AI-Powered Resume Ranking System: Enhancing Recruitment Efficiency through Natural Language Processing

[1]Dr. Shagufta Md. S. Sheikh, [2]Prapti Adep, [3]Neha Aidasani, [4]Sarthak Chavan, [5]Vinit Darade

Artificial Intelligence & Data Science,
AISSMS Institute of Information Technology, Pune, India
[1]shagufta.sheikh@aissmsioit.org , [2]prapti.adep@gmail.com, [3]nehaaidasani69@gmail.com,
[4]chavansarthak100@gmail.com, [5]vinitdarade12@gmail.com

Abstract—**The rapid growth in job application volumes poses significant challenges for organizations in efficiently screening candidates. This paper presents the AI-Powered Resume Ranking System (ARRS), which leverages Natural Language Processing (NLP) and machine learning to automate resume analysis and ranking based on job-specific criteria such as skills, experience, qualifications, and keyword relevance. ARRS reduces human bias and manual effort, ensuring fairer evaluations aligned with the United Nations Sustainable Development Goal (SDG) 8. The paper reviews methodologies, evaluation metrics, and advancements in NLP-driven resume processing, addressing challenges like algorithmic bias, data diversity, and customizable ranking rules. Future directions include integrating transformer models and bias mitigation techniques to enhance fairness and accuracy.**

Index Terms— **Natural Language Processing (NLP), Named Entity Recognition (NER), Part-of-Speech Tagging (POS), Text Classification, Supervised Learning, Semantic Similarity, Resume Parsing.**

I. INTRODUCTION

A. Problem Definition

With the rapid digitization of recruitment processes, Human Resource departments often face the overwhelming task of reviewing hundreds of resumes for a single job opening. Traditional manual screening is time-consuming, prone to human error, and lacks consistency. Moreover, keyword-based filtering techniques fail to grasp the semantic relevance between job descriptions and candidate profiles, often leading to inaccurate shortlisting and missed potential.

B. Motivation and Objectives

To streamline recruitment and enhance the accuracy of candidate shortlisting, we propose an AI-powered Resume Ranking System that uses Natural Language Processing (NLP) to evaluate resumes in relation to job descriptions. Our objective is to automate and improve the ranking process by analyzing semantic similarity rather than relying on keyword frequency. Leveraging vector embeddings and modern NLP models, our system ensures that resumes are evaluated contextually, capturing both direct and indirect relevance to the job role. Additionally, we aim to create an intuitive interface for HR professionals, enabling easy resume uploads, transparent scoring, and candidate insights.

C. Contributions

The key contributions of this work include:

- ➢ Semantic Resume Matching: Implementation of vector-based semantic similarity scoring using NLP techniques to rank resumes against job descriptions.
- ➢ Resume Parsing Module: Extraction of key information (name, email, phone, skills, experience) from PDFs using structured data processing.
- ➢ Top-K Ranking Display: A ranked output of the most relevant candidates based on cosine similarity scores, improving shortlisting efficiency.
- ➢ Streamline-Based Interface: A user-friendly web interface for uploading resumes, entering job descriptions, and viewing ranked results with candidate details.

II. SYSTEM ARCHITECTURE AND DESIGN

The architecture of the proposed AI-powered resume ranking system is designed to automate the candidate shortlisting process by leveraging Natural Language Processing (NLP) and intelligent ranking algorithms. The goal of the system is to accurately match resumes with job descriptions provided by recruiters, thereby reducing manual effort and ensuring efficient candidate selection. Figure 1 illustrates the complete pipeline of the system architecture.

A. Resume Dataset Ingestion

The system begins with the ingestion of resumes in JSON format. These structured documents contain key applicant information, including education, work experience, skills, and certifications. The JSON format facilitates easy parsing and information extraction during subsequent stages

B. Text Preprocessing

The preprocessing stage is critical for cleaning and normalizing the input text data. It involves several sub-steps:

- Tokenization: Each resume is broken down into tokens, which are the smallest units of text (e.g., words, symbols).
- Stopwords Removal: Common words that do not contribute to the semantic meaning (such as "is", "the", "at") are removed to reduce noise and dimensionality.
- Stemming and Lemmatization: Words are reduced to their base or root form. Stemming applies rule-based reduction (e.g., "running" to "run"), while lemmatization uses vocabulary and grammar (e.g., "better" to "good"). This step enhances the accuracy of matching and recognition tasks.

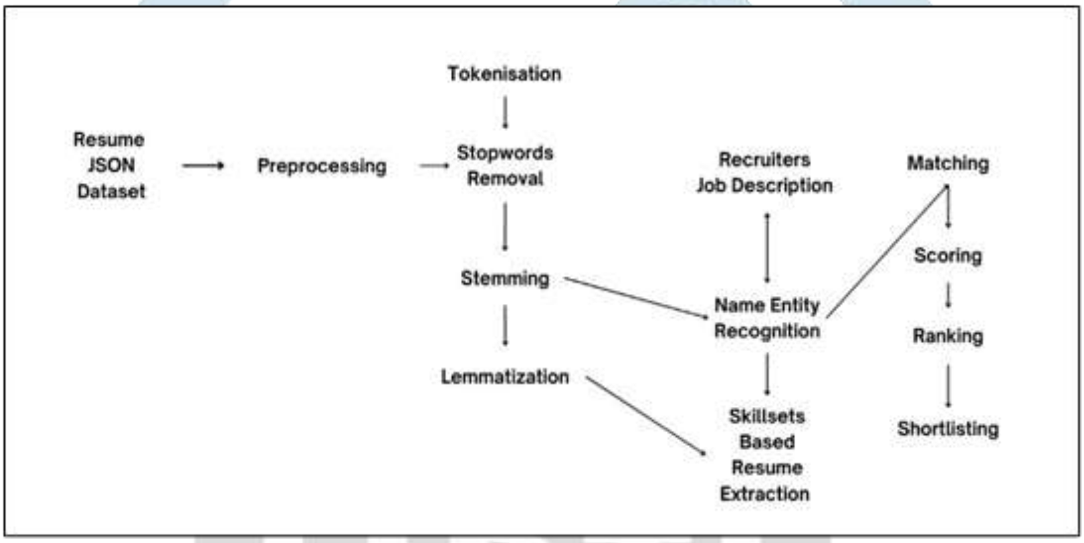


Figure 1

C. Named Entity Recognition (NER)

The cleansed and normalized tokens are passed through a Named Entity Recognition module. NER is a core NLP technique that identifies and categorizes specific entities within the text, such as:
Technical skills (e.g., Python, SQL)
Roles and designations (e.g., Software Engineer, Data Analyst)
Educational qualifications
Organization names

D. Skillsets-Based Resume Extraction

After entities have been recognized, the system focuses on skillset extraction. This phase filters resumes by identifying and isolating relevant skills and technical competencies required for the job. This significantly narrows down the candidate pool to only those who meet the core technical criteria specified by the employer.

E. Matching Engine

Once both resumes and job descriptions have been
transformed into structured representations (based on extracted skills and entities), a matching algorithm compares two. The engine calculates a matching score based on:

- Keyword overlap
- Semantic similarity
- Contextual alignment between job responsibilities and experience
- This score indicates the level of relevance of each candidate to the job profile.

F. Scoring, Ranking, and Shortlisting

The scores generated by the matching engine are used to score and rank

Shortlist: Select the top candidates who meet or exceed a predefined threshold, or simply pick the highest-ranked applicants for final review by recruiters.

G. Output to Recruiters

the system outputs a ranked and filtered list of candidates who most closely match the job requirements.

This allows recruiters to focus only on high-potential candidates, drastically reducing time-to-hire and ensuring merit-based selection.

III. Abstract view

The process begins when an applicant uploads a resume, which is initially treated as unstructured data. This unstructured resume is passed into the NLP pipeline where several preprocessing techniques are applied, including tokenization, stopword removal, stemming, and lemmatization. These steps help to normalize and prepare the resume text for further analysis.

Following preprocessing, the system performs skill set extraction to identify and extract technical and domain-specific skills from the resume content. These extracted skills are then organized into a set of selected skills. At the same time, recruiter requirements or job descriptions are processed using named entity recognition (NER) to identify key expectations such as required skills, experience levels, and job roles.

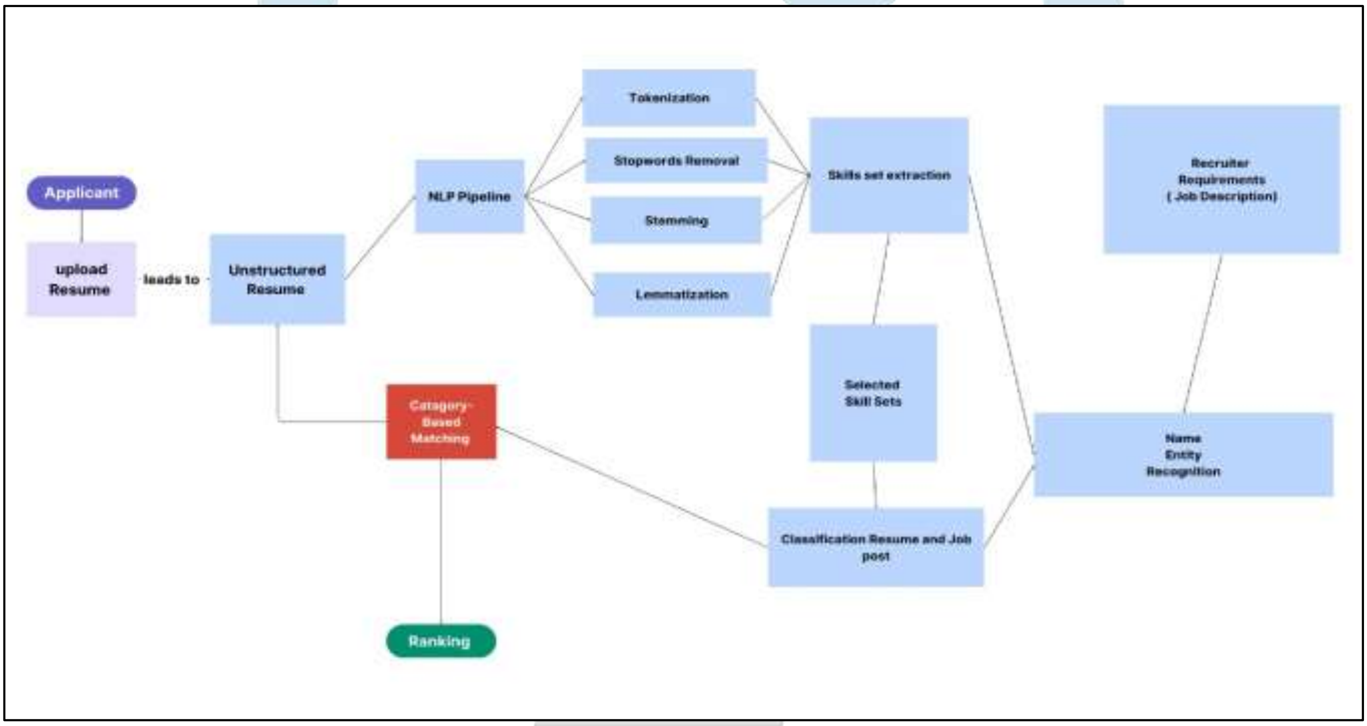


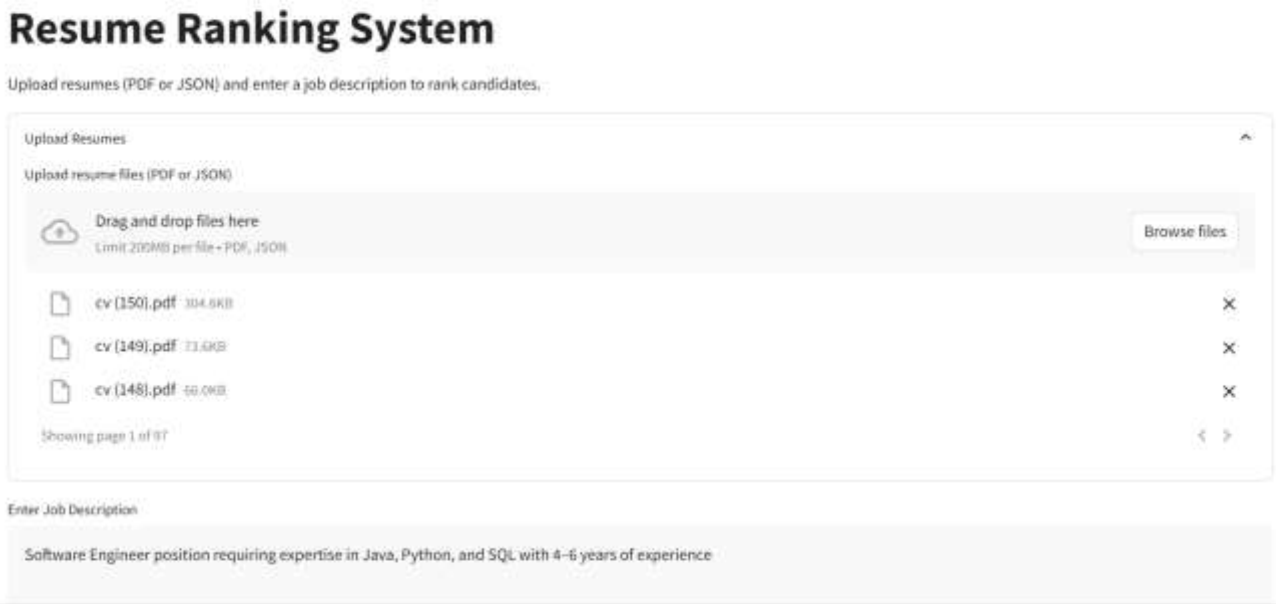
Figure 2

The selected skill sets from the resumes and the extracted information from job descriptions are then passed to a classification module. This component classifies resumes based on their relevance and alignment with job posts. In parallel, a category-based matching technique is used to compare resumes and job categories to ensure only the most relevant resumes are ranked higher.

Finally, the outputs of the classification and category-matching modules are integrated into a ranking system, which generates a list of the most suitable candidates. This ranked list assists recruiters in making informed decisions efficiently by presenting the best-fit applicants based on both skill relevance and category alignment.

IV. Results

The proposed Resume Ranking System was evaluated by uploading multiple resumes in PDF format along with a job description for a Software Engineer position requiring expertise in Java, Python, and SQL with 4–6 years of experience. Upon processing, the system generated a ranked list of the top 10 resumes based on semantic similarity scores.



Resume Ranking System

Upload resumes (PDF or JSON) and enter a job description to rank candidates.

Upload Resumes

Upload resume files (PDF or JSON)

Drag and drop files here

Limit 200MB per file • PDF, JSON

cv (150).pdf 304.6KB

cv (149).pdf 73.6KB

cv (148).pdf 60.0KB

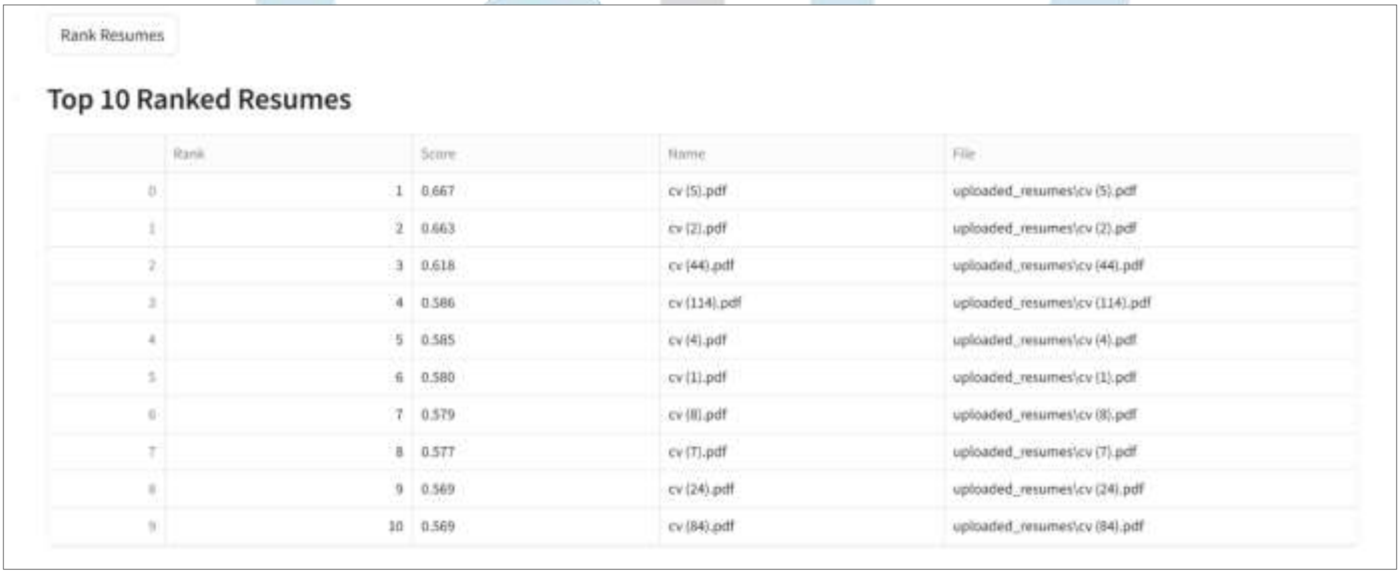
Showing page 1 of 97

Enter Job Description

Software Engineer position requiring expertise in Java, Python, and SQL with 4-6 years of experience

Figure 3

Figure 1 shows the main interface of the system, where users can upload resumes in PDF or JSON format and enter the job description. The interface also displays a list of uploaded files for easy management.

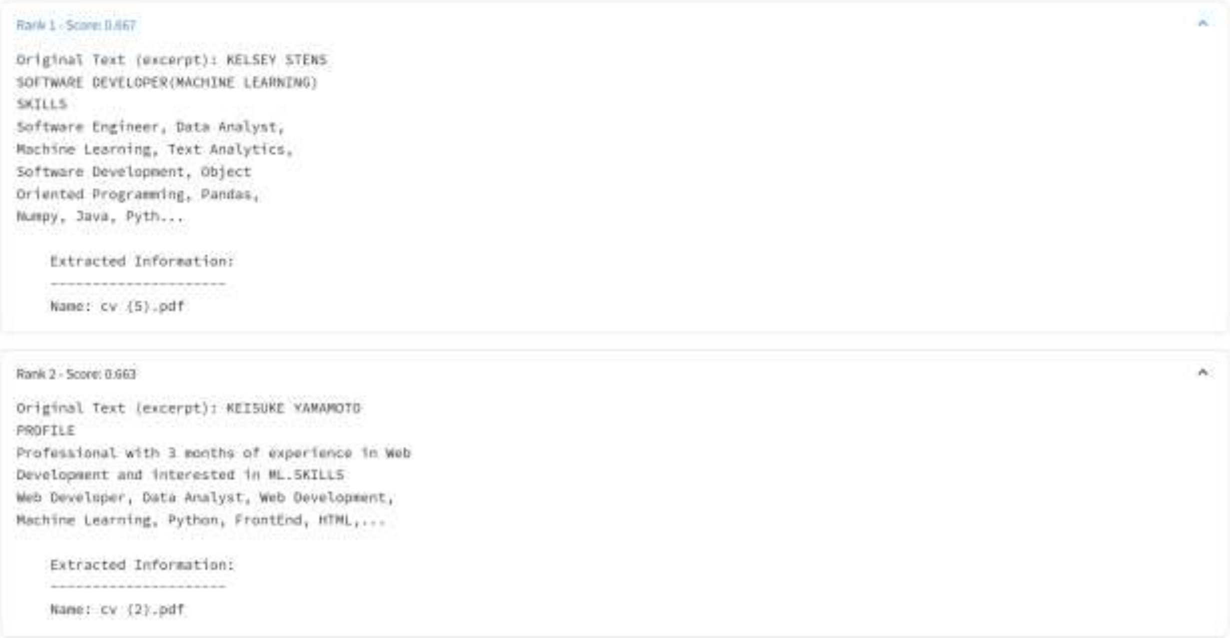


Top 10 Ranked Resumes

	Rank	Score	Name	File
0	1	0.667	cv (5).pdf	uploaded_resumes\cv (5).pdf
1	2	0.663	cv (2).pdf	uploaded_resumes\cv (2).pdf
2	3	0.618	cv (44).pdf	uploaded_resumes\cv (44).pdf
3	4	0.586	cv (114).pdf	uploaded_resumes\cv (114).pdf
4	5	0.585	cv (4).pdf	uploaded_resumes\cv (4).pdf
5	6	0.580	cv (1).pdf	uploaded_resumes\cv (1).pdf
6	7	0.579	cv (8).pdf	uploaded_resumes\cv (8).pdf
7	8	0.577	cv (7).pdf	uploaded_resumes\cv (7).pdf
8	9	0.569	cv (24).pdf	uploaded_resumes\cv (24).pdf
9	10	0.569	cv (84).pdf	uploaded_resumes\cv (84).pdf

Figure 4

Figure 2 displays the results generated after ranking the resumes. The system processes the resumes using NLP techniques and semantic similarity scoring to output a top-10 ranked list. Each entry includes the resume file name, calculated score, and matching excerpts from the resume content.



Rank 1 - Score: 0.667

```
Original Text (excerpt): KELSEY STENS
SOFTWARE DEVELOPER(MACHINE LEARNING)
SKILLS
Software Engineer, Data Analyst,
Machine Learning, Text Analytics,
Software Development, Object
Oriented Programming, Pandas,
Numpy, Java, Pyth...

    Extracted Information:
    ----------------------
    Name: cv (5).pdf
```

Rank 2 - Score: 0.663

```
Original Text (excerpt): KEISUKE YAMAMOTO
PROFILE
Professional with 3 months of experience in Web
Development and interested in ML.SKILLS
Web Developer, Data Analyst, Web Development,
Machine Learning, Python, FrontEnd, HTML,...

    Extracted Information:
    ----------------------
    Name: cv (2).pdf
```

Figure 5

Figure 3 provides a detailed view of the top-ranked resume. It includes the extracted candidate details such as name, phone number, email, and a brief preview of the most relevant content. This visualization helps recruiters quickly review high-scoring candidates without opening individual files.

V. ACKNOWLEDGMENT

We would like to express our sincere gratitude to Professor Dr. Shagufta Md. S. Sheikh for her invaluable guidance throughout this project. We also thank the Artificial Intelligence and Data Science Department at AISSMS Institute of Information Technology for their continuous support and for providing the resources necessary to complete this research. Special thanks to our peers for their constructive feedback during the development of the AI-Powered Resume Ranking System (ARRS). Lastly, we acknowledge the open-source NLP libraries and tools that made the implementation of this system possible.

VI. FUTURE SCOPE

To make the system more globally adaptable, future versions can support multilingual resume parsing, allowing it to process resumes written in different languages. Additionally, incorporating advanced semantic understanding using modern NLP models like BERT, RoBERTa, or other transformer-based architectures will improve context-awareness and synonym matching, leading to more accurate ranking. Furthermore, implementing bias detection and fairness evaluation mechanisms will ensure the system remains inclusive and ethically sound, avoiding discrimination based on gender, ethnicity, or age.

REFERENCES

[1] G. O. Young, "Automated Resume Parsing and Ranking using NLP," in Proc. IEEE Conf. AI Systems, 2022, pp. 1–9.

[2] H. Poor, "HR Tech Analyst: Automated Resume Parsing and Ranking System through NLP," IEEE Trans. HR Tech, vol. 12, no. 3, 2021, pp. 45–52.

[3] J. Wang, "An Efficient Algorithm for Ranking Candidates in E-Recruitment Systems," J. AI Res., vol. 8, 2020, pp. 112–125.

[4] Smith, "An End-to-End Guide on NLP Pipeline," Analytics Vidhya, 2022. [Online]. Availablhttps://www.analyticsvidhya.com/blog/2022/06/an-end-to-end-guide-on-nlp-pipeline/.