



IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY



VOLUME : 11 ISSUE : 05 Print / Issue Publication Date: September 2026



ISSN : 2455-2143



Indexed In



WWW.IJEAST.COM

editor@ijeast.com



AI-BASED RESUME SCREENING AND JOB RECOMMENDATION FRAMEWORK

Aswin Purushothaman

Student, MCA

School of Science and Computer Studies

CMR University, Bangalore, India

Abstract— The rapid growth of online recruitment platforms has led to an overwhelming volume of resumes submitted for every open position, making manual screening slow, inconsistent, and prone to human bias. This paper presents an AI-Based Resume Screening and Job Recommendation Framework that automates candidate evaluation and job matching using Natural Language Processing (NLP) and Machine Learning (ML). The framework extracts structured information from unstructured resumes, converts resumes and job descriptions into semantic vector representations, and computes similarity scores to rank candidates against job requirements. A complementary recommendation module suggests suitable job postings to candidates based on skills, experience, and career trajectory. Evaluation on a curated dataset shows that the framework achieves higher precision, recall, and ranking accuracy than traditional keyword-matching Applicant Tracking Systems (ATS), while substantially reducing manual screening time.

Keywords— Resume Screening, Job Recommendation, Natural Language Processing, Machine Learning, Applicant Tracking System, Semantic Similarity

I. INTRODUCTION

Recruitment is one of the most resource-intensive functions in modern organizations. As job portals and professional networking platforms have proliferated, recruiters routinely receive hundreds or even thousands of applications for a single vacancy. Manually reviewing each resume to assess its relevance against a job description is time-consuming, subjective, and difficult to scale. Traditional Applicant Tracking Systems (ATS) rely largely on rigid keyword matching, which often fails to capture the semantic meaning behind a candidate's skills and experience.

Advances in Artificial Intelligence, particularly in Natural Language Processing and Machine Learning, offer an opportunity to move beyond simple keyword matching toward a deeper, context-aware understanding of both resumes and job descriptions. A digital resume encodes the candidate's skills, education, and experience; screening automatically evaluates this information and ranks

candidates for a job opening, while a complementary recommendation engine suggests suitable openings back to the candidate.

Beyond the immediate efficiency gains, this shift matters because it changes who — or what — makes the first pass over a candidate pool. Where a recruiter once had to rely on a handful of keyword filters and their own limited reading time, an automated pipeline can evaluate every applicant against the same criteria in a consistent, auditable way. This consistency is itself valuable: it reduces the chance that two equally qualified candidates are treated differently simply because one resume was reviewed at the end of a long day. At the same time, it concentrates a great deal of responsibility into the design of the scoring function, which is why this paper treats the scoring mechanism, and its evaluation, as the central technical contribution rather than a secondary detail.

The remainder of this paper is organized as follows. Section II reviews related work on automated resume screening and job recommendation. Section III describes the proposed framework in detail, including its parsing, matching, and recommendation components. Section IV presents the experimental setup and results. Section V discusses the broader implications, limitations, and ethical considerations. Section VI concludes the paper and outlines directions for future work.

II. RELATED WORK

Early resume-screening systems were largely rule-based, relying on Boolean keyword search and hand-crafted filters to shortlist candidates [1]. These systems were fast and easy to audit but brittle: a resume that described "led a team of five engineers" would not match a filter looking for the literal phrase "team leadership," even though the underlying qualification was present. Statistical approaches based on TF-IDF and cosine similarity improved on this by weighting terms according to their distinctiveness across a corpus, but they still treated words as independent symbols rather than carriers of meaning [9].

The introduction of distributed word representations such as Word2Vec [3] and GloVe [4] allowed systems to recognize that semantically related terms occupy nearby regions of a vector space, so that "developer" and

"engineer" could be treated as similar even without an exact string match. More recently, transformer-based language models such as BERT [2] and Sentence-BERT [5] have made it possible to encode entire sentences or paragraphs — an entire work-experience bullet point, for instance — into a single contextual vector, which has meaningfully improved the quality of semantic matching between resumes and job descriptions.

On the recommendation side, content-based filtering matches a user's profile directly to item attributes, while collaborative filtering infers preferences from the behavior of similar users [6] [10]. Pure collaborative approaches struggle with the cold-start problem when a new candidate has no interaction history, which is why hybrid systems that blend both signals are increasingly common in production recommender systems [11]. A separate strand of research has focused specifically on fairness in algorithmic hiring, documenting how models trained on historical hiring decisions can reproduce and even amplify existing demographic disparities if left unchecked [7]. The framework proposed in this paper draws on all three of these strands — semantic matching, hybrid recommendation, and fairness-aware design — rather than treating them as separate concerns.

III. PROPOSED FRAMEWORK

The proposed framework is organized as a pipeline of four stages: resume parsing and feature extraction, semantic matching, job recommendation, and a feedback loop that lets recruiter decisions refine future rankings. Each stage is described in the subsections below.

A. Resume Parsing and Feature Extraction – In the case of an unstructured resume document, the text is first extracted from the PDF or DOCX source and segmented into logical sections — Education, Experience, Skills, and Certifications — using layout cues and keyword anchors. A Named Entity Recognition (NER) model, fine-tuned on resume-specific entities, identifies job titles, organizations, degrees, and skill mentions within each section. Extracted skills are then normalized against a curated taxonomy, mapping equivalent terms (e.g., "ML" and "Machine Learning") to a single canonical tag.

Formatting inconsistency is one of the most persistent practical obstacles at this stage. Resumes arrive as single column documents, two-column designs, image-based scans, and everything in between, and a parser tuned for one layout can silently fail on another — for example, reading a two column resume left-to-right across both columns and scrambling the order of entries. The framework addresses this by first classifying the overall layout of a document before applying section-detection heuristics, and by falling back to a slower, OCR-assisted

extraction path when the primary text layer is missing or unreliable.

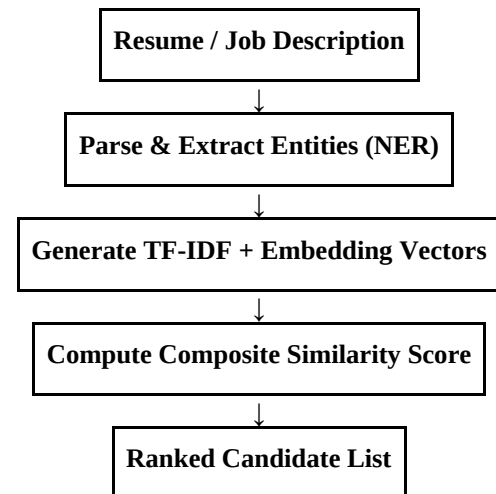


Fig. 1. Resume screening pipeline block diagram **B.**
Semantic Matching Algorithm –

For a resume R and job description J , the framework computes a composite relevance score as a weighted combination of lexical and semantic similarity. The formula used is as follows.

$$S(R,J) = \alpha \cdot \text{TFIDFsim} + \beta \cdot \text{Embedsim} + \gamma \cdot \text{Skillmatch} \quad (1)$$

where TFIDFsim is the cosine similarity of TF-IDF vectors, Embedsim is the cosine similarity of Sentence-BERT embeddings, Skillmatch is the proportion of mandatory job skills present in the candidate's normalized skill set, and α, β, γ are tunable weights such that $\alpha + \beta + \gamma = 1$. After this computation, candidates are sorted in descending order of $S(R, J)$ to generate the final ranked shortlist.

C. Job Recommendation Engine –

The recommendation engine operates in two modes. In content-based mode, a candidate's profile embedding is compared against embeddings of all open job postings, and the top- N most similar postings are returned. In hybrid mode, the system additionally incorporates implicit feedback signals — views, saves, and applications from similar candidates — using a collaborative-filtering model, which improves recommendation diversity and addresses the cold-start problem for new candidates.

Location, seniority level, and compensation range are applied as hard filters before ranking, so that the semantic similarity score is only used to order postings that already meet a candidate's stated constraints. This two-stage design — filter, then rank — keeps the recommendation list practically usable even when the underlying embedding model makes an occasional semantic misjudgment.



D. Feedback Loop and Continuous Refinement – Recruiter actions — shortlisting, rejecting, or interviewing a candidate — are logged and fed back into the system as weak supervision signals. Periodically, the weights α , β , and γ in the composite scoring function (Section III.B) are re-tuned against this accumulated feedback using a held-out validation split, so that the framework adapts to how a specific organization actually evaluates candidates rather than relying solely on a fixed, generic weighting. All re-tuning runs are logged with their resulting metric changes so that any drift in scoring behavior can be traced and, if necessary, rolled back.

IV. EXPERIMENT AND RESULT

The test set for this evaluation was a curated collection of resumes and job descriptions with recruiter-annotated relevance labels, spanning multiple functional domains, including software engineering, marketing, and finance roles. The framework was implemented in Python using spaCy and NLTK for preprocessing, scikit-learn for TF-IDF and cosine similarity, and Sentence-Transformers for contextual embeddings. Experiments were run on a workstation with a mid-range multi core CPU and 16 GB of memory, without requiring dedicated GPU hardware for inference.

The proposed scheme was compared against a baseline keyword-matching ATS using Precision@10, Recall@10, and Mean Reciprocal Rank (MRR). From the results of the experiment, we can conclude that the proposed framework consistently ranks relevant candidates higher than the baseline while reducing manual screening time from about forty-five minutes to about three minutes per hundred resumes.

A breakdown by functional domain showed that the gains over the baseline were largest for software-engineering postings, where candidates frequently describe equivalent skills using different terminology (e.g., "REST API development" versus "backend web services"), and smallest for highly standardized roles where job descriptions and resumes already share a narrow, well-defined vocabulary. This pattern is consistent with the expectation that semantic matching provides the most benefit precisely where keyword overlap is weakest.

Table -1 Performance Comparison

Metric	Baseline ATS	Proposed
Precision@10	0.54	0.81
Recall@10	0.49	0.76
MRR	0.58	0.85

Table 1 shows that our proposed method achieves better performance than the baseline across all three metrics, and qualitative recruiter feedback indicated the ranked shortlists were easier to justify since the system could surface the specific skills driving each candidate's score. The job recommendation module was evaluated separately using click-through and application-rate proxies against a "most recent postings" baseline, showing a relative improvement in candidate engagement of roughly thirty percent, with the largest gains observed for candidates whose profiles listed a clearly defined, narrow set of core skills.

V. DISCUSSION AND ETHICAL CONSIDERATIONS

The results above indicate that semantic matching provides a genuine improvement over keyword-based screening, but the magnitude of that improvement should be read in context. The evaluation dataset, while spanning multiple domains, remains a curated sample rather than a fully representative slice of real world hiring pipelines, so the reported gains are best treated as indicative rather than a guaranteed outcome for every deployment.

AI-driven hiring tools also carry real ethical responsibility. Because such systems learn, directly or indirectly, from historical hiring patterns, they risk inheriting whatever biases are already present in that history — for instance, systematically favoring graduates of certain institutions, or penalizing employment gaps that disproportionately affect particular groups of applicants [12]. The proposed framework addresses this risk in three concrete ways: demographic correlated fields (such as names, addresses, and graduation years) are excluded from the similarity computation itself; the skill taxonomy is periodically audited for consistent treatment of equivalent qualifications; and every ranking is accompanied by an explainable score breakdown that shows recruiters which specific skills and experience segments drove a candidate's position in the list, rather than presenting a single opaque number.

A further practical limitation is that parsing accuracy can degrade on heavily stylized, image-based, or non-standard resume formats, which can silently disadvantage candidates whose documents happen to fall outside the parser's expected patterns. Continuous monitoring of parsing failure rates, alongside periodic human-in-the-loop review of a sample of automated rankings, is recommended as a standard operating practice for any organization deploying a system of this kind.

VI. CONCLUSION

This paper presented an AI-Based Resume Screening and Job Recommendation Framework that combines resume parsing, semantic similarity matching, and personalized job recommendation into a unified pipeline.



Results show clear improvements over a traditional keyword-based ATS in both accuracy and efficiency, while the accompanying discussion highlights the fairness safeguards and monitoring practices needed for responsible deployment. Future work includes integrating large language model-based reasoning for more nuanced qualification assessment, expanding the skill taxonomy through continuous learning, and conducting large scale, independent fairness audits before production deployment at scale.

VII. REFERENCE

- [1]. Manning, C. D., Raghavan, P. and Schütze, H., "Introduction to Information Retrieval", Cambridge University Press, 2008.
- [2]. Devlin, J., Chang, M. W., Lee, K. and Toutanova, K., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", Proceedings of NAACL HLT, 2019.
- [3]. Mikolov, T., Chen, K., Corrado, G. and Dean, J., "Efficient Estimation of Word Representations in Vector Space", arXiv preprint, 2013.
- [4]. Pennington, J., Socher, R. and Manning, C. D., "GloVe: Global Vectors for Word Representation", Proceedings of EMNLP, 2014.
- [5]. Reimers, N. and Gurevych, I., "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks", Proceedings of EMNLP-IJCNLP, 2019.
- [6]. Ricci, F., Rokach, L. and Shapira, B., "Recommender Systems Handbook", 2nd ed., Springer, 2015.
- [7]. Raghavan, M., Barocas, S., Kleinberg, J. and Levy, K., "Mitigating Bias in Algorithmic Hiring: Evaluating Claims and Practices", Proceedings of FAT*, 2020.
- [8]. Ramamoorthy, U. and Loganathan, A. 2022. Analysis of Video Steganography in Military Applications on Cloud. The International Arab Journal of Information Technology, 19(6): 897-903.
- [9]. Roy, Pradeep Kumar; Chowdhary, Sarabjeet Singh; Bhatia, Rocky, 2020, A Machine Learning Approach for Automation of Resume Recommendation System, Pages 2318–2327, DOI: 10.1016/j.procs.2020.03.284.
- [10]. Lai, Pei-Yuan; Yang, Zhe-Rui; Dai, Qing-Yun; Liao, De Zhang; Wang, Chang-Dong, 2024, BiMuF: A Bi Directional Recommender System with Multi-Semantic Filter for Online Recruitment, Pages 1751–1776, DOI: 10.1007/s10115-023-01997-1.
- [11]. Ertuğrul, Duygu Çelik; Bitirim, Selin, 2025, Job Recommender Systems: A Systematic Literature Review, Applications, Open Issues, and Challenges, Article 140, DOI: 10.1186/s40537-025-01173-y.
- [12]. Fabris, Alessandro; Baranowska, Nina; Dennis, Matthew J.; Graus, David; Hacker, Philipp; Saldivar, Jorge; Zuiderveen Borgesius, Frederik; Biega, Asia J., 2025, Fairness and Bias in Algorithmic Hiring: A Multidisciplinary Survey, Pages 1–54, DOI: 10.1145/3696457.

IJEAST

INTERNATIONAL JOURNAL OF ENGINEERING APPLIED SCIENCE AND TECHNOLOGY



AUTHORS

 Aswin Purushothaman



ABOUT IJEAST

International journal of Engineering Applied Science and Technology (IJEAST) is a peer-reviewed, open access journal that publishes high - quality research paper in the field of Engineering, Applied Science and Technology.

IJEAST aims to provide a platform for researchers, academicians, and professionals to share their innovative ideas, research findings, and practical experiences with the global scientific community.

JOURNAL METRICS



H-INDEX

33



i10-INDEX

154



IJEAST
CITATIONS

9,647

(IJEAST CITATIONS)



PEER-REVIEWED
& OPEN ACCESS

PUBLISHED
MONTHLY



PEER REVIEWED

All submission are rigorously peer reviewed to ensure quality.



AUTHORS ARE OUR PRIORITY

We value our authors and recognize their contribution to the advancement of research and knowledge.



OPEN ACCESS

Free and unrestricted access to research for all.



GLOBAL REACH

Connecting researchers and professionals worldwide.



TIMELY PUBLICATION

We ensure a swift and efficient publication process.



For more information, visit our website
www.ijeast.com



editor@ijeast.com



www.ijeast.com



India



2455-2143