

ARTIFICIAL INTELLIGENCE

A Review of AI-Based Job Portal Systems for Intelligent Job Matching and Recruitment

Ganesh Aradkar, B. Gupta

Department of Information Technology, Siddhant College of Engineering (affiliated to Savitribai Phule Pune University),
Sudumbare, Pune, Maharashtra, India

Published October 2026

Abstract

The growth of digital recruitment has created large volumes of job postings, candidate profiles, resumes, applications, and interaction data. Conventional job portals mainly support job publication, keyword search, filtering, application submission, and employer-side candidate management. Although these functions improve access to employment opportunities, they do not always solve the core matching problem: identifying the most relevant job for a candidate and the most suitable candidate for an employer. Artificial Intelligence (AI), Machine Learning (ML), Natural Language Processing (NLP), and recommender-system techniques provide opportunities to improve this process by understanding skills, experience, qualifications, job descriptions, preferences, and historical interactions. This review paper examines research on job recommender systems, e-recruitment recommendation, semantic job matching, resume screening, deep-learning-based talent recommendation, and algorithmic fairness. The literature indicates a progression from content-based and collaborative filtering toward hybrid, semantic, and deep-learning approaches. Recent studies further emphasize explainability, fairness, semantic embeddings, and large-language-model-assisted recruitment. Based on the reviewed literature, this paper proposes an AI-enabled job portal architecture combining a React Native mobile client, Spring Boot backend, MySQL database, and Python-based AI services. The proposed architecture supports role-based access for job seekers, employers, and administrators; resume processing; skill extraction; semantic candidate-job matching; personalized job recommendations; application tracking; employer notifications; and chatbot assistance. The paper identifies major research gaps involving cold-start users, limited public recruitment datasets, explainability, fairness, dynamic job preferences, and integration of multiple recruitment functions into a practical end-to-end platform. Finally, evaluation metrics and an experimental methodology are proposed for assessing recommendation quality, classification performance, response time, usability, and fairness.

Keywords: Artificial Intelligence; Job Portal; Job Recommender System; Machine Learning; Natural Language Processing; Resume Screening; Semantic Matching; Recommendation Systems; Recruitment; Chatbot

1 Introduction

Employment platforms have become an important part of modern recruitment because they reduce the geographic and temporal limitations of traditional job searching. A digital job portal can allow candidates to create profiles, upload resumes, search vacancies, apply for positions, and monitor application status. Employers can publish vacancies, review applicants, communicate with candidates, and manage recruitment activities. However, the large number of vacancies and applications creates an information-overload problem. A candidate may receive many search results containing jobs that share keywords but differ substantially in required skills, seniority, responsibilities,

location, or career direction. Similarly, an employer may receive hundreds of applications that require substantial manual screening.

Recommender systems address information overload by selecting and ranking items that are likely to be useful to a user. General recommender-system research has established content-based, collaborative, knowledge-based, and hybrid approaches as major families of recommendation techniques [1]–[3]. In recruitment, the recommendation object may be a job for a candidate or a candidate for an employer. This creates a reciprocal and high-impact recommendation problem because recommendations can influence careers as well as organizational hiring decisions [4, 5].

Research on job recommendation has demonstrated the value of using profile information, job-posting content, user behavior, and temporal interaction signals. The 2016 ACM RecSys Challenge provided a large-scale real-world setting in which systems predicted jobs that users would positively interact with using anonymized profiles, job postings, and interaction histories [6]. Subsequent research explored gradient boosting, temporal models, semantic matching, attention-based neural networks, and hybrid recommendation strategies [7]–[10].

The recruitment domain also differs from ordinary product recommendation. A job is not simply an item to be consumed; it represents a possible employment relationship. Suitability is therefore multidimensional and includes skills, qualifications, experience, preferences, location, compensation, availability, and employer requirements. Moreover, historical hiring data can contain social and institutional biases. A system that learns directly from biased historical decisions may reproduce or amplify those patterns. Recent multidisciplinary work therefore treats fairness, transparency, privacy, and governance as central requirements for algorithmic hiring [11].

This paper reviews the relevant literature and uses the findings to design an end-to-end AI-based job portal. The focus is not only on a recommendation algorithm but on the integration of recruitment functions into a practical software architecture. The proposed system is intended to support three principal roles: job seeker, employer, and administrator. The AI layer can provide resume analysis, semantic matching, job recommendation, and chatbot-based assistance while conventional web/mobile and backend services provide authentication, data management, vacancy management, applications, notifications, and administration.

2 Objectives of the review

The objectives of this review are: (1) to study the evolution of job recommender systems and e-recruitment platforms; (2) to compare content-based, collaborative, knowledge-based, hybrid, semantic, and deep-learning approaches; (3) to examine how resumes and job descriptions are represented and matched; (4) to identify evaluation methods and metrics used in recruitment recommendation; (5) to examine fairness, explainability, privacy, and cold-start challenges; and (6) to derive requirements for an integrated AI-based job portal using mobile, backend, database, and AI technologies.

3 Background and related concepts

3.1 Recommender systems

Recommender systems generate personalized suggestions from item attributes, user characteristics, interactions, contextual information, or combinations of these sources. Content-based recommenda-

tion compares a user profile with item characteristics. Collaborative filtering learns from interactions among users and items. Knowledge-based approaches use domain knowledge, constraints, or ontologies. Hybrid systems combine multiple techniques to compensate for individual weaknesses [1]–[3].

3.2 Job recommender systems

Job recommender systems apply recommendation principles to employment. The system can perform candidate-to-job recommendation, job-to-candidate recommendation, or both. Literature reviews show that content-based, collaborative, hybrid, and knowledge-based methods have all been used, while hybrid and non-traditional techniques have become increasingly important [4, 5]. The job domain additionally requires consideration of reciprocal preferences, because a successful employment match should satisfy both candidate and employer perspectives.

3.3 Natural language processing for resumes and job descriptions

Resumes and job descriptions are predominantly unstructured or semi-structured text. NLP can convert this text into structured attributes such as skills, education, job titles, years of experience, certifications, tools, responsibilities, and domains. Traditional pipelines commonly use tokenization, normalization, stop-word removal, stemming or lemmatization, TF-IDF, and vector similarity. Transformer-based models such as BERT provide contextual representations that can capture relationships between words and phrases more effectively than purely keyword-based methods. Recent recruitment research reports BERT-based resume screening and semantic matching as promising approaches [12, 13].

4 Literature review

The literature was organized around five themes: recommendation algorithms, semantic matching and resume analysis, deep learning, real-world evaluation, and responsible AI. Earlier work established the foundations of recommendation and job matching, while more recent work has emphasized semantic representations, hybrid methods, and fairness.

Yi, Allan, and Croft investigated matching resumes and jobs using relevance models, demonstrating that information-retrieval techniques can be applied to recruitment documents [14]. Singh, Catherine, and Visweswariah presented PROSPECT, a system for screening recruitment candidates [15]. Al-Otaibi and Ykhlef surveyed job recommender systems and later proposed artificial-immune and hybrid approaches [16, 17]. Their work illustrates the shift from simple similarity mechanisms toward computational-intelligence approaches.

Chala et al. proposed a bidirectional semantic matching framework in which qualifications and skills are matched against vacancies using occupational and job data [18]. This direction is important because recruitment matching should not be one-directional: candidates need suitable jobs, while employers need suitable candidates. Abel et al. documented the ACM RecSys Challenge 2016, where a large anonymized XING dataset was used for job recommendation [6]. Xiao et al. used a Hawkes-process-based approach for the challenge, incorporating user profiles, job content, and historical activities [7]. Pacuk et al. used candidate preselection followed by gradient-boosting ranking [8]. These studies show the importance of scalable candidate generation followed by ranking.

Freire and de Castro conducted a systematic review of e-recruitment recommender systems covering 2012–2020. They reviewed 896 retrieved papers and included 63 works after selection. Their classification identified content-based, collaborative, knowledge-based, hybrid, and other approaches, and

showed that resumes and job posts were among the most important information sources [4]. De Ruijt and Bhulai reviewed job recommender systems from 2011–2021, emphasizing temporal and reciprocal recommendation and observing that fairness received comparatively limited attention in the literature [5].

Mashayekhi et al. provided a challenge-based survey of e-recruitment recommendation systems and highlighted challenges that are more practical than algorithm categories alone, including data sparsity, cold start, privacy, evaluation, and the complex relationship between candidate and recruiter preferences [9]. A newer systematic review by Çelik Ertuğrul and Bitirim analyzed 57 job recommender research articles and reported frequent use of similarity methods, word/sentence vector models, supervised machine learning, regression/classification approaches, and knowledge-based methods [19]. The review also noted that fairness and efficiency remain important improvement areas.

Deep learning has expanded the ability of systems to model complex relationships. Huang et al. proposed an attentive deep neural network for talent recommendation that considered implicit relationships in resumes and latent preferences beyond explicit job requirements [10]. Career recommendation research has also explored transfer learning and LSTM models for predicting career pathways from employment history [20]. These approaches indicate that recruitment systems can move beyond static keyword matching toward representations of skills, roles, experience, and career trajectories.

Recent research has increasingly focused on semantic embeddings and explainability. A 2025/2026 line of work describes semantic-similarity-based recruitment systems that use contextual representations to identify cross-domain relationships between job and resume content. Recent studies have also combined BERT, Sentence-BERT, OCR, named-entity recognition, and large-language-model prompting for resume analysis and job matching. These developments support a hybrid architecture in which conventional structured data is combined with semantic text representations rather than replacing structured recruitment logic entirely.

Finally, fairness has become a critical research requirement. Algorithmic hiring can potentially reduce subjective inconsistencies, but it can also reproduce historical discrimination when biased data or proxy variables are used. A multidisciplinary survey by Fabris et al. reviewed biases, fairness measures, mitigation strategies, datasets, and legal considerations in algorithmic hiring [11]. Therefore, an AI job portal should separate recommendation assistance from final human decisions, avoid unnecessary sensitive attributes, log recommendation reasons, and evaluate performance across relevant groups where legally and ethically appropriate.

5 Literature review comparison

Table 1 compares the main studies reviewed.

Table 1. Comparison of the reviewed literature.

Study	Year	Main Focus	Technique	Data/Input	Key Limitation / Gap
Yi et al. [14]	2007	Resume-job matching	Relevance models	Resume + job text	Limited semantic/contextual representation
Singh et al. [15]	2010	Candidate screening	Recruitment screening	Candidate profiles	Earlier-generation screening architecture

Study	Year	Main Focus	Technique	Data/Input	Key Limitation / Gap
Al-Otaibi & Ykhlef [16]	2012	Job recommender survey	CBR/CF taxonomy	Job/user information	Limited modern deep-learning methods
Al-Otaibi & Ykhlef [17]	2017	Hybrid recommendation	Artificial immune + CF	Applicant/job information	Complexity and limited modern NLP
Chala et al. [18]	2018	Bidirectional matching	Semantic/ML matching	Occupational + vacancy data	Data integration and deployment complexity
Abel et al. [6]	2016	Large-scale job recommendation	Challenge/recommendation	XING profiles, jobs, interactions	Challenge-specific evaluation
Xiao et al. [7]	2016	Temporal recommendation	Hawkes process	Profiles + job content + history	Focus on interaction prediction
Pacuk et al. [8]	2016	Job ranking	Gradient boosting	User/job/interactions	Ranking task rather than full portal
Freire & de Castro [4]	2021	Systematic review	Literature classification	63 e-recruitment studies	Review rather than implementation
de Ruijt & Bhulai [5]	2021	JRS review	Taxonomy/fairness analysis	JRS literature	Fairness still under-developed
Mashayekhi et al. [9]	2024	Challenge-based review	Challenge taxonomy	e-Recruitment literature	Identifies gaps but not a complete product
Huang et al. [10]	2023	Talent recommendation	Attention + deep learning	Resumes/relationships	Model-centric approach
Fabris et al. [11]	2023/2025	Fair algorithmic hiring	Fairness survey	Hiring systems/datasets	Requires domain-specific deployment work
Deshmukh & Raut [12]	2025	Resume screening	BERT/NLP	Resumes + job descriptions	Small evaluation setting
Çelik Ertuğrul & Bitirim [19]	2025	JRS systematic review	Taxonomy + SLR	57 JRS studies	Limited integrated end-to-end systems
Recent semantic matching work	2026	Semantic job matching	Embedding/similarity	Job-resume pairs	Needs broader real-world validation

6 Research gaps

The literature demonstrates substantial progress, but several gaps remain. First, many studies are model-centric: they optimize matching or ranking without integrating authentication, role management, vacancy management, applications, notifications, administration, and conversational assistance into one deployable platform. Second, many recommendation methods depend on historical interaction data, creating a cold-start problem for new users and newly posted jobs. Third, recruitment data is difficult to obtain because resumes, applications, hiring outcomes, and user behavior contain personal and commercially sensitive information. Consequently, published studies often use private, challenge-specific, simulated, or relatively small datasets.

Fourth, keyword-based and even embedding-based matching can produce opaque scores. A candidate should ideally receive an understandable explanation such as “matched skills: Java, Spring Boot,

SQL; experience alignment: 2 years; location: Pune,” rather than only a numerical similarity score. Fifth, fairness is not solved merely by removing an explicitly sensitive feature. Proxy variables, historical hiring patterns, dataset imbalance, and differences in opportunity can still influence results. Sixth, job preferences are dynamic: candidates may change desired locations, salary ranges, technology stacks, or career goals, while vacancies expire and requirements change.

The central research gap addressed by this project is therefore the design of a practical, modular, AI-assisted job portal that combines structured recruitment operations with NLP-based semantic matching, personalized recommendations, resume analysis, explainability, and conversational assistance. Rather than claiming that one algorithm solves recruitment, the proposed system treats AI as a decision-support layer within a controlled software architecture.

7 Proposed AI-based job portal

7.1 System goals

The proposed portal is designed for three primary roles. Job seekers can register, authenticate, maintain profiles, upload resumes, search jobs, receive recommendations, apply for jobs, view application status, and use an AI chatbot. Employers can register, publish vacancies, review applications, inspect candidate profiles, and receive notifications when candidates apply. Administrators can manage users, jobs, applications, and system-level information. The AI layer provides resume parsing, skill extraction, candidate-job similarity, recommendation ranking, and conversational assistance.

7.2 Functional requirements

- User authentication and role-based authorization.
- Job seeker profile creation and resume upload.
- Employer vacancy creation, editing, and management.
- Job search, filtering, job details, and application submission.
- Resume text extraction and skill identification.
- Semantic candidate-job matching and personalized job recommendation.
- Employer notifications when candidates apply.
- Application tracking and interview-related information.
- Administrative management of users, jobs, and applications.
- AI chatbot for job-search and portal assistance.

7.3 Non-functional requirements

The system should provide security, scalability, maintainability, availability, usability, and acceptable response time. Passwords must be stored using secure hashing, authorization must be enforced server-side, uploaded documents should be validated, and sensitive information should be minimized in AI processing. The AI service should be independently deployable so that the portal can continue to perform basic recruitment functions if the AI service is temporarily unavailable.

8 Proposed architecture

The architecture follows a layered client-server model with an AI microservice. React Native provides the mobile interface. Spring Boot provides authentication, role management, REST APIs, business logic, application processing, job management, and notifications. MySQL stores structured entities such as users, jobs, applications, interviews, and recommendation metadata. Python provides AI/NLP services for resume processing and semantic matching.

The logical architecture is: Mobile Client (React Native) → REST API Layer (Spring Boot) → Business Services / Authentication / Job / Application / Notification → MySQL Database ↔ Python AI Service (NLP / Resume Parsing / Embeddings / Matching / Chatbot) → Recommendation Results and Explanations.

The separation of the AI service from the main backend is useful because model development has different dependencies and deployment requirements from ordinary transactional services. The backend can send a resume and job-description representation to the AI service, receive structured skills and similarity scores, and store only the necessary result. A production implementation should additionally define retention policies and access controls for uploaded resumes.

9 Methodology

9.1 Data collection and preparation

The system requires job-posting data, candidate profile data, resume documents, skills, qualifications, application interactions, and optionally explicit feedback. For an academic prototype, publicly available job datasets, synthetic records, or consented sample resumes may be used. Personally identifying information should be removed or replaced when the data is used for experiments. Job and resume text is normalized before feature extraction.

9.2 Resume processing

The resume pipeline can extract text from PDF/DOCX files and perform cleaning, section identification, tokenization, and entity/skill extraction. Structured fields may include education, experience, skills, certifications, job titles, and projects. A hybrid strategy is recommended: deterministic extraction for highly structured information and NLP embeddings for semantic content.

9.3 Job processing

Each job description is transformed into a structured representation containing title, required skills, preferred skills, experience, education, location, salary range, employment type, and description text. Required and preferred skills should be represented separately because an exact match on a mandatory skill may be more important than a similarity score based on descriptive text.

9.4 Candidate-job matching

A practical baseline can combine explicit skill overlap with semantic similarity. Let S_c be the candidate skill set and S_j be the required job skill set. A simple skill overlap score is:

$$\text{SkillScore} = \frac{|S_c \cap S_j|}{|S_j|} \quad (1)$$

Let E_c and E_j denote semantic embeddings for the candidate profile/resume and job description. Cosine similarity is:

$$\text{SemanticScore} = \frac{E_c \cdot E_j}{\|E_c\| \|E_j\|} \quad (2)$$

A hybrid recommendation score can then be defined as:

$$\text{MatchScore} = w_1 \text{SkillScore} + w_2 \text{SemanticScore} + w_3 \text{ExperienceScore} + w_4 \text{LocationScore} + w_5 \text{PreferenceScore} \quad (3)$$

where the weights sum to 1. The exact weights should be tuned on validation data rather than assumed universally.

9.5 Recommendation strategy

For a new candidate, content-based and knowledge-based recommendation can operate immediately because they do not require a long interaction history. As interactions accumulate, collaborative or learning-to-rank features can be added. This hybrid strategy helps reduce cold-start limitations. Recommendations should also account for job freshness and expiry so that old vacancies do not dominate the ranking.

9.6 Chatbot method

The chatbot can use intent classification and retrieval from trusted portal data for common requests such as “show Java jobs,” “what is my application status,” or “how do I upload my resume?” For general career guidance, an LLM can be used with carefully bounded prompts. Actions affecting accounts or applications should be executed through authenticated backend APIs rather than directly by a language model. This separation reduces the risk of unauthorized actions and hallucinated transaction data.

10 Evaluation methodology

Evaluation should be divided into recommendation quality, NLP quality, system performance, usability, and responsible-AI evaluation. For classification tasks such as skill extraction or resume categorization, precision, recall, and F1-score are appropriate. For ranked job recommendations, Precision@K, Recall@K, Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG) can be used. Ranking metrics are especially relevant because users normally inspect only the first few recommended jobs.

A candidate-job matching experiment can compare at least three configurations: (A) keyword/TF-IDF baseline, (B) semantic embedding model, and (C) hybrid model combining skill overlap and semantic similarity. The comparison should use the same test set. Ablation experiments can remove individual features such as location, experience, or skills to determine their contribution.

System-level evaluation should measure API response time, AI inference latency, throughput, failed requests, and resource consumption. Usability can be evaluated with task completion rate, user satisfaction, and a questionnaire. Fairness evaluation should be performed only where suitable data and legal/ethical justification exist; possible measures include selection-rate differences or other group-aware metrics. The objective should be to identify harmful disparities rather than to assume that a single fairness metric is universally sufficient.

11 Security, privacy, and ethical considerations

Resumes contain personal and professional information. The portal should use encrypted transport, secure password hashing, authorization checks, input validation, access-controlled document storage, and audit logging. Resume files should not be publicly accessible through predictable URLs. AI services should receive only the information necessary for the requested task. Data retention and deletion policies should be defined before deployment.

AI recommendations should be treated as decision support rather than automatic hiring decisions. Employers should retain human oversight for consequential decisions. Recommendation explanations can improve transparency by identifying the skills and requirements contributing to a match. Sensitive attributes should not be used for ranking unless there is a specific, lawful, ethical reason, and fairness auditing should be considered during model development. The literature on algorithmic hiring emphasizes that removing obvious sensitive variables does not automatically eliminate discrimination [11].

12 Expected advantages

- Reduces manual effort in searching and preliminary matching.
- Improves relevance by considering skills and semantic meaning rather than exact keywords alone.
- Provides personalized recommendations to job seekers.
- Helps employers organize and prioritize large applicant pools.
- Supports both candidate-to-job and job-to-candidate perspectives.
- Combines conventional portal functions with AI services in one architecture.
- Provides explainable matching factors instead of only opaque scores.
- Can evolve from rule-based baselines toward machine-learning and deep-learning models.

13 Limitations

The proposed architecture has limitations. Recommendation quality depends on the quality and completeness of resumes and job descriptions. Poorly written or misleading information can reduce matching accuracy. Semantic models may also generate plausible but inappropriate matches. Cold-start behavior remains difficult without sufficient profile information or interaction history. A small academic dataset cannot represent the complexity of a national or global labor market. In addition, fairness cannot be guaranteed solely through an algorithm; it requires appropriate data, evaluation, governance, and human oversight. Finally, LLM-based chatbot functions may produce incorrect general advice and therefore should not be allowed to directly make hiring decisions or perform sensitive actions without backend validation.

14 Future scope

Future work can extend the system with multilingual resume processing, OCR for scanned resumes, skill ontologies, knowledge graphs, explainable ranking, learning-to-rank models, graph neural networks, transformer-based cross-encoders, and carefully governed LLMs. Career-path recommendation can use historical employment sequences to recommend possible next roles. Interview assistance can generate role-specific practice questions while keeping the candidate in control. Employer

analytics can identify skill shortages and recruitment bottlenecks. Federated or privacy-preserving learning may be investigated where multiple organizations need to collaborate without sharing raw candidate data. Future studies should also use larger and more representative datasets and conduct longitudinal evaluations rather than relying only on offline ranking metrics.

15 Discussion

The reviewed literature supports a hybrid design rather than a single universal algorithm. Content-based matching is useful when a new user has no interaction history, while collaborative methods become useful after sufficient interaction data is available. Semantic embeddings improve the ability to recognize related concepts, but structured skill constraints remain valuable for mandatory requirements. Deep learning can capture complex patterns, but it can be more difficult to explain and may require more data. Knowledge-based methods can encode domain rules, but building and maintaining ontologies is costly. Therefore, a practical academic system can combine deterministic business rules, semantic similarity, and optional machine-learning ranking.

The proposed Job Portal project contributes at the system-integration level. Instead of focusing solely on a recommendation benchmark, it connects AI capabilities to a complete recruitment workflow. The architecture allows the same candidate profile to support resume analysis, job recommendation, applications, employer notifications, and chatbot interactions. This integration is particularly important for practical deployment because a high-performing model is not useful if the surrounding platform cannot securely manage users, jobs, applications, and data.

The literature also indicates that evaluation must go beyond accuracy. A recommendation that is accurate in an offline dataset can still be unhelpful if it lacks diversity, freshness, explanation, or user trust. Recruitment decisions have high consequences, so fairness and human oversight are especially important. The proposed system therefore treats the AI output as a ranked recommendation with supporting factors, not as a final employment decision.

16 Conclusion

This review examined research in recommender systems, job recommendation, e-recruitment, semantic matching, resume screening, deep learning, and algorithmic fairness. The literature shows an evolution from keyword and information-retrieval techniques toward content-based, collaborative, hybrid, semantic, and deep-learning approaches. Systematic reviews identify hybrid recommendation and richer representations as important directions, while recent work emphasizes semantic embeddings, transformer models, explainability, and responsible AI. Despite these advances, important gaps remain in cold-start handling, data availability, explainability, fairness, dynamic preferences, and end-to-end integration.

Based on these findings, an AI-based Job Portal using React Native, Spring Boot, MySQL, and Python AI services is proposed. The system integrates job search, employer vacancy management, applications, notifications, resume processing, semantic matching, personalized recommendations, and chatbot assistance. A hybrid matching formula combining skill overlap, semantic similarity, experience, location, and user preferences provides an understandable baseline that can later be improved using machine learning. The proposed methodology also establishes measurable evaluation criteria for recommendation quality, NLP performance, system performance, usability, and

responsible AI. Overall, the project demonstrates how AI can enhance a conventional job portal while preserving human control over consequential recruitment decisions.

Appendix: Suggested experimental plan

Dataset split: 70% training, 15% validation, 15% testing for supervised components. Baselines: TF-IDF + cosine similarity; skill-overlap rules; semantic embedding similarity; and the proposed hybrid score. Report Precision@5, Recall@5, NDCG@5, F1 for extraction/classification tasks, average API latency, AI inference latency, and user task completion rate. Conduct ablation tests for skill, semantic, experience, location, and preference features. Maintain a separate test set to prevent information leakage from repeated resumes or duplicate job descriptions.

For a student project, the first implementation should use a transparent hybrid score and a small validated dataset. Once the baseline is stable, a transformer embedding model or supervised ranking model can be introduced and compared against the baseline. This staged methodology makes it easier to demonstrate measurable improvement rather than presenting AI as an unexplained component.

References

- [1] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* 2005;17(6):734–749. <https://doi.org/10.1109/TKDE.2005.99>
- [2] Bobadilla J, Ortega F, Hernando A, Gutiérrez A. Recommender systems survey. *Knowl.-Based Syst.* 2013;46:109–132. <https://doi.org/10.1016/j.knosys.2013.03.012>
- [3] Ricci F, Rokach L, Shapira B. Recommender systems: Introduction and challenges. In: *Recommender Systems Handbook*. Springer. 2015. p. 1–34. https://doi.org/10.1007/978-1-4899-7637-6_1
- [4] Freire MN, de Castro LN. e-Recruitment recommender systems: A systematic review. *Knowl. Inf. Syst.* 2021;63:1–20. <https://doi.org/10.1007/s10115-020-01522-8>
- [5] de Ruijt C, Bhulai S. Job recommender systems: A review. *arXiv:2111.13576* [Preprint]. 2021.
- [6] Abel F, Benczúr A, Kohlsdorf D, Larson M, Pálovics R. RecSys Challenge 2016: Job recommendations. In: *Proc. 10th ACM Conf. Recommender Systems*. 2016. p. 425–426. <https://doi.org/10.1145/2959100.2959207>
- [7] Xiao W, Xu X, Liang K, Mao J, Wang J. Job recommendation with Hawkes process: An effective solution for RecSys Challenge 2016. In: *Proc. Recommender Systems Challenge*. 2016. p. 1–4. <https://doi.org/10.1145/2987538.2987543>
- [8] Pacuk A, Sankowski P, Węgrzycki K, Witkowski A, Wygocki P. RecSys Challenge 2016: Job recommendations based on preselection of offers and gradient boosting. In: *Proc. Recommender Systems Challenge*. 2016. <https://doi.org/10.1145/2987538.2987544>
- [9] Mashayekhi Y, Li N, Kang B, Lijffijt J, De Bie T. A challenge-based survey of e-recruitment recommendation systems. *ACM Comput. Surv.* 2024;56(10). <https://doi.org/10.1145/3659942>
- [10] Huang Y, Liu DR, Lee SJ. Talent recommendation based on attentive deep neural network and implicit relationships of resumes. *Inf. Process. Manag.* 2023;60:103357. <https://doi.org/10.1016/j.ipm.2023.103357>
- [11] Fabris A, Baranowska N, Dennis MJ, Graus D, Hacker P, Saldivar J, et al. Fairness and bias in algorithmic hiring: A multidisciplinary survey. *ACM Trans. Intell. Syst. Technol.* 2025. <https://doi.org/10.1145/3696457>
- [12] Deshmukh A, Raut A. Applying BERT-based NLP for automated resume screening and candidate ranking. *Adv. Data Anal. Classification*. 2025;12(2). <https://doi.org/10.1007/s40745-024-00524-5>
- [13] Kumari Pandit S, Chakraborty K, Rajak S, Biswas S. An explainable deep learning framework for intelligent resume screening using semantic embeddings and attention-based candidate ranking. *Expert Syst Appl.* 2027;133669. <https://doi.org/10.1016/j.eswa.2026.133669>

-
- [14] Yi X, Allan J, Croft WB. Matching resumes and jobs based on relevance models. In: *Proc. 30th Annual Int. ACM SIGIR Conf. Research and Development in Information Retrieval*. 2007. p. 809–810. <https://doi.org/10.1145/127741.1277920>
- [15] Singh A, Catherine R, Visweswariah K. PROSPECT: A system for screening candidates for recruitment. In: *Proc. 19th ACM Int. Conf. Information and Knowledge Management*. 2010. p. 659–668. <https://doi.org/10.1145/1871437.1871523>
- [16] Al-Otaibi ST, Ykhlef M. Survey of job recommender systems. *Int. J. Phys. Sci.* 2012;7(29):5127–5142. <https://doi.org/10.5897/IJPS12.482>
- [17] Al-Otaibi S, Ykhlef M. Hybrid immunizing solution for job recommender system. *Front. Comput. Sci.* 2017;11(3):511–527. <https://doi.org/10.1007/s11704-016-5241-z>
- [18] Chala SA, Ansari F, Fathi M, Tijdens K. Semantic matching of job seeker to vacancy: A bidirectional approach. *Int. J. Manpower.* 2018;39(8):1047–1063. <https://doi.org/10.1108/IJM-10-2018-0331>
- [19] Çelik Ertuğrul D, Bitirim S. Job recommender systems: A systematic literature review, applications, open issues, and challenges. *J. Big Data.* 2025;12:140.
- [20] Next career recommendation in Mississippi with artificial intelligence. *J Comput Appl Math.* 2024;437:115458. <https://doi.org/10.1016/j.cam.2023.115458>