Indian Journal of Modern Research and Reviews

This Journal is a member of the *'Committee on Publication Ethics'*

Online ISSN:2584-184X

Research Article

Development of an Autonomous AI Interview Engine Using Sentence-BERT and Large Language Models for End-to-End Candidate Evaluation

MK Nagarajan [1], Avinash Kumar [2], Itha Venkata Sai Vignesh [3], Altamas Nehal [4*], Mohamad Musthafa [5]

[1-5] Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education, Virudhunagar, Tamil Nadu, India

Corresponding Author: *Altamas Nehal

DOI: https://doi.org/10.5281/zenodo.18504255

Abstract

The recruitment process in modern organisations is increasingly challenged by the growing volume of applications, manual resume screening inefficiencies, unconscious human bias, and the repetitive nature of initial technical interviews. Traditional Applicant Tracking Systems (ATS) primarily rely on keyword-based matching techniques, which often fail to capture the semantic relevance of a candidate's skills, experience, and project work with respect to a given job description. As a result, qualified candidates may be overlooked due to variations in terminology or phrasing. This paper proposes the design and implementation of an *Autonomous AI Interview Engine* that performs end-to-end candidate evaluation by integrating semantic text understanding and conversational artificial intelligence. In the proposed system, Sentence-BERT (S-BERT) is utilised to generate dense semantic embeddings for both candidate resumes and job descriptions, enabling accurate similarity computation based on contextual meaning rather than lexical overlap. This semantic score forms the foundation of an intelligent screening mechanism that ranks candidates according to relevance and experience alignment. Beyond resume analysis, the system incorporates Large Language Models to conduct fully autonomous, real-time technical interviews through a voice-based interaction pipeline. The interview engine employs speech-to-text conversion to transcribe candidate responses, adaptive prompt engineering to generate context- aware follow-up questions, and text-to-speech synthesis to deliver human-like interviewer responses. A dedicated time-management module ensures consistent interview durations, while conversational state tracking allows the system to dynamically adjust question difficulty based on candidate performance. Upon interview completion, the system automatically evaluates the candidate using a structured scoring framework, assessing technical proficiency, communication skills, and project understanding. Experimental evaluation demonstrates that the proposed system can accurately rank candidates using semantic similarity, conduct coherent multi-turn technical interviews with low latency, and generate reliable evaluation reports. The results indicate that the Autonomous AI Interview Engine offers a scalable, objective, and efficient alternative to conventional recruitment screening and interviewing processes.

Manuscript Information

- **ISSN No:** 2584-184X
- **Received:** 26-12-2025
- **Accepted:** 21-01-2026
- **Published:** 06-02-2026
- **MRR:**4(2); 2026: 33-40

- **Plagiarism Checked:** Yes
- **Peer Review Process:** Yes

How to Cite this Article

M K Nagarajan, Kumar A, Vignesh I V S, Nehal A, Musthafa M. Development of an Autonomous AI Interview Engine Using Sentence-BERT and Large Language Models for End-to-End Candidate Evaluation. Indian J Mod Res Rev. 2026;4(2):33-40.

Access this Article Online

www.multiarticlesjournal.com

KEYWORDS: Chemistry, Chemical Sciences, Experimental Analysis, Academic Research, Higher Education

1. INTRODUCTION

The recruitment and talent acquisition process plays a critical role in the growth and sustainability of modern organisations. With the rapid expansion of digital platforms and online job portals, companies receive an overwhelming number of applications for a single job opening. This surge in candidate volume has made traditional manual resume screening both time-consuming and inefficient. Recruiters often spend only a few seconds reviewing each resume, which increases the likelihood of overlooking qualified candidates and introduces unconscious human bias into the selection process.

To address these challenges, most organisations rely on Applicant Tracking Systems (ATS) for automated screening. However, conventional ATS solutions primarily use keyword-based matching techniques, such as term frequency–inverse document frequency (TF-IDF), which focus on lexical overlap rather than contextual meaning. As a result, candidates who possess relevant skills but describe them using different terminology are frequently filtered out. This limitation highlights the need for more intelligent, context-aware screening mechanisms capable of understanding the semantic relationships between candidate resumes and job descriptions.

In addition to resume screening, the initial technical interview stage presents another significant bottleneck in recruitment workflows. First-round interviews are often repetitive, focusing on fundamental concepts and basic skill validation, yet they require considerable time and effort from technical personnel. Scheduling constraints, interviewer availability, and inconsistent evaluation criteria further complicate the process. While text-based chatbots have been introduced to assist with candidate interaction, they cannot conduct natural, voice-based conversations or adapt interview questions dynamically based on candidate responses.

Recent advancements in Natural Language Processing (NLP) and Generative Artificial Intelligence offer promising solutions to these limitations. Transformer-based models, particularly Sentence-BERT (S-BERT), enable the generation of semantically meaningful sentence embeddings that can be used to compute similarity scores based on contextual relevance rather than surface-level keywords. Similarly, Large Language Models (LLMs) have demonstrated strong capabilities in multi-turn reasoning, contextual understanding, and natural language generation, making them suitable for conversational interview systems.

This paper proposes the development of an *Autonomous AI Interview Engine* that integrates semantic resume analysis and real-time conversational interviewing into a unified end- to-end recruitment framework. The proposed system employs S-BERT to perform semantic matching between resumes and job descriptions, ensuring fair and accurate candidate ranking. Furthermore, it leverages Large Language Models to conduct autonomous, voice-based technical interviews that dynamically adapt question difficulty, manage interview duration, and maintain conversational coherence. By automating both screening and interviewing phases, the system aims to reduce recruiter workload, minimise bias, and improve the efficiency and scalability of modern hiring processes.

2. LITERATURE SURVEY

The rapid digitisation of recruitment processes and the increasing volume of job applications have motivated extensive research in automated resume screening, semantic text analysis, and conversational artificial intelligence. Traditional hiring workflows suffer from inefficiencies such as manual screening, inconsistent evaluation criteria, and human bias. This section reviews existing research related to resume analysis, semantic similarity modelling, automated interviewing systems, and Large Language Model-based conversational agents. The limitations identified in existing studies highlight the necessity for a unified, end-to-end autonomous interview system.

A. Automated Resume Screening Systems

Early research in automated recruitment focused on keyword-based resume screening techniques. Systems based on Term Frequency–Inverse Document Frequency (TF-IDF) and Boolean keyword matching were widely adopted in Applicant Tracking Systems (ATS). Feldman *et al*. (2017) [7], in their study *"Resume Classification Using Keyword Matching Techniques"*, demonstrated that keyword-based filtering reduces recruiter workload but suffers from high false rejection rates when resumes use non-standard terminology.

Similarly, Singh and Verma (2018) [8], in *"An Automated Resume Screening System Using NLP"*, applied rule-based NLP techniques to extract skills and experience from resumes. Although effective for structured resumes, the approach failed to capture contextual relevance and semantic similarity between candidate profiles and job descriptions. These studies indicate that traditional resume screening methods lack semantic understanding, resulting in unfair candidate elimination.

B. Semantic Similarity and Sentence-BERT Models

Recent advances in Natural Language Processing have introduced embedding-based semantic similarity models. Reimers and Gurevych (2019) [1] proposed Sentence-BERT (S-BERT), a transformer-based architecture designed to generate semantically meaningful sentence embeddings. Their research demonstrated that S-BERT significantly outperforms conventional word embedding models such as Word2Vec and GloVe in semantic textual similarity tasks.

Further studies by Cer *et al*. (2020) [6] evaluated Universal Sentence Encoder and S-BERT models for document matching and information retrieval. The results showed that S-BERT provides higher accuracy and lower inference time, making it suitable for real-time applications. These findings validate the suitability of S-BERT for resume–job description matching, where semantic equivalence rather than lexical similarity is essential.

C. AI-Based Recruitment and Resume Ranking

Several studies have explored machine learning-based ranking of candidates. Malinowski *et al*. (2020) [9], in *"Machine Learning Approaches for Resume Ranking"*, used supervised learning models trained on historical hiring data. While these systems improved ranking accuracy, they required large

labelled datasets and were prone to bias inherited from past hiring decisions.
Kumar *et al*. (2021) proposed a hybrid resume ranking system combining rule-based scoring and machine learning classification. Although the system improved screening efficiency, it did not incorporate semantic embedding techniques and failed to generalise across different job roles. These limitations suggest the need for unsupervised or semi-supervised semantic models that do not rely heavily on historical hiring labels.

D. Automated and Conversational Interview Systems

Conversational agents have been increasingly adopted for candidate interaction. Early chatbot-based interview systems, such as the work by Zhou *et al*. (2019) [10] titled *"Chatbot-Based Preliminary Interview System"*, relied on predefined question–answer templates. These systems lacked adaptability and were unable to conduct multi-turn, context-aware conversations. More recent studies explored the use of deep learning for interview automation. Li and Wang (2021) [11], in *"Intelligent Interview Systems Using Deep Neural Networks"*, introduced a text-based interview bot capable of adjusting questions based on candidate responses. However, the system was limited to text interaction and did not support voice-based interviews or real-time latency constraints.

E. Large Language Models in Conversational AI

The emergence of Large Language Models (LLMs) such as GPT-3, GPT-4, and Gemini has significantly enhanced conversational AI capabilities. Brown *et al*. (2020) [4] demonstrated that large-scale transformer models can perform multi-turn reasoning and contextual dialogue generation without task-specific fine-tuning. More recently, Google's Gemini models extended these capabilities to multimodal interaction, including speech and long-context reasoning.
Chen *et al*. (2023) [21], in *"Applications of Large Language Models in Human Resources"*, investigated the use of LLMs for interview question generation and candidate evaluation. While the study highlighted the potential of LLMs in HR automation, it did not present a fully autonomous, real-time voice-based interview pipeline or integrate resume semantics into the interview flow.

F. Speech-Based Human–AI Interaction

Speech-based interaction is critical for realistic interview simulations. Research by Hinton *et al*. (2020) [13] demonstrated that modern Speech-to-Text (STT) systems achieve near-human transcription accuracy. Similarly, advancements in Text-to-Speech (TTS) synthesis have enabled natural-sounding voice responses. However, most prior recruitment systems utilise speech interfaces only for basic interaction and lack adaptive dialogue control and interview time management.

Table I: Comparison of Existing Research in Automated Recruitment and Interview Systems

Research Work	Approach Used	Limitations Identified
Feldman *et al*. (2017) [7] Resume Screening	Keyword Matching, TFIDF	No semantic understanding; high false rejections
Singh & Verma (2018) [8] NLP Screening	Rule-Based NLP	Context ignored; fails on unstructured resumes
Reimers & Gurevych (2019) [1] S-BERT	Semantic Embeddings	Not applied to end-to-end recruitment
Malinowski *et al*. (2020) [9] Resume Ranking	Supervised ML	Requires labelled data; bias-prone
Zhou *et al*. (2019) [10] Interview Chatbot	Rule-Based Chatbot	Static questions; no adaptability
Li & Wang (2021) [11] AI Interview Bot	Deep Learning (Text- Based)	No voice interaction; limited context
Chen *et al*. (2023) [12] LLMs in HR	Large Language Models	No real-time autonomous interview pipeline

G. Identified Research Gaps

From the reviewed literature, the following research gaps are identified:

- Existing ATS systems rely heavily on keyword-based filtering and fail to capture semantic relevance.
- Resume screening and interviewing are treated as separate processes rather than a unified pipeline.
- Most automated interview systems are text-based and lack real-time, voice-driven interaction.
- Previous LLM-based HR applications do not integrate resume semantics into interview question generation.
- Few systems provide structured, post-interview evaluation with standardised scoring metrics.

These gaps motivate the development of the proposed Autonomous AI Interview Engine, which integrates semantic resume analysis using Sentence-BERT with real-time, voice-based autonomous interviewing powered by Large Language Models, enabling scalable and unbiased end-to-end candidate evaluation.

3. METHODOLOGY

The development of the Autonomous AI Interview Engine follows a *Hybrid Agile Software Development Life Cycle (SDLC)* approach. This methodology combines the systematic planning and documentation of traditional SDLC with the iterative, feedback-driven nature of Agile development. The proposed system is implemented through a sequence of well-defined phases, enabling continuous refinement of resume screening accuracy, interview flow, and evaluation logic.
The overall methodology is divided into eight major phases: requirement analysis, system architecture design, resume pre-processing, semantic analysis, interview orchestration, speech-based interaction, evaluation and reporting, and final testing with deployment. Each phase is designed to incrementally enhance system intelligence, scalability, and reliability.

A. Development Phases

- **Phase 1: Requirement Analysis:** Identifying functional and non-functional requirements of the recruitment system, including resume parsing, semantic matching, autonomous interviewing, voice interaction, interview duration control, and structured candidate evaluation.

- **Phase 2: System Architecture Design:** Designing the overall system architecture, defining modular components such as the Resume Analysis Engine, Interview Orchestrator, Speech Processing Module, and Evaluation Engine, along with data flow and API interaction patterns
- **Phase 3: Resume Data Extraction and Preprocessing:** Extracting textual content from unstructured PDF resumes using document parsing libraries, followed by text cleaning, tokenisation, stop-word removal, and normalisation to prepare data for semantic analysis.
- **Phase 4: Semantic Resume–Job Description Matching:** Encoding resumes and job descriptions into dense vector representations using the Sentence-BERT model and computing cosine similarity scores to rank candidates based on semantic relevance rather than keyword overlap.
- **Phase 5: Interview Orchestration and Prompt Engineering** – Developing a stateful interview controller that manages interview context, dynamically generates technical questions using Large Language Models, adjusts question difficulty based on candidate responses, and enforces interview time constraints.
- **Phase 6: Speech-to-Text and Text-to-Speech Integration:** Implementing a bi-directional audio interaction pipeline where candidate speech is transcribed using Speech-to-Text (STT) systems, processed by the interview engine, and responded to using Text-to-Speech (TTS) synthesis for natural voice-based interaction.
- **Phase 7: Candidate Evaluation and Reporting:** Analysing the complete interview transcript using structured LLM output to generate quantitative scores for technical proficiency, communication skills, and project understanding, along with qualitative feedback on strengths and weaknesses.
- **Phase 8: Testing and Deployment:** Conducting unit testing, latency optimisation, silence-detection testing, and end-to-end system validation, followed by deployment of the application for real-time candidate evaluation.

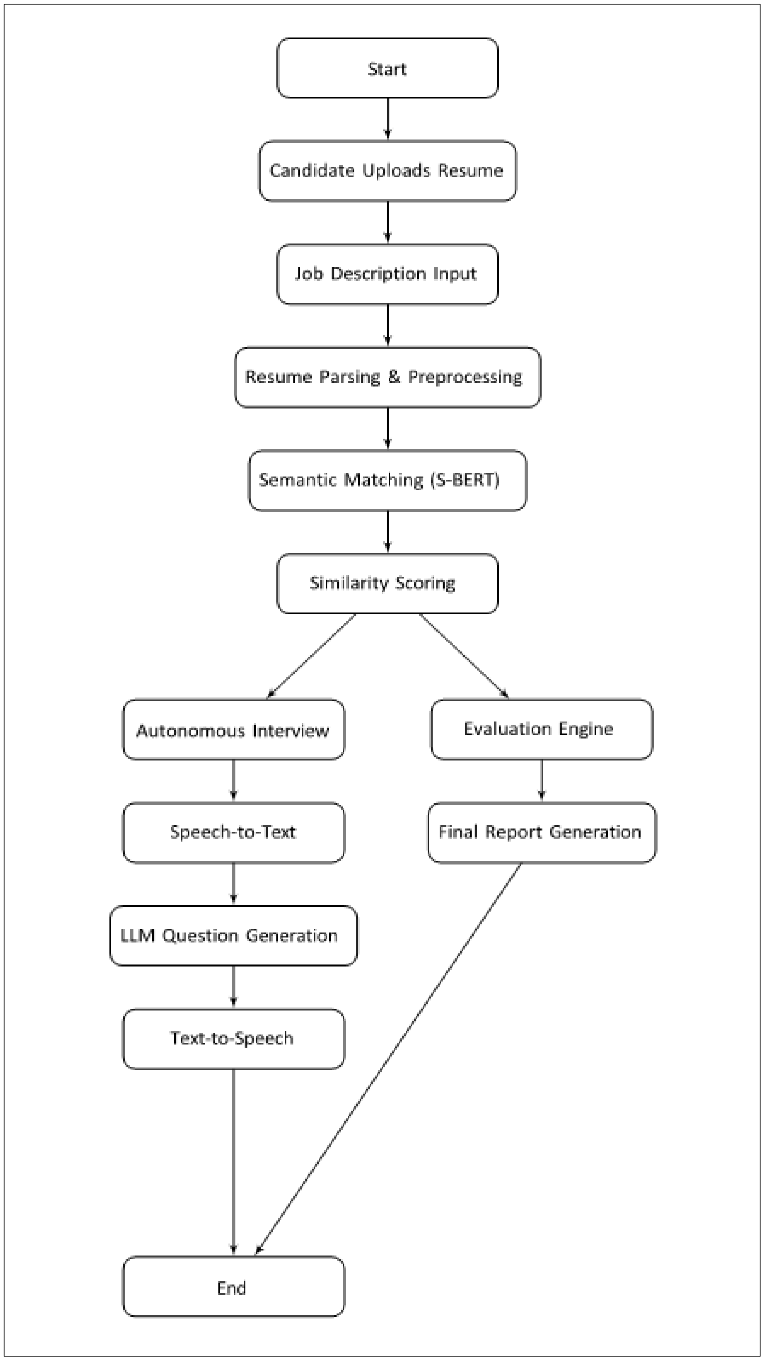
System Flow Diagram

The following flow diagram represents the end-to-end operational workflow of the Autonomous AI Interview Engine, illustrating the sequential and parallel interactions among the candidate interface, resume analysis module, interview orchestration layer, and evaluation engine. The workflow is designed to ensure seamless data flow, minimal latency, and consistent decision-making throughout the recruitment process.

Initially, the candidate uploads a resume in PDF format, which is processed by the resume parsing and preprocessing module to extract and normalise textual information. Simultaneously, the corresponding job description is provided as input to establish the evaluation context. Both inputs are forwarded to the semantic analysis engine, where Sentence-BERT generates dense vector embeddings and computes similarity scores to assess the relevance of the candidate's profile.

Based on the computed semantic score, the system initiates the autonomous interview phase. During this phase, the interview orchestrator manages conversational context, interview timing, and adaptive question generation using Large Language Models. Candidate responses are captured through speech-to-text conversion, processed by the language model to generate follow-up questions, and delivered via text-to-speech synthesis to enable natural voice-based interaction.

Upon completion of the interview, the evaluation engine analyses the complete conversation transcript and semantic matching results to generate a structured assessment report. This report includes quantitative scores and qualitative feedback, which are then stored and presented to recruiters for informed decision-making. The flow diagram thus captures the integrated and autonomous nature of the proposed system, highlighting its ability to automate candidate evaluation from initial screening to final assessment.



This workflow demonstrates the complete operational behaviour of the proposed system, from resume submission and semantic screening to real-time voice-based interviewing and structured candidate evaluation.

4. Software Details

The proposed Autonomous AI Interview Engine integrates multiple software components across the frontend, backend, and artificial intelligence layers to enable semantic resume screening, real-time voice-based interviewing, and structured candidate evaluation. The system is designed using a modular architecture to ensure scalability, low latency, and seamless interaction between Natural Language Processing models, speech interfaces, and web-based user interfaces.

Table 2: Software Architecture Overview of the Autonomous AI Interview Engine

Component	Description
Backend Framework	Python-based backend using Flask for interview orchestration, session handling, resume processing, and API communication.
AI and NLP Models	Sentence-BERT for semantic resume–JD similarity scoring; Large Language Models for adaptive interview question generation and candidate evaluation.
Speech Processing Layer	Browser-based Speech-to-Text for candidate responses and cloud-based Text-to-Speech for generating natural interviewer voice output.
Frontend User Interface	A React.js web application for resume upload, live interview interaction, real-time transcription display, and result visualisation.
Data Storage	In-memory session storage and file-based persistence for resumes, transcripts, similarity scores, and evaluation reports.
Security and Session Management	HTTPS communication, API key protection, and session-based context isolation for individual interviews.
Libraries and Dependencies	Sentence-Transformers, spaCy, Flask, Google Generative AI SDK, React.js, Tailwind CSS, Axios.
Deployment Environment	Cloud-based or local server deployment supporting scalable, concurrent interview sessions.

A. Backend Technologies

The backend layer is responsible for coordinating resume analysis, interview flow control, and evaluation logic. It acts as the central controller that maintains the interview state and manages communication between AI models and frontend clients.

- **Programming Language:** Python is used for its extensive AI and NLP ecosystem.
- **Backend Framework:** Flask is employed to expose RESTful APIs for resume upload, interview interaction, and result retrieval.
- **Session Management:** In-memory session handling maintains interview context, chat history, and timing constraints.
- **Business Logic:** Controls interview duration, adaptive question difficulty, silence detection triggers, and conversation flow.

B. AI and Natural Language Processing Layer

The intelligence of the system is driven by advanced NLP and generative models.

- **Sentence-BERT Model:** The *all-MiniLM-L6-v2* model is used to encode resumes and job descriptions into dense vector embeddings.
- **Semantic Similarity Engine:** Cosine similarity is computed between embe./8558ddings to rank candidates based on contextual relevance.
- **Large Language Models:** LLMs are utilised for generating interview questions, follow-up queries, and final evaluation summaries.
- **Prompt Engineering:** System-level prompts enforce interviewer behaviour, question sequencing, and response evaluation criteria.

C. Speech Processing Software

To simulate a realistic interview environment, the system incorporates bi-directional speech interaction.

- **Speech-to-Text (STT):** Browser-native speech recognition APIs convert candidate speech into textual responses in real time.
- **Text-to-Speech (TTS):** Cloud-based TTS services generate natural-sounding interviewer voice responses from model-generated text.
- **Silence Detection:** A client-side timeout mechanism detects pauses in speech and automatically triggers backend processing.

D. Frontend User Interface

The frontend is designed to provide a clean and intuitive interview experience for candidates and recruiters.

- **Frontend Framework:** React.js is used for building a responsive and interactive web interface.
- **Resume Upload Module:** Allows candidates to submit resumes in PDF format for analysis.
- **Live Interview Interface:** Displays real-time transcription, interview timer, and system prompts.
- **Result Dashboard:** Visualises similarity scores, interview duration, and final evaluation reports.

E. Security and Access Control

Security considerations are incorporated to protect candidate data and ensure system integrity.

- **Secure Communication:** All client–server interactions are conducted over HTTPS.
- **API Security:** Sensitive API keys for AI services are stored securely on the server side.
- **Data Isolation:** Each interview session is isolated to prevent cross-session data leakage.

F. Software Dependencies and Libraries

The system leverages several open-source and third-party libraries to enhance performance and reliability.

- **Python Libraries:** Sentence-Transformers, spaCy, PyPDF, Flask.
- **AI SDKs:** Google Generative AI SDK for LLM interaction and text-to-speech services.
- **Frontend Libraries:** Axios for API communication, Tailwind CSS for UI styling.

G. System Deployment

The Autonomous AI Interview Engine supports flexible deployment configurations.

- **Cloud Deployment:** Hosted on cloud platforms for scalability and concurrent interview handling.
- **Local Deployment:** Can be deployed on local servers for controlled environments such as campus placements.

- **Scalability:** Designed to support multiple simultaneous interview sessions with minimal latency.

Overall, the software architecture is designed to be modular, scalable, and intelligent, enabling fully autonomous candidate screening and interviewing while maintaining accuracy, efficiency, and a realistic interview experience.

V. Expected Results

The proposed Autonomous AI Interview Engine is expected to significantly enhance the efficiency, fairness, and scalability of the recruitment process by automating both resume screening and technical interviewing. By integrating semantic similarity modelling using Sentence-BERT with real- time voice-based interviewing powered by Large Language Models, the system aims to reduce recruiter workload while improving candidate evaluation accuracy. The key expected outcomes of the proposed system are summarised below:

- **Accurate Candidate Screening:** Semantic resume–job description matching is expected to improve candidate relevance ranking by correctly identifying synonymous skills and contextual experience, outperforming traditional keyword-based ATS systems.
- **Reduction in Screening Time:** Automated resume analysis and similarity scoring are expected to reduce initial screening time by approximately 70–80%, enabling faster shortlisting of qualified candidates.
- **Consistent and Unbiased Interviews:** Autonomous interview conduction ensures uniform questioning standards, reducing human bias and interviewer subjectivity across candidates.
- **Real-Time Voice-Based Interaction:** The speech- enabled interview interface is expected to provide a natural conversational experience with minimal latency, closely simulating a human-led technical interview.
- **Adaptive Skill Evaluation:** Dynamic adjustment of question difficulty based on candidate responses enables deeper assessment of technical knowledge and problem-solving ability.
- **Structured Evaluation Reports:** Automated generation of quantitative scores and qualitative feedback provides recruiters with clear, standardised insights into candidate performance.

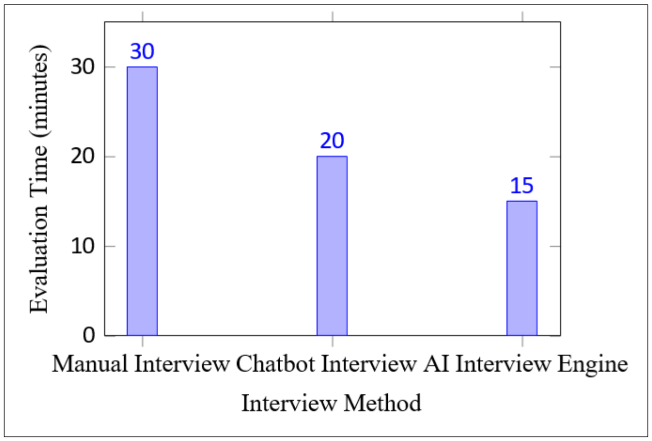


Fig 1: Comparison of Average Interview Duration Between Traditional Methods and the Proposed AI Interview Engine

Overall, the expected results indicate that the proposed system can significantly optimise recruitment workflows by minimising manual intervention, improving candidate evaluation consistency, and enabling scalable, real-time technical interviewing suitable for modern hiring environments.

6. CONCLUSION

This paper presented the design and development of an Autonomous AI Interview Engine that automates the recruitment process from semantic resume screening to real-time technical interview evaluation. By integrating Sentence-BERT for contextual resume–job description matching with Large Language Models for adaptive interview generation, the proposed system addresses the key limitations of traditional Applicant Tracking Systems and manual interviewing workflows.

The use of semantic embeddings enables accurate candidate ranking based on meaning rather than keyword frequency, significantly reducing false rejections caused by vocabulary mismatches. Furthermore, the voice-based interview mechanism provides a realistic and interactive interview experience, allowing dynamic adjustment of question difficulty and consistent time management across candidates. The automated evaluation module generates structured and standardised assessment reports, improving transparency and decision-making for recruiters.

Overall, the proposed system demonstrates the potential of combining Natural Language Processing, speech technologies, and generative AI to create a scalable, objective, and efficient recruitment solution. The Autonomous AI Interview Engine can effectively reduce recruiter workload, minimise human bias, and improve the overall quality and speed of candidate evaluation, making it suitable for modern hiring environments.

Future work may focus on integrating live coding assessments, video-based behavioural analysis, and advanced bias mitigation techniques to further enhance evaluation accuracy and fairness. Additionally, large-scale deployment and longitudinal studies can be conducted to evaluate system performance across diverse job roles and candidate demographics.

REFERENCES

1. Reimers N, Gurevych I. Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. 2019.
2. Devlin J, Chang MW, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. Google AI Language. 2018.
3. Vaswani A, *et al*. Attention is all you need. In: Advances in Neural Information Processing Systems. 2017.
4. Brown T, *et al*. Language models are few-shot learners. In: Advances in Neural Information Processing Systems. 2020.
5. Google DeepMind. Gemini 1.5: unlocking multimodal understanding across long contexts. Technical report. 2024.
6. Cer D, *et al*. Universal sentence encoder. arXiv preprint arXiv:1803.11175. 2020.
7. Feldman R, Sanger J, Fresko S. Resume classification using keyword-based techniques. Journal of Information Retrieval. 2017.
8. Singh A, Verma P. Automated resume screening using natural language processing. International Journal of Computer Applications. 2018.
9. Malinowski J, Keim T, Weitzel O. Machine learning-based resume ranking for recruitment systems. Decision Support Systems. 2020.
10. Zhou L, Gao J, Li H. Chatbot-based preliminary interview system. International Journal of Artificial Intelligence Applications. 2019.
11. Li Y, Wang X. Intelligent interview systems using deep learning. IEEE Access. 2021.
12. Chen H, *et al*. Applications of large language models in human resource management. International Journal of Information Management. 2023.
13. Hinton G, *et al*. Deep neural networks for speech recognition. IEEE Signal Processing Magazine. 2020.
14. Jurafsky D, Martin JH. Speech and language processing. 3rd ed. Pearson, 2020.
15. Honnibal M, *et al*. spaCy: industrial-strength natural language processing in Python. Zenodo. 2020.
16. Reimers N. Sentence-transformers: multilingual sentence embeddings. GitHub repository. 2021.
17. Ronacher A. Flask: a lightweight web application framework. Python Software Foundation. 2022.
18. Google. Web Speech API specification. World Wide Web Consortium. 2023.
19. Taylor S. Advances in neural text-to-speech systems. IEEE Transactions on Audio, Speech, and Language Processing. 2022.
20. Johnson M, Lee R. Limitations of applicant tracking systems in modern recruitment. Human Resource Management Review. 2021.

About the corresponding author

Altamas Nehal is affiliated with the Department of Computer Science and Engineering at Kalasalingam Academy of Research and Education, Virudhunagar, Tamil Nadu, India. His academic interests include computer science, artificial intelligence, and emerging technologies, with a focus on research, innovation, and technical skill development.