

Automated Interview Bot for Hiring Process Optimization Using Machine Learning

Pranali Manohar Rane, Prof. N. V. Alone

¹Student, ²Professor of ME Computer, Gokhale Education Society's R. H. Sapat College of Engineering, Management Studies and Research, Nashik

Abstract - In today's fast-paced recruitment landscape, traditional hiring practices often face challenges such as time consumption, human bias, and inconsistent candidate evaluation. The need for scalable, unbiased, and efficient hiring solutions has become crucial, especially when managing large volumes of applicants. This project, titled "Automated Interview Bot for Hiring Process Optimization Using Machine Learning, addresses these challenges by leveraging Artificial Intelligence (AI) techniques to automate the initial stages of the interview process. The system integrates Machine Learning (ML) and Natural Language Processing (NLP) to conduct dynamic interviews, analyze verbal and technical responses, and generate performance-based reports. Candidate evaluation is performed through two major components: a role-specific Multiple Choice Questionnaire (MCQ) and a verbal interview round. Using sentence transformers and semantic similarity algorithms, candidate responses are evaluated against model answers, while sentiment analysis adds depth to the assessment. The system further generates auto-evaluated reports for both candidates and administrators. The entire process is supported through a Flask-based web application with secure authentication, structured data management using SQLite, and PDF report generation using FPDF. This solution significantly reduces the time and effort required for preliminary screening while enhancing the fairness and consistency of evaluations. It provides actionable insights to recruiters, promotes a bias-free assessment environment, and delivers a user-friendly experience to candidates. This project demonstrates how intelligent systems can modernize and optimize the recruitment workflow, making it more scalable, reliable, and effective.

1. INTRODUCTION

In the era of digital transformation, the recruitment process has shifted from traditional face-to-face interviews to automated and online solutions. With the growing number of applicants and the rise of remote hiring, organizations need systems that can efficiently handle large-scale candidate screening while ensuring fair evaluation. Manual recruitment methods are time-consuming, labor-intensive, and often affected by human bias and inconsistent judgments. As a result, skilled candidates may sometimes be overlooked.

To overcome these challenges, the project titled "Automated Interview Bot for Hiring Process Optimization Using Machine Learning" introduces an AI-based recruitment solution. The system uses Natural Language Processing (NLP) and

Machine Learning (ML) to conduct verbal and technical interviews, analyze candidate responses, and generate automated performance reports.

The proposed system reduces human involvement in the initial recruitment stages, promotes unbiased evaluation, and improves hiring efficiency. By automating the screening process, it helps organizations save time, maintain consistency, and make smarter hiring decisions.

1.1 Problem Definition and Objectives

The traditional hiring process involves several limitations that affect its effectiveness and reliability:

1. Time-consuming procedures due to manual resume screening and interviews.
2. Subjective evaluations that may lead to biased or inconsistent outcomes.
3. Lack of scalability when handling a large volume of applicants.
4. Inability to objectively assess soft skills like confidence, communication, and attitude.
5. Resource dependency on HR personnel for repetitive tasks like initial screening.

OBJECTIVES

To solve the above challenges, the project is designed with the following key objectives:

- Develop a fully automated interview system that conducts MCQ and verbal interviews.
- Implement NLP-based question generation and response interpretation for both technical and behavioral assessments.
- Utilize semantic similarity algorithms to score verbal responses in real-time.
- Provide a dashboard and automated performance reports for recruiters to make data-backed decisions. Ensure the system is secure, scalable, and modular, allowing for future integration of voice-based analysis, multilingual support, or facial expression detection.

1.2 Project Scope

The proposed system enables candidates to register, choose their preferred job role, and participate in an automated interview session. Recruiters or administrators can access an admin dashboard to view candidate scores, monitor individual and overall performance, and download

detailed PDF reports. The platform supports multiple evaluation methods, including MCQ-based tests; verbal interviews, semantic answer scoring, and automated report generation. Role-specific interview questions are dynamically fetched from CSV files, making the system flexible and easy to customize or expand. The project is developed using Flask as the backend framework, SQLite as the database, and integrates powerful libraries such as sentence-transformers, FPDF, and SQLAlchemy to enhance functionality and performance.

2. METHODOLOGIES OF PROBLEM SOLVING

To implement a fully functional automated interview system, a combination of AI methodologies, web technologies, and data processing strategies has been employed:

2.1 Natural Language Processing (NLP)

NLP is the backbone of the verbal response evaluation module. It allows the system to interpret human language, understand sentence structure, and evaluate meaning.

- Models like MiniLM or BERT are used to embed candidate answers and compare them against ideal responses.
- Semantic similarity is calculated using cosine similarity of embeddings, providing a quantitative measure of how closely a candidate's answer matches the expected response.
- TextBlob or VADER can be integrated for sentiment analysis, identifying the tone and confidence in responses to behavioral questions.

2.2 Machine Learning (ML)

ML models are used to provide intelligent scoring of responses.

- Candidate answers are classified or scored based on semantic similarity scores, enabling dynamic and fair evaluation.
- Pre-trained models eliminate the need for large training datasets while still delivering accurate insights.

2.3 Web Application Development (Flask)

Flask serves as the backend web framework, enabling:

- Routing of candidate and admin actions.
- HTML-based rendering of quizzes and interviews.
- SecurAe data handling and user session management.

2.4 Database Management (SQLite + SQLAlchemy)

A relational database is maintained to store:

- Candidate data, interview responses, scores, and reports.
- ORM (Object Relational Mapper) with SQLAlchemy simplifies database operations and supports future migration to cloud-hosted databases like PostgreSQL or MySQL.

2.5 PDF Report Generation

After interview completion:

- The FPDF library is used to generate well-structured candidate and admin reports.
- These reports include performance summaries, observations, and recommendations based on candidate scores.

2.6 CSV-Driven Dynamic Question Loading

- Role-specific MCQ and technical questions are stored in CSV.
- When a candidate selects a role, relevant questions are dynamically fetched for the interview session.
- This method ensures easy customization and modularity.

2.7 Semantic Similarity Scoring

- Using sentence-transformers, embeddings of candidate responses and ideal answers are generated.
- Cosine similarity between the vectors is computed to assign scores (scaled to 10).
- This enables fair, unbiased evaluation without requiring handcrafted answer checking logic.

3. SYSTEM PROPOSED ARCHITECTURE

The proposed architecture for the project "Automated Interview Bot for Hiring Process Optimization Using Machine Learning" is designed to automate and streamline the recruitment process using modern technologies such as Machine Learning (ML), Natural Language Processing (NLP), and Computer Vision. This architecture ensures efficient handling of inputs, intelligent evaluation of candidate responses, secure data storage, and automated reporting.

The system consists of several interconnected components that work together in a pipeline from input to output:

User Input (Candidate Side):

The candidate first registers into the system and selects the desired job role. After registration, the candidate attends the interview process by providing inputs through registration forms, MCQ-based tests, and typed or spoken responses for the verbal interview section.

Processing and Evaluation:

The system automatically evaluates MCQ answers using dynamic scoring techniques. Verbal responses are analyzed with the help of NLP models such as Sentence-BERT to measure semantic similarity and assess answer quality. Additionally, OpenCV-based real-time face detection is used to ensure candidate presence and maintain interview authenticity.

Database Layer:

All candidate information, interview scores, and activity logs are securely stored in a SQLite database. SQL Alchemy ORM is integrated to provide efficient, reliable, and secure communication between the application and the database.

Report Generation:

The admin can monitor candidate performance through the dashboard and generate detailed PDF reports. These reports include registration details, MCQ scores, NLP-based evaluation scores, face detection results, and the final interview outcome or decision.

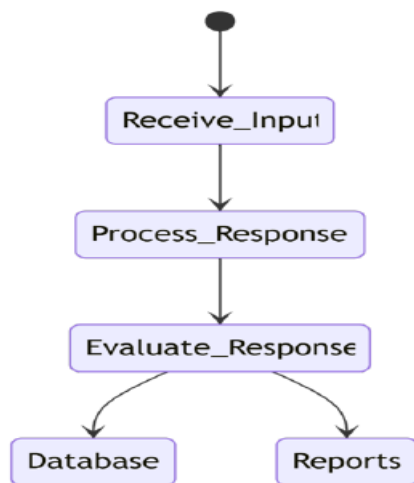


Fig1: Proposed System Architecture of Automated Interview Bot

The architecture follows a linear and modular pipeline:

- Input Phase: Candidate registers → selects job role → answers MCQs and verbal questions.
- Processing Phase:
 - Backend evaluates MCQs instantly.
 - Verbal responses passed to an NLP model for scoring.
 - Face detection ensures only the intended candidate is present.
 - Storage Phase: All inputs and scores are saved in the database.
 - Output Phase: Admins generate reports and view candidate status.

4. REAL-WORLD APPLICATIONS

Government and Public Sector Examinations

Government recruitment agencies can use the system to automate initial candidate assessments for large-scale examinations and hiring processes. It ensures faster and more transparent shortlisting, reduces manual workload and corruption risks, and verifies candidate identity through real-time face detection technology.

Campus Placements and University Hiring Drives

Educational institutions and placement cells can use the system for mock interviews and student pre-screening during campus recruitment drives. The platform provides a standardized method for evaluating students, offers feedback on communication and technical skills, and supports efficient management of large-scale placement activities.

Remote and Global Hiring

The system supports remote hiring by enabling companies to conduct interviews with candidates from different geographical locations. It allows verbal interviews through voice or text responses, performs real-time candidate authentication using webcam monitoring, and removes barriers related to location and time zones.

Skill Assessment Platforms

The Automated Interview Bot can also be integrated into online learning and skill assessment platforms. EdTech companies can use it to conduct MCQ and verbal tests, track learner performance, provide certification-based analysis, and help users prepare for real-world interviews.

Start-ups and Small Businesses

Start-ups and small businesses with limited HR resources can use the system to automate their recruitment process. It helps them reduce dependency on full-time HR staff, focus on business growth, and quickly identify qualified candidates through automated pre-screening and evaluation.

3. CONCLUSIONS

The project was successfully planned and executed over a span of six weeks, adhering closely to the predefined milestones and task dependencies. Each week focused on incremental development, starting from requirement gathering to final testing and documentation. Despite minor delays due to integration challenges and model tuning, the overall effort closely matched the estimated hours, indicating effective planning and resource management.

Risk factors such as technical dependencies and infrastructure concerns were identified early, allowing for timely mitigation without major disruptions. The team maintained strong coordination across different roles, ensuring smooth transitions between tasks and modules.

The final deliverables met the functional and technical objectives set at the beginning of the project, demonstrating the effectiveness of structured planning, agile execution, and proactive risk management. Let me know if you'd like the conclusion tailored for a seminar report, presentation, or project submission.

REFERENCES

- [1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proceedings of NAACL-HLT, 2019.
- [2] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embedding's using Siamese BERT-Networks," EMNLP-IJCNLP, 2019.
- [3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," OpenAI, 2019.
- [4] E. D. Liddy, "Natural Language Processing," Encyclopedia of Library and Information Science, 2001.
- [5] Y. Liu, Z. Tang, and J. Li, "Automated Interview Systems Based on NLP and Machine Learning Techniques," International Journal of Advanced Computer Science and Applications, 2020.
- [6] C. J. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," Proceedings of the International AAAI Conference on Web and Social Media, 2014.
- [7] Armin Ronacher, "Flask Web Development, O'Reilly Media, 2018.