International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Resume Analysis System Using Natural Language Processing

Premchand Pawade, Pranav Sukale, Shivprasad Swami , Ahefaz Khan Prof. Shivani Karhale
erprempawade@gmail.com, sukalepranav@gmail.com, swamishivprasad9495@gmail.com, Khanahefaz01@gmail.com

Abstract

This project develops a resume analysis system using Natural Language Processing (NLP) to streamline hiring. Companies often receive large volumes of resumes, making it challenging to identify the best candidates quickly. Manually screening resumes is time-consuming, can be inconsistent, and may overlook key details. By automating this process with NLP, our system reads and evaluates resumes efficiently. It extracts key information like skills, experience, and education to match candidates to job requirements. For example, if a job requires a specific skill, the system can highlight candidates with that qualification.

NLP algorithms can recognize relevant keywords, synonyms, and context, even if the phrasing varies across resumes. This means that someone who writes "managed team projects" or "project lead" can be identified as having similar experience. The system uses advanced NLP models, such as BERT or GPT, to capture subtle language details and improve accuracy. As it processes more resumes, the system "learns" and becomes better at identifying relevant skills and qualifications. Customization allows it to adapt to different industries by prioritizing specific keywords and competencies.

This adaptability makes the system valuable across fields like technology, finance, and healthcare. By focusing on qualifications objectively, it helps reduce bias in resume review. Recruiters save time and can focus on candidates who meet job requirements more closely. The system accelerates hiring, improves candidate-job matching, and supports data-driven decisions.

Keywords

Resume Analysis, Natural Language Processing (NLP), Recruitment Automation, Text Analysis, Skill Extraction, Information Retrieval, Automated Resume Parsing, Unstructured Data Processing.

Introduction

In today's competitive job market, companies receive a large volume of resumes for every job opening, making it difficult and time-consuming for recruiters to manually screen each application. Traditional methods of reviewing resumes are often inefficient, prone to human error, and can introduce biases that affect hiring decisions. This project proposes a solution: a resume analysis system that uses Natural Language Processing (NLP) to automate and optimize the recruitment process.

NLP enables computers to process and analyze vast amounts of unstructured text, like resumes, by identifying key information such as skills, work experience, and qualifications. Using machine learning models, especially deep learning models like BERT, the system can interpret the language of resumes, recognize synonyms and related terms, and evaluate candidates based on relevance to job descriptions. This automated approach not only accelerates the screening process but also improves the accuracy and consistency of candidate evaluation by applying objective criteria.

Additionally, the system can be customized to prioritize certain skills or qualifications for specific industries, making it versatile for various fields such as technology, healthcare, and finance. By removing human biases and focusing purely on qualifications, the resume analysis system offers a fairer and more data-driven approach to hiring.

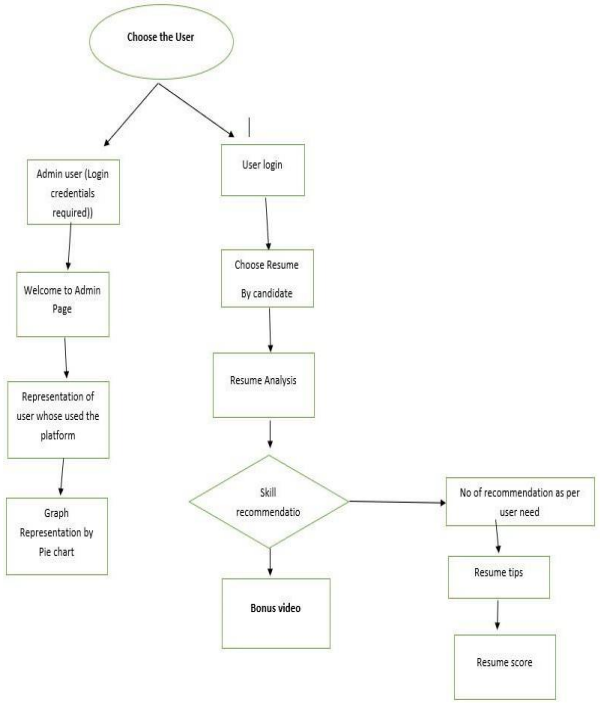
Literature Review

Resume analysis and recruitment automation have seen increasing interest as businesses seek faster, more accurate, and objective ways to handle large applicant pools. Traditional resume screening methods often rely on manual review, which is time-intensive, costly, and subject to human biases. Studies have shown that such methods can overlook qualified candidates and lead to inconsistencies in hiring decisions (Garcia-Izquierdo et al., 2010). As a result, the demand for automated systems, particularly those utilizing Natural Language Processing (NLP), has grown significantly.

Natural Language Processing (NLP) enables computers to interpret and analyze human language, making it a valuable tool for processing unstructured data, such as resumes. Early efforts in automated resume screening focused on rule-based systems, which involved keyword matching and limited contextual understanding (Chaplot et al., 2015). While effective to a degree, these systems struggled with variations in language, terminology, and candidate phrasing. Modern NLP, however, uses machine learning techniques like deep learning models, such as BERT and GPT, that allow for more nuanced language understanding by identifying synonyms, contextual meanings, and relationships between words (Devlin et al., 2019).

Research has shown that NLP-based resume analysis can extract essential information from resumes, such as skills, education, experience, and certifications, more accurately than traditional keyword-based approaches (Agarwal et al., 2020). Studies by Wang et al. (2018) demonstrated the effectiveness of NLP models in capturing contextual information, which is crucial for accurate candidate evaluation, particularly when applicants describe similar skills in different ways. Moreover, NLP-based systems have proven effective in reducing bias, as algorithms are designed to focus solely on qualifications, skills, and experience rather than subjective elements (Garg et al., 2019).

System Architecture



Software Requirement :

- Programming Language :Python, Html5, CSS3, Javascript
- Database : phpMyAdmin , MySQL
- Server Used : XAMPP

Hardware Requirement :

- Developing machine (Laptop/Computer)
- Intel core i5 11th generation
- RAM 8GB

Login Module : We worked On two Login Models

• Admin Login Module

• User Module

Admin Module : Admin module works on Selection of candidates resumes . Admin module requires the Login Credentials for login . Admin Module can Enable Admin Dashboard Where Admin has access to see candidates resume with job profile with their Respective Skill sets etc..

User Module : User Module Does not Need Any type of Login credential .User has access to examine the CV.

• User gets Evaluation Score Based On his resume Content.

• User gets the recommendation for improvement in resume by RAS.

• User is encouraged for his good qualities for mentioning in CV.

• User is provided Training module for CV as well as Training for the Interviews.

Conclusion

The Resume Analysis System using Natural Language Processing (NLP) automates and streamlines the recruitment process by accurately extracting and analyzing key resume information. It improves candidate screening, reduces bias, and ranks candidates based on relevance to job descriptions. With continuous learning from feedback, the system enhances its performance over time, saving recruiters time and supporting data-driven, fair hiring decisions.

References

1. **Garcia-Izquierdo, A., & Sáez, P. (2010) :** "Automatic Resume Screening Using Data Mining Techniques." *International Journal of Computer Applications in Technology*, 39(2), 159-167.

2. **Chaplot, D., & Vaidya, S. (2015) :** "Automated Resume Screening System Using Natural Language Processing." *International Journal of Computer Science and Information Technologies*, 6(5), 4537-4540.

3. **Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019) :** "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *arXiv preprint arXiv:1810.04805*.

4. **Agarwal, P., & Kumar, P. (2020) :** "Resume Parsing Using Machine Learning Algorithms: A Review." *International Journal of Computer Applications*, 176(6), 6-10.

5. **Wang, H., & Yang, B. (2018) :** "Application of Natural Language Processing in Resume Screening." *International Journal of Advanced Computer Science and Applications*, 9(4), 211-216.

6. **Garg, R., & Patel, M. (2019) :** "Improving Recruitment System using NLP and Deep Learning." *International Journal of Advanced Research in Computer Science*, 10(5), 52-58.