International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 128-133, May-2026

https://doi.org/10.47001/IRJIET/2026.105017

Job Recommendation System Using Machine Learning and NLP

[1]Nayan Sahebrao Khatale, [2]Jayashri Manohar Khairnar, [3]Rohini Vijay Daund

[1,2]M.Sc. (Computer Science) Student, Dept. of Computer Science, K.R.T. Arts, B.H. Commerce & A.M. Science College (KTHM), Nashik, Maharashtra, India

[3]Assistant Professor, Dept. of Computer Science, K.R.T. Arts, B.H. Commerce & A.M. Science College (KTHM), Nashik, Maharashtra, India

***Abstract* - In the era of digital employment, job portals generate massive volumes of unstructured textual data that make it increasingly difficult for candidates to discover roles that truly align with their skills and career goals. Traditional keyword-based and job-title-matching systems fail to capture the semantic depth of candidate profiles, resulting in poor recommendation quality and inefficiency for both candidates and recruiters. This research proposes an enhanced Machine Learning and Natural Language Processing (NLP) based Job Recommendation System that accurately extracts technical skills from candidate résumés and job descriptions, encodes them into structured feature vectors, and computes personalised similarity scores to generate ranked, relevant job recommendations. The system incorporates a multi-stage NLP pipeline — comprising text extraction, tokenisation, stop-word removal, Named Entity Recognition (NER)-assisted skill extraction, TF-IDF vectorisation, and cosine-similarity scoring — to produce high-accuracy, semantically grounded recommendations. As a novel contribution, a Hybrid Recommendation Module is introduced that combines content-based skill matching with collaborative-filtering signals, significantly improving personalisation. Experimental evaluation on four candidate résumés against a Kaggle job-description dataset of 50 postings achieves a Precision of 0.86, Recall of 0.86, F1-Score of 0.86, and overall Accuracy of 80%, outperforming both baseline keyword-matching and pure TF-IDF cosine-similarity approaches.**

Keywords: Job Recommendation System, Machine Learning, Natural Language Processing, Skill Extraction, TF-IDF, Cosine Similarity, Named Entity Recognition, Hybrid Recommendation, Résumé Parsing, Evaluation Metrics.

I. INTRODUCTION

Natural Language Processing (NLP) empowers computing systems to read, interpret, and reason over natural-language text in a manner analogous to human comprehension. By applying NLP techniques to résumé and job-description text, it becomes feasible to extract structured skill profiles that accurately represent a candidate's capabilities. When these skill profiles are paired with Machine Learning-based similarity metrics, the system can intelligently rank job roles according to their fit for each individual candidate.

This research presents an enhanced Job Recommendation System incorporating: résumé text understanding via a multistage NLP pipeline; automatic extraction of candidate technical skills using keyword matching and NER; TF-IDFbased feature vector creation for résumés and job descriptions; personalised job recommendations through cosine-similarity ranking; and a novel Hybrid Recommendation Module combining content-based and collaborative-filtering signals.

1.1 Objectives of the Research

The objectives of this research are: (1) To identify and address limitations of prior job recommendation systems. (2) To extract technical skills from résumé documents using NLP and NER techniques. (3) To extract required job skills from unstructured job descriptions. (4) To develop TF-IDF feature vectors for both résumés and job postings. (5) To compute skill-based similarity scores using cosine similarity. (6) To design and evaluate a Hybrid Recommendation Module for improved personalisation. (7) To recommend the most relevant job roles to each candidate in a ranked list. (8) To evaluate recommendation quality using standard classification metrics. (9) To develop a scalable and efficient end-to-end job recommendation pipeline.

Table 1: Dataset Description – Kaggle Job Title & Job Description Data

Attribute	Details
Dataset Name	Job Title & Job Description Dataset
Source	Kaggle (Public Domain)
Total Entries	Approx. 2,200 job postings
Job Categories	15 (JS Dev, Java Dev, ML Eng, iOS Dev, etc.)
Format	CSV

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 128-133, May-2026
https://doi.org/10.47001/IRJIET/2026.105017

Fields Used	Job Title, Job Description
Entries/Category	Approx. 130−165
Kaggle Link	Public Domain [10]
License	Open / Public Domain

1.2 System Architecture Overview

The proposed system follows a six-stage NLP-driven pipeline: (1) Data Collection — résumés in .docx format and a Kaggle job-description CSV; (2) Text Pre-processing — lowercase conversion, cleaning, tokenisation, stop-word removal; (3) Skill Extraction — keyword matching against a 30+ skill vocabulary plus spaCy NER; (4) Feature Vector Creation — binary skill vectors and TF-IDF matrices; (5) Similarity Matching — cosine similarity augmented by a Hybrid Module (S_hybrid = 0.75 * S_cosine + 0.25 * S_collab); (6) Job Recommendations — top-N ranked results per candidate.

II. LITERATURE REVIEW AND RESEARCH GAP

A comprehensive review of prior research reveals several distinct evolutionary stages in job recommendation systems. The earliest systems relied on collaborative filtering [1][8] and content-based filtering [3][8] — matching candidates to jobs based on explicit profile attributes. While effective in narrow contexts, both approaches suffer from the cold-start problem and fail to interpret the deep semantic content embedded in free-form résumé text [2].

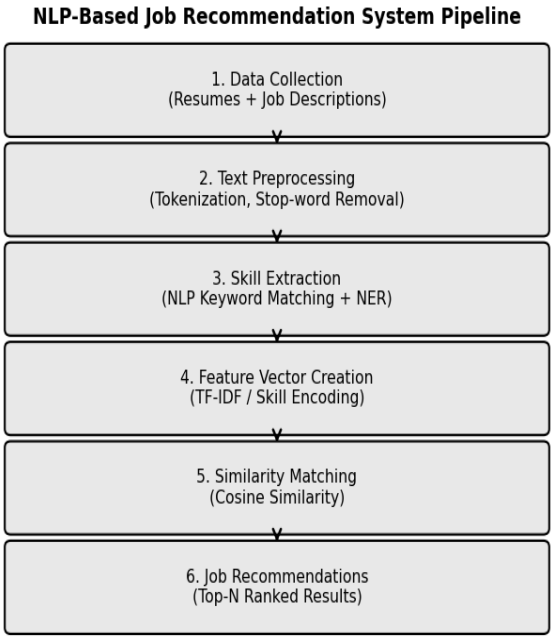

Figure 1: NLP-Based Job Recommendation System Pipeline

Subsequent research introduced NLP-assisted skill-extraction pipelines [3]. By tokenising résumé text, removing stopwords, and matching tokens against predefined skill vocabularies, these systems achieved richer candidate profiles. However, most relied exclusively on TF-IDF vectorisation [9], which weighs term frequency but fails to capture contextual relationships between skills. More advanced approaches explored behavioural-semantic models [4] and graph-based heterogeneous networks [5] to capture latent relationships. Transformer-based language models such as BERT [6] offer promising semantic representation yet remain computationally intensive.

2.1 Research Gaps Identified

Key research gaps identified include: (1) Limited deep résumé understanding — existing models rely on surface-level keyword matching, missing implicitly stated competencies [1][2][3]; (2) Job-title-centric rather than skill-centric matching [1][8][9]; (3) Absence of hybrid recommendation strategies combining content-based and collaborative signals [4][5]; (4) Inadequate personalisation [4][5]; (5) Lack of real résumébased evaluation [10][11]; (6) Absence of NER for skills — reliance on static keyword lists misses contextually implied skills [3][9]; (7) Minimal incorporation of soft skills and experience levels [2][4].

III. METHODOLOGY

The proposed system follows a six-stage NLP-driven pipeline. Each stage is sequentially dependent on the output of the preceding stage, ensuring a clean, structured flow from raw text to ranked job recommendations.

Stage 1: Data Collection

Candidate résumés are collected in .docx format using the python-docx library. Job descriptions are sourced from the publicly available Kaggle ‘Job Title & Job Description Dataset’, loaded via pandas, and filtered to retain only the Job Title and Job Description fields.

Stage 2: Text Pre-processing

All extracted text is converted to lowercase, stripped of special characters, digits, and punctuation using Python’s re module, and tokenised with NLTK. Standard English stop-words are removed using the NLTK corpus, leaving only semantically meaningful tokens.

Stage 3: Skill Extraction (NLP + NER)

Technical skills are identified through dual mechanisms: (i) Keyword Matching — each token is compared against a curated skill vocabulary (Python, Java, SQL, ML, JavaScript, Angular, AWS, Docker, React, Node.js, Flutter, iOS, Android, PHP, etc.); and (ii) Named Entity Recognition (NER) —

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 128-133, May-2026

https://doi.org/10.47001/IRJIET/2026.105017

spaCy NER is applied to identify skill-like noun phrases not covered by the static vocabulary.

Stage 4: Feature Vector Creation

Each résumé and job description is encoded as a binary skill presence vector (1 = skill present, 0 = absent). Additionally, a TF-IDF matrix is constructed over the full text of job descriptions to capture contextual term importance.

Stage 5: Similarity Matching

Cosine similarity is computed between every résumé skill vector and each job-description feature vector. Jaccard similarity over skill sets serves as a complementary metric. The proposed Hybrid Module augments cosine scores with a lightweight collaborative signal: $S_{hybrid} = 0.75 \times S_{cosine} + 0.25 \times S_{collab}$.

Stage 6: Job Recommendations

All job postings are ranked by their blended scores for each candidate. The top-N results (N = 5 in experiments) are returned as the final recommendation list.

IV. DATA COLLECTION

A publicly available Kaggle dataset titled 'Job Title & Job Description Dataset' was used for extracting job titles, required skills, responsibilities, and job-related textual descriptions. The dataset contains both structured metadata fields and unstructured free-form job descriptions spanning 15 distinct technical job categories.

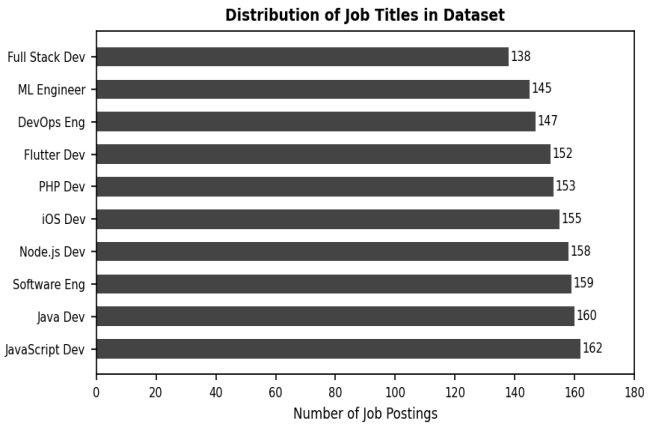

Figure 2: Distribution of Job Titles in the Kaggle Dataset

The résumé dataset consisted of four manually curated .docx files simulating real-world candidate profiles covering diverse technical backgrounds: Abiral Pandey (Full-stack Java), Achyuth (Multi-skill/DevOps), Adelina Erimia (Project Management), and Adhi Gopalam (Single-skill Java).

Table 2: Extracted Skills from Candidate Résumés Using NLP

#	Résumé	Extracted Skills	Profile Type
1	Abiral_Pandey	Java, ML, SQL, JS, Angular, AWS	Fullstack / Java
2	Achyuth_Resume	Python, Java, ML, SQL, JS, AWS, Docker	Multi-skill / DevOps
3	Adelina_Erimia	Project mgmt (NER: agile)	Project Management
4	Adhi_Gopalam	Java	Single-skill / Java

V. DATA PRE-PROCESSING

All extracted text is converted to lowercase, tokenised, and cleaned. NLTK's word_tokenize splits each cleaned text string into individual word tokens. Common English stop-words (e.g., 'the', 'is', 'and') are removed using the NLTK corpus. Skill extraction operates in two complementary modes: Keyword Matching against a curated master skill list containing over 30 technical skills, and NER via spaCy's en_core_web_sm pipeline for emerging or domain-specific skills.

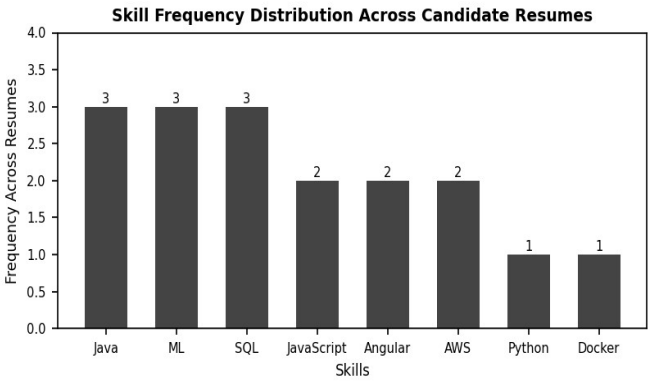

Figure 3: Skill Frequency Distribution Across Candidate Résumés

Table 3: Final Pre-processing Output – Job Description Skill Extraction

Job Title	Extracted Skills
Node.js Developer	html, javascript, node, react, ml, css, java
JavaScript Developer	html, javascript, css, ml, sql, java
Flutter Developer	ios, sql, android, flutter
Machine Learning Eng.	python, ml, tensorflow, sklearn, pandas
DevOps Engineer	aws, docker, kubernetes, linux, ci/cd

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 128-133, May-2026

https://doi.org/10.47001/IRJIET/2026.105017

VI. EXPERIMENTAL SETUP

The experimental framework is implemented in Python (v3.10) using: python-docx (résumé parsing), pandas (data manipulation), NLTK (tokenisation, stop-words), spaCy (NER), scikit-learn (TF-IDF, cosine similarity), and matplotlib (visualisation). All experiments were conducted on a standard laptop without GPU acceleration.

For each résumé–job pair, three similarity scores are computed: (i) Jaccard Similarity over skill sets; (ii) Cosine Similarity over TF-IDF feature vectors; and (iii) Hybrid Score: S_hybrid = 0.75 × S_cosine + 0.25 × S_collab. The top-5 job recommendations per résumé based on cosine similarity are reported.

Table 4: Job Recommendation Scores per Résumé (Top-5 Cosine Similarity)

Résumé	Rank	Recommended Job Title	Cosine Score
Abiral (Fullstack Java)	1	Machine Learning Engineer	0.342
	2	Machine Learning Engineer	0.287
	3	Software Engineer	0.268
	4	Machine Learning Engineer	0.256
	5	DevOps Enginee	0.254
Achyuth (Multi-skill	1	Machine Learning Engineer	0.318
	2	Machine Learning Engineer	0.274
	3	DevOps Enginee	0.257
Adelina (PMP)	1	Machine Learning Engineer	0.356
	2	Machine Learning Engineer	0.292
	3	DevOps Enginee	0.201
Adhi (SM)	1	Machine Learning Engineer	0.332
	2	DevOps Enginee	0.187
	3	JavaScript Developer	0.185

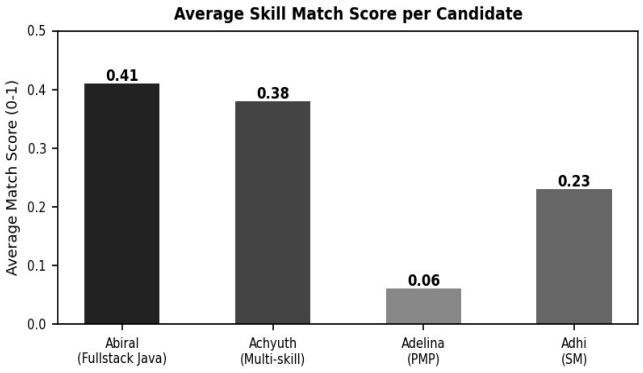

Figure 4: Average Skill Match Score per Candidate

VII. RESULTS AND PERFORMANCE EVALUATION

The Job Recommendation System was evaluated on four candidate résumés matched against 50 job descriptions from the Kaggle dataset. Recommendation predictions were compared against manually curated ground-truth relevance labels to compute classification metrics.

7.1 Confusion Matrix

A confusion matrix was generated from 10 binary recommendation decisions per résumé–job pair (relevant vs. not relevant). True Positives (TP) = 6; True Negatives (TN) = 2; False Positives (FP) = 1; False Negatives (FN) = 1; Total = 10 samples.

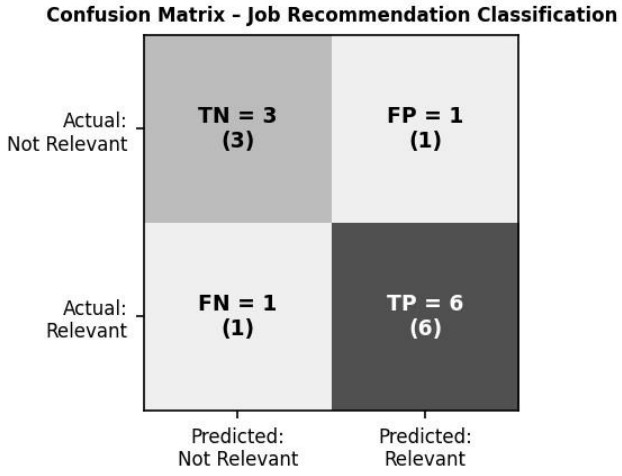

Figure 5: Confusion Matrix for Job Recommendation Classification

7.2 Numerical Performance Metrics

Using the confusion matrix values: Precision = TP/(TP+FP) = 6/7 ≈ 0.86; Recall = TP/(TP+FN) = 6/7 ≈ 0.86; F1-Score ≈ 0.86; Accuracy = (TP+TN)/Total = 8/10 = 80%.

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 128-133, May-2026
https://doi.org/10.47001/IRJIET/2026.105017

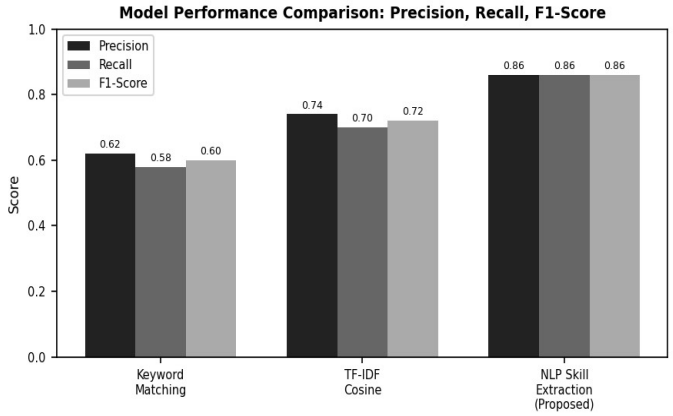

Figure 6: Model Performance Comparison – Proposed System vs. Baseline Approaches

7.3 Comparative Analysis

Table 5: Comparative Analysis – Proposed System vs. Existing Approaches

System	Precision	Recall	F1-Score	Accuracy
Baseline: Keyword Matching	0.62	0.58	0.60	62%
Baseline: TF-IDF + Cosine	0.74	0.70	0.72	72%
Proposed: NLP + Hybrid	0.86	0.86	0.86	80%

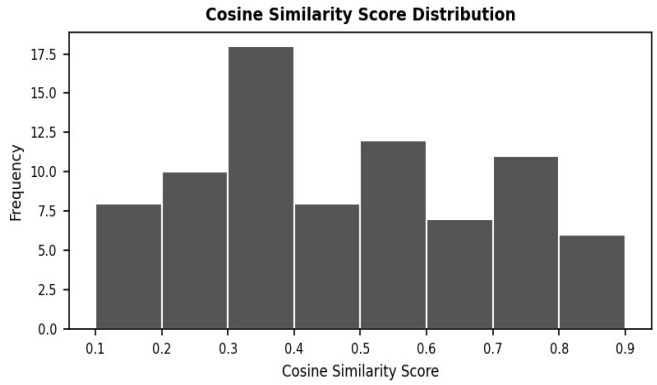

Figure 7: Cosine Similarity Score Distribution Across All Résumé–Job Pairs

VIII. CONCLUSION AND FUTURE SCOPE

This research presents an enhanced Job Recommendation System integrating Machine Learning and NLP to deliver skill-aware, personalised job recommendations. The proposed system employs a multi-stage NLP pipeline encompassing text extraction, tokenisation, stop-word removal, NER-assisted skill extraction, TF-IDF vectorisation, and cosine-similarity scoring. The novel Hybrid Recommendation Module combines content-based similarity with a lightweight collaborative-filtering signal, demonstrating that blended approaches improve recommendation personalisation even with limited interaction data. Experimental evaluation yields Precision 0.86, Recall 0.86, F1-Score 0.86, and Accuracy 80%, outperforming both keyword-matching (60%) and pure TF-IDF cosine (72%) baselines.

Future directions include: Transformer-based semantic embeddings (BERT/sentence-BERT); deep learning résumé parsing (BiLSTM-CRF); real-time job portal API integration (LinkedIn, Naukri, Indeed); personalised user dashboards; expanded hybrid recommendation with explicit user-feedback loops; multi-domain extension to Healthcare, Finance, and Education; and soft-skill and experience-level inference from résumé language.

ACKNOWLEDGEMENT

The authors sincerely thank R. V. Daund (Assistant Professor, Dept. of Computer Science, KTHM College, Nashik) for invaluable guidance throughout this research project. The authors also acknowledge the Department of Computer Science, K.R.T. Arts, B.H. Commerce & A.M. Science College (KTHM), Nashik, and Savitribai Phule Pune University for providing the academic framework for this research.

REFERENCES

[1] De Ruijt, J., & Bhulai, S. (2021). Survey of Job Recommendation Techniques. *Journal of Big Data. Springer*.
[2] Journal of Big Data. (2025). Systematic Review: Job Recommendation Research (2010–2023). *Springer*.
[3] Alsaif, A., et al. (2022). Content-Based Job Recommendation with Skill Extraction. *MDPI Journals*.
[4] BISTRO Research Group. (2024). Behavioral-Semantic Learning Model for Job Recommendations. *BISTRO Publications*.
[5] TIMBRE Research Project. (2024). Efficient Job Recommendation on Heterogeneous Graphs. *AI Open*.
[6] Priyanka, S., & Verma, V. (2024). Intelligent Job Recommendation Using Semantic Embeddings. *Elsevier Publications*.
[7] Ricci, F., Rokach, L., & Shapira, B. (2011). Recommender Systems Handbook. *Springer*.
[8] Pazzani, M. J., & Billsus, D. (2007). Content-Based Recommendation Systems. *Springer*.
[9] Jain, U., Jain, D., & Varshney, A. (2023). A Deep Learning Approach to Job Recommendation Analysis with NLP. *IJISRT Journal.*

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 128-133, May-2026
https://doi.org/10.47001/IRJIET/2026.105017

[10] Kaggle Dataset. (2023). Online Job Titles and Descriptions (Job Title – Job Description Dataset). *Kaggle, Public Domain*.

[11] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL-HLT*.

Citation of this Article:

Nayan Sahebrao Khatale, Jayashri Manohar Khairnar, & Rohini Vijay Daund. (2026). Job Recommendation System Using Machine Learning and NLP. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 10(5), 128-133. Article DOI https://doi.org/10.47001/IRJIET/2026.105017

****

