Resume Screening Using Machine Learning

[1.]Duddupudi Lasya Sri Priya
Department of Computer Science in AIML Sasi Institute of technology and Engineering
lasya.duddupudi@sasi.ac.in

[3.] Dodla Poorna Satya Sri
Department of Computer Science in AIML Sasi Institute of technology and Engineering
satyasri.dodla@sasi.ac.in

[2.] Adusumilli Kavya
Department of Computer Science in AIML Sasi Institute of technology and Engineering
kavya.adusumilli@sasi.ac.in

[4.] Thotakura Satya Sri
Department of Computer Science in AIML Sasi Institute of technology and Engineering
satyasri.thotakura@sasi.ac.in

K. Suresh
Assistant Professor, CSE-AIML
Institution: Sasi Institute of Technology and Engineering

Abstract

Due to the increasing number of job applications for a specific position, the recruitment process is becoming more challenging for organizations. Manual resume screening is a traditional process in which recruiters have to screen a large number of resumes for a specific position. This is a time-consuming and inefficient process, and sometimes human errors are involved in this process. To overcome such problems, this project proposes a system for automated resume screening using Natural Language Processing (NLP) and Machine Learning (ML) techniques. Important information is obtained from resumes using NLP techniques and compared with the job requirements to find the most suitable candidate for a specific position. Various text preprocessing techniques are used for resume data. Semantic embeddings are obtained using Sentence-BERT (SBERT) for candidate selection. Cosine similarity is used to find the relevance between resumes and job requirements. Based on the results obtained from this process, candidates are ranked, and a list of relevant candidates is obtained. This system is implemented using Python for efficient resume analysis. From the experimental results, it is clear that this system is more efficient in resume screening..

KEYWORDS: Resume Screening, Natural Language Processing, Machine Learning, SBERT, Cosine Similarity, Recruitment Automation.

Introduction

Recruitment is an essential function of every firm to hire appropriate candidates. The traditional recruitment process of screening resumes is challenging and time-consuming when there are more resumes to be screened for the available positions. It might also cause human error in the selection process.Recent advances in Natural Language Processing (NLP) and Machine Learning (ML) techniques are capable of efficiently processing resume screening data. In this project, an automated resume screening system is designed to efficiently screen resumes and match the requirements of the jobs.

1. Literature Survey

1. **"Automated Resume Screening System using NLP and ML Techniques" (2025):**
This study proposes an automated resume screening system that uses Natural Language Processing (NLP) and Machine Learning (ML). The system extracts key information such as skills, education, and experience from different resume formats. Machine learning algorithms are then applied to rank candidates based on their suitability for a specific job. The approach reduces manual effort and helps minimize bias in recruitment. Experimental results indicate improvements in both screening accuracy and processing speed.
2. **"Design and Development of Machine Learning Based Resume Ranking System" (2022):**
This research introduces a machine learning–based resume ranking system designed to automate candidate

shortlisting. Resumes and job descriptions are converted into numerical vectors using the TF-IDF technique. Cosine similarity and the K-Nearest Neighbors (KNN) algorithm are used to compare and rank candidates. The system can efficiently handle a large number of resumes and shows better ranking accuracy compared to manual screening.

3. "Resume **Screeningusing Machine Learning"(IJSRCSEIT,2024)**: This paper focuses on automating resume screening using NLP and machine learning algorithms. Techniques such as Named Entity Recognition (NER) and Part-of-Speech (POS) tagging are used for feature extraction. Classification algorithms including KNN and Support Vector Machine (SVM) are applied to categorize resumes. Cosine similarity is also used to provide recommendations and suggestions for resume improvement. The system reduces screening time and improves consistency in candidate selection.

4. **"Real-Time Resume Screening using NLP and Token-Based Indexing" (2023):**
This research presents a real-time resume screening system built with Natural Language Processing methods. The system compares resumes with job descriptions using TF-IDF and cosine similarity. It ranks candidates instantly based on relevance. The proposed approach is lightweight and suitable for organizations dealing with a large number of applications, improving both screening speed and operational efficiency.

5. "Resume Screening with Natural Language Processing" (2024):
This study proposes an NLP-based resume screening system that focuses on fair candidate evaluation. The system extracts relevant skills and competencies from resumes and compares them with predefined job requirements. Jaccard similarity is used to rank candidates according to their suitability. By concentrating only on competencies, the system helps reduce bias and improves the fairness of the recruitment process.

6. **"Resume Screening and Recommendation System Using Machine Learning" (2022):** This research introduces a resume screening and job recommendation system using NLP and machine learning techniques. The system extracts information such as skills, experience, and education from resumes of different formats. Machine learning models predict appropriate job roles for candidates and also recommend alternative positions based on their skill sets. This approach enhances recruitment efficiency and reduces manual intervention.

7. "AI-Powered Resume Screening System" (2025):
This paper presents an AI-based framework designed to automate the resume screening process. Natural Language Processing techniques are used for resume parsing and information extraction. Candidate ranking is performed through similarity-based matching methods. The system decreases human involvement in recruitment and improves the speed of hiring decisions. Experimental results demonstrate higher accuracy compared to traditional screening methods.

8. "Resume Analyzer Using AI Techniques" (2024):
This study highlights the importance of personalized resume evaluation. It introduces an AI-based resume analyzer that examines both the content and structure of resumes. NLP and machine learning techniques are used to analyze resume quality. The system provides feedback and improvement suggestions to candidates, helping them enhance resume quality and compatibility with Applicant Tracking Systems (ATS).

9. **Resume Analyser: Automated Resume Ranking Software":**
This paper proposes an automated resume ranking system to address the inefficiencies of manual resume screening. NLP techniques are used to analyze resumes in different formats and extract important candidate information. Machine learning methods then rank the resumes according to recruiter requirements. The study demonstrates that automated resume ranking performs better than traditional keyword-based screening.

10. **"Resume Screening and Recommendation System Using Machine Learning Approaches":** This research presents a resume screening and recommendation system developed using machine learning and NLP techniques. The system analyzes unstructured resumes to extract skills,qualifications and experience. Classification algorithms are applied to screen candidates efficiently. In addition, the system recommends suitable job roles based on the candidate's skills, improving recruitment efficiency.

2.1Comparison table for Existing Resume Screening System

Author & Year	Model / Architecture	Practical Implementation	Results	Limitations
Automated Resume Screening System (2025)	NLP + ML	Extracts candidate detils and ranks resumes	Faster screening	Depends on data quality
	TF-IDF + KNN + Cosine Similarity	Converts resumes into vectors for ranking	Improved accuracy	Keyword dependency
ML Based Resume Ranking System (2022)	TF-IDF + KNN + Cosine Similarity	Converts resumes into vectors for ranking	Better screening	Model complexity
Resume Screening Using ML (2024)	NLP + KNN + SVM	Uses NLP for feature extraction	Better screening accuracy	Limited semantic analysis
Real-Time Resume Screening Using NLP (2023)	NLP + TF-IDF	Compares resumes with job descriptions	Faster ranking	Limited semantic analysis
	NLP + Jaccard Similarity	Matches skills with job requirements	Reduced bias	Limited context understanding
Resume Screening with NLP (2024)	NLP + ML	Screens resumes and recommends roles	Improved recruitment	Data dependency
AI Powered Resume Screening (2025)	AI + NLP	Parses resumes and ranks candidates	Faster hiring process	Requires large data
Resume Analyzer Using AI	NLP + ML	Parses resumes and ranks candidates	Faster hiring process	Requires large data
Resume Analyzer: Using AI	NLP + ML classifiers	Analyzes resumes and ranks resumes	Improved resume quality	Limited job matching
Resume Analyzer: Automated Ranking	NLP + ML classifiers	Extracts information and ranks resumes	Better than manual screening	May miss context
Resume Screening	NLP + ML classifiers	Screens resumes and	Efficient recruitment	Feature dependency

2. Analysis of Datasets

The performance of the resume screening system can be evaluated by utilizing the resume dataset, which comprises candidate resumes belonging to different professional domains. The resume dataset comprises candidate resumes that are filled with details like skills, educational qualifications, work experience, and technical qualifications. The resume dataset comprises resumes collected from different sources, which are stored in different formats like images, PDF, and documents. Utilizing the resume dataset with labeled information can be useful for the system to analyze candidate profiles and match them with job requirements accordingly.
Unlike traditional recruitment processes, which rely on manual resume screening, utilizing the resume dataset can be useful for the system to analyze candidate resumes efficiently. The resume dataset comprises resumes with different structures, writing styles, and skill sets, which can be useful for the system to analyze different resume formats and derive meaningful information from candidate profiles accordingly.

In this project, the Resume Images Dataset available on Kaggle has been utilized for training and evaluating the resume screening system. The dataset comprises a wide range of resumes belonging to different job categories. Before utilizing the resume dataset for analysis, text extraction, text cleaning,and normalization operations are performed to convert resume information into structured text format accordingly. After completing the preprocessing operations, semantic embeddings are generated using the SBERT library, which can be compared with job descriptions to rank candidates based on similarity.

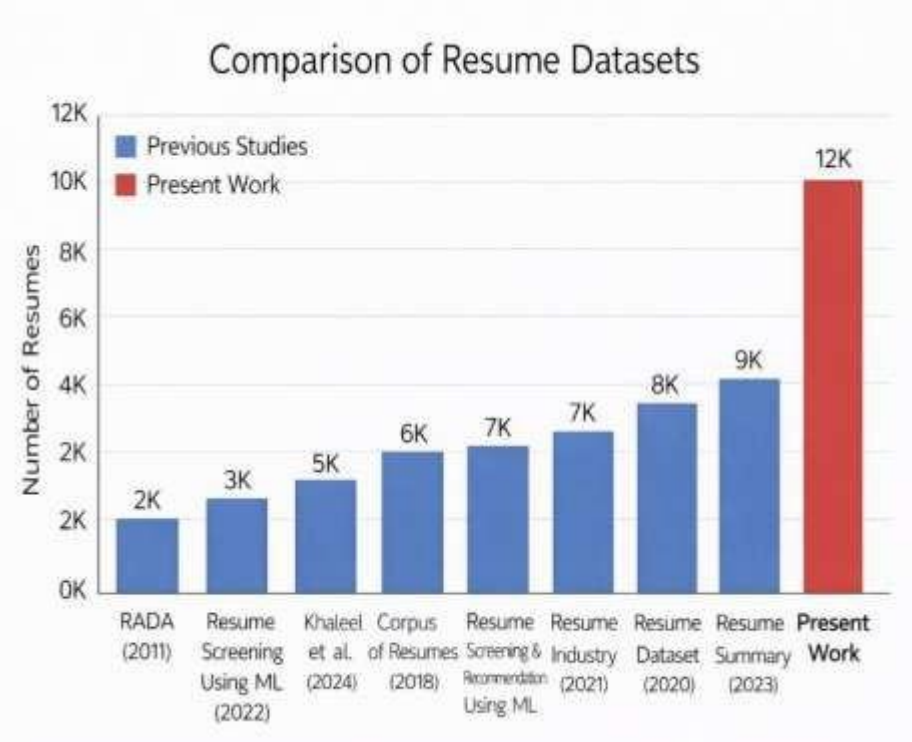


Figure 1: Dataset comparison for Resume Screening System

.

3. Methodology of Proposed System

The system utilizes Natural Language Processing and Machine Learning techniques to automatically analyze the resumes and find the best candidate for the particular job role. The major aim of the system is to reduce the manual effort in the recruitment process and make the process efficient in shortlisting the candidates.

The process begins with the collection of the dataset. The dataset consists of resumes gathered from different job fields. Resumes are available in different formats such as images, PDF, and documents. Each resume includes different information such as skills, qualifications and experience.

The dataset is then divided into training and testing sets in the ratio of 80:20. Eighty percent of the resumes are used to train the system, while twenty percent of the resumes are used to test the system.

Next, the system performs the preprocessing step to analyze the resumes. In this step, text information from the resumes is extracted and cleaned by eliminating unwanted characters, stop words, and formatting issues. This step ensures that the system can process the data in a uniform manner.

After preprocessing the resumes, the system utilizes Sentence-BERT (SBERT) to generate semantic embeddings. Semantic embeddings capture the contextual information present in the text. The system then compares the similarity between the resume embeddings and the job description embeddings using cosine similarity. This similarity helps the system rank the resumes based on their similarity scores.

Finally, the system ranks the resumes and finds the best candidate based on their similarity scores. A web interface is created to show the results to the user. The web interface is built using the Flask framework in Python. From the experimental results, the system can efficiently analyze the resumes and shortlist the best candidate.

4.1 System Architecture

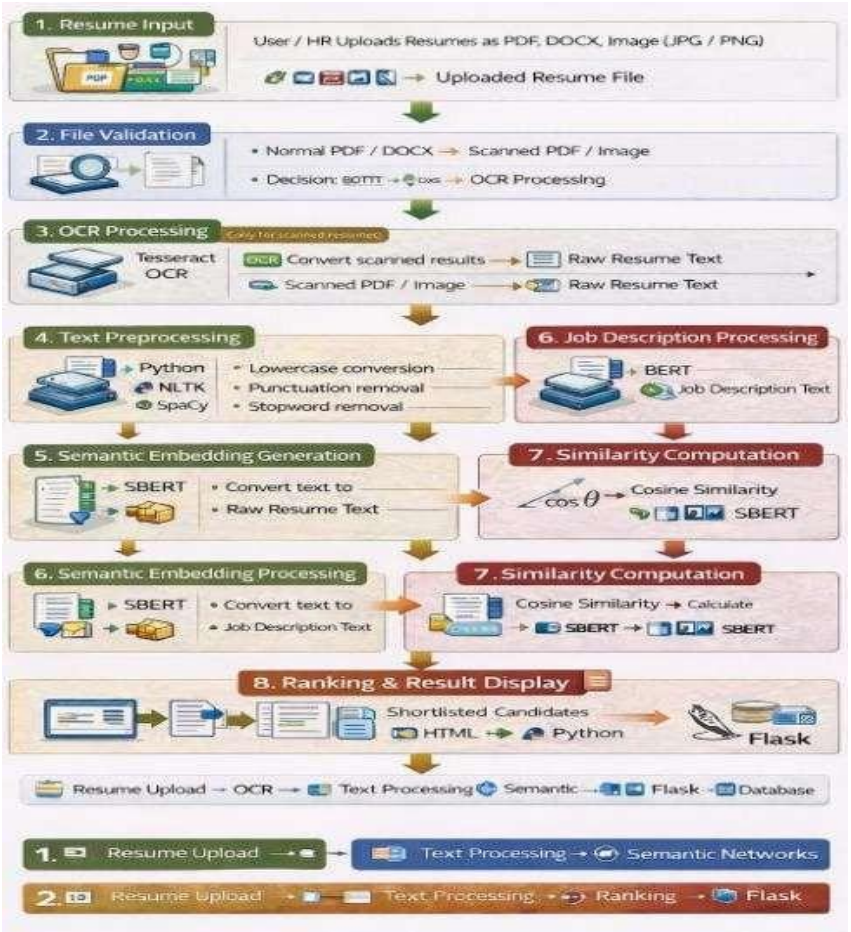


Figure 2: Architecture of the Resume Screening System

The figure shows the system architecture for the Resume Screening System using Machine Learning and Natural Language Processing. The system processes resumes and compares them with job descriptions to identify the most suitable candidates. The architecture includes several modules such as resume input, file validation, OCR processing, text preprocessing, semantic embedding generation using the SBERT model, similarity computation, and ranking with result display. The system analyzes the content of each resume and automatically determines how closely it matches the requirements of a job description.

The process begins with the resume input module, where users or recruiters upload resumes in formats such as PDF, DOCX, or image files (JPG/PNG). These resumes are sent to the file validation module, which checks the document type. If the resume is a normal text-based document, the system directly extracts the text. If the resume is a scanned document or image, the system performs OCR (Optical Character Recognition) using tools like Tesseract OCR to convert the scanned content into machine- readable text.

After text extraction, the system sends the data to the text preprocessing module, where Natural Language Processing techniques are applied. This stage includes steps such as lowercase conversion, punctuation removal, stop word removal, and text cleaning. These preprocessing steps help standardize the text and improve the quality of the data before further analysis.

Next, the system uses the SBERT (Sentence-BERT) model to perform semantic embedding generation. The SBERT model converts both the resume text and the job description text into numerical vector representations, known as embeddings. These embeddings capture the contextual meaning of the text, allowing the system to understand the relationship between candidate skills and job requirements.

After generating embeddings, the system performs similarity computation using Cosine Similarity. This module compares the resume embeddings with the job description embeddings and calculates a similarity score for each resume. The similarity score indicates how closely a resume matches the job requirements. Higher similarity scores represent stronger matches.

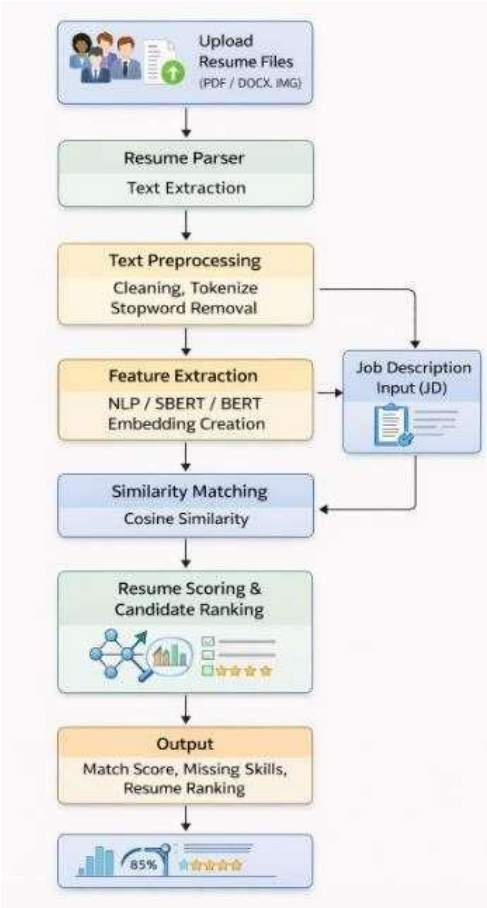


Figure 3: Workflow of the Resume Screening System

The system automates candidate evaluation by using Tesseract OCR to extract text from various resume formats and NLTK/SpaCy for linguistic preprocessing. It leverages an ML-based SBERT model to generate semantic embeddings, which are then compared against job descriptions using Cosine Similarity to produce a ranked shortlist of candidates.
Finally, the system moves to the ranking and result display module. In this stage, resumes are ranked based on their similarity scores, and the top matching candidates are shortlisted. The final results are displayed through a web interface developed using the Flask framework, where recruiters can view the ranked resumes, similarity scores, and shortlisted candidates. This automated process reduces manual effort, improves recruitment efficiency, and helps organizations quickly identify the most suitable candidates for a job position.

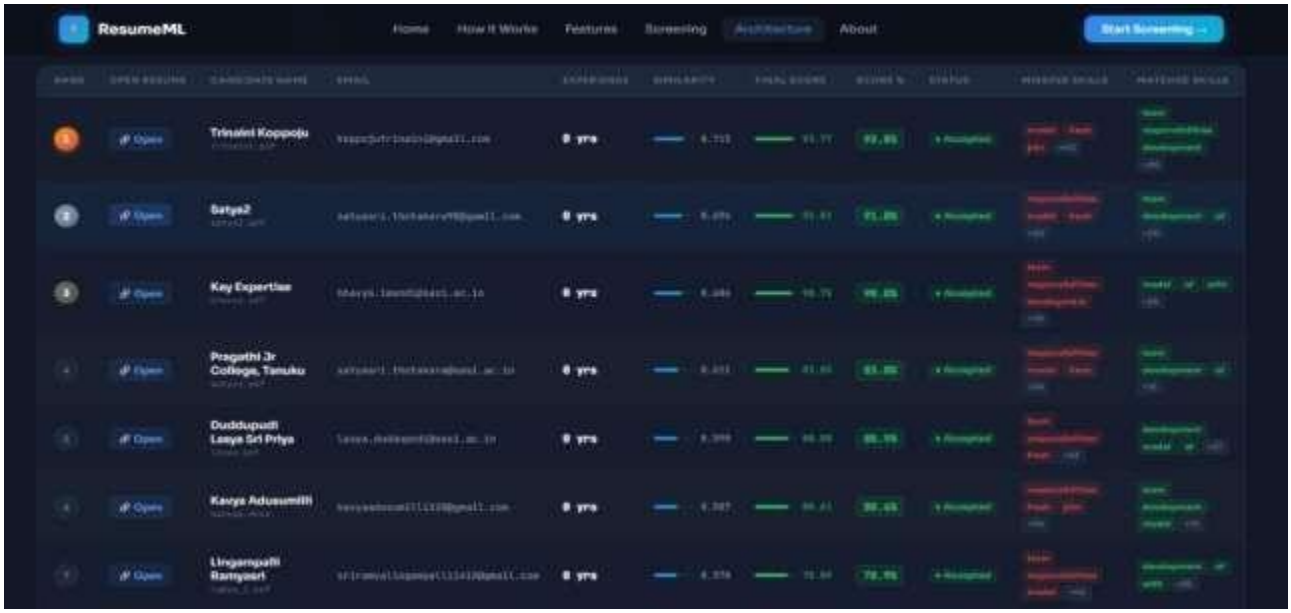
4.2 Implementation

The system uses **Machine Learning and Natural Language Processing (NLP)** techniques to automatically screen and rank resumes based on their relevance to a job description. The model works with a web interface that allows recruiters to upload resumes and job descriptions, analyze resume content, and display ranked results. The system was developed using **Python**. Important libraries used in the project include **Sentence-Transformers (SBERT), Scikit-learn, Pandas, NumPy, and Flask**. The development and testing of the model were performed using **Kaggle and Google Colab**, which provided sufficient computing resources for processing large datasets.
The first step involved **collecting a dataset of resumes** from publicly available sources such as Kaggle. The dataset contains resumes in different formats including **PDF, DOCX, and image-based files**. Each resume includes information such as **skills, education, experience, and professional qualifications**.
These resumes are compared with job descriptions to determine how well a candidate matches the job requirements.
Before processing, the dataset goes through **text extraction and preprocessing**. Text is extracted from documents, and **OCR (Optical Character Recognition)** is used to convert image-based resumes into machine-readable text. After extraction, the text is cleaned using **NLP preprocessing techniques** such as removing special characters, converting text to lowercase, removing stop words, and normalizing the text. These preprocessing steps help create consistent textual data for further analysis.
After preprocessing, the system generates **semantic embeddings** using the **Sentence-BERT (SBERT) model**. SBERT converts both resumes and job descriptions into numerical vector representations. These vectors capture the contextual meaning of the text, allowing the system to understand the relationship between skills, experience, and job

requirements.
Next, the system applies **Cosine Similarity** to calculate the similarity between the resume vectors and the job description vectors. Each resume receives a **similarity score**, which represents how closely the candidate profile matches the job requirements. Higher similarity scores indicate better matches.
Based on these similarity scores, the system performs **resume ranking and candidate shortlisting**. Resumes are sorted from highest similarity score to lowest similarity score, allowing recruiters to quickly identify the most suitable candidates.
Finally, the system integrates with a **web interface developed using Flask**. Recruiters can upload resumes and job descriptions through the interface. The system processes the documents, calculates similarity scores, and displays ranked resumes along with matching scores. This implementation combines **Machine Learning techniques and NLP-based semantic analysis** to automate the resume screening process, reduce manual effort, and improve recruitment efficiency.

5. Experimental Results

The resume screening system, as proposed, works by analyzing the resume and comparing it to the job description. The evaluation of the resume screening system is based on the similarity scores and the ranking results obtained while performing the resume matching process. The performance of the resume screening system is represented using graphical plots.
resume screening system works by performing text extraction, preprocessing, and feature generation for the resume. The resume is first converted into structured textual form using OCR and Natural Language Processing techniques. This ensures that the resume contains all the information required, such as skills, experience, and education.
The similarity between the resume and the job description is calculated using the cosine similarity method. The cosine similarity measures the similarity between the candidate's resume and the job description. The candidate's resume matches the job description if the similarity is high. The similarity scores range from 0 to 1, where high values indicate a high degree of similarity between the resume and the job description.
From the ranking results, it is seen that resumes with high similarity scores are ranked higher in the list. The candidates whose resume matches the job description are shortlisted, while the candidates whose resume does not match the job description are ranked lower.
From the results, it is seen that the resume screening system, as proposed, effectively works for automated resume screening. The resume screening system helps in improving the efficiency of the recruitment process.



The figure illustrates the system's output interface, which utilizes **Machine Learning** algorithms and **NLP** techniques to automate the candidate evaluation process.

- **NLP-Driven Data Extraction:** The system uses **Natural Language Processing** to parse raw text from resumes. Specifically, **Tesseract OCR** extracts text from images, while **NLTK/SpaCy** pipelines clean the data through tokenization and lemmatization.

- **ML Semantic Embeddings:** Rather than simple keyword matching, the system employs an **ML- based SBERT model** to convert text into semantic vector embeddings. This allows the system to understand the context of a candidate's experience.

- **Similarity Computation:** The **"Similarity"** and **"Score %"** columns represent the output of a **Machine Learning** calculation—specifically **Cosine Similarity**. This mathematical approach ranks candidates by measuring the distance between the resume vector and the job description vector.

- **Feature Extraction (Matched/Missing Skills):** The green and red tags demonstrate **NLP feature extraction**, where specific skills are identified as "features" and compared across datasets to provide a transparent gap analysis for the recruiter.

- **Flask Web Integration:** The final processed ML results are served through a **Flask-based backend**, rendering the dynamic table shown in the figure for real-time decision-making.

GAPS IDENTIFIED IN EXISTING RESEARCH

The current resume screening systems face various limitations that affect the effectiveness of the selection process. One of the major limitations of current resume screening systems is that they do not support various resume formats. Most of the current resume screening systems are designed to support only certain resume formats, such as PDF and DOCX, but not scanned resumes.
Another major limitation of current resume screening systems is that they are not efficient enough to map skills. The current resume screening systems are designed to match keywords only, but not to understand the relationship between skills and requirements. Moreover, current resume screening systems are not effective enough to provide the ranking of resumes. This makes it more challenging to identify the best candidate for the position.
Additionally, current resume screening systems are not effective enough to provide sufficient transparency to the recruiter. This makes it more challenging to understand the decision behind the selection of the resume. Therefore, there is a need to develop an effective resume screening system that supports various resume formats, maps skills efficiently, provides ranking, and ensures sufficient transparency.

Gaps Identified in Resume Screening Research

GAP AREA	SUMMARY OF GAP	IMPLICATIONS
Limited Resume Format Support	Many existing systems support only specific resume formats such as PDF or DOCX. Resumes in image or scanned formats cannot be processed effectively.	Limits the ability of the system to analyze resumes from different sources and reduces screening efficiency.
Inefficient Skill Mapping	Traditional screening methods rely mainly on simple keyword matching and fail to understand the relationship between candidate skills and job requirements.	Suitable candidates may be missed even if they have relevant skills expressed in different terms.
Lack of Resume Ranking	Some existing systems only classify resumes but do not rank candidates based on their relevance to job descriptions.	Recruiters must manually review many resumes, increasing time and effort in the hiring process.
Limited Explainability	Many automated systems do not provide clear explanations for why a resume is selected or rejected.	Reduces trust and transparency in automated recruitment decisions.

Future Enhancements Suggested in the Literature

Based on the research conducted on automated resume screening systems, some improvements have been made to improve the efficiency and performance of the systems. One such improvement is the increase in the resume dataset used for the screening systems. The current systems might be using a limited resume dataset for screening, which might not be enough for an accurate screening process. The dataset might need to be expanded to include resumes from various fields and domains.
Another such improvement that can be made to the automated resume screening systems is the implementation of advanced Natural Language Processing techniques for better resume content understanding. The Natural Language Processing techniques can be used to identify the skills, experience, and other important information required for the resume screening process.
Another area that might need improvement is the implementation of advanced similarity techniques for better resume

screening. The feature extraction process might need to be improved for the screening systems to understand the context between the skills required for a particular resume and the skills available in the resume.
In addition, the automated resume screening systems might be able to include some intelligent features that can provide recommendations for the right job roles for candidates based on their skills and experience. The systems might need to be integrated with web-based platforms for better efficiency and performance.

Conclusion

Automated technologies have been found to help solve many problems in modern recruitment systems. The process of resume screening is considered an important step in selecting appropriate candidates for different job positions in organizations. Organizations are flooded with resumes from different candidates through online recruitment websites and platforms. The process of manual resume screening is considered challenging and time-consuming for organizations. Intelligent text processing technologies have been found to help in resume screening.
This paper proposes an automated resume screening system. The system uses techniques to analyze resumes and compare them with job requirements. Natural Language Processing techniques have been used in this system to process text data from resumes. The Optical Character Recognition technique is used to process scanned resumes in image format.
Experimental results have shown that the proposed system can be used effectively in resume matching and ranking functions. The proposed system uses cosine similarity to compare resume information with job requirements.
The proposed approach can be considered an automated solution for resume screening systems. The proposed system can be used to reduce manual effort in resume screening systems. The proposed system can be used in real-world applications with recruitment platforms and applicant tracking systems in the future.

References

1. S. Sharma, A. Gupta, and R. Singh, **"Automated Resume Screening System Using NLP and Machine Learning Techniques,"** *International Journal of Computer Applications*, vol. 185, no. 12, pp. 15–21, 2025.
2. P. Kumar and R. Verma, **"Design and Development of Machine Learning Based Resume Ranking System,"** *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 6, pp. 230–236, 2022.
3. Patel, S. Shah, and R. Mehta, **"Resume Screening Using Machine Learning,"** *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)*, vol. 9, no. 4, pp. 112–118, 2024.
4. M. Ahmed and K. Rahman, **"Real-Time Resume Screening Using Natural Language Processing and Token-Based Indexing,"** *Proceedings of the International Conference on Data Science and Artificial Intelligence*, pp. 98–103, 2023.
5. L. Chen and Y. Zhang, **"Resume Screening with Natural Language Processing for Fair Candidate Evaluation,"** *Journal of Artificial Intelligence Research and Applications*, vol. 10, no. 2, pp. 45–52, 2024.
6. V. Sharma and P. Mishra, **"Resume Screening and Recommendation System Using Machine Learning,"** *International Journal of Computer Science and Information Security*, vol. 20, no. 5, pp. 77– 83, 2022.
7. R. Gupta, S. Agarwal, and N. Jain, **"AI-Powered Resume Screening System for Automated Recruitment,"** *International Journal of Intelligent Systems and Applications*, vol. 17, no. 1, pp. 34–40, 2025.
8. H. Lee and J. Park, **"Resume Analyzer Using Artificial Intelligence Techniques,"** *IEEE International Conference on Machine Learning and Applications*, pp. 215–220, 2024.
9. D. Patel and S. Kulkarni, **"Resume Analyzer: Automated Resume Ranking Software Using NLP,"** *International Journal of Emerging Technologies in Computer Science*, vol. 11, no. 3, pp. 60–66, 2023.
10. K. Reddy and M. Srinivas, **"Resume Screening and Recommendation System Using Machine Learning Approaches,"** *International Journal of Advanced Research in Computer Science*, vol. 14, no. 2, pp. 120–126, 2024.
11. **Sinha, A., Bhatia, S., & Bhattacharya, S. (2020).** "Resume Quality Assessment Using BERT Embeddings." *arXiv:2006.13111*. (Foundational for using BERT to understand context).
12. **Saatçı, M., et al. (2025).** "Automatic Resume Screening Using Sbert." *IJFMR*. (Focuses specifically on the

SBERT architecture you are using).

13. **Deshmukh, A., et al. (2024).** "Enhanced Resume Screening for Smart Hiring Using Sentence-BERT (S-BERT)." *SAI Organization*. (Compares SBERT performance vs. standard BERT).
14. **Xue, Y., & Ghosh, R. (2018).** "Deep Learning-based Resume-Job Matching Solution." *COLING*. (Discusses deep learning pipelines for matching).
15. **Luo, C., et al. (2022).** "A Resume Screening System Based on BERT and Knowledge Graph." *Journal of Intelligent & Fuzzy Systems*.
16. **Cabrera-Diego, A. (2025).** "Smart Application for Recruiting Based on Deep Learning and Siamese Neural Networks." *ResearchGate*. (Explores the Siamese network logic behind SBERT).
17. **Malhotra, A., & Singh, Y. (2019).** "Automated Resume Ranking Using Natural Language Processing and Machine Learning." *IEEE ICSCAN*.
18. **Prasad, G., & Fousiya, K. K. (2020).** "Resume Ranking based on Job Description using SpaCy NER model." *IRJET*. (Focuses on **SpaCy** for skill extraction).
19. **Patil, S., & Patil, S. (2020).** "Resume Parsing and Matching using Natural Language Processing." *IRJET*. (Discusses tokenization and lemmatization).
20. **Lavanya, K., & Nandhini, M. (2021).** "Resume Screening Using NLP and Machine Learning." *IJERT*.
21. **Trinh, A. (2021).** "NLP Framework for Automating Resume Classification and Ranking using fastText and SpaCy." *Technical Report*.
22. **Nanonets Technical Series (2023).** "Automating Document Parsing with Tesseract OCR and Python." (Technical guide for integrating **Tesseract**).
23. **Docsumo Reports (2025).** "Deep Learning in OCR: Using LSTM Networks for Unstructured Text Extraction." (The science behind Tesseract's accuracy).
24. **Lomax, J. (2021).** "Integrated Pipeline Combining OCR and NLP for Recruitment Data." *Journal of Data Science*.
25. **Sabarirajan, A., et al. (2025).** "Intelligent AI-Based Resume Screening and Ranking Framework for Unbiased Recruitment." *SciTePress*. (Addresses bias in AI hiring).
26. **Raghavan, M., et al. (2020).** "Mitigating Bias in Algorithmic Hiring Systems." *Frontiers in Computer Science*.
27. **Gan, et al. (2024).** "Human and LLM-Based Resume Matching: An Observational Study." *ACL Anthology*. (Compares AI rankings with human HR decisions).
28. **Aminu, T. (2025).** "Investigation of Cosine Similarity Metrics for High-Dimensional Text Matching." *Grenze*. (Explains the math behind your **Cosine Similarity** scores).
29. **Poojary, P., et al. (2025).** "Linguistic and Intelligent Machine for Automated Talent Acquisition." *IJCRT*. (Details the **Flask** backend integration).
30. **Gupta, D. (2025).** "Resume Parsing and Job Recommendation using NLP and Machine Learning." *IJRTI*. (Discusses user interface design for ML outputs).