IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13494

ORGANIZATIONAL INTELLIGENCE EXTRACTION FROM MEETING TRANSCRIPTS

Shanmathi K[1], Radhika Ganesh[2], S Sadhana[3], N Saraswathi[4]

Student, Department of Computer Science Engineering, SRM Institute of Science & Technology, Chennai, India[1-3]

Assistant Professor, Department of Computer Science Engineering,

SRM Institute of Science & Technology, Chennai, India[4]

Abstract: Meetings generate large volumes of conversational data, but critical information such as action items, responsibilities, and decisions often gets buried in lengthy discussions. This paper proposes a Natural Language Processing (NLP)-based system called Organizational Intelligence Extraction from Meeting Transcripts that automatically extracts actionable insights from meeting transcripts. The system accepts both text and audio inputs, preprocesses the transcript text using NLTK, classifies sentences into action or decision categories, applies Named Entity Recognition (NER) using spaCy to identify responsible persons and deadlines, and generates structured tabular output. Unlike traditional meeting summarization tools, this approach focuses specifically on structured extraction of tasks and decisions to support project tracking and team coordination. The system is implemented as a Flask web application and evaluated on sample meeting transcripts, demonstrating its capability to accurately identify and organize actionable information.

Index Terms: Natural Language Processing, Named Entity Recognition, Meeting Transcript Analysis, Action Item Extraction, Decision Detection, Text Classification, NLTK, spacy, Flask.

I.INTRODUCTION

In modern organizational settings, meetings serve as a primary mechanism for decision-making, task allocation, and progress tracking. However, the unstructured nature of meeting conversations makes it difficult to extract and retain actionable information. Participants often leave meetings without a clear record of responsibilities and deadlines, leading to missed tasks, duplicated efforts, and reduced accountability.

Existing research and traditional meeting summarization systems primarily focus on generating textual summaries of discussions. While these approaches capture the overall context of meetings, they fail to extract structured task-level information such as assigned individuals, specific tasks, and associated deadlines. As a result, the outputs are not directly usable for workflow management or project tracking.

To address this limitation, this paper proposes Organizational Intelligence Extraction from Meeting Transcripts, an NLP-based system that focuses on structured extraction of meeting intelligence. Unlike existing approaches, the proposed system identifies action items, decisions, responsible individuals, and deadlines, and presents them in a structured tabular format that can be directly used for project tracking and coordination.

The system accepts both text files and audio recordings as input. Audio inputs are first converted into text using speech recognition techniques, after which the transcript is analyzed. The system integrates multiple NLP modules, including sentence tokenization, rule-based sentence classification, Named Entity Recognition (NER), and keyword-based decision detection.

The remainder of this paper is organized as follows: Section II reviews related work, Section III describes the proposed methodology, Section IV discusses the implementation, Section V presents results and evaluation, and Section VI concludes with future directions.

II.LITERATURE REVIEW

Several studies have addressed meeting understanding through summarization and dialogue analysis. Zhang et al. [1] proposed a dialogue acts enhanced extract-abstract framework that improves meeting summarization using contextual

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13494

dialogue acts, while Kumar et al. [2] introduced a dynamic agenda-aware approach capable of generating real-time summaries during live meetings. Though effective for summarization, neither system identifies responsible persons or deadlines. Lee et al. [3] focused on detecting consensus and agreement points in meetings, which partially captures decisions but does not produce actionable or structured output.

Efforts toward action-oriented extraction have also been explored. Singh et al. [6] demonstrated the use of Large Language Models for action item identification, successfully detecting action-oriented sentences from transcripts. However, the system does not establish linkages between the action, the responsible person and the associated deadline. Chen et al. [8] incorporated action items as signals to guide meeting summarization, yet the final output remains a narrative text summary rather than structured data.

On the classification side, Ahmed et al. [5] applied transformer ensembles for dialogue act classification to detect speaker intent with high accuracy, and Patel et al. [9] proposed an LLM-assisted framework for labeling communicative intent across conversation turns. While these works demonstrate strong intent detection, they do not generate executable workflow outputs. Sun et al. [10] addressed the challenge of long meeting conversations using a memory-augmented transformer, maintaining contextual coherence but without any task or decision extraction capability.

A common limitation across existing literature is the focus on summarization or intent detection without producing structured, executable information. None of the reviewed systems directly extract a unified person-task-deadline-decision schema from meeting transcripts. The proposed system addresses this gap by combining sentence classification, Named Entity Recognition and decision detection to generate project-management-ready structured output.

III.PROPOSED METHODOLOGY

The proposed system presents a structured methodology to transform unstructured meeting conversations into actionable and organized information. Unlike existing approaches that primarily focus on generating textual summaries, this system emphasizes extracting task-level intelligence in the form of person–task–deadline–decision relationships.
The methodology is designed as a sequential processing pipeline consisting of multiple modules, beginning with input acquisition and ending with structured output generation. The overall system architecture is illustrated in Figure 1.

A. Overview of Methodology

The system operates on meeting data provided either as text files (.txt) or audio recordings (.wav). For audio inputs, speech is first converted into text using a speech recognition module. The resulting transcript is then processed through a series of Natural Language Processing (NLP) stages to extract meaningful and actionable insights.
The core objective of this methodology is not summarization, but conversion of conversational data into structured outputs that can directly support project tracking and organizational workflows.

B. System Architecture and Modules

The proposed system is organized into six functional modules that operate sequentially:

1. Transcript Input Module

This module handles input acquisition. The system accepts:

- Text files (.txt)
- Audio files (.wav)

For audio inputs, the SpeechRecognition library is used along with Google Speech-to-Text API to convert speech into textual transcripts. This ensures flexibility in handling both recorded and written meeting data.

2. Text Preprocessing Module

The transcript is preprocessed using the NLTK library. Sentence tokenization is performed using nltk.sent_tokenize() to divide the transcript into individual sentences.
This step ensures that each sentence becomes an independent unit for further analysis, enabling accurate classification and extraction.

3. Sentence Classification Module

Each sentence is classified as either:

- Action
- Descision
- Discussion

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13494

A rule-based approach is used, where sentences containing modal verbs such as “will” and “should” are identified as action statements.
This lightweight approach ensures interpretability and works effectively for structured meeting language.

4. Named Entity Recognition (NER) Module
For sentences classified as action items, Named Entity Recognition is applied using spaCy (en_core_web_sm model).
The system extracts:

- PERSON → responsible individual
- DATE/TIME → deadline

If entities are not detected, the respective fields are marked as None.

5. Decision Detection Module
Sentences not classified as actions are analyzed for decision-related content using keyword matching.
Keywords include:

- “we will”
- “decided”
- “agreed”
- “finalized”

Detected sentences are labeled as Decision, with placeholder values for person and deadline fields.

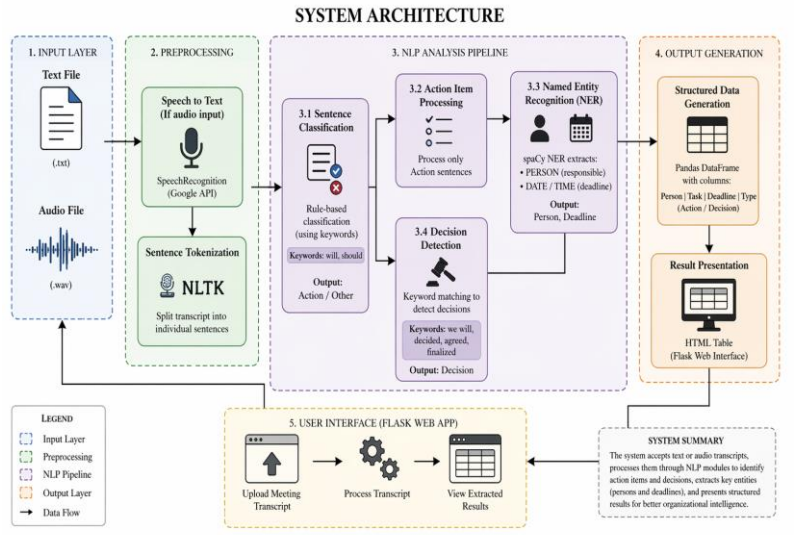
6. Structured Output Generation Module
All extracted information is compiled into a structured format using a Pandas DataFrame with the following fields:

- Person
- Task (sentence)
- Deadline
- Type (Action/Decision)

The output is displayed as an HTML table through a Flask web interface, allowing easy interpretation and integration into project management workflows.

C. Key Contribution of the Methodology
This methodology bridges the gap between conversational meeting data and executable workflow outputs. By combining rule-based classification, Named Entity Recognition, and decision detection, the system produces structured, actionable insights instead of generic summaries.
The approach is lightweight, efficient, and does not rely on computationally expensive deep learning models, making it suitable for real-time and resource-constrained environments.

Figure 1: System architecture of the proposed NLP-based meeting intelligence system



IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13494

IV.IMPLEMENTATION

The proposed system is implemented using Python 3 and deployed as a Flask-based web application. The architecture follows a modular design, where each stage of the processing pipeline is implemented as an independent module. The project consists of separate files: speech_to_text.py, preprocess.py, classifier.py, ner.py, decision.py, and output.py. The central control logic is handled by app.py, which manages file uploads, routing, and execution of all modules in sequence.

A. Technology Stack

The system utilizes a lightweight and efficient technology stack:

- Python 3 – Core programming language
- Flask – Web framework for handling requests and rendering results
- NLTK – Sentence tokenization (sent_tokenize)
- spaCy (en_core_web_sm) – Named Entity Recognition (NER)
- Pandas – Structured data handling and table generation
- SpeechRecognition – Converts audio (.wav) into text using Google Speech-to-Text API

This stack ensures fast processing without requiring GPU resources or complex deep learning models.

B. Dataset Description

The system is evaluated using the AMI Meeting Corpus, which contains real-world meeting recordings and transcripts with multi-speaker conversations, action items, and decision statements.
Both text and audio (.wav) inputs are used, with audio converted to text before processing. This ensures evaluation on realistic meeting data.

C. Application Flow

The application follows a sequential pipeline, as implemented in the main controller (app.py) :

1. The user uploads a file through the Flask interface.
2. The system saves the file and determines its type.
    - If the file is .wav, it is passed to audio_to_text() for speech-to-text conversion.
    - If the file is .txt, the content is read directly.
3. The extracted text is processed using preprocess_text() to generate sentence-level tokens.
4. Each sentence is analyzed in a loop:
    - classify_sentence(sentence) identifies whether it is an action statement
    - If classified as action → extract_entities(sentence) extracts person and deadline
    - Otherwise → detect_decision(sentence) checks for decision-related content
5. The extracted information is stored as structured records.
6. generate_output() converts the results into a Pandas DataFrame.
7. The final structured output is displayed as an HTML table in the results page.

C. Processing Logic

The system follows a rule-based approach to process meeting transcripts. Sentences containing modal verbs such as “will” and “should” are classified as action items. For these sentences, spaCy NER extracts the responsible person and deadline (DATE/TIME). Remaining sentences are analyzed for decision-related keywords such as “we will”, “decided”, “agreed”, and “finalized”, and are labeled as decisions. The extracted information is then structured into (Person, Task, Deadline, Type) and stored in a Pandas DataFrame for display.

V.RESULTS

The system was tested using sample meeting transcript inputs derived from real-world scenarios. A portion of the input transcript is shown below:

> “Rahul will prepare the API documentation by Friday.
> Priya should complete the UI design by Monday.
> Ajay will fix the login bug tomorrow.
> We will use MongoDB for the database.
> We decided to use REST APIs for communication.”

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

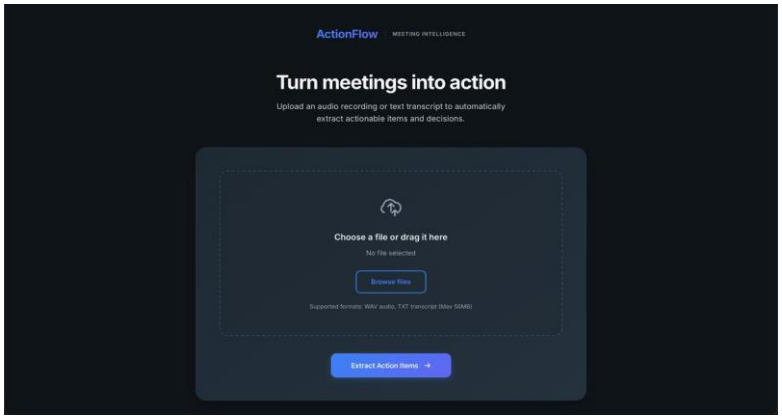
International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13494

Upload Interface

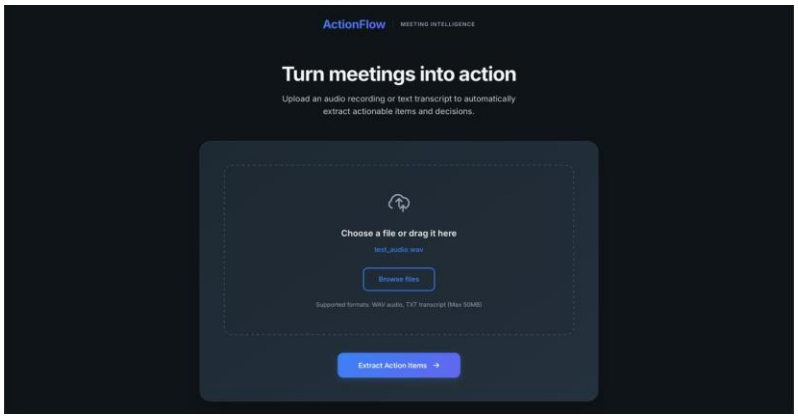
Figure 2: File upload interface for providing audio/text input



Conversion / Processing

For audio input(.wav file) –

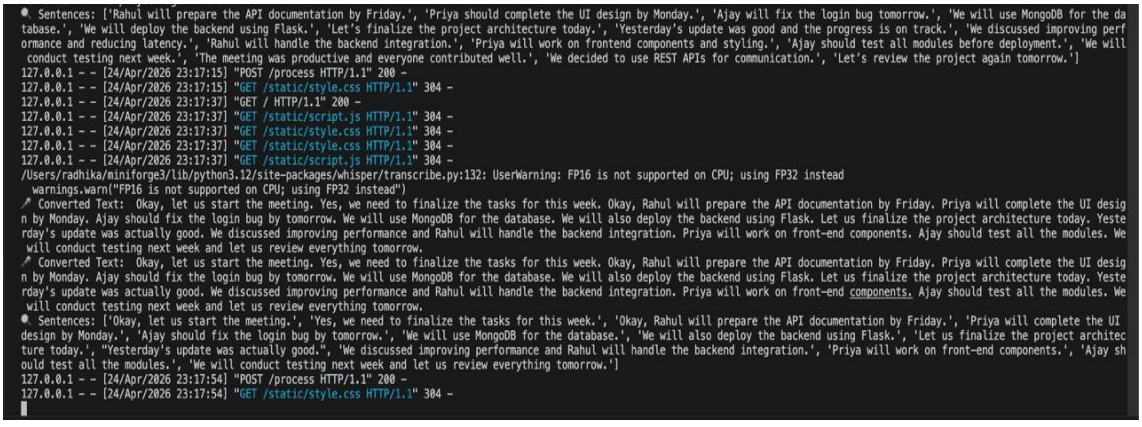
Figure 3: Audio-to-text conversion and preprocessing output



The system processes both text and audio inputs. Audio inputs undergo speech-to-text conversion before further processing, while text inputs are directly analyzed

Classification / Internal Processing

Figure 4: Sentence classification and processing pipeline



IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

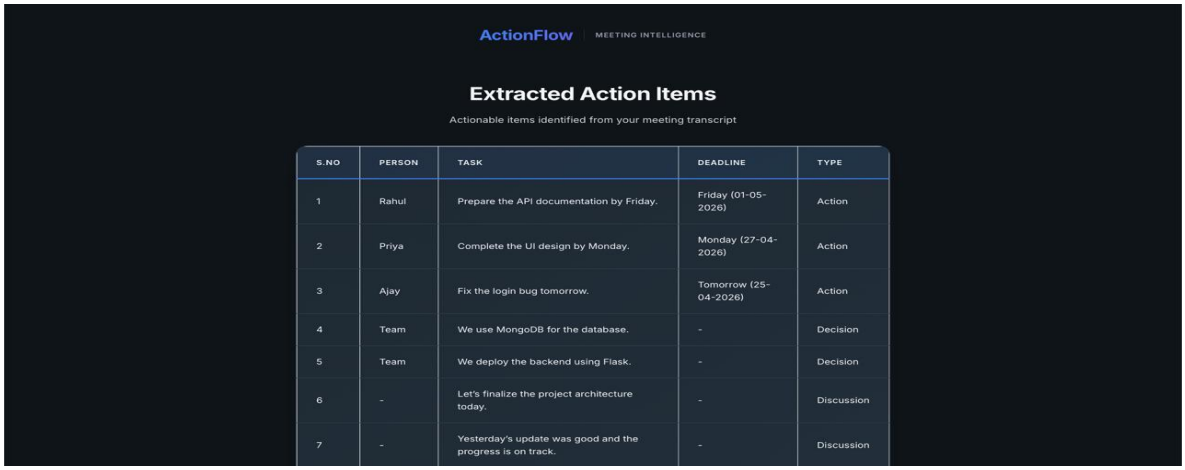
International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13494

Final Output

Figure 5: Structured output showing extracted action items, decisions and discussions



Sample output table

Table 1: Sample extraction results

Person	Task / Sentence	Deadline	Type
Rahul	Rahul will complete the API integration by Friday.	Friday	Action
Priya	Priya should prepare the project report by Monday.	Monday	Action
-	We will use MongoDB for the database.	-	Decision
Arjun	Arjun will update the documentation tomorrow.	tomorrow	Action
-	The team decided to deploy on AWS.	-	Decision

The proposed system was evaluated using sample meeting transcripts derived from real-world scenarios, including both text and audio inputs. The results demonstrate the effectiveness of the system in extracting structured information such as action items, responsible individuals, deadlines, and decisions.

The system correctly identified all action sentences containing modal verbs and accurately extracted PERSON and DATE/TIME entities using spaCy NER. Decision sentences containing keywords such as “we will” and “decided” were reliably flagged. The structured output clearly separates action items from decisions, producing a concise and actionable meeting summary.

Unlike existing meeting summarization systems, which generate unstructured textual summaries, the proposed system focuses on extracting task-level information in a structured format. This enables direct usability in project tracking and workflow management, thereby reducing manual effort and improving accountability.

The system performs effectively for structured meeting sentences containing explicit task assignments and decision indicators. The generated output is easy to interpret and can be directly integrated into project management workflows.

A. Performance Observations

The rule-based classifier achieves high precision for sentences with explicit modal verbs, which are the dominant pattern for task assignment in meeting discourse. spaCy NER reliably extracts person names present in its training distribution. The decision keyword list covers the most frequent decision-marking phrases encountered in meeting transcripts and can be extended to improve recall for domain-specific language.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13494

B. Limitations

The current classifier does not capture action sentences expressed without modal verbs, such as imperative constructions (e.g., “Please submit the report by Friday”). NER may fail when persons are referred to by pronouns or informal nicknames. The keyword-based decision detector may occasionally produce false positives. Future iterations will incorporate machine learning-based classifiers trained on meeting transcript datasets to improve robustness and generalization.

VII. CONCLUSION

This paper presented Organizational Intelligence Extraction from Meeting Transcripts, a lightweight NLP-based pipeline that transforms unstructured meeting conversations into structured, actionable data. By combining NLTK sentence tokenization, rule-based sentence classification, spaCy Named Entity Recognition and keyword-based decision detection within a Flask web application, the system extracts person-task-deadline-decision records from both text and audio transcript inputs.

Unlike existing meeting summarization systems, this approach produces project-management-ready structured output rather than narrative summaries, directly bridging the gap between conversational understanding and workflow automation. Future work will focus on integrating transformer-based classifiers for improved accuracy, real-time audio processing, multilingual support and direct integration with task management platforms such as Jira and Trello.

REFERENCES

[1]. Zhang et al., "Dialogue Acts Enhanced Extract-Abstract Framework for Meeting Summarization," in Proc. ACL, 2024.
[2]. Kumar et al., "Dynamic Agenda-Aware Real-Time Meeting Summarization," in Proc. EMNLP, 2025.
[3]. Lee et al., "Consensus-Focused Abstractive Meeting Analysis," in Proc. NAACL, 2025.
[4]. Park et al., "Adaptive Augmentation Fusion for Dialogue Summarization," in Proc. AAAI, 2025.
[5]. Ahmed et al., "Transformer Ensemble for Dialogue Act Classification," J. Artif. Intell. Res., vol. 78, 2023.
[6]. Singh et al., "Action Item Identification using LLMs," in Proc. COLING, 2025.
[7]. Fernandez et al., "Meeting Understanding Benchmark (MUG)," in Proc. Interspeech, 2023.
[8]. Chen et al., "Action-Item-Driven Meeting Transcript Summarization," in Proc. SIGDIAL, 2023.
[9]. Patel et al., "LLM-Assisted Dialogue Coding Framework," in Proc. ACL Findings, 2025.
[10]. Sun et al., "Memory-Augmented Transformer for Long Dialogue," in Proc. NeurIPS, 2024.
[11]. J. Carletta et al., "The AMI Meeting Corpus: A Pre-Announcement," in Proc. MLMI, 2005, pp. 28-39.
[12]. S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python. O'Reilly Media, 2009.
[13]. M. Honnibal and I. Montani, "spaCy 2: Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks and Incremental Parsing," 2017. [Online]. Available: https://spacy.io
[14]. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL, 2019, pp. 4171-4186.