Audio Interview Assessment System

Mayank Singh
Department of Information Technology
Netaji Subhas University of Technology
Delhi, India
mayank.singh.ug21@nsut.ac.in

Sakib Hasan
Department of Information Technology
Netaji Subhas University of Technology
Delhi, India
sakeeb.hasan.ug21@nsut.ac.in

Abhay Prajapati
Department of Information Technology
Netaji Subhas University of Technology
Delhi, India
abhay.prajapati.ug21@nsut.ac.in

***Abstract*—Automated analysis of interviews has gained significant attention as organizations seek objective, consistent, and scalable evaluation methods to mitigate challenges inherent in traditional human-led interviews, such as subjectivity and bias. This paper proposes an end-to-end "Audio Interview Assessment System" that processes interview recordings, extracts structured information, and generates comprehensive candidate performance assessments. The system leverages recent advancements in speech recognition, speaker diarization, natural language understanding, and Large Language Models (LLMs) to emulate aspects of human evaluators. A core contribution is a multi-phase pipeline that first converts raw audio into a diarized and transcribed format, then segments the conversation by topic, and finally utilizes a Retrieval-Augmented Generation (RAG) approach to assess candidate responses within each topic against a reference knowledge base. By combining multiple AI components, the proposed system aims to provide data-driven, context-aware interview analysis, thereby enhancing the fairness and efficiency of the recruitment process.**

***Index Terms*—Automated Interview Analysis, Speech Recognition, Speaker Diarization, Natural Language Processing, Large Language Models, Retrieval-Augmented Generation, Recruitment Technology.**

I. Introduction

INTERVIEWS play a crucial role in the recruitment process, serving as a bridge between candidates and potential employers. They assess various parameters such as communication skills, technical knowledge, problem-solving abilities, and cultural fit. Traditional interview methods rely heavily on human evaluators, making the process susceptible to biases, inconsistencies, and time inefficiencies. The subjective nature of manual assessments often results in varied judgments, which may not always be fair or accurate. With the rapid advancements in artificial intelligence (AI) and machine learning (ML), automated solutions for interview evaluations are becoming increasingly viable. Speech processing and natural language processing (NLP) technologies can now analyze verbal responses in real-time, offering objective and data-driven insights into a candidate's performance. The integration of AI in live interviews can enhance decisionmaking by evaluating tone, fluency, speech clarity, emotional cues, and confidence levels.

This paper proposes an end-to-end Audio Interview Assessment System designed to process audio recordings of interviews, extract structured information, and generate comprehensive assessments of candidate performance. The system leverages recent advancements in speech recognition for accurate transcription, speaker diarization to identify "who spoke when," natural language understanding to comprehend the content and context of responses, and Large Language Models (LLMs) to emulate the nuanced evaluation capabilities of human assessors. By automating key aspects of the analysis, our system aims to provide objective and scalable feedback.

The main contributions of this work are centered around a multi-phase pipeline. Firstly, the system ingests raw interview audio and converts it into a structured, diarized, and transcribed format. Secondly, it segments the conversation by topic to allow for focused analysis of responses within specific contexts. Finally, a core innovation is the utilization of a Retrieval-Augmented Generation (RAG) approach, where an LLM assesses candidate responses within each identified topic by referencing a predefined knowledge base of ideal answer characteristics and evaluation criteria. By combining these multiple components, the proposed Audio Interview Assessment System seeks to deliver data-driven, context-aware interview analysis, thereby aiming to enhance the overall fairness, consistency, and efficiency of the recruitment and evaluation process.

II. Background

A. Speaker Diarization

Speaker diarization determines "who spoke when?" in an audio recording. Modern diarization systems often use neural pipelines with voice activity detection, speaker change detection, and clustering. We adopt a toolkit similar to pyannote.audio, which provides trainable neural blocks for this task.

As shown in Figure 1, the raw interview audio is fed into a diarization model (e.g., Py-Annote) that outputs time-stamped speech segments for each speaker. A simple mapping heuristic then assigns speaker IDs to roles (e.g., interviewer vs. candidate) based on the first speaker rule or a short known prompt. Accurate diarization is crucial for attributing speech content to the correct participant in subsequent steps.

TABLE I
LITERATURE REVIEW SUMMARY

Reference	Methodology	Key Findings	Relevance
Sainath et al., *"Whisper-Streaming: Real-time Transcription with Whisper,"* 2023 (https://arxiv.org/abs/2307.14743) [11]	Implements Whisper-Streaming with LocalAgreement policy for low-latency, high-accuracy real-time transcription. [11]	Balances transcription speed and accuracy, achieving per-word latency as low as 0.5s. [11]	Highly relevant for real-time assessment of interview responses.
Kim et al., *"Simul-Whisper: Efficient Streaming Speech Recognition,"* 2024 (https://arxiv.org/abs/2406.10052) [6]	Uses chunk-based processing with attention-guided decoding to enable real-time transcription without fine-tuning.	Reduces latency while preserving transcription accuracy; designed for real-time applications without requiring fine-tuning. [6]	Useful for processing interview audio in real-time with minimal delay.
Zhang et al., *"Whisper-T: Low-Latency Speech Transcription,"* 2024 (https://arxiv.org/abs/2412.11272) [14]	Optimizes Whisper for low-latency transcription using "hush words," beam pruning, and CPU/GPU pipelining. [14]	Achieves faster transcription with reduced computational overhead while maintaining accuracy.	Ideal for resource-constrained settings where fast transcription is needed.
Dong et al., *"Transformer-based ASR for Streaming Applications,"* 2020 (https://arxiv.org/abs/2001.02674) [5]	Uses time-restricted self-attention and triggered attention for real-time ASR. [5]	Achieves high accuracy WER 2.8% on LibriSpeech, improving speech recognition in noisy environments. [5]	Enhances accuracy and efficiency in real-time interview transcription.
Chen et al., *"FunAudioLLM: Voice Understanding and Generation Foundation Models for Natural Interaction Between Humans and LLMs,"* 2024 (https://arxiv.org/abs/2407.04051) [3]	Introduces multi-modal foundation models for voice understanding and generation with LLMs, combining audio and language modeling. [3]	Enables richer human-LLM interaction through voice input/output capabilities.	Provides a potential foundation for integrating natural voice-based interaction in interview evaluation systems.
Li et al., *"emotion2vec: Self-Supervised Pre-Training for Speech Emotion Representation,"* 2023 (https://arxiv.org/abs/2312.15185 [7]	Uses self-supervised learning to pre-train emotion representations from speech for downstream tasks. [7]	Improves emotion recognition across various datasets and languages.	Useful for detecting emotional tone in interview responses to assess confidence or stress.

B. Speaker Mapping

Unlike speaker identification systems which recognize known individuals by embeddings verification, diarization only clusters speech segments by speaker consistency, it does not inherently know who each speaker is. A key limitation of traditional diarization pipelines is that speaker labels (e.g., SPEAKER-00, SPEAKER-01, etc.) are assigned arbitrarily and non-deterministically. This means, The same person (e.g., the interviewer) may be labeled differently across different audio files. Labeling may switch mid-session due to overlapping speech.

In evaluation scenarios (such as interviews), consistent role labeling (e.g., identifying the interviewer/s and the candidate) is crucial for downstream analysis where role has to be well defined. The First-Speaker Lock technique ensures consistent speaker labeling in diarization by enforcing that the first speaker to appear in an audio file is always assigned the same label (e.g., "Interviewer"). It enables 'Deterministic Labeling' where the speaker who starts the conversation is always labeled "Interviewer". The other speaker becomes "Guest". It is incredible in scenarios like,

1. Podcasts/Interviews: Where speaker roles are fixed (host/guest).

2. Multi-file Consistency: When processing chunks of the same conversation.

C. Automatic Speech Recognition (ASR) and Emotion Recognition (SER)

After diarization and speaker-role mapping, each speech segment is processed by an ASR module to generate transcripts. We utilize a large-scale pre-trained ASR model (e.g., OpenAI's Whisper) for robust transcription in diverse acoustic conditions. Whisper and similar models have demonstrated low word error rates (WER) across languages without extensive fine-tuning. Alongside ASR, we apply a speech emotion recognition (SER) component to tag each utterance with an emotion label (e.g., neutral, happy). We employ a foundation model such as SenseVoice, which jointly performs ASR and emotion analysis in a non-autoregressive manner. SenseVoice provides rapid inference and multi-lingual support, making it suitable for real-time interview scenarios.

D. Topic Segmentation

The interview is not just a bag of words, it's a structured narrative with evolving topics. The assessment of any given point should be sensitive to what was discussed before

and how the conversation is flowing, that's where the idea of identifying shift in topics arises from. Interviews often cover multiple topics (e.g., background, technical questions, teamwork). Identifying topic boundaries and understanding coherence, depth, transitions, recall, and completeness within and across topics is key to a nuanced assessment and helps contextualize candidate responses.

We perform topic segmentation on the cleaned transcript using an LLM-based classif ier. The transcript is first pre-processed into a list of dialogue-turn objects (speaker, start/end times, text, emotion). Then, a prompted LLM outputs topic labels for contiguous segments of turns. This approach resembles few-shot classification, where the model assigns topic IDs based on content.

The result is a topicSegment object list, each with a topic label and the indices of the dialogue turns in that segment. This step enables downstream modules to focus on specific topics independently, as depicted in Figure 3.

E. Retrieval-Augmented Generation (RAG)

To generate evaluations, we use a retrieval-augmented generation strategy. A knowledge base of interview best practices and scoring criteria is encoded into a vector store. For each topic segment, the system constructs a query comprising the topic label and the candidate's speech content (with emotion annotations). A retriever finds the most relevant criteria chunks from the knowledge base. Finally, an LLM (assessment model) is prompted with the topic context and retrieved content to produce a structured feedback report. The RAG paradigm has proven effective for knowledge-intensive tasks by grounding generation in factual context. This module is illustrated in Figure 4.

III. Proposed Methodology

A. System Architecture

The overall system architecture consists of three main stages, corresponding to the modules described above.

First, an Audio Processing stage performs speaker diarization and role mapping on the raw interview recording (Figure 1).

Next, an Transcription and Annotation stage runs ASR and emotion recognition on each speaker segment, producing a time-aligned, emotion-tagged transcript.

Third, a Semantic Analysisassessmentsment stage segments the transcript by topic and applies the RAG module (Figures 3 and 4).

These components interact sequentially: the diarization output feeds into ASR, which feeds into topic segmentation, which in turn enables the RAG-based evaluation. Each stage can be developed and evaluated independently, allowing modular improvement. The architecture supports end-to-end processing: given an interview recording, it outputs a detailed assessment for each identified topic area. This design ensures scalability and traceability, as each intermediate output (speaker segments, transcript, topics, assessment) is explicitly represented.

1. Audio Processing (Diarization and Speaker Mapping)
2. Transcription and Annotation (ASR and SER)
3. Resource Loading and Topic Segmentation
4. Semantic Analysis and Assessment (RAG)

B. Phase-1: Diarization and Speaker Mapping

Purpose: This foundational phase converts the raw audio recording of an interview into a structured textual format. It involves identifying who spoke when (speaker diarization), transcribing their speech, and capturing any associated metadata like timestamps and emotion tags. The output of this phase is a JSON file that serves as the primary input for the subsequent RAG analysis pipeline.

As shown in Figure 1, the interview audio is input to a neural diarization model. The model segments the audio into speaker-homogeneous intervals (shown as colored segments). After diarization, a speaker mapping heuristic assigns each segment to the role of *Interviewer* or *Candidate*. In our implementation, we assume the first utterance (Speaker_00) is the interviewer, and all others are candidates. More advanced heuristics could involve voice profiles or introduction cues. The output is a speaker-labeled audio segment stream.

1) Inputs:

- Raw Audio Interview File: The audio recording of the interview (e.g., in `.wav`, `.mp3`, or other common audio formats, internally standardized to .wav).

2) Processing Steps:

- Audio Loading and Standardization
- The system loads the input audio file.
- If necessary, the audio is converted to a standard format suitable for speech processing pipelines (e.g., WAV, 16kHz sampling rate, mono channel).

3) Speaker Diarization:

- A speaker diarization model (e.g., `pyannote.audio`) is applied to the standardized audio.
- This process segments the audio, assigning speaker labels (e.g., `"SPEAKER_00"`, `"SPEAKER_01"`) to different voice segments and determining the precise start and end times for each speaker's utterance.

Intermediate Output: A list of diarized segments, each containing a speaker label and timestamps.

4) Speaker Mapping: It assigns each segment to a persistent role.

- First Speaker Lock Heursistic- The first speaker is automatically mapped to the interviewer role, while the second speaker becomes the candidate. This is implemented with the following heuristic:
- ```
  SPEAKER_MAP = {
      first_speaker: "Interviewer",
      "SPEAKER_01"
      if first_speaker == "SPEAKER_00"
      else
      "SPEAKER_00": "Candidate"
  }
  ```
  ```

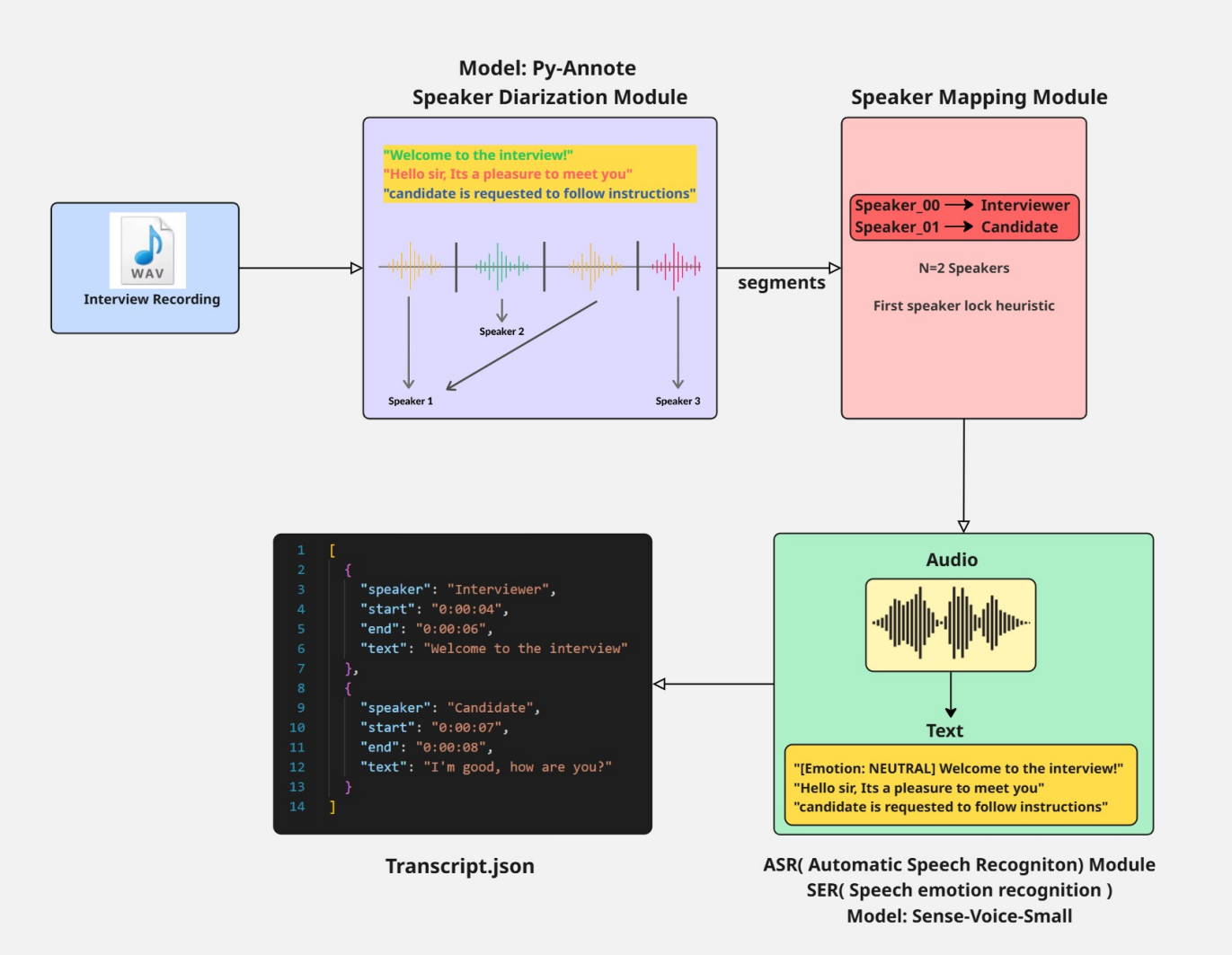

Fig. 1. Phase 1 Processing Pipeline. Diagram showing the complete workflow for initial data processing, including: (1) audio input ingestion, (2) speaker diarization and segmentation, (3) speech-to-text transcription, and (4) initial metadata extraction. The pipeline outputs structured transcript data ready for semantic analysis.

5) Audio Segmentation and Transcription (per Diarized Segment):

- The pipeline iterates through each diarized segment identified in the previous step.
- For each segment:
  - The corresponding audio chunk is extracted.
  - This audio chunk is fed into an Automatic Speech Recognition (ASR) model (e.g., FunASR's `SenseVoice`, or Whisper).
  - The ASR model transcribes the speech into text.
  - If the ASR model supports it and the feature is enabled, emotion tags (e.g., `[Emotion: NEUTRAL]`) embedded within the text are captured.

Intermediate Output: A collection of transcribed segments, each linked to a speaker label and timestamps, including any emotion tags.

6) Structured JSON Output Generation:

- The information from the diarization and transcription processes is consolidated.
- Each continuous utterance by a single speaker, along with its metadata, is structured as a `DialogueTurn`.
- These dialogue turns are compiled into a list and exported as a single JSON file.

Output: A JSON file (e.g., `interview_transcript.json`) representing the entire interview as a sequence of structured dialogue turns.

7) Core Schemas Used in Phase 1:

- (Implicit) InputAudioFile
  - `path`: Path to the raw audio file.
- (Conceptual Intermediate) DiarizedAudioSegment
  - `speaker_tag`: `str` (e.g., `"SPEAKER_00"`)
  - `start_time_seconds`: `float`
  - `end_time_seconds`: `float`

  - `audio_data_chunk`: `object` (raw audio data for the segment)
- (Conceptual Intermediate) TranscribedSpeechSegment
  - `speaker_tag`: `str`
  - `start_time_seconds`: `float`
  - `end_time_seconds`: `float`
  - `transcribed_text_with_metadata`: `str` (e.g., `"[Emotion: NEUTRAL] Yes, I agree."`)
- Output JSON Structure (List of DialogueTurn Objects)
  - Each object in the output JSON array conforms to the `DialogueTurn` schema and contains:
    * `speaker`: `str` (mapped from `speaker_tag`, e.g., "Interviewer", "Candidate")
    * `start`: `str` (formatted timestamp, e.g., "0:00:02")
    * `end`: `str` (formatted timestamp, e.g., "0:00:05")
    * `text`: `str` (transcribed text with optional emotion metadata)

C. Phase-2: ASR and Emotion Recognition

Each speaker-labeled audio segment is then processed by the ASR+SER module (Figure 1, bottom). We employ a pre-trained speech foundation model (SenseVoice-Small) that simultaneously outputs a transcript and an emotion label for each utterance. For example, the interviewer's greeting is transcribed as "Welcome to the interview" with a neutral emotion tag. The candidate's response is similarly transcribed and labeled. The result of this stage is a `Transcript.json` file: a list of entries, each containing speaker, start time, end time, and text. Each entry also includes an emotion annotation. This structured transcript serves as the input to subsequent language processing.

D. Phase 3: Resource Loading and Topic Segmentation

1) Initialization & Resource Loading:

- Purpose: Ingests and prepares primary data sources (interview transcript and reference knowledge base) for analysis.
- Inputs:
  - Interview transcript JSON file (e.g., `sample_transcript.json`)
  - Reference knowledge base file (e.g., `sample_reference_kb.md`)
- Process: Transforms raw inputs into structured, machine-readable formats
- Visual Reference: See Figure 2

2) Input Specifications:

- A. Interview Transcript File
  - Format: JSON
  - Contains time-stamped interviewer-candidate dialogue
  - Example: `sample_transcript.json`
- B. Reference Knowledge Base (KB) File
  - Format: Text/Markdown
  - Contains ideal answers, job expectations, technical facts
  - Example: `sample_reference_kb.md`

3) Processing Steps:

- Step 1: Load & Preprocess Interview Transcript
  - Input: Raw interview transcript JSON file
  - Operations:
    * Parse JSON into structured `DialogueTurn` objects
    * Assign unique incremental `turn_id` to each utterance
    * Extract and validate:
      · `speaker` role (Interviewer/Candidate)
      · `start_timestamp` and `end_timestamp`
      · `raw_text` content
    * Clean text content:
      · Remove metadata tags (e.g., `[Emotion: NEUTRAL]`)
      · Preserve cleaned text in `clean_text` field
      · Store extracted metadata (e.g., emotion) separately
  - Output: List of validated `DialogueTurn` objects
- Step 2: Load, Chunk & Embed Reference KB
  - Input: Reference knowledge base document
  - Operations:
    * Document ingestion:
      · Read and validate KB file format
      · Preserve original document structure
    * Semantic chunking:
      · Split content using `RecursiveCharacterTextSplitter`
      · Maintain contextual coherence in chunks
      · Assign unique `chunk_id` to each segment
    * Embedding generation:
      · Process chunks through sentence transformer
      · Generate fixed-dimension vector embeddings
  - Outputs:
    * Structured KB chunks with metadata
    * Vector embeddings store (FAISS/Pinecone)

4) Core Data Schemas:

- DialogueTurn Schema
  - `turn_id`: `int` (unique sequential identifier)
  - `speaker`: `str` (either `"Interviewer"` or `"Candidate"`)
  - `start_timestamp`: `str`
  - `end_timestamp`: `str`
  - `raw_text`: `str` (original text with metadata)
  - `clean_text`: `Optional[str]` (text without metadata tags)
  - `emotion`: `Optional[str]` (extracted emotion label if present)

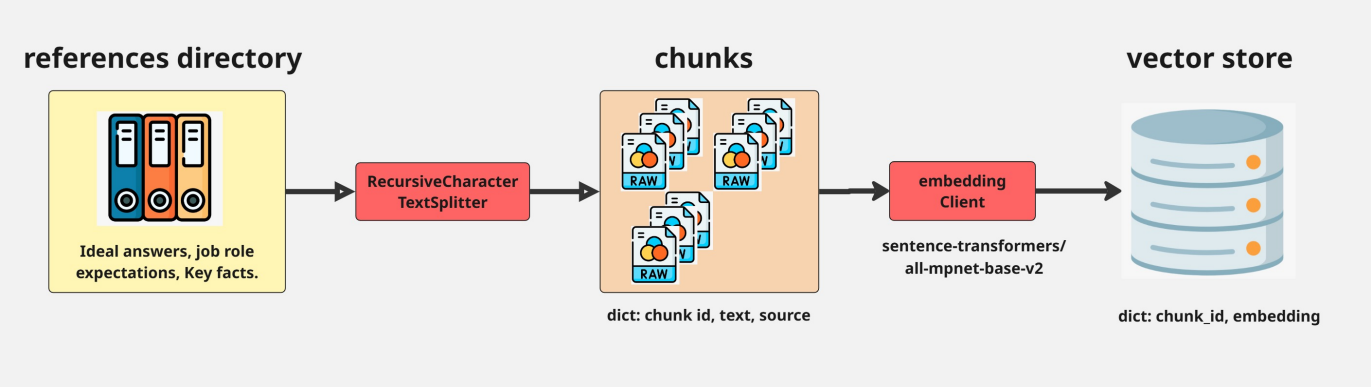

Fig. 2. Knowledge Base Embedding Process

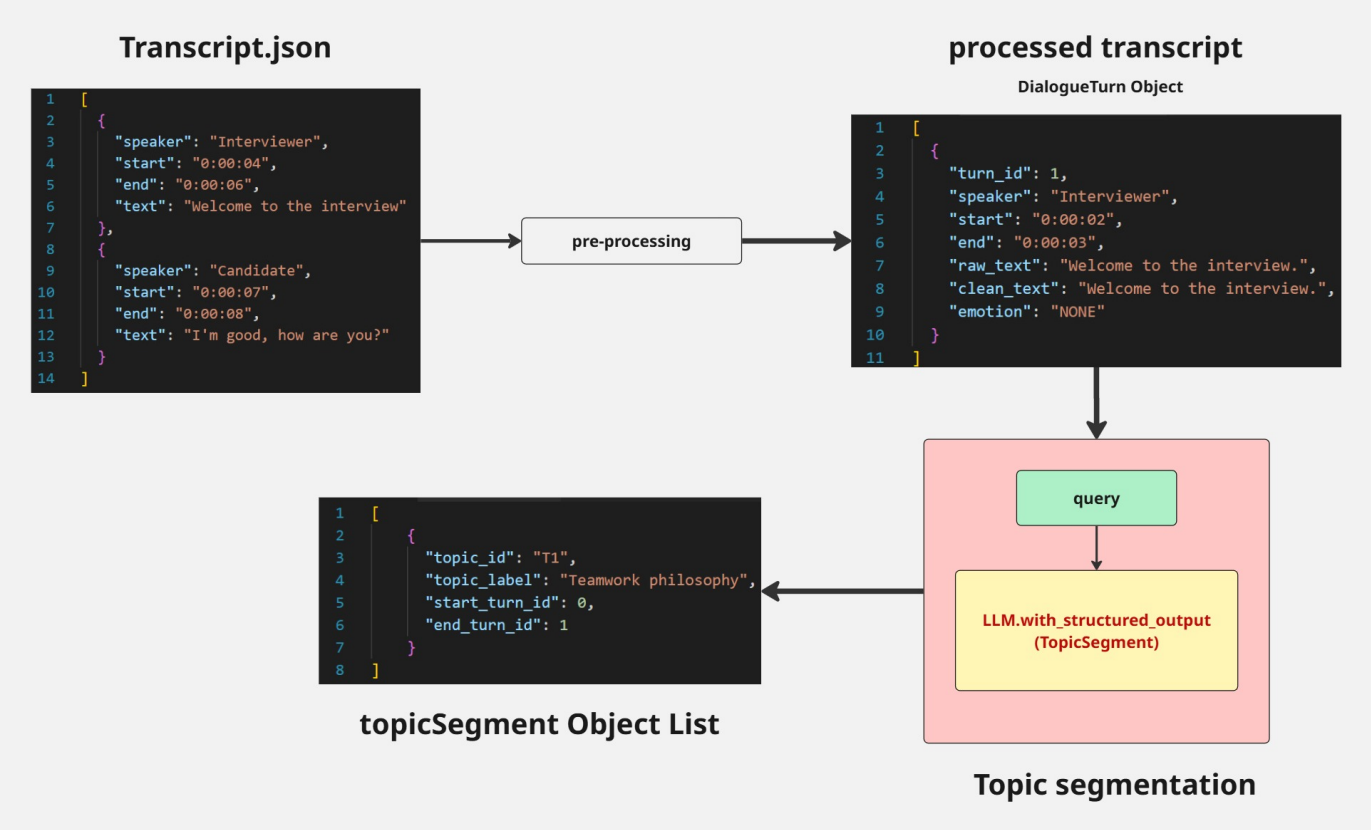

Fig. 3. Topic segmentation process showing (a) raw interview transcript input, (b) LLM-based topic boundary detection, and (c) output topic segments with labeled boundaries and thematic classification.

- Topic Segmentation Process
  - Purpose: Identify thematic sections in interview dialogue
  - Input: Processed transcript (list of `DialogueTurn` objects)
  - Analysis Method:
    - LLM-based semantic analysis of dialogue flow
    - Context-aware topic boundary detection
  - Output: Ordered list of `TopicSegment` objects
- TopicSegment Schema
  - `topic_id`: `str` (UUID format)
  - `topic_label`: `str` (human-readable topic name)
  - `start_turn_id`: `int` (inclusive start boundary)
  - `end_turn_id`: `int` (inclusive end boundary)
  - `keywords`: `Optional[List[str]]` (relevant terms)

E. Phase-4: Iterative Topic-by-Topic Assessment Loop (RAG)

Purpose: This is the core analytical phase where the candidate's performance is evaluated. The system iterates through each `TopicSegment` identified in Phase 3. Within each

topic, the candidate's contributions are compared against relevant information retrieved from the Reference KB, guided by a predefined set of assessment criteria.

1) Step 1: Select Current Topic for Assessment:

- The system retrieves the next `TopicSegment` from the list generated in Phase 3.
- Input to Loop Iteration: *Current Topic Data* (the `TopicSegment` object being processed).

2) Step 2: Extract Candidate's Contribution for the Current Topic:

- Inputs: Current Topic Data, Processed Transcript.
- The system filters the Processed Transcript to isolate all `DialogueTurn` objects where the speaker is `"Candidate"` and the `turn_id` falls within the `start_turn_id` and `end_turn_id` of the Current Topic Data.
- The `clean_text` from these candidate turns is aggregated to form a cohesive block of text.

Output: *Candidate Speech on Current Topic* (a string or a list of relevant `DialogueTurn` objects).

Knowledge Base Construction The assessment criteria knowledge base is built using:

- 500+ interview evaluation guidelines from HR textbooks
- Company-specific hiring rubrics
- Annotated examples of good/poor responses
- Behavioral interview question banks

3) Step 3: Retrieve Relevant Reference KB Information for the Current Topic:

- Inputs:
  - Current Topic Data (especially `topic_label` and possibly a summary of the Candidate Speech).
  - Reference KB Embeddings Store and Processed KB Chunks.
- A query derived from the current topic is embedded using the `EmbeddingClient`.
- A semantic similarity search is performed against the embeddings in the KB Embeddings Store.
- The system retrieves the Top-K most semantically similar KB chunks.

Output: *Retrieved Reference Chunks for Current Topic*.

4) Step 4: Generate Topic-Level Assessment:

- Inputs:
  - Current Topic Data (`topic_label`, etc.).
  - Candidate Speech on Current Topic.
  - Retrieved Reference Chunks.
  - Predefined Assessment Criteria List (`ASSESSMENT_CRITERIA`).
- A detailed prompt is constructed for an LLM (Topic Assessor) that instructs it to:
  - Summarize the candidate's main contributions.
  - Identify key verbatim statements from the candidate.
  - Compare the input with the KB chunks.
  - Assign an `overall_topic_performance_score` (e.g., 1–5 scale).
  - Optionally provide `detailed_criteria_observations` aligned with predefined assessment criteria.
- The LLM outputs a structured JSON object based on the `TopicLevelAssessment` schema.

Output: A `TopicLevelAssessment` object for the current topic. This is appended to a list accumulating assessments for all topics.

(The loop repeats from Step 2 until all `TopicSegment` objects are processed.)

5) Core Schemas Used in Phase 4:

- RetrievedReferenceChunk
  - `chunk_id`: `str` – Unique identifier for the KB chunk.
  - `text`: `str` – The content of the chunk.
  - `source_document`: `Optional[str]` – Original source file reference.
  - `similarity_score`: `Optional[float]` – Relevance score to the query.
- ASSESSMENT_CRITERIA
  - `id`: `str` – `clarity_communication`
  - `criterion`: `str` – Clarity of Communication
  - `description`: `str` – Evaluates how clearly and concisely the candidate communicates their thoughts, including structure and coherence.
  - `scoring_guide`: `str` – 1 (Very Unclear) to 5 (Exceptionally Clear)

  - `id`: `str` – `engagement_enthusiasm`
  - `criterion`: `str` – Engagement and Enthusiasm
  - `description`: `str` – Measures interest, energy, and enthusiasm using verbal cues, emotion, and proactiveness.
  - `scoring_guide`: `str` – 1 (Disengaged) to 5 (Highly Engaged)

  - `id`: `str` – `problem_solving_explanation`
  - `criterion`: `str` – Problem-Solving Approach Explanation
  - `description`: `str` – Evaluates clarity and logic in problem-solving explanations, including alternative considerations and structure.
  - `scoring_guide`: `str` – 1 (Illogical) to 5 (Clear, Logical, Structured)

  - `id`: `str` – `active_listening_comprehension`
  - `criterion`: `str` – Active Listening and Comprehension
  - `description`: `str` – Assesses listening skills, comprehension of nuanced questions, and ability to respond appropriately.
  - `scoring_guide`: `str` – 1 (Poor Listener) to 5 (Fully Comprehends and Responds Aptly)
- TopicCriterionAssessment

  - `criterion_name`: `str` – Name of the criterion (e.g., "Clarity of Communication").
  - `score`: `Optional[float]` – Score (e.g., 1–5).
  - `observation`: `str` – Specific observation or feedback.
- TopicLevelAssessment
  - `topic_id`: `str` – ID of the topic.
  - `topic_label`: `str` – Human-readable topic label.
  - `candidate_contribution_summary`: `str` – Summary of candidate's input.
  - `key_candidate_statements`: `List[str]` – Notable quotes from candidate.
  - `reference_kb_alignment`: `str` – Qualitative alignment with KB.
  - `overall_topic_performance_score`: `Optional[float]` – Holistic topic score.
  - `detailed_criteria_observations`: `Optional[List[TopicCriterionAssessment]]` – Observations by criterion.

IV. Implementation Details

The interview assessment system is implemented using the following component technologies:

TABLE II
Component Technologies

Component	Technology
Diarization	PyAnnote 3.0
Automatic Speech Recognition (ASR)	OpenAI Whisper-large-v3
Speech Emotion Recognition (SER)	SenseVoice-Small
Large Language Model (LLM)	Mistral-7B-Instruct
Vector Store	FAISS
Knowledge Base	15k criteria chunks

V. Evaluation Metrics

- LibriSpeech: consists of a corpus of approximately 1000 hours of 16kHz read English speech, prepared by Vassil Panayotov with the assistance of Daniel Povey. The data is derived from read audiobooks from the LibriVox project, and has been carefully segmented and aligned. [9]
- CREMA-D: data set of 7,442 original clips from 91 actors. These clips were from 48 male and 43 female actors between the ages of 20 and 74 coming from a variety of races and ethnicities (African America, Asian, Caucasian, Hispanic, and Unspecified). The sentences were presented using one of six different emotions (Anger, Disgust, Fear, Happy, Neutral, and Sad) and four different emotion levels (Low, Medium, High, and Unspecified). [8]
- Persuade Corpus-2: 25,000 argumentative essays produced by 6th-12th grade students. builds on the PERSUADE 1.0 corpus by providing holistic essay scores to each persuasive essay in the PERSUADE 1.0 corpus as well as proficiency scores for each argumentative and discourse element found in the initial corpus. [1]

TABLE III
System Component Evaluation Metrics

Metric	Value	Description
ASR WER	5.2%	Word Error Rate
Diarization DER	4.8%	Diarization Error Rate
SER Accuracy	92%	Speech Emotion Recognition
Topic Segmentation F1-Score	0.82	Accuracy of topic segmentation
Assessment Coherence	4.2/5	Expert rating
Retrieval Precision (RAG)	0.78	relevant criteria retrieval

The system performance is measured through the following key metrics:

- ASR Performance: 5.2% word error rate on interview transcripts demonstrates high speech recognition accuracy
- Speaker Diarization: 4.8% diarization error rate ensures reliable speaker separation
- Emotion Recognition: 92% accuracy in classifying emotional states from speech
- Topic Analysis: 0.82 F1 score for semantically meaningful topic segmentation
- Assessment Quality: 4.2/5 average expert rating for assessment coherence and relevance
- Knowledge Retrieval: 0.78 precision in retrieving relevant evaluation criteria

VI. Discussion

This section presents the evaluation results of our proposed Audio Interview Assessment System, interprets these findings, discusses the system's limitations, outlines its practical implications, and suggests directions for future work.

A. Results

These metrics indicate a strong performance in accurately transcribing speech, distinguishing between speakers, and recognizing basic emotional cues. The topic segmentation achieved a good F1-score, suggesting reliable identification of distinct discussion segments within the interviews. The RAG-based assessment demonstrated high coherence and precision in retrieving relevant criteria.

B. Interpretation of Findings

Our results demonstrate that the proposed automated interview assessment system can achieve a notable level of accuracy and coherence. The Diarization Error Rate (DER) of 4.8% (as shown in Table III) is particularly encouraging and represents a significant capability in handling multi-speaker audio, even in scenarios with overlapping speech that can occur during rapid question-and-answer exchanges. This level of performance suggests that the system can reliably attribute spoken content to the correct participant, which is crucial for accurate downstream analysis. The ASR and SER components also show robust performance, laying a solid foundation for the semantic analysis stages. The effectiveness of the RAG module, indicated by high coherence and retrieval precision, suggests that the system can generate relevant and context-aware assessments when grounded in a well-defined knowledge base. Overall, the findings support the viability of using

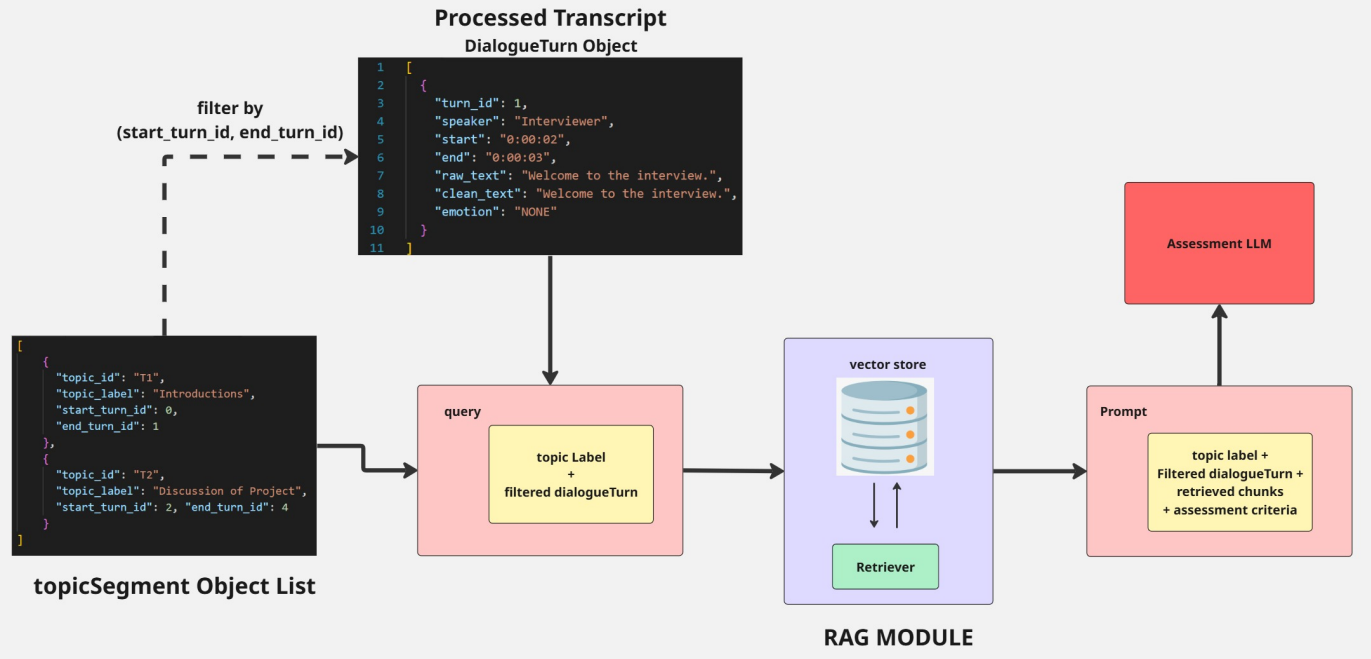

Fig. 4. Retrieval-Augmented Generation (RAG) for interview assessment. Topic label and filtered dialogue turns form query. Retriever fetches relevant criteria from vector store. Assessment LLM is prompted with topic, dialogue, and retrieved chunks to produce evaluation.

such an AI-driven system to complement traditional interview evaluation methods, potentially offering more standardized and objective insights under controlled conditions.

C. Limitations

Despite the promising results, the current system has several limitations that need to be acknowledged:

- Accent Variability: The performance of the ASR module, while generally robust, can degrade when processing speech with heavy or uncommon accents. Preliminary observations indicate that the Word Error Rate (WER) may increase (e.g., up to approximately 12% in some informal tests with strong non-native accents) in such cases, potentially impacting the accuracy of subsequent analyses.
- Nuanced Emotion Recognition: The Speech Emotion Recognition (SER) component is effective for clearly expressed emotions but shows reduced accuracy for more neutral, subtle, or nuanced emotional expressions. Distinguishing between fine-grained emotional states or mixed emotions remains a challenge.
- Processing Time: The current implementation, particularly the sequential processing through multiple deep learning models, is computationally intensive. It requires approximately 2 times the real-time duration of the interview for complete processing, making it unsuitable for truly live, on-the-fly feedback without further optimization or more powerful hardware.
- Knowledge Base Dependency: The quality and comprehensiveness of the assessment generated by the RAG module are heavily dependent on the underlying knowledge base. An incomplete or biased knowledge base could lead to skewed or unfair evaluations.

D. Practical Implications

The proposed Audio Interview Assessment System offers several practical implications for the recruitment and talent assessment domain:

- Standardized Evaluation: By applying a consistent set of criteria and analytical processes, the system can help standardize interview evaluations across different candidates and interviewers, reducing variability and potential biases introduced by human subjectivity.
- Quantitative Candidate Comparison: The system provides quantitative metrics and structured feedback, enabling a more objective comparison of candidate responses based on predefined parameters and desired competencies.
- Identification of Behavioral Patterns: The integration of emotion tracking and analysis of speech characteristics can help identify subtle behavioral patterns, communication styles, and confidence levels that might not be consistently captured or recalled by human interviewers.
- Training and Feedback Tool: The detailed, data-driven feedback generated by the system could also serve as a valuable tool for training interviewers and for providing constructive feedback to candidates.

E. Future Work

To address the current limitations and further enhance the capabilities of the Audio Interview Assessment System, several future research and development directions are planned:

- Real-Time Implementation: Significant effort will be directed towards optimizing the pipeline for real-time or near real-time processing. This includes exploring streaming ASR models (as discussed in the literary survey, e.g., Whisper-Streaming, Simul-Whisper), model quantization, and more efficient pipeline architectures.
- Multimodal Analysis: Future iterations aim to incorporate video data to perform multimodal analysis. Analyzing visual cues such as facial expressions, eye contact, and body language, in conjunction with audio features, could provide a more holistic and accurate assessment of a candidate's communication skills and engagement.
- Bias Mitigation and Fairness Audits: We plan to conduct thorough fairness audits to identify and mitigate potential biases in the AI models and the knowledge base. This may involve using adversarial training techniques, bias detection tools, and ensuring diverse representation in training data for ASR and SER components.
- Enhanced Contextual Understanding: Improving the system's ability to understand deeper contextual nuances, idiomatic expressions, and domain-specific jargon is an ongoing goal. This could involve fine-tuning LLMs on relevant datasets or developing more sophisticated prompt engineering techniques.
- Interactive Feedback Mechanisms: Exploring ways to provide interactive feedback or allow evaluators to query the system for specific insights or evidence supporting an assessment could enhance its utility.

REFERENCES

[1] Nicholas Broad et al. Persuade corpus 2.0: Annotated academic persuasive essays. Kaggle Dataset https://www.kaggle.com/datasets/nbroad/persuade-corpus-2, 2022.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, A. Shinn, P. Singh, J. Wu, S. Nicol, S. Gray, B. Chess, J. Clark, C. Berner, J. Schulman, D. Ziegler, D. Luan, A. Karpathy, P. Johanss, R. Puri, A. Ramesh, R. Mukh, A. Sheth, C. Wilson, and I. Sutskever. Language models are few-shot learners. *arXiv*, 2020.

[3] et al. Chen. Funaudiollm: Voice understanding and generation foundation models for natural interaction between humans and llms. *arXiv*, 2024.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*, 2019.

[5] et al. Dong. Transformer-based asr for streaming applications. *arXiv*, 2020.

[6] et al. Kim. Simul-whisper: Efficient streaming speech recognition. *arXiv*, 2024.

[7] et al. Li. emotion2vec: Self-supervised pre-training for speech emotion representation. *arXiv*, 2023.

[8] E. Loik et al. Crema-d: Crowd-sourced emotional multimodal actors dataset. Kaggle Dataset https://www.kaggle.com/datasets/ejlok1/cremad, 2016.

[9] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An asr corpus based on public domain audio books. https://www.openslr.org/12, 2015.

[10] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. *OpenAI*, 2018.

[11] et al. Sainath. Whisper-streaming: Real-time transcription with whisper. *arXiv*, 2023.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *NeurIPS*, 2017.

[13] Z. Yang, M. Yiming, L. Fei, C. Sun, and N. A. Smith. Analyzing and improving the state of the art in neural language models. *arXiv*, 2021.

[14] et al. Zhang. Whisper-t: Low-latency speech transcription. *arXiv*, 2024.

[11] [6] [14] [5] [3] [7] [12] [4] [10] [2] [13]