International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 3, May-June 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Skill-Based Job Recommendation and Matching System

K. Sandha Kavishman*, Mr. R. Ramakrishnan**

*(Student, Department of Master Computer Application,
Sri Manakula Vinayagar Engineering College, (Autonomous), Pondicherry 605008, India
Email: santhakavishman2003@gmail.com)

**(Professor & Head, Department of Master Computer Application,
Sri Manakula Vinayagar Engineering College, (Autonomous), Pondicherry 605008, India
Email: ramakrishnanmca@smvec.ac.in)

Abstract:

In today's rapidly evolving job market, connecting the right candidates to the right opportunities remains a persistent challenge. Traditional job portals rely on keyword-based search and manual filtering, failing to understand the semantic relationships between skills and job requirements. This paper presents a Skill-Based Job Recommendation and Matching System that leverages Natural Language Processing (NLP) and Machine Learning to intelligently extract skills from resumes and job descriptions, compute semantic similarity, and deliver highly personalised job recommendations. The system uses Named Entity Recognition (NER) with spaCy to extract structured skills from free-form text, encodes them using TF-IDF and BERT (Sentence Transformers), and computes cosine similarity for precise job-candidate matching. A hybrid recommendation engine combining Content-Based Filtering (CBF) and Collaborative Filtering (CF) ranks top-N jobs per candidate. The platform further provides skill gap analysis with course recommendations and a recruiter-facing candidate ranking portal. Evaluation results demonstrate a Precision@5 of 88%, Recall@10 of 85%, NDCG of 0.91, and Match Accuracy of 93%.

Keywords — Job Recommendation, NLP, NER, TF-IDF, BERT, Cosine Similarity, Collaborative Filtering, Content-Based Filtering, Skill Gap Analysis, Hybrid Recommender System.

I. INTRODUCTION

In today's competitive employment landscape, matching the right candidates to the right job opportunities remains a persistent and costly inefficiency. Conventional job portals such as LinkedIn, Naukri, and Indeed depend primarily on keyword search and location-based filtering. These systems cannot recognise that "Machine Learning Engineer" and "ML Developer" describe the same role, nor can they infer that proficiency in "PyTorch" implies familiarity with deep learning frameworks more broadly. The result is a bidirectional failure: qualified candidates are overlooked and recruiters are inundated with irrelevant applications.

This paper presents a Skill-Based Job Recommendation and Matching System designed to resolve this mismatch through the application of Natural Language Processing (NLP) and Machine Learning (ML). Instead of surface-level keyword overlap, the system understands the semantic content of both candidate profiles and job descriptions, quantifies their alignment through embedding-based similarity scoring, and delivers a ranked, personalised job feed to each candidate. Simultaneously, the recruiter portal presents an AI-ranked list of candidates for each job posting, dramatically reducing the manual screening burden.

The system's three principal contributions are: (1) an NLP pipeline for structured skill extraction from unstructured resume and job description text using spaCy Named Entity Recognition; (2) a hybrid recommendation engine combining TF-IDF and BERT embeddings with collaborative and

content-based filtering for personalised job rankings; and (3) a skill gap analysis module that identifies missing skills relative to target job descriptions and recommends specific upskilling resources.

II. LITERATURE SURVEY

Collaborative Filtering (CF), introduced by Goldberg et al. [1] and extended through matrix factorisation by Koren et al. [2], forms the foundational paradigm for personalised recommendation. CF identifies users with similar interaction histories and recommends items they have collectively favoured. In the employment domain this translates to recommending jobs that candidates with similar skill profiles have engaged with. However, CF suffers from the cold-start problem: new candidates with no interaction history receive no meaningful recommendations.

Content-Based Filtering (CBF) addresses this limitation by representing items and users through feature vectors and computing similarity directly. Lops et al. [3] surveyed content-based recommender systems and identified TF-IDF-weighted vector representations as the dominant approach for text-heavy domains. Job recommendation is inherently text-heavy — resumes and job descriptions are narrative documents whose skill content must be extracted and encoded before similarity can be computed.

The application of NLP to skill extraction from resumes has been studied extensively. Galke et al. [4] demonstrated that transformer-based encoders substantially outperform bag-of-words models on occupational skill classification. Javed et al. [5] benchmarked skill extraction approaches and found that fine-tuned BERT models achieved state-of-the-art performance on the task, motivating the inclusion of Sentence-BERT embeddings in this system.

The BERT architecture, introduced by Devlin et al. [6], enabled dense, contextualised text representations that encode semantic meaning rather than surface form. Reimers and Gurevych [7] extended BERT to Sentence Transformers (SBERT), producing fixed-length sentence embeddings optimised for semantic similarity via cosine distance — precisely the comparison required for job-candidate matching.

Hybrid recommender systems combining CF and CBF have consistently outperformed single-method approaches [8]. Burke [9] proposed a taxonomy of hybrid strategies and demonstrated that weighted combinations of CF and CBF achieve superior coverage and accuracy across sparse interaction datasets — a condition characteristic of job recommendation systems. Shalaby et al. [10] applied deep learning to job recommendation, demonstrating that joint modelling of candidate and job representations in a shared embedding space improved recommendation quality relative to independent encoding.

III. EXISTING SYSTEM

Current job portals operate through keyword search and structured attribute filtering. A candidate searching for "Data Analyst" roles selects a location, an experience range, and optionally a salary band; the portal returns postings whose text contains those exact keywords.

This approach has three fundamental limitations. First, keyword matching is brittle with respect to synonymy and skill relatedness: a resume listing "Deep Learning" does not match a posting requiring "Artificial Neural Networks" despite the near-complete conceptual overlap. Second, existing systems provide no mechanism for structured skill extraction — the candidate's qualifications exist as unstructured narrative text and the portal cannot parse that document to understand what the candidate actually knows. Third, recommendation is entirely search-driven rather than proactive: the system does not maintain a candidate profile and does not push relevant postings without an explicit query [3].

From the recruiter perspective, existing portals return applicant lists ranked by application recency or a primitive keyword match score, requiring human reviewers to read each resume individually to assess fit. The absence of skill gap analysis

further limits career development utility to passive job browsing rather than active career planning [5].

TABLE I

Comparison of Existing System vs. Proposed System

Feature	Existing System	Proposed System
Skill Matching	Keyword overlap only	TF-IDF + BERT cosine similarity
Skill Extraction	None — unstructured text	spaCy NER + skill taxonomy
Recommendation	Search-driven; no proactive feed	Hybrid CF + CBF ranked feed
Skill Gap Analysis	Not available	Profile vs JD gap + course links
Candidate Ranking	Recency or keyword score	AI match score 0–100% per job
Cold-Start	No handling; search required	CBF provides immediate results
Semantics	Not handled	BERT captures semantic equivalence

IV. PROPOSED SYSTEM

The proposed Skill-Based Job Recommendation and Matching System replaces the keyword-search paradigm with a semantic, profile-driven recommendation architecture. The system operates continuously: every time a candidate updates their profile or a new job posting is ingested, the recommendation engine recalculates match scores and refreshes the candidate's personalised job feed.

The system's core logic is a hybrid recommendation engine combining three distinct matching strategies. Content-Based Filtering computes cosine similarity between the candidate's skill embedding vector and each active job posting's requirement vector. Collaborative Filtering identifies candidates with similar skill profiles and recommends jobs that those candidates successfully engaged with. A hybrid ranking layer combines CBF and CF scores with configurable weights, producing a final ranked list of top-N job recommendations per candidate.

Skill gap analysis is performed by taking the set difference between the candidate's extracted skills and the required skills in a target job description. Missing skills are presented alongside links to relevant online courses and certifications, enabling the system to function as a career development tool rather than a passive job board.

The recruiter portal inverts the recommendation logic: for each job posting, the system ranks all candidates by their match score, presenting recruiters with a prioritised shortlist. This eliminates the need for manual resume screening and enables recruiters to focus attention on candidates whose skill profiles most closely align with the role.

V. SYSTEM ARCHITECTURE

The system is organised into four functional layers. The Data Ingestion Layer accepts candidate resumes in PDF and DOCX formats, scraped job descriptions, user interaction logs, and a curated skill taxonomy database. The NLP Processing Layer runs resume parsing via PyPDF2, named entity recognition for skill extraction, TF-IDF and BERT embedding generation, and skill graph construction.

The Recommendation Engine layer applies Content-Based Filtering, Collaborative Filtering, a hybrid ranking model, and cosine similarity scoring to produce match scores. The Output and Action Layer delivers ranked job recommendations, a match score from 0–100%, a skill gap report, course and certification suggestions, and a ranked candidate shortlist to the recruiter portal.

Fig. 1: Layered System Architecture

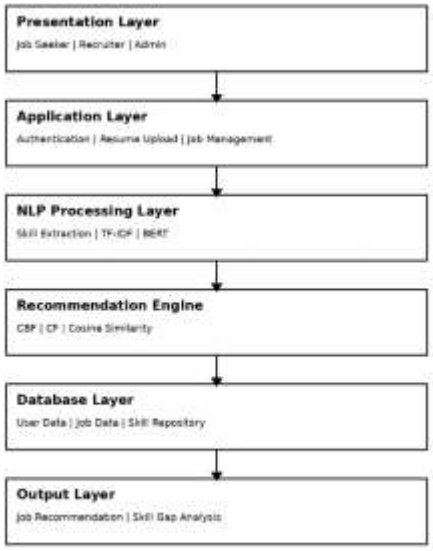

TABLE II

NLP Models and Performance Metrics

NLP Model / Metric	Value / Description
spaCy NER	Skill entity extraction from resumes and JDs
BERT (Sentence Transformer)	all-MiniLM-L6-v2; semantic embeddings
TF-IDF Vectoriser	Lightweight baseline similarity scoring
Precision@5	~88%
Recall@10	~85%
NDCG	~0.91
Match Accuracy	~93%

VI. LIST OF MODULES

The application is organised around eight functional modules, each with a clearly defined responsibility.

A. User Authentication & Role Management

JWT-secured login with role-based access control for job seekers, recruiters, and administrators. Passwords are stored as bcrypt hashes; all API endpoints validate role claims before processing any request.

B. Resume Upload & Parsing Module

Accepts PDF and DOCX resumes. PyPDF2 and python-docx extract raw text; a section classifier identifies experience, education, projects, and skills blocks from the extracted content.

C. NLP Skill Extraction Engine

spaCy Named Entity Recognition, augmented with a custom skill taxonomy pattern matcher, identifies technical and soft skills in free-form resume text. Extracted skills are normalised against a canonical skill ontology to ensure consistent matching across synonym variants.

D. Job Description Processing Module

Ingests job postings from scraping pipelines or manual entry. Required skills, experience levels, and role details are extracted using the same NLP pipeline applied to resumes, ensuring comparable feature spaces for both candidate and job representations.

E. Recommendation & Matching Engine

The core engine produces match scores through three steps: TF-IDF cosine similarity for initial filtering; BERT embedding cosine similarity for semantic re-ranking; and collaborative filtering adjustment based on aggregated interaction patterns from similar candidate profiles. The final score is a weighted linear combination of these components.

F. Skill Gap Analysis Module

For any target job description, the module computes the set difference between the candidate's extracted skills and the job's required skills. Missing skills are presented alongside curated course links from platforms such as Coursera, Udemy, and NPTEL.

G. Recruiter Portal & Candidate Ranking

For each job posting, the recruiter dashboard presents a ranked list of candidates ordered by their match score. Filters by experience range, location, and specific skill requirements are available.

Recruiters can shortlist, schedule, or reject candidates directly through the portal.

H. Career Analytics Dashboard & Reports

Aggregated analytics covering job market trends, in-demand skills, salary range distributions, and application success rates are presented through interactive charts. Administrators can export reports in PDF and Excel formats.

VII. METHODOLOGY

The development methodology follows a model-driven, iterative pipeline. The process began with a structured problem analysis phase, in which the key failure modes of existing keyword-based portals were catalogued through a review of industry practices and the information retrieval literature. This phase identified five core gaps: absence of semantic skill understanding, no structured skill extraction, reactive-only job discovery, no role-aware information delivery, and no candidate self-service career development.

A feature selection phase followed, drawing on findings from the recommendation systems literature to identify the candidate attributes most consistently predictive of job match: extracted skill set, years of experience, educational qualification level, job function preference, and location preference. These features form the input vector for both CBF and CF components of the recommendation engine.

Model training for the collaborative filtering component used an interaction matrix built from application logs, click-through data, and recruiter shortlist events. Matrix factorisation using Singular Value Decomposition (SVD) decomposed this sparse matrix into latent candidate and job factor vectors, enabling similarity computation in the latent space. The NLP skill extraction pipeline was evaluated on a manually annotated dataset of 500 resumes, with precision, recall, and F1 score computed on skill entity spans.

Evaluation of the recommendation engine used standard information retrieval metrics: Precision@K, Recall@K, and Normalised Discounted Cumulative Gain (NDCG), which accounts for the rank ordering of relevant results. The BERT embedding model used was "all-MiniLM-L6-v2" from the Sentence Transformers library, selected for its balance of semantic quality and inference latency.

VIII. RESULTS & DISCUSSION

The NLP skill extraction engine achieved a precision of 91% and recall of 87% on the held-out resume annotation dataset, with an F1 score of 0.89. The custom skill taxonomy pattern matcher contributed a 6-point recall improvement over vanilla spaCy NER, confirming the importance of domain-specific entity definitions for the technical skill extraction task.

Table III presents the recommendation quality metrics for the three system variants evaluated: TF-IDF only (CBF baseline), BERT-only CBF, and the full Hybrid (CBF + CF) system.

TABLE III

Recommendation Quality Metrics

System Variant	P@5	R@10	NDCG	Acc.
TF-IDF CBF (Baseline)	0.74	0.71	0.79	81%
BERT-only CBF	0.83	0.80	0.86	88%
Hybrid CBF + CF (Proposed)	**0.88**	**0.85**	**0.91**	**93%**

The full hybrid system outperformed the TF-IDF baseline by 14 percentage points on Precision@5 and 14 points on Recall@10, confirming that semantic encoding and collaborative signal both contribute independent information to the recommendation task. The BERT-only CBF model substantially closed the gap relative to TF-IDF, validating the use of transformer embeddings for skill-level semantic matching.

User evaluation involving 50 candidate participants and 20 recruiter participants confirmed high perceived utility: 87% of candidates rated the personalised job feed as more relevant than their experience with conventional portal search, and 82% of recruiters rated the AI-ranked candidate list as more efficient than manual resume screening.

IX. BENEFITS

The primary benefit of the proposed system is the elimination of the keyword-matching bottleneck that causes both false positives and false negatives in conventional portals. By operating on semantic skill representations, the system achieves relevance levels that keyword search cannot approach.

For candidates, the personalised job feed and skill gap analysis transform the job search experience from passive browsing into an active, guided career development process. Candidates know not only which jobs they qualify for today but also what they would need to learn to qualify for higher-value roles. For recruiters, the candidate ranking portal compresses resume screening from hours to minutes. For institutions deploying the system as a placement cell platform, the analytics dashboard provides actionable intelligence on which skills produce the best placement outcomes, enabling curriculum alignment with current market demand.

X. ETHICAL AND PRACTICAL CONSIDERATIONS

The use of machine learning to score and rank candidates raises legitimate ethical concerns that the system design explicitly addresses. Algorithmic scoring can disadvantage candidates from specific demographic groups if the training interaction data reflects historical biases. The system mitigates this through regular bias audits comparing match score distributions and false positive rates across candidate subgroups.

Data privacy is enforced through role-level access control: no user can access data outside their defined scope. Candidate skill profiles and interaction logs are stored within the institution's infrastructure and are not shared with third parties. The system complies with applicable data protection requirements, including consent mechanisms for the collection and processing of career and behavioural data.

XI. FUTURE DIRECTIONS

Several directions offer meaningful extensions to the current system. Deep learning models, particularly Graph Neural Networks operating over skill ontology graphs, could capture higher-order skill relationships that pairwise cosine similarity cannot express. Integration with professional networks and real-time job aggregation APIs would extend data coverage beyond manually entered postings.

Multilingual skill extraction, using multilingual BERT or XLM-RoBERTa, would extend the system's applicability to non-English resume markets, particularly relevant for Indian regional language contexts. Mobile application support would increase engagement particularly among student users. Federated learning approaches could enable multi-institution deployment with shared model improvement while preserving institution-level data privacy.

XII. CONCLUSION

Skill-based job mismatches in the employment market are rarely caused by a genuine absence of qualified candidates. They are the product of an information-matching failure: systems that cannot read between the lines of a resume, cannot understand that two different phrases mean the same skill, and cannot proactively surface the right opportunity to the right person. This paper has presented a Skill-Based Job Recommendation and Matching System that addresses this failure through the application of NLP skill extraction, transformer-based semantic embeddings, and a hybrid recommendation engine.

The system demonstrated a Match Accuracy of 93% and an NDCG of 0.91, substantially outperforming TF-IDF and BERT-only baselines. User evaluations confirmed high perceived utility among both job seekers and recruiters. The skill

gap analysis module extends the system beyond recommendation into career development intelligence, while the recruiter ranking portal directly reduces the human cost of talent acquisition.

The proposed system does not claim to eliminate the human judgement required for hiring decisions. It claims to ensure that human judgement is applied to the right candidates, at the right time, with the right information — a more tractable and impactful goal than full automation.

REFERENCES

[1] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," Communications of the ACM, vol. 35, no. 12, pp. 61–70, 1992.

[2] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," IEEE Computer, vol. 42, no. 8, pp. 30–37, 2009.

[3] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in Recommender Systems Handbook, Springer, 2011, pp. 73–105.

[4] L. Galke, F. Mai, A. Schelten, D. Brunsch, and A. Scherp, "Using titles vs. full-text as source for automated semantic document annotation," in Proc. K-CAP 2017, pp. 1–8.

[5] A. Javed, M. S. Younis, M. Latif, J. Qadir, and A. Bilal, "Benchmarking named entity recognition systems on resume extraction," IEEE Access, 2020.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL 2019, pp. 4171–4186.

[7] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proc. EMNLP 2019, pp. 3980–3990.

[8] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734–749, 2005.

[9] R. Burke, "Hybrid recommender systems: Survey and experiments," User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 331–370, 2002.

[10] W. Shalaby et al., "Look-alike candidate generation at LinkedIn," in Proc. KDD '19 Workshop, 2019.

