International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

AI-Driven Talent Matching System Using BERT and Recommendation Algorithms

Umesh Shingare[1], Saurav Sultane[2], Om Autade[3], Om Taskar[4], Kalpesh Wagh[5], Prof D. S. Shingate[6]

Department of Information Technology / MET's Bhujbal Knowledge City IoE / Nashik, India

Corresponding Author Email: sultanesaurav@email.com

How to Cite this Article:
Shingare, U., Sultane, S., Autade, O., Taskar, O. & Wagh, K. (2026). AI-Driven Talent Matching System Using BERT and Recommendation Algorithms. International Journal of Creative and Open Research in Engineering and Management, 02(05). https://doi.org/10.55041/ijcope.v2i5.016

https://doi.org/ 10.55041/ijcope.v2i5.016

Abstract—

In today's competitive job market, organizations struggle with the overwhelming volume of resumes received for each job posting, making manual screening inefficient and prone to bias. This research directly addresses these pain points by presenting an AI-Based Resume Shortlisting and Job Recommendation System. The system's main thesis is to leverage advanced AI and ML technologies to automate and optimize recruitment, enabling faster, fairer, and more accurate hiring decisions. The system is designed to serve both candidates and HR administrators through a dual-interface architecture. Candidates can register, upload their resumes, and receive real-time evaluation in the form of a resume score, personalized skill enhancement suggestions, and job recommendations aligned with their profile. The system also enables candidates to directly apply for relevant job roles, thereby streamlining the job search process. On the administrative side, HR users can create and manage job postings, analyze candidate resumes, and leverage the AI model to automatically shortlist the most suitable applicants based on predefined criteria.

Keywords—Artificial Intelligence, Feature Extraction, Cosine Similarity, Resume Screening, Transformer Models / BERT, Recruitment Automation, Candidate Evaluation.

I. INTRODUCTION

In the contemporary recruitment landscape, organizations face an overwhelming number of resumes per job opening, making manual screening inefficient, inconsistent, and biased. At the same time, candidates often lack clarity on role suitability and necessary skill enhancements. To address these challenges, this work proposes an AI-Based Resume Shortlisting and Job Recommendation System using Python and the Django framework. The proposed system leverages advanced Natural Language Processing (NLP) techniques for automated resume parsing, information extraction, and text normalization. Unstructured resume data is transformed into structured representations through preprocessing steps such as tokenization, lemmatization, and stop-word removal. For semantic understanding, the system incorporates Transformer-based models such as BERT (Bidirectional Encoder Representations from Transformers) to generate contextual embeddings of both resumes and job descriptions. These embeddings enable accurate semantic similarity computation, allowing the system to capture contextual relationships beyond simple keyword matching. The architecture supports two primary user roles: Candidate and Admin (HR). Candidates can upload resumes and receive a quantitative resume score, skill gap analysis, and personalized job recommendations generated through similarity metrics such as cosine

similarity and machine learning-based ranking algorithms. On the administrative side, HR users can create job postings, analyze candidate profiles, and utilize the trained model to automatically shortlist candidates based on relevance and predicted job fit. Furthermore, the system integrates a feedback-driven refinement mechanism, where recruiter decisions are used to iteratively improve model performance and recommendation accuracy. By combining NLP-driven feature extraction, BERT-based semantic modeling, and machine learning techniques, the proposed system enhances recruitment efficiency, reduces human bias, and enables data-driven hiring decisions.

II. LITERATURE REVIEW

The literature survey of AI-based resume shortlisting and job recommendation systems, as summarized in Table 2.1. Rohit Jain, Neha Sharma, and Priya Verma (2024) proposed an AI-driven recruitment solution that applies NLP techniques with TF–IDF to automate resume screening and improve candidate–job matching accuracy. Karan Patel, Deepika Singh, and Ravi Nair (2023) developed a job recommendation system that generates personalized job suggestions by analyzing candidate profiles and mapping them to job requirements. Ahmed Mohamed, Fatima Khan, and John Lewis (2022) designed a web-based recruitment platform that integrates Django with AI methods to automate screening and streamline candidate evaluation. Vikas Kumar, Sneha Joshi, and Arjun Rao (2024) introduced BERT-based contextual understanding for resumes to strengthen semantic analysis and enhance job-role matching performance. Aditi Singh and Meena Iyer (2023) highlighted the broader impact of AI in recruitment, emphasizing automation, reduced manual effort, and improved hiring decisions, collectively motivating the proposed system.

III. METHODOLOGY

The system is organized as a layered, modular pipeline that takes uploaded candidate resumes and job descriptions as input and produces ranked candidate shortlists with tailored job recommendations. A brief account of each module follows:

1) Input and User Interface Module: HR This module serves as the entry point for both HR administrators and job seekers. It accepts resumes in PDF format and captures related information such as user profiles, job postings, and company details. The interface is designed to be clean and user-friendly, ensuring smooth interaction even for non-technical users.

2) **Data Extraction and Preprocessing Module:** This combined module focuses on extracting and structuring data from resumes. It parses text from different document formats and applies preprocessing techniques such as tokenization, stop-word removal, and normalization. Advanced NLP methods, including Named Entity Recognition (NER), are used to identify key elements like skills, education, job roles, and organizations.

3) Semantic Matchmaking Module: This module evaluates the similarity between candidate profiles and job descriptions. It converts textual data into numerical vector representations using techniques such as TF-IDF and transformer-based embeddings like BERT. Similarity measures such as cosine similarity and the Jaccard index are used to determine how closely a candidate's profile aligns with job requirements.

4) Machine Learning Scoring Engine: The final module computes a comprehensive match score by combining semantic similarity with additional factors like experience, education, and skill relevance. Based on this score, candidates are ranked for specific job postings, and suitable job recommendations are generated for job seekers.

IV. PROPOSED SYSTEM

As shown in Fig. 1, the proposed system follows a layered workflow where candidates and HR

interact through a web interface, and all requests are handled by the Django application layer for authentication, form processing, and routing. The uploaded resumes are passed to the AI/NLP module for PDF text extraction, cleaning, and skill identification, after which the ML engine performs TF-IDF-based multi-label classification to predict suitable job categories. The scoring and recommendation module then computes resume quality and job match scores, generating personalized recommendations and shortlisting support. Finally, the database stores applicant

profiles, job postings, applications, and AI outputs, enabling the system to display results and tracking information back to users through the portal.

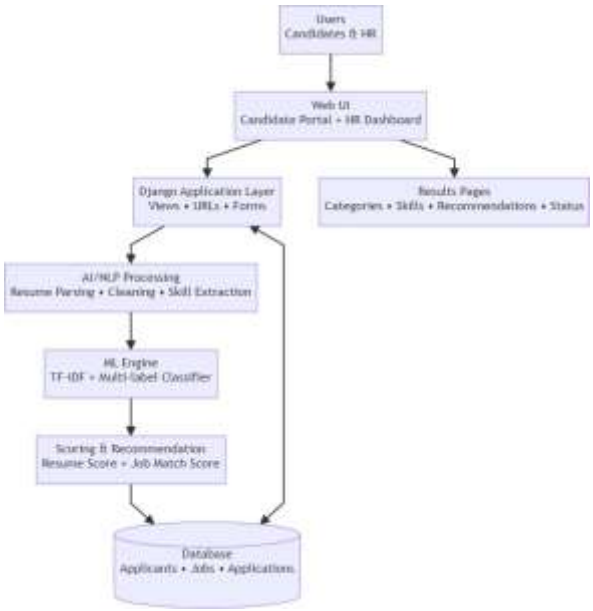

Fig 1: System Workflow Diagram

V. ARCHITECTURE

Figure 2 illustrates the high-level architecture of the proposed system, organized into four main layers: presentation, application, AI/ML processing, and data storage. In the presentation layer, the Candidate Portal supports resume upload, result viewing, and job applications, while the HR Dashboard enables job posting, applicant monitoring, and shortlisting. These interfaces communicate with the Django application layer, which manages URL routing, authentication, and form-based request handling, and forwards resumes for analysis. The AI/NLP and ML layer performs PDF-to-text extraction, preprocessing, skill extraction, multi-label classification using TF-IDF with a One-vs-Rest KNN model, and finally generates resume scores and job match recommendations. The data layer persists users, applicants, jobs, and applications in a database, stores uploaded resumes in file storage, and loads trained model artifacts (e.g., model.pkl, vectorizer.pkl) to ensure consistent predictions across the system.

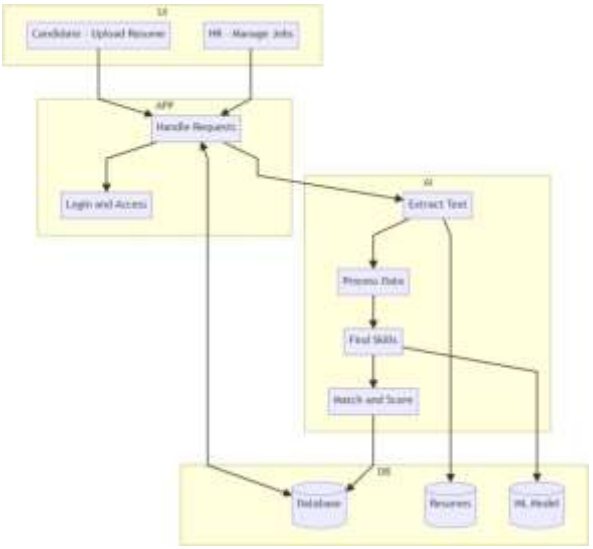

Fig 2: Architecture Diagram

VI. METHEMATICAL MODEL OF SYATEM

Let U be the set of users (candidates/HR), R the set of resumes, J the set of jobs, and $C = \{c_1, c_2, \dots, c_m\}$ the set of job categories.

1) Resume Representation

For a resume $r \in R$, let t_r be the extracted and cleaned text.
A TF–IDF vectorizer $\phi(\cdot)$ maps text to a feature vector:

$$\mathbf{x}_r = \phi(t_r) \in \mathbb{R}^d$$

2) Multi-Label Category Prediction

Each resume can belong to multiple categories. Define a label vector:

$$\mathbf{y}_r = [y_{r1}, y_{r2}, \dots, y_{rm}], y_{rk} \in \{0,1\}$$

A trained multi-label classifier $f(\cdot)$ outputs confidence scores:

$$\mathbf{p}_r = f(\mathbf{x}_r) = [p_{r1}, p_{r2}, \ldots, p_{rm}], p_{rk} \in [0,1]$$

Using a threshold τ, predicted labels are:

$$\hat{y}_{rk} = \begin{cases} 1, & p_{rk} \geq \tau \\ 0, & p_{rk} < \tau \end{cases}$$

and the predicted category set is:

$$\hat{C}_r = \{c_k \in C \mid \hat{y}_{rk} = 1\}$$

3) Skill Extraction Model

Let S be the global skill vocabulary. The system extracts a skill set from the resume:

$$S_r = g(t_r) \subseteq S$$

For a job $j \in J$, let required skills be:

$$S_j \subseteq S$$

4) Job Match Score

A skill-overlap score (Jaccard similarity) is:

$$M_{\text{skill}}(r,j) = \frac{|S_r \cap S_j|}{|\cup_r S_j|}$$

A category-alignment score is:

$$M_{\text{cat}}(r,j) = \max_{c_k \in C_j} p_{rk}$$

where $C_j \subseteq C$ are categories associated with job j. Final match score:

$$M(r,j) = \alpha\, M_{\text{skill}}(r,j) + (1-\alpha)\, M_{\text{cat}}(r,j), 0 \leq \alpha \leq 1$$

5) Resume Score

Let the resume score be a weighted combination of quality factors:

$$Q(r) = w_1 q_{\text{skills}}(r) + w_2 q_{\text{edu}}(r) + w_3 q_{\text{exp}}(r) + w_4 q_{\text{proj}}(r) + w_5 q_{\text{struct}}(r)$$

with $w_i \geq 0$ and $\sum_{i=1}^{5} w_i = 1$. The final score is normalized to 0–100:

$$\text{Score}(r) = 100 \cdot Q(r)$$

6) Recommendation Function

For a candidate resume r, recommended jobs are the top-K jobs by match score:

$$\text{Rec}(r) = \text{TopK}_{j \in J}\ M(r,j)$$

This mathematical model captures the system's main pipeline: Resume → TF–IDF features → multi-label prediction → skills + matching → scores + recommendations.

VII. RESULTS AND DISCUSSION

In this work, the proposed AI-based resume shortlisting and job recommendation system demonstrates that integrating NLP driven skill extraction with multi-label classification can effectively automate initial screening and improve consistency in candidate evaluation. The system successfully parses uploaded PDF resumes, predicts one or more relevant job categories, and generates confidence scores that help rank candidates for specific roles. In addition, the resume scoring module provides interpretable feedback by highlighting detected skills and recommending missing skills, which supports candidate profile improvement and increases transparency of the decision support process. From the recruiter's perspective, the HR dashboard enables faster filtering and shortlisting by combining predicted categories, match scores, and application status tracking, thereby reducing manual effort and turnaround time. Overall, the results indicate that the proposed approach is practical for real-world recruitment workflows, while future improvements can focus on expanding the dataset, reducing bias further, and enhancing

recommendation quality using advanced transformer-based embeddings and larger-scale evaluation.

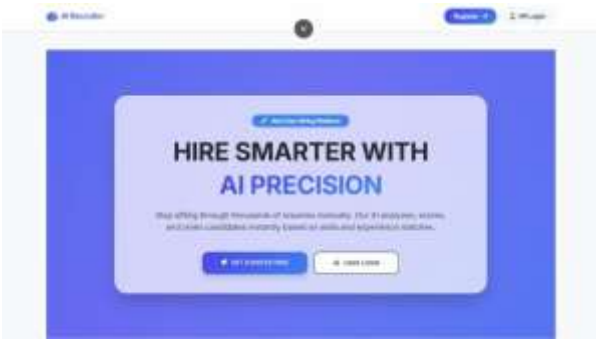

Fig 3: Home Page

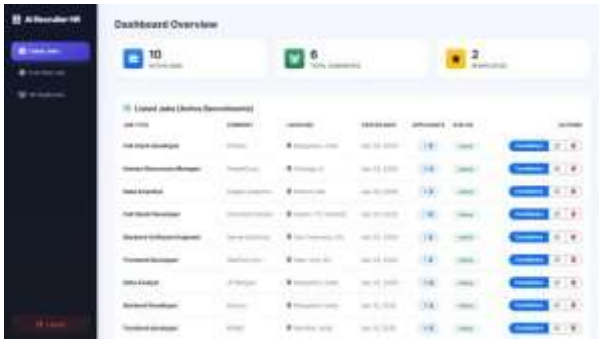

Fig 4: HR Dashboard

Fig 5: Job Posting

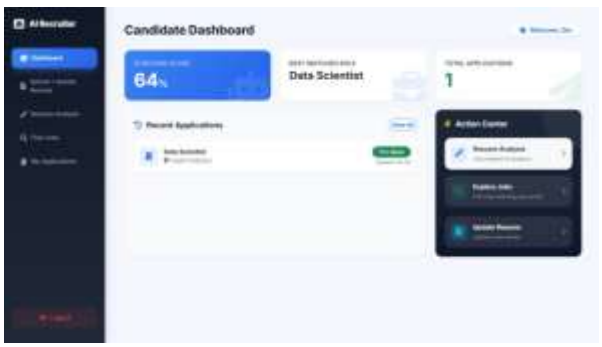

Fig 6: Candidate Dashboard

Fig 7: Listed Job

VIII. FUTURE ENHANCEMENTS

The current system demonstrates effective functionality using standard NLP techniques for resume parsing and skill matching; however, several enhancements are planned for future development. One of the major improvements includes the integration of advanced Large Language Models (LLMs) such as BERT and GPT architectures. This will enable the system to understand the deeper semantic meaning of a candidate's experience rather than relying only on keyword matching, thereby improving the accuracy of candidate evaluation and job recommendations. In addition, the platform will be expanded beyond its current web-based implementation through the development of a dedicated mobile application for both Android and iOS platforms. This will allow users to upload resumes, receive real-time job notifications, and track application statuses from anywhere, significantly improving accessibility and user experience. Future versions will also incorporate AI-powered chatbots capable of conducting preliminary screening interactions with candidates. Alongside this, automated interview scheduling will be introduced by integrating the system with HR calendars, eliminating manual coordination and streamlining the recruitment workflow. Another key enhancement will be the support for multi-language resume parsing, enabling the system to process resumes written in various regional and international languages. This will make the platform more inclusive and suitable for global recruitment scenarios.

IX. CONCLUSION

The proposed AI-based recruitment platform presents an efficient and practical alternative to fully manual resume screening by strategically leveraging NLP and machine learning to generate actionable insights, rather than relying on exhaustive human review of complete resume content. This approach improves scalability and consistency, making it suitable for large applicant pools and time-sensitive hiring without compromising decision support quality. Furthermore, the integration of multi-label classification, skill extraction, and resume scoring mitigates common recruitment challenges such as inconsistent shortlisting and missed transferable skills, while providing transparent outputs like confidence scores and recommended skills. The HR dashboard and candidate portal enable an end-to-end workflow for job posting, applications, filtering, and tracking, reducing operational workload for recruiters and improving candidate experience. Overall, the system demonstrates a balanced integration of automation, explainability, and deployable web architecture, making it a promising solution for modern AI-assisted recruitment and job recommendation frameworks.

REFERENCES

[1] Rohit Jain, Neha Sharma, and Priya Verma. (2024). AI-Powered Recruitment: Enhancing Resume Screening Using Natural Language Processing. Focuses on automating resume shortlisting using NLP to extract candidate skills and match them with job descriptions.

[2] Karan Patel, Deepika Singh, and Ravi Nair. (2023). Automated Job Recommendation System Using Machine Learning and User Profiling. Develops a recommendation engine that suggests jobs based on resume data, interests, and candidate skill sets. [3] D. D. Author et al., "Conference Paper Title," in Proc. IEEE Conf., Year, pp. 1–6.

[3] Ahmed Mohamed, Fatima Khan, and John Lewis. (2022). Intelligent Hiring System Using Django Framework and Machine Learning Algorithms. Integrates Django web framework

with AI to streamline the recruitment and candidate evaluation process.

[4] Vikas Kumar, Sneha Joshi, and Arjun Rao. (2024). A Study on Resume Classification Using Deep Learning Approaches. Compares deep learning models like CNN and BERT for skill extraction, job fit prediction, and automated shortlisting.

[5] Aditi Singh and Meena Iyer. (2023). AI in Human Resource Management: An Overview of Recruitment Automation. Explores how artificial intelligence transforms HR operations by improving decision- making in hiring.

[6] Rajesh Gupta and Harshita Malhotra. (2022). TextMining Techniques for Resume Screening: Challenges and Opportunities. Analyzes NLP-based resume parsing techniques and their role in automating candidate evaluation.

[7] Li Chen, Maria Santos, and David Park. (2024). Design and Implementation of a Job Matching Platform Using Python and SQLite. Describes the backend development and database structure for AI- driven job recommendation systems.

[8] George Thomas and Aishwarya Das. (2023). Skill Gap Analysis Using Machine Learning for Career Recommendation. Proposes an AI model that identifies missing skills in candidate profiles and provides personalized improvement advice.

[9] S. R. Raoand K. Ramesh (2025). Django-Based Web Application for Automated Recruitment Process. Presents a full stack implementation for resume upload, job posting, and candidate ranking using AI models.

[10] A. Krizhevsky, I. Sutskever, and G. Hinton. (2017). ImageNet Classification with Deep Convolutional Neural Networks. Introduced foundational deep learning concepts later adapted for NLP-based resume classification systems.