT and computing the cosine similarity between these embeddings. A higher ICE indicates better retention of important information.

Redundancy: Redundancy quantifies the amount of repeated content within the summary. It is defined as:

$$\text{Redundancy} = 1 - \frac{\text{Number of unique } n\text{-grams}}{\text{Total number of } n\text{-grams}} \tag{1}$$

A lower redundancy score means that the summary is more concise and free from unnecessary repetition.

Abstractivity: Abstractivity evaluates the extent to which the summary is generated using new, rephrased content rather than copying segments of the original text. It is calculated as:

$$\text{Abstractivity} = \frac{\text{Number of novel words}}{\text{Total words in summary}} \tag{2}$$

A higher abstractivity score reflects the model's ability to effectively paraphrase and generate novel expressions.

Table 2: Testing pre-trained LLMs on a custom dataset of English transcripts to identify the best performing model from each model family

Setting	Model	ROUGE			BERT Score			BLEU	GLEU
		R1	R2	RL	Precision	Recall	F1		
0-shot inference	Mistral (7B)	0.3823	0.1791	0.3099	0.9243	0.8819	0.9025	0.0506	0.1315
	LLaMA 3 (8B)	0.4318	0.1863	0.3432	0.9277	0.8926	0.9098	0.0901	0.1657
	LLaMA 3.2 (3B)	0.3377	0.1410	0.2503	0.9187	0.8759	0.8967	0.0274	0.1045
	LLaMA 3.1 (8B)	0.4830	0.2374	0.3677	0.9341	0.9004	0.9169	0.1324	0.2058
	Mistral v0.3 Instruct (7B)	0.4920	0.2439	0.3891	0.9232	0.9082	0.9155	0.1557	0.2221
	Deepseek LLaMA (8B)	0.2115	0.3528	0.3528	0.9230	0.9112	0.9169	0.1382	0.2105
1-shot inference	Mistral (7B)	0.5059	0.2321	0.3916	0.9116	0.9108	0.9111	0.1354	0.1892
	LLaMA 3 (8B)	0.5037	0.2646	0.4130	0.9349	0.9057	0.9200	0.1540	0.2224
	LLaMA 3.2 (3B)	0.5369	0.2557	0.4114	0.9456	0.9239	0.9346	0.2114	0.2744
	LLaMA 3.1 (8B)	0.5503	0.3145	0.4611	0.9381	0.9170	0.9274	0.2213	0.2758
	Mistral v0.3 Instruct (7B)	0.6041	0.3596	0.5182	0.9389	0.9272	0.9329	0.2642	0.3118
	Deepseek LLaMA (8B)	0.5321	0.2902	0.4343	0.9322	0.9149	0.9234	0.2117	0.2665
3-shot inference	Mistral (7B)	0.5554	0.3250	0.4759	0.9431	0.9200	0.9313	0.2310	0.2935
	LLaMA 3 (8B)	0.5554	0.3250	0.4759	0.9431	0.9200	0.9313	0.2309	0.2935
	LLaMA 3.2 (3B)	0.5369	0.2557	0.4114	0.9456	0.9239	0.9346	0.2114	0.2744
	LLaMA 3.1 (8B)	0.5194	0.2926	0.4414	0.9369	0.9114	0.9239	0.2023	0.2593
	Mistral v0.3 Instruct (7B)	0.5964	0.3428	0.4963	0.9287	0.9319	0.9302	0.2629	0.3086
	Deepseek LLaMA (8B)	0.5691	0.3144	0.4646	0.9366	0.9246	0.9305	0.2176	0.2822

N-gram Ratio: The N-gram Ratio measures lexical diversity by comparing the number of novel n-grams in the summary to the total number of n-grams:

$$\text{N-gram Ratio} = \frac{\text{Number of novel } n\text{-grams}}{\text{Total } n\text{-grams in summary}} \quad (3)$$

A higher ratio indicates greater linguistic variety, showing that the model uses a richer vocabulary.

Conciseness: Conciseness is determined by comparing the length of the summary to that of the original transcript:

$$\text{Conciseness} = \frac{\text{Number of words in summary}}{\text{Number of words in original text}} \quad (4)$$

A lower value indicates that the summary is succinct, retaining only the most important content.

These extra measures offer a thorough system for assessing summary quality in a multilingual setting. They make sure that the summaries are varied, clear, and able to convey all vital information across English, Hindi, and Marathi in addition to correctness and fluency. The findings and metric values are presented in Table 4

5 OUTCOMES

The outcomes of this study are discussed in three main parts: the initial model evaluation, the fine-tuning results, and the multilingual performance analysis.

5.1 Performance of Model Families

Table 2 presents the evaluation of pre-trained large language models on English transcripts across zero-shot, one-shot, and three-shot inference settings. The evaluation metrics ROUGE, BERT Score, BLEU, and GLEU are used to determine how well each model captures essential content and maintains fluency in the generated summaries.

- ***Mistral Family*** Within the Mistral family, the base Mistral (7B) model achieves moderate scores in the zero-shot setting, while the Mistral v0.3 Instruct (7B) variant shows noticeable improvements in both one-shot and three-shot scenarios. This suggests that instruction tuning has a positive impact on its summarization capabilities, yielding higher overlap with reference summaries and better semantic alignment.

- ***LLaMA 3 Family*** For the LLaMA 3 family, three variants were tested. LLaMA 3 (8B) and LLaMA 3.2 (3B) deliver competitive results; however, LLaMA 3.1 (8B) consistently stands out. It produces the highest ROUGE scores, indicating superior content retention, and achieves the best BERT Score F1, reflecting strong semantic similarity with the reference summaries. LLaMA 3.1 (8B) shows improved BLEU and GLEU scores, which imply that the summaries are both fluent and well-structured.

- ***DeepSeek LLaMA (8B)*** It reaches competitive ROUGE and BERT Score values in the three-shot setting. Although its performance is notable, its overall scores are slightly lower compared to the top variants from the LLaMA and Mistral families.

Based on the results, the following models were chosen for further fine-tuning:

(1) **Mistral family:** Mistral v0.3 Instruct (7B)
(2) **LLaMA 3 family:** LLaMA 3.1 (8B)
(3) **DeepSeek family:** DeepSeek LLaMA (8B)

Table 3: Performance Comparison of Fine-tuned Models on English Transcripts

Model	ROUGE-L	BERT F1	BLEU	GLEU	Inference Time
LLaMA 3.1 (8B)	0.5178	0.9378	0.3253	0.3334	12.3s
Mistral v0.3 Instruct (7B)	0.4796	0.9350	0.2745	0.3496	16.5s
DeepSeek LLaMA (8B)	0.4527	0.7903	0.3103	0.2396	19.2s

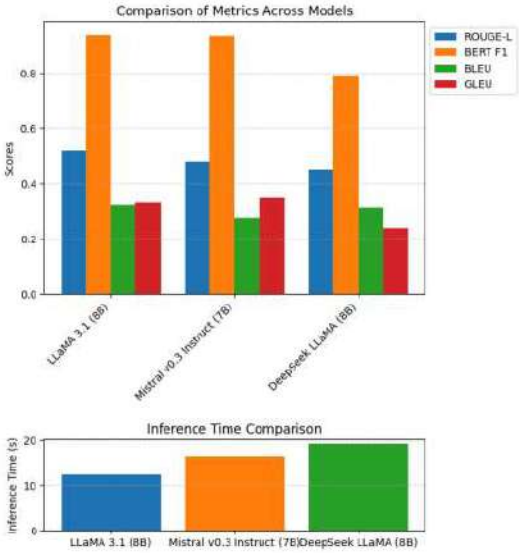


Figure 1: Performance comparison of fine-tuned models on English transcripts.

5.2 Results of Fine-Tuning Process and Comparative Analysis

The fine-tuning was executed as outlined in Section 4.3. Table 3 shows a detailed comparison of the three fine-tuned models, and Figure 1 shows a graphical comparison on English meeting transcripts.

- **LLaMA 3.1 (8B):** This model achieved a ROUGE-L score of 0.5178 and a BERT Score F1 of 0.9378. It also reached a BLEU of 0.3253 and a GLEU of 0.3334. These results reflect an improvement of approximately 7% in ROUGE-L and more than 35% in BERT Score F1 compared to some pre-fine-tuning results.
- **Mistral v0.3 Instruct (7B):** This model recorded a ROUGE-L of 0.479% and a BERT Score F1 of 0.9350 along with a BLEU of 0.2745 and a GLEU of 0.3496.
- **DeepSeek LLaMA (8B):** While DeepSeek LLaMA (8B) achieved a moderate BLEU score of 0.3103, its overall performance was lower, with a ROUGE-L of 0.4527 and a BERT Score F1 of 0.7903.

5.2.1 Inference Time Trade-offs. Practical uses also depend on inference time, which is as important as correctness. With an average time of 12.3 seconds per transcript, LLaMA 3.1 (8B) not only offered the best accuracy, but also attained the fastest inference. In contrast, DeepSeek LLaMA (8B) was the slowest at 19.2 seconds, while Mistral v0.3 Instruct (7B) needed 16.5 seconds. This trade-off between speed and accuracy is significant; quicker inference allows real-time summarization, which is absolutely vital for interactive systems. LLaMA 3.1 (8B) is the most practical option for deployment given the balance of high performance and low inference time.

DeepSeek LLaMA (8B) encountered challenges during fine-tuning. While it performed reasonably well on transcripts it had seen during training, it struggled to generate meaningful summaries for unseen transcripts. Adjustments to training parameters, such as reducing the maximum sequence length and modifying dropout rates, did not resolve this issue. As a result, DeepSeek was not selected for further testing. This discovery underlines the importance of a model's ability to generalize beyond the training data, a factor that is critical for real-world applications.

The comparison shows that the summary quality improved significantly as a result of fine-tuning. Compared to their pre-fine-tuning outputs presented in Table 2, the models produced more accurate and coherent summaries with faster processing times. The metrics make it evident that accuracy and speed must be traded off; LLaMA 3.1 (8B) provides the fastest inference time and high accuracy (with notable improvements in ROUGE-L and BERT Score F1), achieving the best overall balance. This led to LLaMA 3.1 being chosen for further multilingual fine-tuning and analysis.

5.3 Multilingual Performance Evaluation

After identifying LLaMA 3.1 (8B) as the top model on English transcripts, this model was further fine-tuned on an expanded dataset that includes English, Hindi and Marathi transcripts as explained in Section 4.1. A detailed analysis of the performance of the model in these languages is presented in Table 4 and Figure 2 using the metrics explained in Section 4.5.

The evaluation shows that the model delivers consistent performance in all languages. For example, Hindi transcripts achieved a slightly higher BERT Score (0.7281) than English (0.6533) and Marathi (0.6807), indicating a strong ability to capture semantic meaning. Low Redundancy values confirm that the summaries avoid repetitive content, while high Abstractivity scores demonstrate the model's ability to paraphrase and generate novel expressions while retaining essential information. Furthermore, the elevated N-gram Ratio and solid Conciseness scores attest that the summaries are both varied in vocabulary and succinct.

Table 4: Multilingual Evaluation of the Fine-tuned Model

Language	BERT Score	ICE	Redundancy	Abstractivity	N-gram Ratio	Conciseness
English	0.6533	0.4043	0.0347	0.8007	0.9690	0.6788
Hindi	0.7281	0.7702	0.1064	0.5361	0.8113	0.6203
Marathi	0.6807	0.5688	0.0677	0.8207	0.9663	0.6649

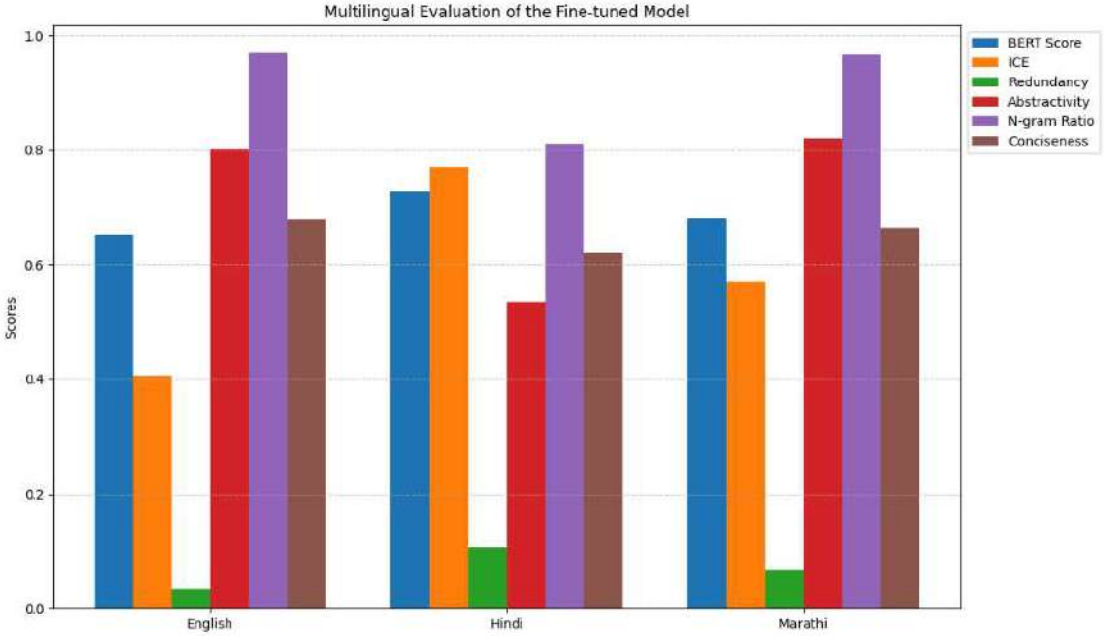


Figure 2: Multilingual Evaluation Metrics for LLaMA 3.1 (8B) across English, Hindi, and Marathi transcripts

6 CONCLUSION

The research in this paper presents a systematic evaluation and optimization procedure for meeting summarization models in a multilingual environment. Three model types were thoroughly tested with zero-shot, one-shot, and three-shot prompting techniques on a specially created dataset of career guidance meeting transcripts. ROUGE-L, BERT Score F1, BLEU, and GLEU were used as metrics to evaluate and determine which model worked best for each type. Fine-tuning experiments with Unsloth's 4-bit quantization architecture and Parameter-Efficient Fine-Tuning (PEFT) with LoRA validated that the LLaMA 3.1 (8B) model not only produced correct summaries, but also performed with exceptional efficiency and pace.

Furthermore, applying the best-performing model to handle multiple languages demonstrated its ability to extract important information in English, Hindi, and Marathi transcripts. The results of the experiment emphasize the requirement of domain-specific fine-tuning for domain-related tasks and indicate the potential of open-source LLMs to develop working and resource-effective AI tools for career guidance.

Future research could investigate comparative tests between supervised fine-tuning and reinforcement learning or reward-based optimization techniques to optimize structured, customized summarization. Future research can also examine trade-offs between instruction tuning and fine-tuning, optimization for live summarization, and more pervasive applications in other professional domains. In general, the study reflects a clear methodology from model choice to effective multilingual summarization, and with it, establishes a solid foundation for next-generation AI-powered communication tools.

REFERENCES

[1] Jay Peters. Google's Meet teleconferencing service now adding about 3 million users per day – theverge.com. https://www.theverge.com/2020/4/28/21240434/google-meet-three-million-users-per-day-pichai-earnings. [Accessed 14-10-2024].

[2] Tom Warren. Zoom grows to 300 million meeting participants despite security backlash – theverge.com. https://www.theverge.com/2020/4/23/21232401/zoom-300-million-users-growth-coronavirus-pandemic-security-privacy-concerns-

response. [Accessed 14-10-2024].

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

[4] Md Tahmid Rahman Laskar, Xue-Yong Fu, Cheng Chen, and Shashi Bhushan TN. Building real-world meeting summarization systems using large language models: A practical perspective, 2023. URL https://arxiv.org/abs/2310.19233.

[5] Fei Ge. *Fine-tune Whisper and transformer large language model for meeting summarization*. PhD thesis, UCLA, 2024.

[6] Aatman Vaidya, Tarunima Prabhakar, Denny George, and Swair Shah. Analysis of indic language capabilities in llms, 2025. URL https://arxiv.org/abs/2501.13912.

[7] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, and et. al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[8] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, and et. al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

[9] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

[10] Michael Han Daniel Han and Unsloth team. Unsloth, 2023. URL http://github.com/unslothai/unsloth.

[11] Nima Sadri, Bohan Zhang, and Bihan Liu. Meetsum: Transforming meeting transcript summarization using transformers!, 2021. URL https://arxiv.org/abs/2108.06310.

[12] Sumedh S Bhat, Uzair Ahmed Nawaz, Sujay M, Nameesha Tantri, and Vani Vasudevan. Jotter: An approach to summarize the formal online meeting. In *2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE)*, pages 1–6, 2023. doi: 10.1109/AIKIIE60097.2023.10390455.

[13] Lakshmi Prasanna Kumar and Arman Kabiri. Meeting summarization: A survey of the state of the art, 2022. URL https://arxiv.org/abs/2212.08206.

[14] Medha Wyawahare, Madhuri Shelke, Siddharth Bhorge, and Rohit Agrawal. Ai powered multilingual meeting summarization. In *2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 86–91, Jan 2024. doi: 10.1109/Confluence60223.2024.10463307.

[15] Aseem Srivastava, Tharun Suresh, Sarah P. Lord, Md Shad Akhtar, and Tanmoy Chakraborty. Counseling summarization using mental health knowledge guided utterance filtering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 3920–3930, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539187. URL https://doi.org/10.1145/3534678.3539187.

[16] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, K. McKeown, and Tatsunori Hashimoto. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57, 2023. doi: 10.1162/tacl_a_00632.

[17] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantizedllms, 2023. URL https://arxiv.org/abs/2305.14314.

[18] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

[19] Geetanjali Singh, Namita Mittal, and Satyendra Singh Chouhan. Hindisumm: A hindi abstractive summarization benchmark dataset. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 23(12), November 2024. ISSN 2375-4699. doi: 10.1145/3696207. URL https://doi.org/10.1145/3696207.

[20] Daisy Monika Lal, Paul Rayson, Krishna Pratap Singh, and Uma Shanker Tiwary. Abstractive Hindi text summarization: A challenge in a low-resource setting. In Jyoti D. Pawar and Sobha Lalitha Devi, editors, *Proceedings of the 20th International Conference on Natural Language Processing (ICON)*, pages 603–612, Goa University, Goa, India, December 2023. NLP Association of India (NLPAI). URL https://aclanthology.org/2023.icon-1.58/.

[21] Tong Xiao and Jingbo Zhu. Foundations of large language models, 2025. URL https://arxiv.org/abs/2501.09223.

[22] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

[23] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

[24] Maxime Peyrard, Teresa Botschen, and Iryna Gurevych. Learning to score system summaries for better content selection evaluation. In Lu Wang, Jackie Chi Kit Cheung, Giuseppe Carenini, and Fei Liu, editors, *Proceedings of the Workshop on New Frontiers in Summarization*, pages 74–84, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4510. URL https://aclanthology.org/W17-4510/.

[25] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020. URL https://arxiv.org/abs/1904.09675.

Unsloth PEFT based Multilingual Meeting Summarization with Open-Source LLMs

ORIGINALITY REPORT

7%	5%	4%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	arxiv.org Internet Source	3%
2	www.riverpublishers.com Internet Source	2%
3	R. N. V. Jagan Mohan, B. H. V. S. Rama Krishnam Raju, V. Chandra Sekhar, T. V. K. P. Prasad. "Algorithms in Advanced Artificial Intelligence - Proceedings of International Conference on Algorithms in Advanced Artificial Intelligence (ICAAAI-2024)", CRC Press, 2025 Publication	1%
4	El Habib Nfaoui, Hanane Elfaik. "Evaluating Arabic Emotion Recognition Task Using ChatGPT Models: A Comparative Analysis between Emotional Stimuli Prompt, Fine-Tuning, and In-Context Learning", Journal of Theoretical and Applied Electronic Commerce Research, 2024 Publication	1%
5	Zhang, Ruiru. "MARS: MedicAl thRead Summarization Dataset Based on IIYI With Comparative Analysis of Large Language Models", University of Washington, 2025	1%