© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715449

AI-Powered Automated Behavioral Mock Interview System with Star-Based Evaluation and Leadership Scoring

KAYALVIZHI.S[1], DR.K. PONMOZHI[2]
[1] *PG Student, Department of Computer Application, SRM Valliammai Engineering College, Kattankulathur, Chennai, Tamil Nadu, India*
[2] *Associate Professor, Department of Computer Applications, SRM Valliammai Engineering college, Kattankulathur. Chennai, Tamil Nadu, India*

Abstract- This paper presents an AI-powered Interview Bot, a real-time conversational platform designed to automate and enhance the interview process using advanced speech recognition and language models. Traditional interview systems rely heavily on manual evaluation and lack scalability and consistency in candidate assessment. The proposed system integrates WebRTC-based audio capture, WebSocket-based real-time streaming, and AI-driven evaluation models to conduct dynamic and interactive interviews. Audio inputs from candidates are processed using Deepgram for speech-to-text conversion, while Groq-powered large language models analyze responses based on structured evaluation metrics such as scoring, leadership assessment, and STAR-based feedback. The system also incorporates resume parsing using PDF processing techniques to provide contextual questioning. Real-time feedback is continuously updated on the dashboard, enabling immediate performance insights. By combining real-time communication, AI evaluation, and scalable architecture, the system provides an efficient, automated, and intelligent solution for modern recruitment processes.

Index Terms- AI Interview Bot; Speech-to-Text; WebRTC; WebSocket; Deepgram; Groq LLM; Resume Parsing; Real-Time Processing; Automated Evaluation; Candidate Scoring; NLP; Conversational AI

I. INTRODUCTION

Recruitment processes in many organizations still depend on manual interviews, which are time-consuming, inconsistent, and difficult to scale. Evaluating candidates objectively becomes challenging due to human bias and limited monitoring capabilities. Additionally, traditional systems lack real-time feedback and structured performance analysis, reducing the efficiency of the hiring process.

To address these challenges, an automated interview system that can conduct real-time conversations, analyze candidate responses, and provide structured evaluation is essential. The integration of speech recognition and artificial intelligence enables dynamic interaction and objective assessment without human intervention.

The proposed AI-powered Interview Bot adopts a real-time processing approach using WebRTC for audio capture and WebSocket-based communication for continuous data streaming. Speech-to-text conversion is performed using Deepgram, while Groq-based language models evaluate candidate responses based on predefined metrics. This system enables scalable, efficient, and intelligent interview automation.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the system architecture. Section IV explains the methodology. Section V presents the database design. Section VI discusses real-time processing. Section VII presents results and discussion. Section VIII concludes with future enhancements.

II. LITERATURE REVIEW

A. Speech Recognition in Customer Service Systems

Modern speech recognition systems have evolved significantly with the adoption of deep learning models. Cloud-based APIs such as Deepgram provide real-time transcription with high accuracy and low

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715449

latency, enabling efficient processing of streaming audio data.

B. Sentiment Analysis and Natural Language Processing

Natural Language Processing techniques are widely used for analyzing textual responses. Advanced transformer-based models enable context-aware evaluation of candidate answers, improving the accuracy of automated assessments.

C. Real-Time Communication Systems

WebRTC and WebSocket technologies enable low-latency communication between client and server systems. These technologies are essential for building real-time applications such as conversational AI systems.

D. AI-Based Evaluation Systems

Large Language Models (LLMs) are increasingly used for automated evaluation of textual inputs. These models can generate structured feedback and scoring metrics, improving consistency in candidate assessment.

E. Cloud-Based Processing and Scalable Architectures

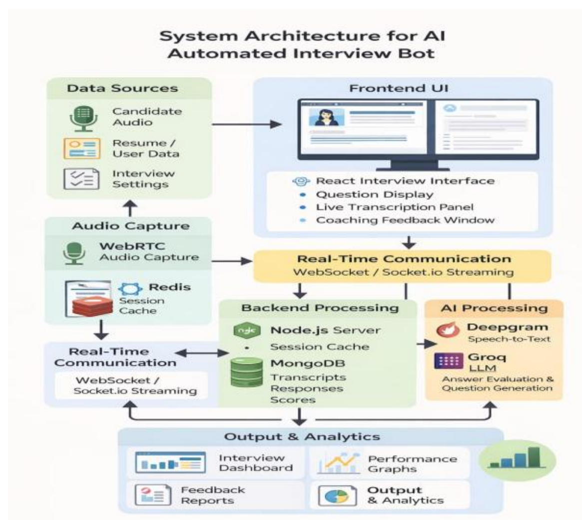
Cloud computing platforms have become widely adopted for processing and storing large volumes of interaction data. These platforms provide real-time data processing, scalability, and cost efficiency. The use of cloud-based architectures allows systems to handle multiple concurrent analyses and ensures seamless integration with existing customer service platforms, making them suitable for modern call center environments.

III. SYSTEM ARCHITECTURE

The proposed system is designed as a modular and scalable architecture that enables efficient processing and analysis of customer care interactions using artificial intelligence techniques. The system operates as a web-based application, where the presentation layer provides an interactive user interface for uploading and analyzing customer service calls. The processing layer handles speech recognition, natural language processing, and voice emotion detection, while the data layer manages storage and retrieval of call records and analysis results. External tools and libraries are integrated for audio processing, visualization, and report generation.

The architecture is divided into three main tiers: (1) Client Tier — a user interface that allows supervisors and quality assurance teams to upload audio recordings, view transcripts, and analyze results through dashboards; (2) Processing Tier — responsible for converting speech to text, performing sentiment analysis, and detecting emotional tone from audio signals using machine learning models; (3) Data Tier — a centralized database that stores audio files, transcripts, sentiment scores, emotion labels, and performance reports for efficient retrieval and analysis.

The system achieves efficient processing by combining textual and acoustic analysis in a unified workflow.



A. Role-Based Access Control

The system defines multiple user roles such as admin and candidate to ensure controlled access to system functionalities. each user role is stored in the database and validated during authentication using secure token-based mechanisms. role-based access control is implemented at both the application and database levels, ensuring that users can only access features and data permitted to their assigned role. this approach prevents unauthorized access and maintains system security. the frontend enforces access

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715449

restrictions through route protection and dashboard-level controls, while backend validation ensures that sensitive operations and data access are securely managed.

Role	Key Capabilities
Admin	System configuration, interview management, analytics dashboard, user monitoring
Candidate	Participate in interview, respond to questions, view feedback and scores

B. Frontend-Architecture

The frontend of the system is developed as a browser-based interface that enables candidates to participate in interviews and interact with the system in real time. It captures audio input using WebRTC APIs and displays dynamically generated interview questions, live transcripts, and evaluation results through interactive dashboards. Efficient rendering techniques are used to handle continuous updates without affecting performance. The interface is designed to be user-friendly, with clear navigation and structured layouts for displaying scores, feedback metrics, and interview progress. Visualization components are optimized to support real-time data updates and ensure a smooth user experience during the interview session.

C. Backend-Architecture

The backend is responsible for handling real-time audio streaming, processing, and evaluation using artificial intelligence models. It manages speech-to-text conversion using Deepgram and performs response analysis using Groq-based large language models. The backend also handles authentication, session management, and communication between different system components through WebSocket connections. Data such as transcripts, scores, and feedback are stored in the database for further analysis. Secure access mechanisms ensure that only authorized users can perform specific operations. Backend services are optimized to process streaming data efficiently while maintaining low latency and high reliability.

D. Notification-Layer

The system includes a real-time feedback mechanism that updates users with analysis results and interview progress. Feedback is generated dynamically after each response and displayed instantly on the dashboard. Notifications may include performance scores, improvement suggestions, and evaluation summaries. This continuous feedback loop helps candidates understand their performance and make improvements during the interview. The system ensures timely delivery of updates without requiring manual refresh, enabling a seamless and interactive interview experience.

IV. IMPLEMENTATION METHODOLOGY

The system is implemented as a browser-native web application using a modern technology stack designed for real-time communication and intelligent processing. This approach eliminates the need for specialized hardware, reducing deployment cost and enabling accessibility across standard devices. The system operates using browser-based audio capture and leverages cloud-based technologies for speech recognition, natural language processing, and AI-driven evaluation. By integrating real-time streaming and scalable backend services, the system ensures efficient interview processing and continuous feedback generation.

Layer	Technology	Purpose
Frontend	React, HTML, CSS, JavaScript	Web interface, interview dashboard, user interaction
Backend	Node.js / Express	Real-time processing, API handling, system control
AI Models	Deepgram, Groq LLM	Speech-to-text, response evaluation, NLP processing
Communication	WebRTC, Socket.io	Real-time audio streaming and data transfer
Database	MongoDB	Storage of transcripts, sessions, and evaluation results
Visualization	Dashboard	Display of scores,

Layer	Technology	Purpose
	Components / Charts	feedback, and analytics
Resume Parser	Multer, PDF-Parse	Resume upload and content extraction
Caching	Redis	Performance optimization and fast data retrieval

A. Audio-Input-Preprocessing

The system interface allows candidates to participate in interviews through a browser-based application that captures real-time audio using webrtc apis. each audio input is streamed continuously and associated with metadata such as timestamp and session identification. the input is validated for quality and format before processing. basic preprocessing steps such as noise handling and normalization are applied to ensure clear audio signals. the processed audio is then forwarded to the streaming and transcription modules for further analysis.

B. Transcription-Processing-Pipeline

After audio capture, the system streams the audio data to the backend through WebSocket communication. The audio is processed using Deepgram speech recognition models, which convert spoken responses into real-time textual transcripts. The transcription process is optimized to handle variations in speech patterns, accents, and speaking speeds. The generated text is stored in the database and used for further evaluation. This pipeline ensures accurate and continuous conversion of audio input into structured text format.

C. Textual-Sentiment-Evaluation

The transcribed text is analyzed using natural language processing techniques powered by groq-based large language models. the system evaluates responses based on structured criteria such as relevance, clarity, and completeness using frameworks like the star method. it generates performance metrics including scores, feedback, and communication quality indicators. these results help assess the candidate’s response effectiveness and overall interview performance.

D. Acoustic-Emotion-Detection

In parallel with textual evaluation, the system generates detailed feedback using ai models. it analyzes speech characteristics such as response length, fluency, and filler word usage to provide additional insights. the final results include structured feedback, performance scores, and improvement suggestions. these outputs are continuously updated on the dashboard, providing candidates with real-time insights and enabling a comprehensive evaluation of interview performance.

V. DATABASE DESIGN

The database schema is implemented using a structured data model designed to efficiently support the operational workflow of the interview system. It consists of multiple interconnected collections that maintain relationships between users, interview sessions, transcripts, and evaluation results. These relationships ensure data integrity and enforce system logic at the database level, reducing reliance on application-layer validations. Each collection stores specific information such as candidate details, audio transcripts, evaluation metrics, and feedback results, ensuring organized and efficient data management. Proper referencing between collections maintains consistency and enables seamless data flow across different modules of the system.

TABLE III Database Schema Summary

Table	Key Columns
users	id (PK), name, email, role, created_at
sessions	id (PK), user_id (FK), start_time, end_time, status
resumes	id (PK), user_id (FK), file_path, parsed_data
transcripts	id (PK), session_id (FK), transcript_text, timestamp
evaluation_results	id (PK), session_id (FK), score, feedback, rating
metrics	id (PK), session_id (FK), speech_rate, filler_words, confidence_score

Foreign key relationships enforce referential integrity throughout the system by linking related entities at the data level. Sessions reference users to associate each interview with a specific candidate, while transcripts are linked to their corresponding sessions.

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715449

Evaluation results and metrics reference sessions to maintain consistency across all stages of the interview process. Resume data is also connected to users to provide contextual information during interviews. These relationships ensure structured data flow and prevent inconsistencies across modules. The

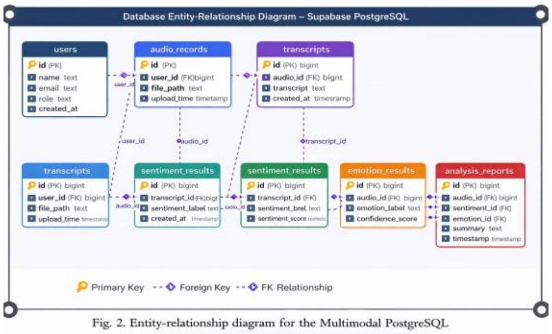


Fig. 2. Entity-relationship diagram for the Multimodal PostgreSQL

system supports real-time updates through WebSocket communication, where newly generated transcripts and evaluation results are immediately propagated to the client dashboard, enabling continuous feedback without delays.

VI. REAL-TIME INTERVIEW PROCESSING ENGINE

Live interview environments may not always be available in academic or testing scenarios, yet the interview system must remain fully functional without such constraints. the system addresses this limitation by supporting both real-time audio capture and controlled evaluation workflows that replicate complete interview interactions. this approach enables full testing of the system's capabilities using standard computing environments without requiring physical interview setups or human interviewers.

A. Input-Mode

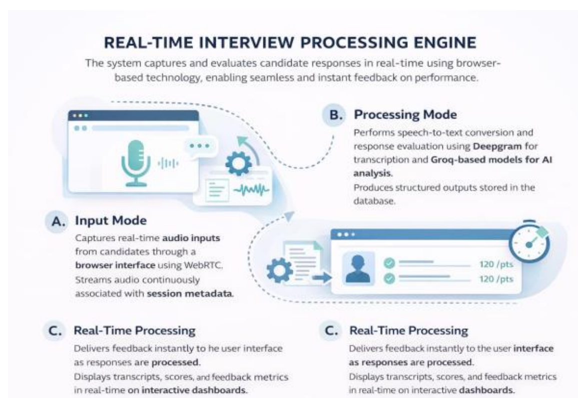
The system captures candidate responses in real time through a browser-based interface using WebRTC APIs. Audio is streamed continuously from the user's microphone and associated with session metadata such as timestamp and user identification. This mode enables realistic interview interaction where candidates respond to dynamically generated questions. The continuous streaming ensures seamless communication between client and backend without interruptions.

B. Processing-Mode

In processing mode, the system performs real-time speech-to-text transcription and response evaluation using integrated AI models. The audio stream is converted into text using Deepgram, while Groq-based models analyze the response for scoring, feedback, and structured evaluation metrics. Both processes operate simultaneously and generate structured outputs that are stored in the database. This ensures accurate and efficient analysis of candidate responses during the interview session

C. Real-Time-Processing

The system enables near real-time analysis by processing uploaded audio immediately after submission. the results, including transcripts, sentiment scores, and detected emotions, are delivered to the user interface without delay. data is dynamically updated and displayed in dashboards, allowing users to view analysis results instantly. the system ensures proper handling of data flow and prevents duplication by managing processing states efficiently within the application.



VII. AUDIO PROCESSING AND VALIDATION

A. Audio-Input-Handling

When a candidate provides a response through the microphone, the system captures the audio stream and assigns a unique session identifier to the input. The audio is associated with relevant metadata such as user details, session ID, and timestamp. The input is validated for quality and continuity before processing. This ensures that only clear and usable audio streams are passed into the processing pipeline.

The captured audio acts as the primary input for transcription and response evaluation modules.

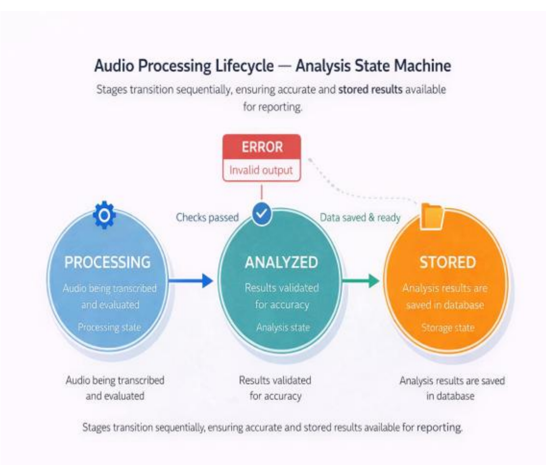
B. Processing-Interface

The system processes the captured audio using real-time speech recognition and AI evaluation models. The transcription module converts spoken responses into text using Deepgram, while the evaluation module analyzes the transcript using Groq-based language models. The system performs validation checks to ensure accuracy and completeness of the generated output. The processed data is then used to generate structured evaluation results such as scores, feedback, and performance metrics. All operations are executed within a controlled workflow to maintain consistency and reliability.

C. Result-Lifecycle

The analysis results follow a defined lifecycle consisting of stages such as processing, evaluated, and stored. Initially, the response is marked as processing while transcription and evaluation are performed. Once analysis is completed, the results are marked as evaluated and stored in the database. These results can then be accessed through the dashboard for review and feedback. The system ensures that each stage follows a proper sequence, preventing incomplete or inconsistent data from being used in performance evaluation

VIII. REAL-TIME ANALYSIS AND RESULTNOTIFICATIONS



A. Response Evaluation Threshold Detection

On each processed response event, the system evaluates performance metrics generated from the candidate's answer using predefined thresholds. The transcribed text is analyzed for relevance, clarity, and completeness, while additional parameters such as speech rate and filler word usage are also considered. When the computed evaluation scores fall below or exceed defined thresholds, the system identifies the response as critical. This evaluation is performed automatically within the processing pipeline, ensuring efficient detection of performance patterns during the interview.

The threshold values are designed to balance accuracy and reliability, ensuring that meaningful responses are identified without generating unnecessary alerts. These thresholds are configurable parameters and can be adjusted based on evaluation criteria without modifying the system architecture or core processing logic.

B. Automated-Result-Handling

When threshold conditions are met, the system automatically updates the evaluation status and stores the results in the database. the system records relevant details such as scores, feedback, and response timestamps. this process does not require manual intervention and ensures that all candidate responses are properly logged and available for further review.

to maintain data consistency, the system includes validation checks that prevent duplicate updates or incorrect state transitions. each evaluation result follows a defined sequence, ensuring that processing, analysis, and storage occur in a controlled and reliable manner.

C. Notification-Mechanism

The system generates real-time feedback notifications to inform candidates about their performance during the interview session. alerts are triggered when specific conditions such as low scores, incomplete responses, or excessive filler words are detected. these notifications provide immediate insights into areas that require improvement.

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715449

notifications are displayed directly on the dashboard and updated dynamically as responses are processed. this ensures that users receive continuous feedback without requiring manual refresh, enabling an interactive and responsive interview experience.

IX. ANALYTICS DASHBOARD

the analytics dashboard presents multiple categories of performance metrics related to candidate interview sessions. interaction metrics display the total number of completed interviews over a given time period, segmented by performance levels such as high, medium, and low scores. this provides an overall view of candidate performance trends and helps identify patterns in response quality. performance metrics aggregate evaluation scores across different sessions and time intervals, enabling comparison and identification of areas requiring improvement. response analysis reports highlight factors such as speech clarity, relevance, and confidence level, offering deeper insights into candidate communication skills.

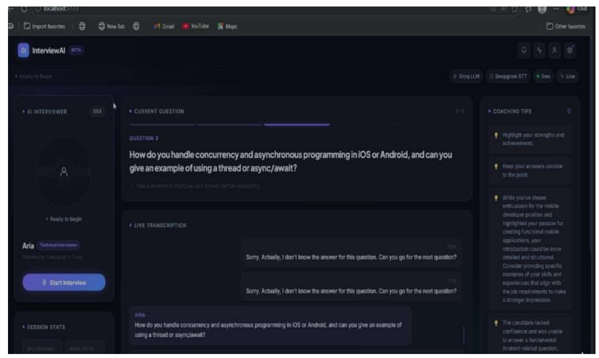
all metric categories are generated using backend processing functions that compute aggregated results directly from stored session data. these functions return structured outputs that are efficiently delivered to the frontend interface for visualization. performing aggregation at the system level reduces data transfer overhead and ensures consistency across multiple users accessing the dashboard simultaneously.

X. RESULTS AND DISCUSSION

The proposed AI-powered Interview Bot (EMU) was evaluated through end-to-end functional testing of the system using the Admin dashboard. The system successfully processed complete workflows including audio capture, real-time transcription, response evaluation, and feedback generation. The integrated modules worked efficiently to provide accurate and consistent results without requiring complex infrastructure. The system demonstrated its capability to handle continuous audio inputs and generate real-time insights for monitoring candidate performance. The following subsection presents the key output from the Admin dashboard.

A. Admin Dashboard – Analysis and Monitoring

The Admin dashboard serves as the central interface for managing and monitoring interview analysis. It allows users to access candidate responses, view transcripts, and analyze evaluation results including scores and feedback metrics. The dashboard presents summarized insights such as performance distribution and response quality across multiple sessions, helping in identifying trends in candidate performance. It also provides structured reports and visual representations that enable effective evaluation and support decision-making in recruitment processes.



XI. COMPARATIVE ANALYSIS

Table IV compares the proposed AI-powered Interview Bot with existing interview and evaluation approaches across key functional aspects addressed by the system. The most significant differentiators include the integration of real-time speech recognition, natural language processing, and AI-based response evaluation within a single unified platform. Unlike traditional interview methods that rely on manual assessment or basic question-answer formats, the proposed system provides automated, structured, and consistent evaluation of candidate responses.

Additionally, the system supports real-time processing and feedback generation, eliminating the need for manual review and reducing evaluation time. It also incorporates contextual questioning through resume parsing, enhancing the relevance of interview interactions

TABLE IV Feature Comparison

Feature	Proposed System	Manual Interviews	Basic Online Tests	Recorded Interviews
Automated interview process	Yes	No	Partial	No
Real-time speech-to-text	Yes	No	No	No
AI-based response evaluation	Yes	No	Partial	No
Dynamic question generation	Yes	No	No	No
Real-time feedback	Yes	No	No	No
Resume-based questioning	Yes	Partial	No	No
Performance scoring	Yes	Partial	Yes	Partial
Scalable architecture	Yes	No	Yes	No
Centralized data storage	Yes	No	Yes	Partial

XII. LIMITATIONS AND CHALLENGES

The system relies on speech recognition models to convert candidate responses into text, which may be affected by background noise, accents, or unclear speech. In noisy environments or unstable network conditions, transcription accuracy may decrease, leading to incorrect evaluation results and feedback. Improving audio input quality or using more advanced models can help reduce these limitations.

The evaluation process depends on natural language processing and large language models, which may not always fully capture the true intent or context of a candidate's response. Variations in speaking style, language proficiency, or incomplete answers can affect the accuracy of generated scores and feedback. As a result, certain evaluations may not always reflect the candidate's actual capabilities.

Although the system is designed for efficient real-time processing, handling continuous audio streams and AI-based evaluation requires significant computational resources. Large numbers of concurrent interview sessions may increase system load and impact processing speed, especially when deployed on limited infrastructure.

XIII. CONCLUSION

This paper presented an ai-powered interview bot (emu), a web-based system that integrates real-time speech recognition, natural language processing, and ai-driven evaluation to automate the interview process. the system provides a comprehensive approach by analyzing candidate responses using structured evaluation frameworks, enabling accurate and objective assessment of communication skills and performance. the centralized architecture allows efficient data processing, storage, and visualization through an interactive dashboard.

the proposed system overcomes the limitations of traditional interview methods by automating evaluation and providing real-time feedback to candidates. by analyzing response content, speech patterns, and performance metrics, the system improves the reliability and consistency of candidate assessment and supports better decision-making in recruitment processes.

XIV. ACKNOWLEDGMENT

The author expresses sincere gratitude to Dr. K. Ponmozhi, Associate Professor, Department of Computer Applications, SRM Valliammai Engineering College, for valuable guidance, continuous support, and encouragement throughout the development of this project. The author also acknowledges the contributions of open-source technologies and tools used in this work, including speech recognition and natural language processing frameworks, which played a significant role in the successful implementation of the system.

REFERENCES

[1] B. Schuller, S. Steidl, and A. Batliner, "The INTERSPEECH 2009 Emotion Challenge," in Proc. INTERSPEECH, 2009, pp. 312–315.

[2] D. Jurafsky and J. H. Martin, "Speech and Language Processing," 3rd ed., Pearson Education, 2021.

[3] T. Brown et al., "Language Models are Few-Shot Learners," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2020, pp. 1877–1901.

[4] A. Graves, A. Mohamed, and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), 2013, pp. 6645–6649.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[6] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New Avenues in Opinion Mining and Sentiment Analysis," IEEE Intelligent Systems, vol. 28, no. 2, pp. 15–21, 2013.

[7] R. Cowie, E. Douglas-Cowie, and C. Cox, "Emotion Recognition in Human-Computer Interaction," IEEE Signal Processing Magazine, vol. 18, no. 1, pp. 32–80, 2001.

[8] S. Poria, E. Cambria, and A. Gelbukh, "Deep Convolutional Neural Network Textual Features and Multiple Kernel Learning for Multimodal Sentiment Analysis," in Proc. EMNLP, 2015, pp. 2539–2544.