G H Patel College of Engineering & Technology
(A Constituent College of CVM University)

Automated Meeting Analyzer

A mini project report submitted
In partial fulfilment of the requirements for
the Degree of Bachelor of Technology

In
Computer Engineering (CP)
Semester – VII

By

Mitali Sharma (12202040501040)

Ruchi Tandel (12202040501055)

Faculty Guide
Prof. Vraj Hingu

Natural Language Processing (202047809)

A.Y. 2025-26 (ODD TERM)

Department of Computer Engineering
G H Patel College of Engineering & Technology
Bakrol Road, Vallabh Vidyanagar

Automated Meeting Analyzer

1. Abstract

In many organizations, critical information, decisions, and actionable tasks discussed in meetings are often lost or poorly documented, leading to missed deadlines and a decline in productivity. This project introduces an automated system built with Natural Language Processing (NLP) to analyze meeting transcripts. The system leverages Latent Dirichlet Allocation (LDA) for topic modeling to identify key discussion themes and employs a rule-based approach with spaCy for Named Entity Recognition (NER) to extract action items, assignees, and deadlines. The final output is a structured summary that makes meeting outcomes easily accessible. This tool improves efficiency by ensuring that crucial information and responsibilities are captured, helping teams stay aligned and accountable.

2. Introduction

Meetings are fundamental to business collaboration, but the manual process of summarizing discussions and tracking outcomes is inefficient and prone to human error. Key action items can be overlooked, and the core themes of a conversation can be misinterpreted. This project addresses the gap between conversation and action by creating an intelligent tool that automates the analysis of meeting transcripts.

The **Automated Meeting Analyzer** is designed to process raw text from meetings and generate concise, actionable insights. By using advanced NLP techniques, the system identifies the primary topics of discussion and extracts a clear list of tasks, ensuring that important decisions and responsibilities are not forgotten. This streamlines post-meeting workflows and empowers teams to be more organized and productive.

3. System Architecture and Methodology

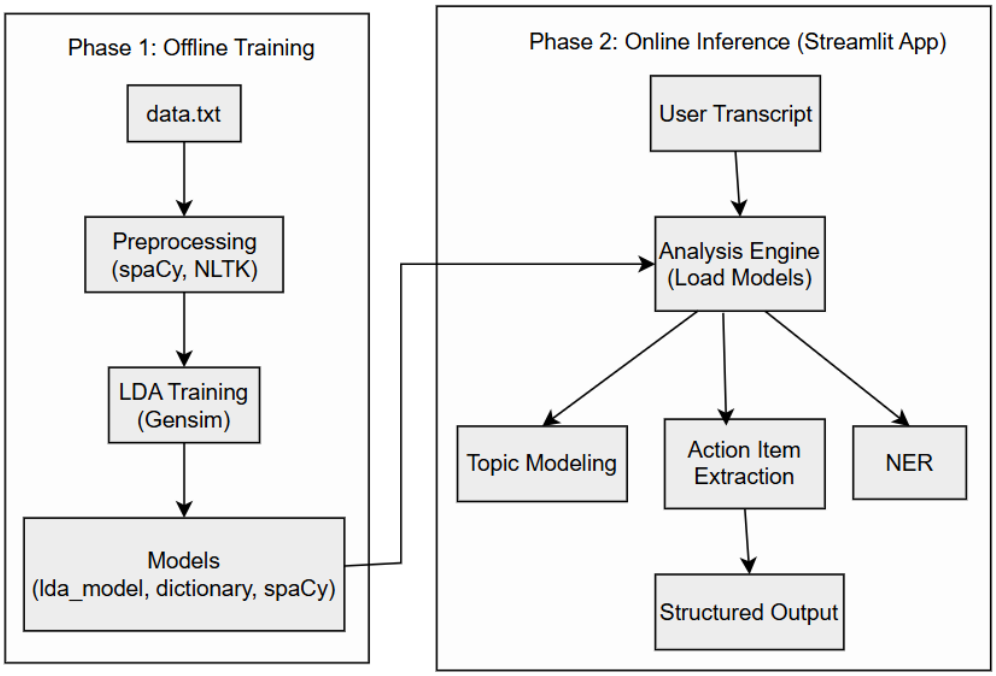
The system is designed with a two-phase architecture: an offline phase for model training and an online phase for real-time analysis via a web application.

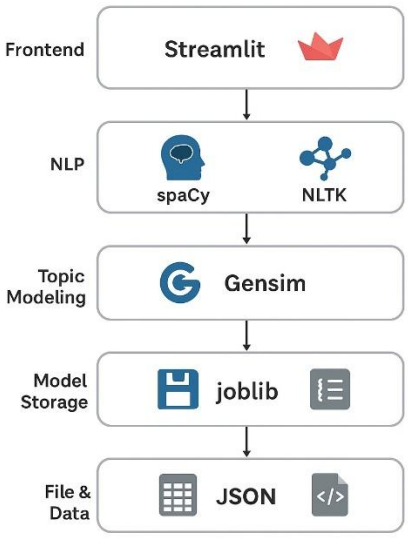
Phase 1: Offline Training

- **Data Parsing and Preprocessing:** The raw text from the transcripts is parsed to separate dialogues by speaker. Using NLP libraries like **spaCy** and **NLTK**, the text is cleaned by converting it to lowercase, removing punctuation and stop words, and performing lemmatization to reduce words to their root forms.
- **Topic Modeling:** The preprocessed text is then used to train a Latent Dirichlet Allocation (LDA) model with **Gensim**. The model was configured to identify five distinct topics from the meeting data, allowing it to categorize discussions into themes such as "Rural Customer Strategy," "Customer Support Automation," and "Sales Team Training."
- **Model Storage:** The trained LDA model and its corresponding dictionary are serialized and saved as model files using the **joblib** library for later use in the application.

Phase 2: Online Inference (Streamlit Application)

- **Frontend:** A web-based interface was developed using **Streamlit**. This allows users to paste a meeting transcript directly into a text box for analysis.
- **Analysis Engine:** The Streamlit application loads the pre-trained LDA model, dictionary, and spaCy model. When a user submits a transcript, the application runs the same preprocessing pipeline used during training.
- **Insight Extraction:**
    - **Topic Prediction:** The cleaned text is passed to the LDA model to identify the dominant topic and associated keywords.
    - **Action Item Extraction:** A rule-based function scans the text for specific keywords and linguistic patterns (e.g., "we will," "next step is," modal verbs like "should") to identify and extract sentences that represent actionable tasks.
    - **Named Entity Recognition (NER):** spaCy's NER model is used to identify and extract key entities such as people (PERSON), organizations (ORG), and dates (DATE), providing context for the discussion and action items.
- **Structured Output:** The extracted topics, action items, and entities are presented to the user in a clean, organized format.

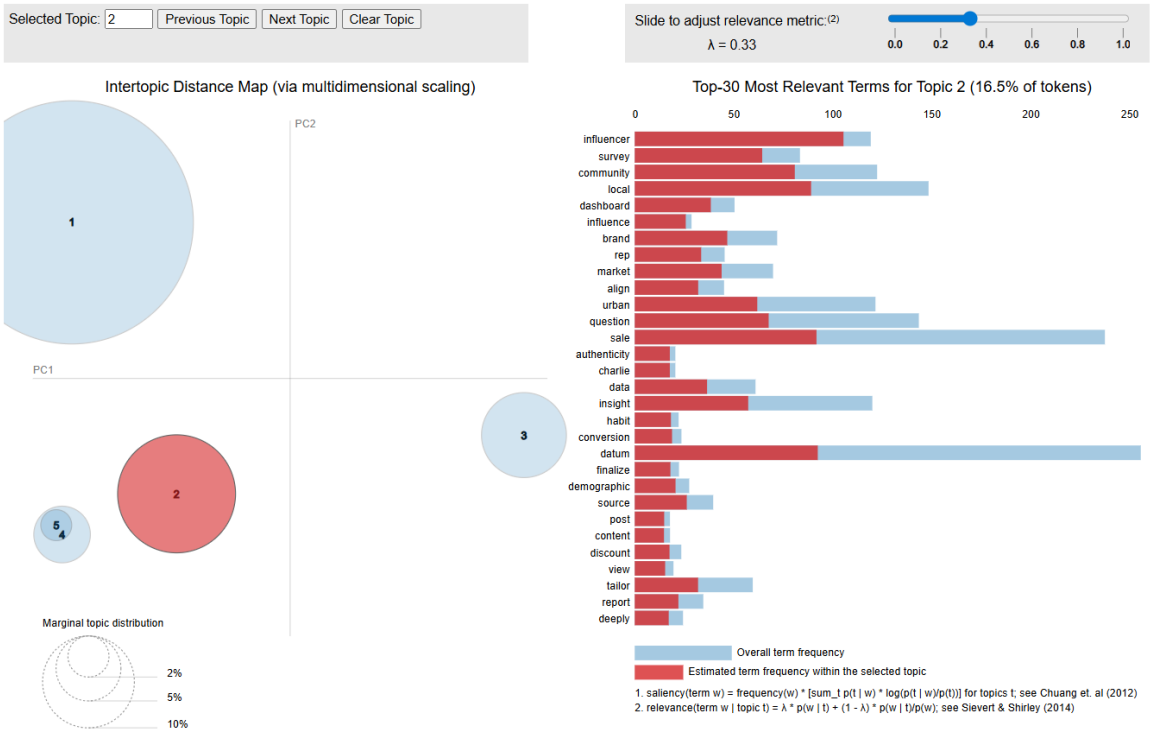




Topic Visualization and Interpretation

To interpret the topics generated by the LDA model, the pyLDAvis library was used to create an interactive visualization. This was a key step in the development process, allowing for a deeper understanding of the model's output. The visualization provides:

- An **Intertopic Distance Map**, which plots topics as circles in a 2D space. The size of each circle represents its overall prevalence, and the distance between circles indicates their similarity.
- A **Salient Terms Bar Chart**, which displays the most relevant keywords for a selected topic. This interactivity is crucial for assigning a meaningful name and interpretation to each computer-generated topic.



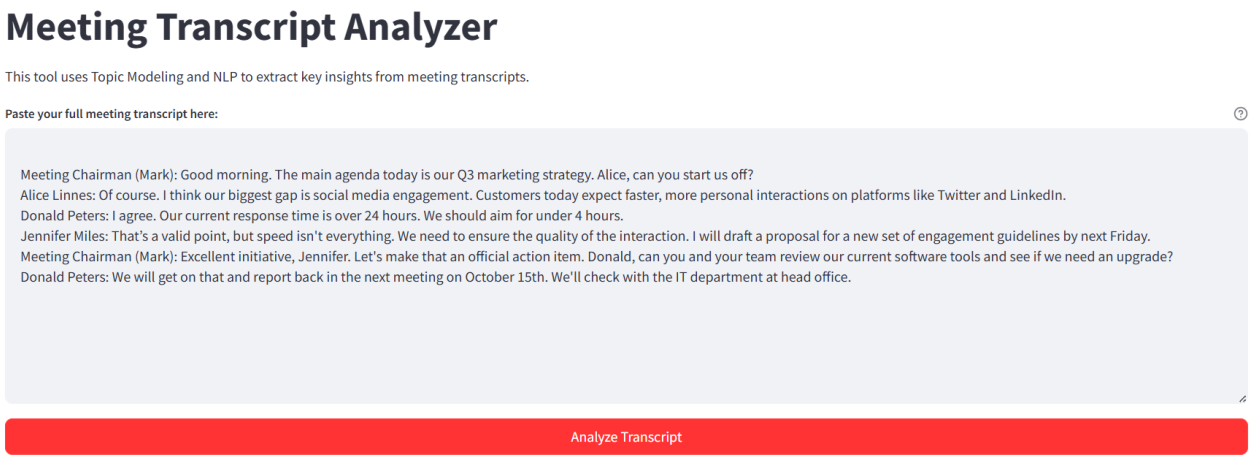
4. Models Chosen

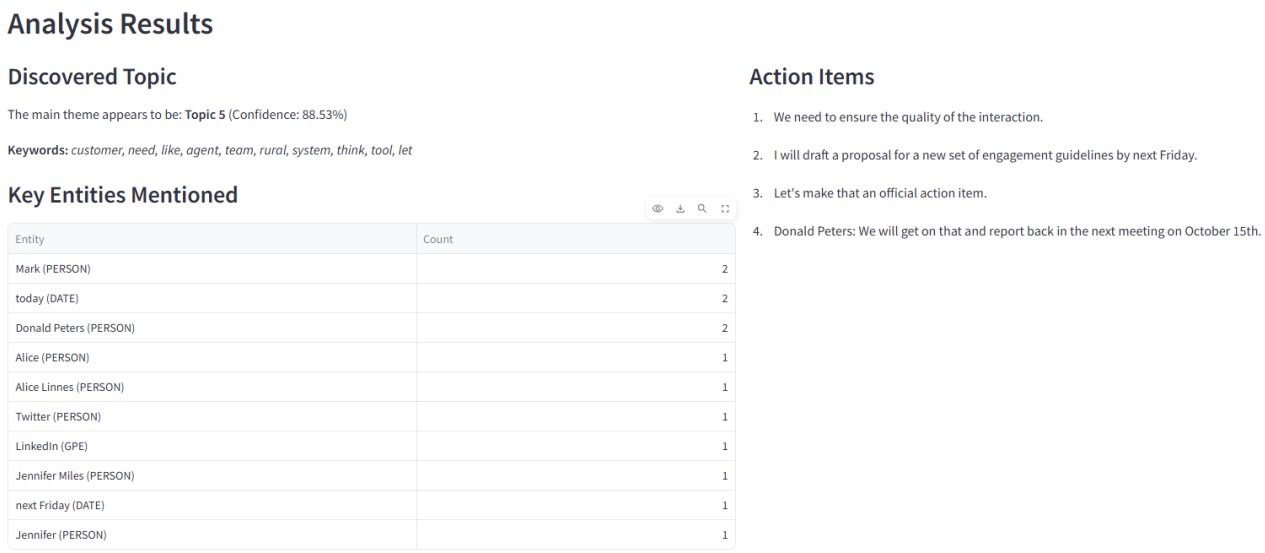
- **Topic Modeling: Latent Dirichlet Allocation (LDA) with Gensim**
    - **What it is:** LDA is a probabilistic topic model that discovers abstract topics within a collection of documents. It assumes that each document is a mixture of topics and that each topic is a mixture of words.
    - **Why it was chosen:** LDA is an industry-standard for unsupervised topic modeling and is highly effective for identifying latent thematic structures in text data like meeting transcripts. The **Gensim** library was chosen for its efficient implementation of LDA and its seamless integration with other Python data science tools.

5. Results and Discussion

The system was trained and evaluated on a dataset of meeting transcripts, yielding the following results:

- **Meetings Analyzed:** 11
- **Dialogue Turns Processed:** 650
- **Vocabulary Size:** 1,727 unique words
- **Topics Generated:** 5
- **Action Items Identified:** 75
- **Named Entities Extracted:** 112
- **Total Processing Time:** 8.24 seconds





Analysis Results

Discovered Topic

The main theme appears to be: **Topic 5** (Confidence: 88.53%)

Keywords: *customer, need, like, agent, team, rural, system, think, tool, let*

Key Entities Mentioned

Entity	Count
Mark (PERSON)	2
today (DATE)	2
Donald Peters (PERSON)	2
Alice (PERSON)	1
Alice Linnes (PERSON)	1
Twitter (PERSON)	1
LinkedIn (GPE)	1
Jennifer Miles (PERSON)	1
next Friday (DATE)	1
Jennifer (PERSON)	1

Action Items

1. We need to ensure the quality of the interaction.
2. I will draft a proposal for a new set of engagement guidelines by next Friday.
3. Let's make that an official action item.
4. Donald Peters: We will get on that and report back in the next meeting on October 15th.

6. Challenges and Limitations

- **Data Quality:** The model's performance is highly dependent on the quality of the transcript. Ambiguous language or transcription errors can affect the accuracy of topic modeling and action item extraction.
- **System Integration:** As discussed in the meetings, the operational effectiveness of the sales and support teams is hampered by fragmented IT systems. While the analyzer can identify these issues, it cannot solve the underlying technical debt.
- **Rule-Based Limitations:** The action item extractor relies on predefined keywords and patterns. It may miss nuanced tasks or incorrectly flag sentences that are not true action items.

7. Applications and Future Scope

Applications:

- **Corporate:** Automated generation of Minutes of Meeting (MoM).
- **Project Management:** Tracking tasks and deliverables from team check-ins.
- **Customer Support:** Analyzing sales and support calls to identify pain points and opportunities.
- **Education:** Summarizing research discussions or group study sessions.

Future Scope:

- **Advanced Action Item Assignment:** Develop a more sophisticated model to automatically assign tasks to the correct individuals identified in the meeting.
- **Sentiment Analysis:** Incorporate sentiment analysis to gauge the mood of the meeting or identify customer frustration in support calls.
- **Integration with Project Management Tools:** Create APIs to automatically export extracted action items to platforms like Jira, Asana, or Trello.

- **Real-time Analysis:** Explore the possibility of integrating the tool with live meeting platforms for real-time transcription and analysis.

8. Conclusion

This project successfully developed an end-to-end NLP system for automating the analysis of meeting transcripts. By combining Latent Dirichlet Allocation for topic modeling with a rule-based approach for information extraction, the **Automated Meeting Analyzer** effectively transforms unstructured conversations into structured, actionable insights.

The tool not only identifies key discussion themes, such as the strategic focus on rural markets and internal system integration challenges, but also extracts a clear list of action items, ensuring that critical tasks are not overlooked.

Ultimately, this project provides a robust and practical solution to a common organizational problem. By reducing the manual effort required to document meeting outcomes, the analyzer enhances productivity, promotes accountability, and ensures that the valuable information shared in meetings is effectively captured and utilized.

9. References

- Streamlit Documentation: https://docs.streamlit.io/
- Gensim: Topic Modeling for Python: https://radimrehurek.com/gensim/
- spaCy: Industrial-Strength Natural Language Processing: https://spacy.io/
- NLTK: Natural Language Toolkit: https://www.nltk.org/
- pyLDAvis: Interactive Topic Model Visualization: https://pyldavis.readthedocs.io/
- Joblib Documentation: https://joblib.readthedocs.io/