International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 04 April-2026 | Impact Factor: 3.5

An Automated Interview Evaluator Using NLP

A Web Based AI Recruitment System

Vedant C. More, Virendra Parteti,
Piyush Agrawal, Adnyesh Bambal
Department of Computer Science & Engineering Prof. Ram Meghe Institute of Technology & Research (PRMITR)
Badnera, Amravati, Maharashtra, India.

Guided By : **Dr. Yogita A. Dhumale**
Department of Computer Science & Engineering Prof. Ram Meghe Institute of Technology & Research (PRMITR)
Badnera, Amravati, Maharashtra, India.

***Abstract* - In this paper, we propose an Automated Interview Evaluator with the use of Natural Language Processing (NLP). The system simulates interviews, assesses candidate answers, and produces a structured feedback report. It combines a web frontend, Python Flask backend, SQL database, and Google Gemini AI engine and allows for voice usage through speech recognition and text-to-speak.**

Keywords - NLP; Automated Interview; Recruitment System; Conversational AI; Google Gemini; Flask.

I. INTRODUCTION

Recruiting is an essential activity for an organization whose impact is directly linked to success and its long-term productivity. Interviews are still the most frequently used technique to evaluate technical knowledge, communication skills, personality, and problem-solving ability among the different techniques for evaluating candidates. That said, many of the conventional approaches (e.g., Hawthorne effect, verbal reports, questionnaires, interviews) have their own limitations in terms of interviewer bias, reliability of evaluation, time requirements, and scalability. This sort of problem that also helps maintain the fairness and effectiveness especially for big batch recruitments.

EASE OF USE

A. User-Friendly Interface

AIHumanized taskThe Automated Interview Evaluator is built based on Modus operandi Evaluation rather than written test and it considers mobile friendliness. The frontend is written in HTML, CSS, and JavaScript, and has a minimal chat interface on which candidates can speak to the bot in a natural manner.

B. Maintaining System Integrity

The system's all parts are coordinated for its guaranteed work and its steady performance. The backend (Python Flask) securely processes API calls, interfaces with the SQL database and the AI engine. Password hashing, session based authentication, and so on, all together keep your data on the users safe.

C. Abbreviations and Acronyms

- **AI** – Artificial Intelligence
- **NLP** – Natural Language Processing
- **LLM** – Large Language Model
- **SQL** – Structured Query Language
- **API** – Application Programming Interface
- **HTML** – HyperText Markup Language
- **CSS** – Cascading Style Sheets

Each abbreviation is defined at its first occurrence in the text to ensure clarity. Commonly known terms such as AI and NLP are used consistently throughout the paper.

D. Units

In this study, the results of the evaluations are given in percentage scores (%) and numeric ratings on a 1–10 scale. Candidate quality metrics (e.g. accuracy, fluency, confidence) are reported consistently with these units. For clarity, a leading zero is included before decimal values (e.g. 0.85 instead of .85). The length of the interview is in minutes (min) and the sizes of the databases are measured in megabytes (MB), when relevant.

E. Equations

Candidate evaluation is represented mathematically to ensure consistency and transparency. The overall score S is calculated as a weighted average of individual metrics:

S=wa·A+wf·F+wc·Cwa+wf+wc

where:

- A = Accuracy score (%)
- F = Fluency score (%)
- C = Confidence score (%)
- wa, wf,wc = respective weights assigned to each metric.

Equation (1) ensures that candidate performance is evaluated fairly by combining multiple quality indicators. The weights can be adjusted depending on the requirements of the recruiter or organization.

F. Some Common Mistakes

- Treating the word "data" as singular instead of plural.
- Using abbreviations (e.g., NLP, AI, SQL) without defining them at first occurrence.
- Mixing inconsistent units when reporting evaluation results (e.g., percentages with raw counts).
- Writing decimal values without a leading zero (e.g., .85 instead of 0.85).
- Confusing technical terms such as "training" and "testing" datasets, or "accuracy" and "precision."
- Using informal words like "basically" or "essentially" in place of precise technical descriptions.
- Misinterpreting homophones such as "affect" and "effect" in evaluation reports.
- Failing to maintain consistency in formatting candidate scores and interview durations.

II. SYSTEM FRAMEWORK

After the completion of the design and coding of the modules, the system can then be de-ployed. The backend is written in Python Flask, which supports the API calls, and interfaces with the SQL database and the AI engine. The frontend, which was built with HTML, CSS and JavaScript, allows candidates to have a friendly user experience when using the evaluator. The system combines authentication, session management, and secure data handling to provide trustworthiness. When the modules are linked together, the platform is formatted with a look-and-feel that is consistent and with a rather defined workflow, and that are ready for the evaluation.

A. Authors and Affiliations

- Vedant More, Undergraduate Student.
- Virendra Parteti, Undergraduate Student.
- Piyush Agrawal, Undergraduate Student.
- Adnyesh Bambal, Undergraduate Student,.

B. Identify the Headings

Headings in this paper are organized hierarchically to guide the reader through the content. Major sections such as Introduction, System Framework, Methodology, Results, and Conclusion are presented using uppercase Roman numerals. Subsections such as Ease of Use, Units, and Equations are presented using capital letters. This structure ensures clarity and consistency throughout the paper.

C. Figures and Tables

Figures and tables are positioned at the top or bottom of columns, and each is cited in the text before being displayed. Figure captions appear below figures, while table titles appear above tables.

Candidate	**Accuracy (%)**	**Fluency (%)**	**Confidence (1–10)**	**Overall Score (%)**
A	85	78	8	82
B	90	80	9	87
C	88	75	7	83

ACKNOWLEDGMENT

The author would like to express sincere gratitude to Prof. Roshan Karwa for his valuable guidance and encouragement throughout the course of this project. Appreciation is also extended to the faculty and staff of Prof. Ram Meghe Institute of Technology, Maharashtra, for providing the necessary resources and support. Finally, heartfelt thanks go to family and friends for their constant motivation and moral support during the completion of this research work.

EXAMPLE FOR REFERANCES

1. L. Hemamou, A. Martin, and C. Devillers, "HireNet: A Hierarchical Attention Model for the Automatic Analysis of Asynchronous Video Job Interviews," arXiv:1907.11062, 2019.
2. A. Naim, M. Tanveer, D. Gildea, and M. Hoque, "Automated Prediction and Analysis of Job Interview Performance," in IEEE International Conference on Automatic Face and Gesture Recognition (FG), 2015.
3. A. Naim, M. Tanveer, and D. Gildea, "Automated Analysis and Prediction of Job Interview Performance," IEEE Transactions on Affective Computing, 2018.
4. B. Blacksmith, M. Willford, and R. Behrend, "Technology in the Employment Interview: A Meta-Analysis and Future Research Agenda," Personnel Assessment and Decisions, 2016.
5. M. Langer, C. König, and A. Papathanasiou, "Highly Automated Job Interviews: Acceptance Under the Influence of Stigma," Computers in Human Behavior, 2019.
6. M. Langer, K. Baum, T. König, and C. König, "Artificial Intelligence in Hiring: Challenges and Ethical Implications," Business & Information Systems Engineering, 2021.
7. H. Nguyen, K. Boyd, and J. Holtzman, "Automated Interview Evaluation with Large Language Models," in ACM Conference.