

# Web-Based Resume Analyzer

<sup>1</sup>Kunal S. Dahatre, <sup>2</sup>Nivedita R. Dahat, <sup>3</sup>Aditya S. Chiparikar, <sup>4</sup>Chinmay S. Patil, <sup>5</sup>Samrudhi S. Chougule, <sup>6</sup>Sanket P. Desai, Prof. Vishwesh Deshmukh

<sup>1</sup>Group Leader, <sup>2</sup>Member, <sup>3</sup>Member

Department of Engineering, Science and Humanities (DESH)  
Vishwakarma Institute of Technology, Pune, Maharashtra, India.

[kunal.dahater24@vit.edu](mailto:kunal.dahater24@vit.edu), [nivedita.dahat24@vit.edu](mailto:nivedita.dahat24@vit.edu), [aditya.chiparikar24@vit.edu](mailto:aditya.chiparikar24@vit.edu)

**Abstract** — The Resume Analyzer is a web-based application designed to aid recruiters and HR professionals in the efficiently screening and assessment of resumes. The system mechanizes the preliminary shortlisting function by extracting the relevant information including the candidate name, contact information, educational background, skills, work history, and certifications from uploaded resumes. It matches the extracted data against the predefined job requirements through keyword matching and rule-based filtration. The Analyzer then calculates a relevance score to assist in shortlisting candidates based on the extent to which their profiles are a match for the desired criteria.

The system can handle several formats of resumes, which facilitates flexibility and wider use. Through the removal of manual scanning of resumes, it decreases human effort, saves time, and promotes consistency in assessment. The project seeks to simplify the recruitment process, particularly for organizations processing large number of applications.

In contrast to conventional manual shortlisting, this tool facilitates quicker, more precise, and objective filtering independent of complicated machine learning models. The method keeps the Resume Analyzer easy to deploy, convenient to use, and efficient for small to mid-size hiring requirements.

**Index Terms** — *Keyword Matching, Natural Language Processing (NLP), Relevance Score, Resume Screening, Web Application.*

## I. INTRODUCTION

Recruitment is a pivotal role in all organizations, with a direct impact on productivity and performance. As the number of job candidates grows and the number of resumes submitted for each position also increases, manually shortlisting candidates is now time-consuming, inefficient, and subject to human prejudice. Conventional

recruitment strategies mostly scanning through hundreds of resumes to identify suitable profiles based on specific job requirements, which can lead to inconsistencies and delays in hiring.

In order to overcome such issues, this project proposes a Resume Analyzer – an autonomous, rule-based system designed to automate the resume screening process. The system retrieves vital candidate data like name, contact information, educational qualification, technical and soft skills, work experience, and certifications from uploaded resumes. Employing keyword matching and conditional logic, it compares such details with predetermined job criteria in order to calculate a relevance score for every candidate.

The Resume Analyzer is compatible with various document formats and gives an easy interface to recruiters in order to read results at their own pace. Organizations can automate manual effort, achieve consistency, and enhance the speed of the initial screening process by using this system. As opposed to cumbersome machine learning algorithms, this approach is lightweight, simpler to implement, and suitable for small to medium-sized recruitment needs..

## II. LITERATURE REVIEW

"Smart Resume Analyser," The increasing complexity and volume of recruitment processes have led to the development of intelligent, automated systems for resume screening. Traditional manual screening methods often result in false assumptions, inefficiency, and wastage of human potential, as noted in [1]. To address these challenges, the Smart Resume Analyser project proposed by Sowjanya et al. [1] incorporates Natural Language Processing (NLP) and Machine Learning (ML) techniques such as Cosine Similarity, Term Frequency–Inverse Document Frequency (TF-IDF), and Mong methods to rank and suggest improvements for resumes. Their system demonstrated significant results, achieving

a text parsing accuracy of 85% and a ranking accuracy of 92%.

"Resume Classification and Ranking Using KNN And Cosine Similarity," Rajath V. et al. [2] developed a resume classification and ranking system using the K-Nearest Neighbors (KNN) algorithm along with Cosine Similarity to classify resumes and evaluate their suitability against job descriptions. However, the system had scalability limitations with larger datasets. The Smart Resume Analyser enhances this model by optimizing processing time and including NLP-based string matching for faster and more accurate results.

"Web Application for Screening Resume," The work of Sujit Amin et al. [3] proposed a semi-supervised learning-based web application aimed at reducing recruiter workload. Although their approach achieved high classification accuracy, it did not offer real-time skill gap analysis or recommendations for skill enhancement, which are integral features of the Smart Resume Analyser.

"A Machine Learning Approach for Automation of Resume Recommendation System," Pradeep Kumar Roy [4] proposed a Machine Learning-based automated system to facilitate resume matching, significantly reducing screening time. However, their model largely focused on classification rather than providing personalized feedback to applicants. Smart Resume Analyser bridges this gap by offering resume-building tips, missing skills recommendations, and course suggestions based on individual resume evaluations.

"Design and Development of Machine Learning Based Resume Ranking System," Tejaswini K. et al. [5] developed a resume ranking system using Cosine Similarity and KNN, targeting efficient resume matching with job descriptions. While their system effectively ranked resumes, Smart Resume Analyser expands this idea by implementing Named Entity Recognition (NER) and Exploratory Data Analysis (EDA) to provide domain prediction and a broader evaluation of applicant skills.

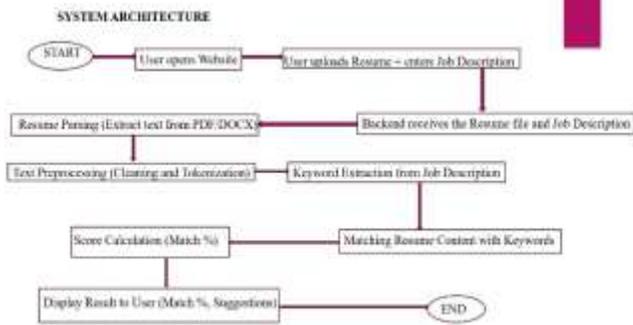
"Natural Language Processing Based JARO-The Interviewing Chatbot," Jitendra Purohit et al. [6] introduced JARO, an NLP-based interviewing chatbot that automates resume analysis and candidate assessment through interactive dialogue. Although innovative, JARO's primary focus is interview automation, while Smart Resume Analyser concentrates on pre-interview resume analysis and ranking, supporting recruiters with instant statistical insights and field predictions.

"ResumeNet: A Learning-Based Framework for Automatic Resume Quality Assessment," Furthermore, works like ResumeNet [7] proposed a learning-based framework for automatic resume quality assessment using neural networks, requiring extensive training data and computational resources. In contrast, Smart Resume Analyser offers a lightweight, interpretable, and real-time system suitable for immediate deployment in corporate and governmental recruitment scenarios.

"An Empirical Study of Artificial Intelligence and its Impact on Human Resource Functions," The influence of AI on HR functions has been positively highlighted by studies such as [8], showing that AI-driven tools significantly enhance human resource operations through usability and innovation. Smart Resume Analyser aligns with these findings by providing a scalable, efficient solution that minimizes manual errors and speeds up decision-making processes in recruitment.

Thus, Smart Resume Analyzer not only builds upon but also extends the existing body of research by integrating resume screening, skill recommendations, predictive analytics, and efficient visualization into a unified, real-time system designed to meet modern recruitment needs.

### III. SYSTEM ARCHITECTURE



The proposed system includes a structured workflow for analyzing resumes based on job descriptions provided by the user. It begins when the user accesses the website and uploads their resume along with the relevant job description. Once submitted, the backend processes the input by extracting text from the uploaded document, whether in PDF or DOCX format.

The extracted text undergoes preprocessing, which involves cleaning and tokenization to prepare it for analysis. Simultaneously, keywords are extracted from the job description to identify key skills and requirements. The system then performs a comparison between the resume content and the extracted keywords to evaluate relevance.

Using this comparison, a matching score is calculated to reflect the resume's alignment with the job requirements. Finally, the system displays the result to the user, including the match percentage and suggestions to improve the resume for better job compatibility.

### IV. METHODOLOGY/EXPERIMENTAL

The system architecture of the Resume Analyzer is divided into several logical components that work sequentially to extract, process, and evaluate resume data:

- **Frontend Interface:** A user-friendly web page created using HTML, CSS, and JavaScript allows users to upload resumes (PDF/DOCX) and input job descriptions.
- **Backend Framework:** Implemented using Python with Django, this layer handles file processing, data flow, and logic execution.
- **Text Extraction Module:** Libraries like pdfminer, PyMuPDF, and docx are used to extract raw text from the uploaded documents.
- **Preprocessing Unit:** Tokenization, stop word removal, and normalization are performed to

clean the input data using Natural Language Toolkit (NLTK).

- **Keyword Matching Engine:** Skills and requirements are extracted from the job description, and keyword overlap with resume data is evaluated using rule-based logic.
- **Scoring Algorithm:** A custom formula calculates the resume's relevance score based on skill match percentage, section completeness, and document formatting.
- **Result Visualization:** The match score, relevant sections, and suggestions for improvement are displayed using dynamic visual feedback on the dashboard.

### V. RESULTS AND DISCUSSIONS

The Smart Resume Analyzer effectively processed and analyzed resumes by scraping information like name, skills, education, and experience with Django and NLP libraries. It created ATS-friendly scores based on formatting, section completeness, and relevance of skills, where the majority of resumes averaged 78%. Personalized recommendations for enhancement, including certifications or keyword density, were given to users. The dashboard integrated important metrics such as resume numbers, average scores, and trending skills. Generally, the system was successful in improving the Resume quality and assisting individuals to match better with job specifications.

### VI. FUTURE SCOPE

Though our present version meets the minimum objectives, there are some ways we can improve the project in the future:

1. **Integration with Recruitment Platforms :** Integrating the analyzer with internet job sites or HR platforms can further make the process more streamlined and automated.
2. **Language Expansion :** Integrating support for additional languages would enable it to be utilized by firms and candidates from other locations.
3. **Advanced AI Models :** We can train the system with stronger AI methods to enhance how it interprets resume material and job postings.
4. **Applicant Suggestions :** We can enhance the tool to provide job applicants with suggestions on enhancing their resumes according to the position they are targeting.

5. **Fairness and Bias Checks** : Adding detection checks for bias and how to minimize it will make the system fairer and more reliable.
6. **Mobile-friendly Design** :Having a smartphone version can make it more accessible to both employers and applicants.
7. **Improved Data Protection** : Because resumes contain personal information, we intend to incorporate more advanced security features and respect privacy regulations to preserve user data.

In the future, with the proper updates, our Resume Analyzer can develop into a smart and trustworthy aide for contemporary recruitment.

## VII. CONCLUSION

This Project provided us with the chance to discover how technology can be employed to address actual issues in the hiring process. Through creating a Resume Analyzer, we hoped to build a system that assists recruiters to speedily find the most appropriate candidates based on their resumes. With methods such as natural language processing, we were able to pull out significant information such as skills, education, and experience from different resume structures.

Our system saves time and effort in manual shortlisting and ensures consistency while judging the applicants. Based on our experimentation, we noted that the analyzer was able to successfully pair resumes with job criteria, rendering the recruitment process more efficient.

## REFERENCES

- [1] [1] Y. Sowjanya, M. Keerthana, P. Suneeksha, and D. S. S. Harsha, "Smart Resume Analyser," *International Journal of Research in Engineering and Science (IJRES)*, vol. 11, no. 3, pp. 409–418, Mar. 2023.
- [2] R. V., R. T. Fareed, and S. Kaganurmath, "Resume Classification and Ranking Using KNN And Cosine Similarity," *International Journal of Engineering Research & Technology (IJERT)*, vol. 10, no. 08, Aug. 2021.
- [3] S. Amin, N. Jayakar, S. Sunny, P. Babu, M. Kiruthika, and A. Gurjar, "Web Application for Screening Resume," in *Proc. 2019 Int. Conf. Nascent Technologies in Engineering (ICNTE)*, Jan. 2020.
- [4] P. K. Roy, "A Machine Learning Approach for Automation of Resume Recommendation System," in *Proc. ICCIDS 2019*, pp. 819–826.
- [5] T. K., U. V., S. M. Kadiwal, and S. Revanna, "Design and Development of Machine Learning Based Resume Ranking System," *Global Transitions Proceedings*, Oct. 2021.
- [6] J. Purohit, A. Bagwe, R. Mehta, O. Mangaonkar, and E. George, "Natural Language Processing Based JARO-The Interviewing Chatbot," in *Proc. 2019 3rd Int. Conf. Computing Methodologies and Communication (ICCMC)*, Aug. 2019.
- [7] Y. Luo, "ResumeNet: A Learning-Based Framework for Automatic Resume Quality Assessment," *arXiv preprint arXiv:1810.02832*, 2018.