International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

WWW.IJRPR.COM

AUTOMATED RESUME ANALYZER USING AI

Dhivya K , Mukesh Kumar R, Naveen Kumar P , Santhosh S, Surendiran MG

Computer Science and Engineering (Cyber Security), Sri Shakthi Institue of Engineering and Technology, Tamil Nadu, India.

ABSTRACT:

the conventional hiring process typically requires manually screening massive amounts of resumes, which is time-consuming and susceptible to prejudice. In an effort to combat these problems, this project recommends an AI-based resume screening system that can automate the shortlisting of candidates depending on job-related criteria like job title, number of years of experience, and needed certifications. The system is constructed with the Flask web framework for frontend and backend integration, enabling recruiters to upload multiple resumes and obtain structured, real-time analysis results. Natural Language Processing (NLP) methods are utilized to extract principal features from resume text, including skills, experience, and certifications, which are then assessed by a rule-based or machine learning scoring algorithm. The system sorts the resumes based on their level of relevance to the given job description, enhancing the efficiency and accuracy of the recruitment process. This methodology not only minimizes human workload and screening time but also maximizes objectivity in candidate selection. The modular architecture enables future integration with applicant tracking systems and ongoing model refinement by recruiter feedback.

Keywords:Resume Screening,Artificial Intelligence (AI),Machine Learning,Natural Language Processing (NLP),Flask Web Framework,Automated Recruitment,Skill Extraction,Candidate Ranking,Human Resource Technology (HRTech),Document Analysis,

1. INTRODUCTION

With the current competitive labor market, companies tend to get hundreds or thousands of resumes for a single position. The traditional process of checking every resume is not only time-consuming but also prone to human error and subconscious bias. This is why automated systems that can make the recruitment process faster and more efficient are becoming increasingly important.Artificial Intelligence (AI), especially when coupled with Natural Language Processing (NLP) and machine learning methods, has proven to be an effective solution in tackling this issue. Such technologies allow systems to scan, understand, and extract relevant information from resumes, including work history, educational qualification, and skill sets. By matching the information with pre-defined job specifications, AI systems are able to shortlist candidates efficiently.This project proposes an artificial intelligence-based resume filtering system that was created through the use of the Flask web development framework. Recruiters can upload various resumes and set principal job specifications such as preferred job title, number of years of experience, and required certification. Every resume undergoes analysis in a backend function where the resume's similarity to the job requirements is scored based on feature extraction and scoring methods. The outcomes are subsequently presented in an easy-to-view format to help recruiters make informed hiring decisions.By automating screening, the system not only saves time and labor but also enhances the fairness and consistency of candidate assessment. It is a pragmatic move toward human resource modernization through the implementation of smart technologies.

2. LITERATURE SURVEY

As the number of job applications rises, companies have to deal with the issue of screening resumes effectively and equally. Manual screening is time-consuming, tends to be inconsistent, and prone to unconscious bias. To overcome such issues, artificial intelligence has been the game-changer in recruitment, particularly in automating the resume screening process.Natural Language Processing (NLP) is key to allowing machines to read and understand resumes. By applying methods like keyword extraction, named entity recognition, and part-of-speech tagging, systems can detect key features such as candidate skills, work history, education level, and certifications. The features extracted are later utilized to match candidates with set job specifications.

Machine learning algorithms refine the process further by learning from past hiring data and making predictions of candidate fit. Decision trees, support vector machines, and random forests are algorithms that can categorize resumes into relevance for the job. Even more sophisticated systems incorporate semantic analysis and embedding models to capture meaning and context in resumes, enhancing accuracy in matching candidates to job postings.

New developments have seen deep learning models such as BERT, which offer even better contextual awareness of resume information. These models enable systems to better understand the finer points of language, allowing more precise candidate assessment.Furthermore, contemporary resume screening software is becoming more ethical-oriented. To ensure trust, developers are adding transparency mechanisms and fairness auditing to prevent the perpetuation of past biases. The development of AI-driven resume screening systems is a response to an emerging necessity for intelligent, scalable, and impartial recruitment tools that can flexibly respond to changing hiring conditions.

3. ARCHITECTURAL METHODOLOGY

The proposed AI-based resume screening system follows a modular architecture that integrates web-based interfaces with backend AI processing components. The overall workflow is designed to ensure scalability, accuracy, and usability for recruiters.

3.1.User Interface (Frontend Layer):

The system provides a web-based interface built using Flask. Users (recruiters) can upload multiple resumes in PDF or DOCX format and enter job-specific criteria such as job title, years of experience, and required certifications. This interface is rendered using HTML templates and handles form submissions and file uploads.

3.2. *Resume Upload & Preprocessing:*

Uploaded resumes are saved securely on server. the Each document is parsed using a resume parser module, which extracts structured data including name, contact details, skills, experience, and education. Preprocessing steps include text normalization, tokenization, and stop-word removal

3.3. Session Management Layer:

User inputs such as job title, experience, and resume paths are temporarily stored in Flask sessions. This enables smooth data transfer across different pages and request cycles without re-entry.

3.4. Resume Analysis Engine:

Each resume is analyzed using a function (e.g., analyze_resume) that likely applies NLP techniques like keyword matching, tokenization, and entity recognition to extract relevant candidate features.

3.5. *Matching and Scoring Engine:*

The system compares extracted resume features against the input job criteria. It calculates a similarity score using techniques such as cosine similarity or ML models trained on past hiring data. Each resume is scored and ranked accordingly.

3.6. Result Rendering Layer:

After processing, results are displayed on a separate page showing analysis summaries and scores. This helps recruiters quickly evaluate top candidates in a structured, visual format.

3.7. Confirmation Workflow:

A verification page is included before processing begins. Users can confirm job details and uploaded files, reducing errors and ensuring the analysis is based on correct information. The system is modular and can be extended with ML models, cloud deployment, HR dashboards, or integration with Applicant Tracking Systems (ATS) for enhanced enterprise use.

4. CONCLUSION AND FUTURE ENHANCEMENT

The proposed AI-based resume screening system offers an efficient, user-friendly solution for automating the candidate shortlisting process. By leveraging natural language processing and structured form inputs, it enables recruiters to analyze multiple resumes based on job-specific criteria such as experience, certifications, and job title. The integration of a confirmation step adds an extra layer of validation, ensuring accurate data input before analysis. This system significantly reduces manual screening time, increases consistency in evaluation, and enhances decision-making accuracy. Its modular design allows for ease of use and adaptability, making it suitable for small to mid-sized recruitment workflows.

Future Enhancements

To further improve the system's performance and scalability, the following enhancements can be considered:

1. **Machine Learning Integration: Incorporate ML models trained on historical hiring data to predict candidate-job fit more accurately.**
2. **Cloud Deployment: Host the application on cloud platforms (e.g., AWS, Azure) to support scalability and remote access for large organizations.**
3. **Multilingual Resume Support: Add OCR and translation tools to handle resumes in different languages and formats.**
4. **Advanced Scoring Algorithms: Implement dynamic weighting of resume sections (e.g., skills vs. experience) based on role type or recruiter preference.**
5. **Feedback Loop: Enable recruiter feedback to refine scoring models through supervised learning.**

6. **Analytics Dashboard: Develop a dashboard for visualizing candidate data, filtering, and exporting reports for HR teams.**
7. **Integration with ATS: Allow API-based integration with Applicant Tracking Systems for seamless recruitment workflows.**
8. **Email Notifications: Add automatic email alerts to notify shortlisted candidates or recruiters after analysis.**

REFERENCES

1. Kashif, M. & Kumar, P. (2024). *Resume Parser Using NLP*. Int'l Journal of Adv. Research in Comp. & Comm. Eng., 13(9).
2. Zinjad, S.B. *et al.* (2024). *ResumeFlow: An LLM-facilitated Pipeline for Personalized Resume Generation and Refinement*. (ArXiv preprint).
3. Sripriya, S. *et al.* (2024). *Resume Analysis and Suggestion System Using NLP*. Int. Research Journal of Modernization in Engg. Tech. & Science, 6(5).
4. Pokharel, P. (2022). *Hireability: Information Extraction from Resume* (M.Sc. Thesis).
5. Daryani, C. *et al.* (2020). *Automated Resume Screening System Using NLP and Similarity*. Topics in Intelligent Computing and Industry Design, 2(2), pp. 99-103.
6. Wankhede, D. *et al.* (2024). *Resume Analyser: Analysing and Filtering System*. J. Emerging Tech. & Innovative Research, 11(7).
7. Pokala, N. (2025). *Implementing Resume Analysis with Google's Gemini AI Model*. *Medium* blog.
8. Insight-Services-APAC (2023). *CV Analysis Tool*. GitHub repository.