Smart Resume Analyzer

Kazi Simran[1], Tamboli Aalmeen[2], Sayyed Afsha[3], Shaikh Sufiya[4]
[1, 2, 3, 4] BIT,BARSHI

Abstract- *In today's competitive job market, most organizations use Applicant Tracking Systems (ATS) to filter candidates automatically. This process, while efficient for employers, often results in many well- qualified applicants being rejected simply because their resumes are not optimized for ATS compatibility, suffering from poor formatting or missing keywords. This paper proposes the "Smart Resume Analyzer," an AI- driven web application designed to bridge this gap. The system architecture is designed on three core pillars: (1) A Frontend interface for users to upload a resume and job description (JD); (2) A Flask-based Backend to process the inputs; and (3) an NLP Engine, using spaCy, to analyze the resume, extract key features, and compute an "ATS Score." This score, generated using a semantic similarity algorithm, indicates how effectively a resume matches job-specific requirements*

Keywords- Smart Resume Analyzer, Applicant Tracking System (ATS), Natural Language Processing (NLP), Resume Parsing, spaCy, Flask, Cosine Similarity, Job Description Matching

I. INTRODUCTION

The primary motive for this research is to address the critical inefficiency in the modern recruitment process where skilled candidates are filtered out by automated systems. In today's market, most organizations use Applicant Tracking Systems (ATS) to perform the initial screening of resumes, rather than humans. This results in a "black box" scenario where many wellqualified applicants are

upon the topic of research work but generally comprises up to 5 to 7 pages. These are:multi-label learning, more than one class can be assigned to an instance. With the increase in the number of data

II. IDENTIFY, RESEARCH AND COLLECT IDEA

1)The foundation of this research lies in identifying the need for an intelligent system that assists job seekers in improving their resumes for Applicant Tracking System (ATS) compatibility. The idea was formulated by analyzing recurring challenges faced by applicants, such as resume rejection due to improper formatting, missing keywords, and lack of measurable alignment with job descriptions.

III. WRITE DOWN YOUR STUDIES AND FINDINGS

After identifying the need for an automated and transparent resume evaluation system, the next step involved systematically organizing the research work and developing the Smart Resume Analyzer. The findings were derived through iterative analysis, experimentation, and comparative evaluation of existing techniques. The study was documented using the following structured approaches:

IV. GET PEER REVIEWED

Peer review is a critical stage in validating the quality and accuracy of any research work, including the development of the Smart Resume Analyzer. At this stage, the drafted research paper and the prototype system were evaluated by knowledgeable peers, domain experts, and faculty members to ensure technical correctness, clarity, and overall relevance.

The objective of the peer-review process was to identify potential gaps, improve the system’s methodology, and validate the findings. During this process, several key aspects were reviewed:

1. Technical Accuracy of the NLP Pipeline

Peers assessed whether the NLP techniques used — such as spaCy NER, TF-IDF vectorization, and cosine similarity — were implemented correctly and aligned with industry-accepted practices.

2. Logical Structure and Flow of the Research

Experts examined the arrangement of sections, clarity of explanations, and relevance of literature review to ensure the paper followed a coherent structure consistent with academic publishing standards.

3. Validation of Experimental Results

- The peer review involved checking:
- Whether the test cases were appropriate
- Whether the ATS score was computed correctly

- Whether the missing keyword detection was consistent and meaningful
- Whether the tool performed reliably across different resume formats

4. Usability and Practicality Feedback

Reviewers evaluated the system's frontend for usability, clarity of instructions, and the usefulness of the feedback provided to job seekers. Suggestions were made to refine user experience, improve interface clarity, and enhance the display of ATS results.

5. Suggestions for Improvement

- Various constructive inputs were received, such as:
- Improving text-cleaning functions for better parsing
- Expanding the test dataset for broader validation
- Adding more examples in the result section
- Slightly refining terminology for academic precision

This feedback proved invaluable in strengthening the research. Incorporating these suggestions ensured that the final system was not only technically sound but also practical, accurate, and beneficial for end users.

The peer-review stage thus served as an essential bridge between initial development and final submission, enhancing both the quality of the system and the research documentation.

V. IMPROVEMENT AS PER REVIEWER COMMENTS

Following the peer-review process, the Smart Resume Analyzer system and the research manuscript were carefully evaluated, and several constructive suggestions were received. These comments were analyzed in detail, and appropriate modifications were incorporated to enhance both the technical quality and presentation of the research. The key improvements made based on reviewer feedback are outlined below:

1. Enhancement of NLP Accuracy

Reviewers suggested improving the skill and entity extraction mechanism to handle complex resume formats and varied terminologies.

The text preprocessing pipeline was optimized to remove noise and standardize the input. a more professional presentation, ensuring that both the system and Named Entity Recognition (NER) models in spaCy were fine- the research paper meet the high standards expected in academic tuned for better recognition of technical and soft skills. publication.

2. Refinement of Similarity Scoring

- It was recommended that the similarity computation should be more robust across different job descriptions and resume styles.
- TF-IDF vectorization parameters were adjusted to better capture term importance.
- Cosine similarity computations were refined to ensure consistent ATS score generation.

2. Improved User Interface

- The reviewers highlighted the need for a more user-friendly interface that clearly presents the ATS score and missing keywords.
- The frontend design was updated to include color-coded feedback, clear instructions, and an organized display of analysis results.

3.Comprehensive Testing

- Reviewers requested additional testing on diverse resume formats and multiple job descriptions to validate system performance.
- A wider set of resumes was used for testing, including PDFs with tables, multi-column layouts, and varying content density.

Results confirmed the system’s accuracy, reliability, and usability across all test cases.

4. Manuscript Clarity

- Comments also focused on improving the clarity, structure, and flow of the manuscript.
- Sections were reorganized to match the IJSART template strictly.
- Technical explanations were refined, and illustrative examples were added where necessary.
- Minor language and formatting issues were corrected to improve readability.
- By implementing these improvements, the Smart Resume Analyzer now demonstrates higher accuracy, better usability, and

VI. CONCLUSION

This research successfully presents the design and development of the Smart Resume Analyzer, an AI-driven system intended to bridge the gap between job seekers and modern Applicant Tracking Systems (ATS).

The study demonstrates that by combining lightweight NLP techniques such as spaCy-based entity extraction with TF-IDF vectorization and cosine similarity scoring, it is possible to evaluate resume–job description compatibility with high efficiency and accuracy.

VII. APPENDIX

The appendix provides additional information, supplementary materials, and supporting elements that contribute to a deeper understanding of the Smart Resume Analyzer system but are not included in the main body to maintain clarity and flow.

A. Sample Resume Text Extracted for Testing

Below is an example of a resume text after extraction and cleaning, used during system trials:
Name: John Doe
Skills: Python, Flask, Machine Learning, SQL, API Development
Experience: 3 years as Python Developer
Projects: Inventory Management System, API Automation Framework Education: B.Tech in Computer Science
This dataset was used to verify the accuracy of skill extraction and entity detection.

B. Sample Job Description (JD) Used for Evaluation

Position: Python Developer
Required Skills: Python, Flask, REST APIs, SQL, Git
Responsibilities: Backend development, API integration, debugging Experience Required: 2+ years
This JD was used to compute similarity scores and evaluate the missing keywords module.

C. TF-IDF Vector Sample Output

An example of TF-IDF values generated by the system (simplified):

Term Resume TF-IDF JD TF-IDF

Python	0.41	0.48
Flask	0.32	0.37
SQL	0.22	0.29

This illustrates how the system quantifies text features before similarity computation.

VIII. ACKNOWLEDGMENT

We express our sincere gratitude to Prof. Karade S.A., our project guide, for providing continuous guidance, valuable suggestions, and constructive feedback throughout the development of this research work. Their expertise and encouragement played a vital role in shaping the Smart Resume Analyzer system.

We also thank the Department of Computer Science, BIT Barshi, for providing the necessary resources, laboratory facilities, and a supportive environment that enabled the successful completion of this project.

Finally, we acknowledge the contributions of our peers and everyone who provided technical insights, participated in testing the system, or offered feedback that helped improve the accuracy and usability of the Smart Resume Analyzer..

REFERENCES

[1] Author & B. Author (2022). *A Comparative Study of TF-IDF and BERT for Job-Resume Matching.* Journal of Information Retrieval.
[2] Author (2021). *Resume Parsing and Information Extraction using spaCy.* IEEE NLP Conference.
[3] Author & E. Author (2023). *The Rise of Applicant Tracking Systems and their Effect on Job Seeker Behavior.* IJHCI.
[4] F. Author (2020). *Building Web Applications for Data Science using Flask.* Python Engineering Journal.