International Journal of Engineering Research Updates

Journal homepage: https://orionjournals.com/ijeru/

ISSN: 2783-0187 (Online)

ORION SCHOLAR JOURNALS

(RESEARCH ARTICLE)

Check for updates

Recruit AI: An AI-powered intelligent recruitment management system

S. S. Walunj, Bhakti Adhav *, Samruddhi Bhogawade, Sharvari Devikar and Tanuja Salke

Department of Information Technology, Sanjivani Pratisthan Institute of Technology (S.P.I.T.) Polytechnic, Kurund, Tal-Parner, Dist. – Ahilyanagar, India.

International Journal of Engineering Research Updates, 2026, 09(01), 009-020

Publication history: Received on 12 February 2026; revised on 18 March 2026; accepted on 20 March 2026

Article DOI: https://doi.org/10.53430/ijeru.2026.9.1.0015

Abstract

The recruitment process in modern organizations faces significant challenges including high volume of applications, time-consuming manual screening, and subjective candidate evaluation. This paper presents Recruit AI, an intelligent web-based recruitment management system that leverages artificial intelligence and machine learning to automate and optimize the hiring workflow. The system implements a hybrid resume matching algorithm combining TF-IDF vectorization and cosine similarity to achieve accurate candidate-job matching with scores ranging from 0–100%. Additionally, it integrates Samba Nova's Meta-Llama-3.1-8B-Instruct model for conversational AI assistance and employs automated email workflows using Django Signals. Experimental results demonstrate that the AI-powered matching algorithm achieves an average accuracy of 85% in identifying suitable candidates, reducing manual screening time by approximately 70%. The system serves dual user roles (candidates and recruiters) through a responsive web interface built with Django 5.x framework, featuring profile management, application tracking, and comprehensive analytics dashboards. This research contributes to the field of HR technology by presenting a scalable, cost-effective solution that enhances recruitment efficiency while maintaining transparency and fairness in candidate evaluation.

Keywords: Recruitment Management; Artificial Intelligence; Resume Matching; Tf-Idf; Cosine Similarity; Natural Language Processing; Django Framework; Machine Learning

1 Introduction

1.1 Background and Motivation

The recruitment industry has undergone a significant transformation in recent years, driven by technological advancements and changing workforce dynamics. Traditional recruitment methods, characterized by manual resume screening and subjective candidate evaluation, are increasingly inadequate for handling the volume and complexity of modern hiring processes. Organizations receive hundreds to thousands of applications for single job postings, making manual review time-consuming, expensive, and prone to human bias [1].

Recent studies indicate that recruiters spend an average of 6–7 seconds on initial resume screening [2], leading to potential oversight of qualified candidates. Furthermore, the COVID-19 pandemic accelerated the shift toward remote hiring, necessitating digital solutions that can efficiently manage virtual recruitment workflows [3].

1.2 Problem Statement

1.2.1. Current recruitment systems face several critical challenges

- Scalability Issues: Manual screening cannot efficiently process large application volumes

* Corresponding author: Bhakti Adhav

- Subjectivity and Bias: Human reviewers may introduce unconscious bias in candidate evaluation
- Time Inefficiency: Recruiters spend 60–70% of their time on administrative tasks rather than strategic activities [4]
- Poor Candidate Experience: Lack of transparency and delayed communication frustrate applicants
- Limited Matching Accuracy: Traditional keyword matching fails to capture semantic similarities between job requirements and candidate qualifications

1.2.2. Research Objectives

This research aims to address these challenges by developing an intelligent recruitment management system with the following objectives:

- Design and implement an AI-powered resume matching algorithm that accurately scores candidate-job fit
- Develop an automated workflow system that reduces manual intervention in routine recruitment tasks
- Create a user-friendly interface serving both candidates and recruiters with role-specific functionalities
- Integrate conversational AI to provide real-time assistance and improve user experience
- Evaluate system performance through quantitative metrics including matching accuracy, processing time, and user satisfaction

1.3 Contributions

1.3.1. The key contributions of this work include

- A hybrid matching algorithm combining keyword-based and semantic similarity techniques
- Integration of large language models (LLM) for intelligent chatbot assistance
- Automated email notification system using event-driven architecture
- Comprehensive web application with modern UI/UX design principles
- Empirical evaluation demonstrating significant improvements in recruitment efficiency

1.4 Paper Organization

The remainder of this paper is organized as follows: Section II reviews related work in automated recruitment and resume matching. Section III presents the system architecture and design. Section IV details the AI-powered matching methodology. Section V describes implementation specifics. Section VI presents experimental results and evaluation. Section VII discusses findings and limitations. Sections VIII and IX outline future work and conclusions respectively.

2 Literature Review

2.1 Automated Recruitment Systems

Automated Applicant Tracking Systems (ATS) have been studied extensively in academic literature. Keim et al. (2014) proposed a visual analytics approach for resume screening, while Faliagka et al. (2012) developed a ranking system using machine learning classifiers [5], [6]. However, these systems often rely on rigid keyword matching, limiting their effectiveness in capturing semantic relationships.

2.2 Resume Matching Techniques

Several approaches have been proposed for automated resume-job matching. Traditional systems use Boolean queries and exact keyword matching. While computationally efficient, this approach suffers from vocabulary mismatch problems [7]. Supervised learning methods including Support Vector Machines (SVM), Random Forests, and Neural Networks have been applied to resume classification tasks [8], [9]. These methods require labeled training data and may not generalize well to unseen job categories.

Recent work has explored NLP techniques including Named Entity Recognition (NER) for skill extraction and word embeddings for semantic matching [10], [11]. Our approach builds upon these foundations by combining TF-IDF vectorization with cosine similarity for robust matching.

2.3 Conversational AI in HR

The application of chatbots in human resources is an emerging research area. Studies have shown that AI-powered chatbots can improve candidate engagement, answer frequently asked questions, and provide 24/7 support [12]. Recent advances in large language models, particularly the Llama family of models, have enabled more natural and context-aware conversations [13].

2.4 Research Gap

While existing literature addresses individual components of recruitment automation, there is limited research on integrated systems that combine hybrid matching algorithms with real-time conversational AI assistance, automated workflow management, and comprehensive dual-role portals. RecruitAI addresses this gap by providing an end-to-end solution with these integrated capabilities.

3 System Architecture

3.1 Architectural Overview

Recruit AI follows a Model-View-Template (MVT) architecture pattern inherent to the Django framework. The system comprises four primary layers:

- Presentation Layer: Responsive web interface built with HTML5, CSS3, Bootstrap 5, and JavaScript
- Application Layer: Django views, forms, and business logic implementing core functionality
- Data Layer: SQLite (development) or PostgreSQL (production) database with Django ORM for data persistence

Integration Layer: External services including Cloudinary for cloud storage, SendGrid for email delivery, and Samba Nova AI for chatbot capabilities

3.2 Core System Components

3.2.1. User Management Module

The system implements a custom user model extending Django's Abstract User with role-based fields (is HR, is candidate). Authentication utilizes Django's built-in system with OTP-based password reset functionality. Authorization is enforced through custom decorators for role-based access control.

3.2.2. Profile Management Module:

Candidate Profile stores skills, education (JSON format), projects (JSON format), resume files via Cloudinary integration, and social media links. Profile maintains company information including industry, size, location, and professional links. The system includes dynamic profile completeness tracking to encourage comprehensive data entry.

3.2.3. Job Management Module

Job Post model supports rich text descriptions using Coeditors, comma-separated required skills, and eligibility criteria including academic percentages and age limits. The system provides automated eligibility validation and complete CRUD operations for HR users.

3.2.4. Application Management Module

Application model links candidates to jobs with comprehensive status tracking following the workflow: PENDING → SHORTLISTED → SECOND ROUND → SELECTED/REJECTED. The system supports bulk actions for simultaneous processing of multiple applications.

3.2.5. AI Matching Engine

This critical component performs resume parsing using PyPDF2 for PDF text extraction, implements the hybrid matching algorithm detailed in Section IV, and provides score normalization to a 0–100 range for intuitive interpretation.

3.2.6. Chatbot Module

Integration with SambaNova's Meta-Llama-3.1-8B-Instruct model provides conversational AI assistance. The system implements context management through domain-specific system prompts and delivers real-time responses via AJAX-based asynchronous communication.

3.2.7. Notification Module

Event-driven email automation utilizes Django Signals with post_save and post_delete triggers. Professional HTML email templates ensure consistent communication, while SendGrid integration provides reliable delivery with tracking capabilities.

3.2.8. Technology Stack

Table 1 presents the comprehensive technology stack employed in RecruitAI.

Table 1 Recruital Technology Stack

Layer	Technology	Purpose
Frontend	HTML5, CSS3, JS	User interface
Framework	Bootstrap 5	Responsive design
Backend	Django 5.x	Web framework
Database	PostgreSQL	Data persistence
ML Library	Scikit-learn	TF-IDF, similarity
AI Model	SambaNova Llama	Conversational AI
Storage	Cloudinary	Resume/media CDN
Email	SendGrid	Email delivery
Task Queue	Celery + Redis	Async processing
Server	Gunicorn	Production WSGI

3.3 Database Design

The system employs five core models with defined relationships: CustomUser maintains one-to-one relationships with both CandidateProfile and HRProfile. Each HR user can create multiple JobPosts (one-to-many). Applications create many-to-many relationships between candidates and jobs, storing AI matching scores and application status for each pairing.

4 AI-Powered Resume Matching Methodology

4.1 Algorithm Design Philosophy

The resume matching algorithm implements a hybrid approach combining explicit keyword matching with semantic similarity analysis. This design addresses the limitations of pure keyword matching (vocabulary mismatch) while maintaining computational efficiency superior to deep learning approaches.

4.2 Input Processing

4.2.1. Job Requirements Extraction

The system extracts two primary components from each job posting

- S_j: Required skills as comma-separated string
- D_j: Job description as rich text (HTML format)

4.2.2. Candidate Data Extraction

- For each candidate, the system processes
- Sc: Candidate skills as comma-separated string
- Pc: Project descriptions in JSON-encoded format
- Rs: Resume text extracted from PDF document

4.2.3. Text Preprocessing Pipeline

The preprocessing stage implements four critical transformations

- HTML tag stripping from job descriptions to obtain plain text
- Case normalization through lowercase conversion for case-insensitive matching
- Whitespace handling and normalization to ensure consistent token separation
- Text concatenation combining candidate skills, projects, and resume into unified representation: $C_{i\varphi\chi\chi}$ = Sc ⊕ Pc ⊕ RC

4.3 Dual-Component Matching Strategy

4.3.1. Component 1: Keyword Matching

For a given target text T (skills or description) and candidate text C, the keyword matching score is computed as

$$K = \text{split}\ (T, \text{delimiter} = \text{‘,’})\ (1)$$

$$M_{aeoxnt} = \Sigma^{L}\ (k \in C)\ (2)$$

$$\text{Score}_{kem}{}^{\circ}C_{lone} = (M_{aeoxnt} / |K|) \times 100 (3)$$

where ^{L} is the indicator function returning 1 if keyword k appears in candidate text C, and 0 otherwise.

4.3.2. Component 2: Semantic Similarity

The semantic component utilizes TF-IDF (Term Frequency-Inverse Document Frequency) vectorization. For a term t in document d within document collection D:

$$TF\ (t, d) = f,{}^{d} / \Sigma\ f_{t}',{}^{d}\ (4)$$

where $f,{}^{d}$ represents the frequency of term t in document d.

$$IDF\ (t, D) = \log\ (|D| / |\ \{d \in D: t \in d\}\ |\ (5)$$

The combined TF-IDF weight is:

TF-IDF (t, d, D) = TF (t, d) × IDF (t, D) (6)

Cosine similarity between job vector v_j and candidate vector v_e is computed as:

$\text{Score}_{semantie} = (v_j \cdot v_e) / (||v_j|| \times ||v_e||) \times 100$ (7)

4.4 E. Score Aggregation Strategy

4.4.1. Skills-Based Scoring:

For explicit skills matching, keyword precision is prioritized (60%) as exact skill names are critical indicators of qualification:

$$\text{Score}_{ski\chi\chi s} = 0.6 \times \text{Score}_{ski\chi\chi s}\text{\_kw} + 0.4 \times \text{Score}_{ski\chi\chi s}\text{\_sem} \quad (8)$$

4.4.2. Description-Based Scoring:

Job descriptions contain contextual information where semantic understanding provides greater value (70%):

$$\text{Score}^{d}_{ese} = 0.3 \times \text{Score}^{d}_{ese}\_kw + 0.7 \times \text{Score}^{d}_{ese}\_sem \quad (9)$$

4.4.3. Final Score Computation:

Skills are weighted as the primary matching criterion (80%), with job description providing supplementary context (20%):

$$\text{Score}^{f}_{ina\chi} = 0.8 \times \text{Score}_{ski\chi\chi s} + 0.2 \times \text{Score}^{d}_{ese} \quad (10)$$

The final score is bounded to the range [0, 100] and rounded to two decimal places for interpretability.

4.5 Algorithm Specification

Algorithm 1 presents the complete matching procedure in pseudocode format.

Algorithm 1: Hybrid Resume Matching Algorithm

procedure CALCULATESCORE (Sj, Dj, Sc, Pc, Rc)

Cfull ← Concatenate (Sc, Pc, Rc). lower ()

if Sj ≠ ∅ then

kwskills ← KeywordMatch(Sj, Cfull)

semskills ← TF-IDF-Cosine (Sj, Cfull)

Scoreskills ← 0.6 × kwskills + 0.4 × semskills

else

Scoreskills ← 0

end if

if Dj ≠ ∅ then

Dclean ← StripHTML(DJ)

kwdesc ← KeywordMatch(Declan, Cfull)

semdesc ← TF-IDF-Cosine (Dclean, Cfull)

Scoredesc ← 0.3 × kwdesc + 0.7 × semdesc

else

Scoredesc ← 0

end if

if Sj ≠ ∅ and Dj ≠ ∅ then

Scorefinal ← 0.8 × Scoreskills + 0.2 × Scoredesc

else if Sj ≠ ∅ then

Scorefinal ← Scoreskills

else

Scorefinal ← Scoredesc

end if

return min (100.0, round (Scorefinal, 2))

end procedure

4.6 Computational Complexity Analysis

The algorithm exhibits favourable computational characteristics

- Keyword Matching: O (n × m) where n represents the number of keywords and m denotes candidate text length
- TF-IDF Vectorization: O (V × D) where V represents vocabulary size and D indicates the number of documents
- Cosine Similarity: O(V) for sparse vector representations
- Overall Complexity: O (V × D + n × m), demonstrating linear scalability with input size

This complexity profile enables processing of approximately 37 applications per minute on standard hardware configurations (Intel i5 processor, 8GB RAM).

4.7 Advantages of Hybrid Approach

The proposed methodology offers several significant advantages

- Robustness: Combines exact string matching with semantic understanding to handle diverse resume formats
- Flexibility: Effectively handles vocabulary variations and synonyms (e.g., "ML" vs "Machine Learning")
- Interpretability: Separate keyword and semantic scores facilitate algorithm debugging and validation
- Tunability: Weighted combination parameters can be optimized for domain-specific requirements
- Efficiency: TF-IDF computation is significantly faster than deep learning alternatives while requiring no training data

5 Implementation details

5.1 Development Environment

The system was developed using Python 3.8+ as the primary programming language, Django 5.x as the web framework, Git for version control, Visual Studio Code as the integrated development environment, and pip with requirements.txt for package management.

5.2 Security Implementations

Security is enforced through multiple mechanisms: CSRF protection via Django's built-in middleware, password hashing using PBKDF2 with SHA256, SQL injection prevention through Django ORM parameterized queries, XSS protection via automatic template escaping, file upload validation with type and size restrictions, and sensitive data management using environment variables stored in .env files.

5.3 User Interface Design Principles

The interface follows modern UX principles including responsive mobile-first design using Bootstrap 5 grid system, clear visual hierarchy distinguishing primary and secondary actions, consistent color scheme (gradient: #667eea to #764ba2), WCAG 2.1 AA accessibility compliance with semantic HTML, and performance optimization through lazy loading, minified assets, and CDN integration.

5.4 Deployment Architecture

Development deployment utilizes Django Development Server (manage.py runserver) with SQLite database and local file storage. Production deployment employs Gunicorn WSGI Server, Nginx reverse proxy, PostgreSQL database, Cloudinary CDN for media files, and Redis cache for performance optimization.

6 Experimental Results and Evaluation

6.1 Experimental Setup

The evaluation utilized a comprehensive dataset comprising 150 job postings across 5 categories (Software Engineering, Data Science, Marketing, Sales, HR), 500 candidate profiles with varying skill sets, and 1,200 applications submitted over a 3-month evaluation period.

Evaluation metrics included matching accuracy (percentage of correctly identified suitable candidates), processing time (average computation time per resume), user satisfaction (measured via 1–5 Likert scale survey responses), and system efficiency (measured as reduction in manual screening time).

6.2 Matching Algorithm Performance

6.2.1. Accuracy Evaluation

Ground truth was established through manual expert evaluation of 200 candidate-job pairs. The algorithm achieved 156 true positives, 14 true negatives, 18 false positives, and 12 false negatives. Table 2 presents the comprehensive performance metrics.

Table 2 Matching algorithm performance metrics

Metric	Value
True Positives	156
True Negatives	14
False Positives	18
False Negatives	12
Accuracy	85.0%
Precision	89.7%
Recall	92.9%
F1-Score	91.2%

Score Distribution Analysis:

Analysis of 1,200 applications revealed a mean matching score of 62.4% with standard deviation of 18.7%. Scores ranged from 12% to 98% with a median of 64.0%. The distribution showed 18% of applications as excellent matches (80–100%), 42% as good matches (60–79%), 28% as moderate matches (40–59%), and 12% as poor matches (0–39%).

6.3 Processing Time Analysis

Table 3 presents detailed processing time breakdown.

Table 3 Resume processing time breakdown

Operation	Avg Time	Max Time
PDF Extraction	1.2s	3.5s
TF-IDF Vectorization	0.3s	0.8s
Cosine Similarity	0.1s	0.2s
Total Matching	1.6s	4.5s

6.4 Chatbot Performance

Evaluation of 100 user interactions with the AI chatbot revealed 87% relevant responses, 10% partially relevant responses, and only 3% irrelevant responses. Average response time was 2.3 seconds with user satisfaction rating of 4.2/5.0. Common query categories included application status (32%), profile completion (24%), job search tips (18%), technical issues (15%), and general information (11%).

6.4.1. System Efficiency Gains

Table IV presents comprehensive comparison with traditional manual recruitment processes.

Table 4 Efficiency comparison (per 100 applications)

Task	Manual	Recruit AI	Reduction
Resume Screening	8.0 hrs	2.5 hrs	68.8%
Shortlisting	4.0 hrs	0.5 hrs	87.5%
Email Communications	3.0 hrs	0.1 hrs	96.7%
Tracking	2.0 hrs	0.0 hrs	100.0%
Total	17.0 hrs	3.1 hrs	81.8%

6.4.2. User Satisfaction Survey

Survey of 50 recruiters yielded average ratings of 4.3/5.0 for ease of use, 4.6/5.0 for time savings, 4.1/5.0 for matching accuracy, and 4.4/5.0 for overall satisfaction. Survey of 200 candidates produced ratings of 4.2/5.0 for application process, 4.4/5.0 for profile management, 4.5/5.0 for status transparency, 4.0/5.0 for chatbot helpfulness, and 4.3/5.0 for overall satisfaction.

Qualitative feedback highlighted positive themes including “intuitive interface,” “fast application process,” and “transparent status updates.” Areas for improvement included requests for more detailed rejection feedback, advanced search filters, and native mobile applications.

6.4.3. Comparative Analysis

Table 5 compares RecruitAI with existing commercial ATS solutions.

Table 5 comparison with existing ats solutions

Feature	RecruitAI	Comp A	Comp B
AI Matching	Hybrid	Keyword	ML
Chatbot	LLM-based	None	Rule-based
Auto Emails	Yes	Yes	Yes
Accuracy	85%	72%	81%
Processing Speed	1.6s	2.8s	1.9s
Cost	Free	$99/mo	$149/mo

7 Discussion

7.1 Key Findings

7.1.1. Hybrid Matching Superiority

The combination of keyword and semantic matching demonstrates superior performance, outperforming pure keyword systems by 13 percentage points in accuracy. This validates the hypothesis that combining explicit and implicit matching strategies addresses vocabulary mismatch problems while maintaining computational efficiency.

7.2 Significant Efficiency Gains

The system reduces manual screening time by 81.8%, enabling recruiters to focus on strategic activities such as candidate interviews, employer branding, and talent pipeline development rather than administrative tasks.

7.3 Strong User Acceptance

High satisfaction scores (4.3–4.4/5.0) across both candidate and recruiter user groups indicate strong acceptance and usability of the system. This validates the dual-role interface design and user experience optimization efforts.

7.4 Demonstrated Scalability

Linear computational complexity enables processing of large application volumes without performance degradation. The system maintains consistent processing times even with increasing dataset sizes, demonstrating production-ready scalability.

7.5 Limitations and Challenges

7.5.1. Training Data Independence

The TF-IDF approach operates without learning from historical hiring decisions, unlike supervised machine learning methods. This independence provides advantages in interpretability and zero-shot capability but sacrifices potential accuracy improvements from domain-specific training.

7.5.2. Domain Specificity Constraints

The system is optimized for technical roles with clearly defined skill requirements. Performance may vary for creative positions, executive roles, or positions where qualifications are more subjective and context-dependent.

7.5.3. Resume Quality Variance

Poorly formatted resumes, non-standard terminology, or creative resume designs can reduce matching accuracy. The system performs optimally with standard resume formats containing clearly articulated skills and experiences.

7.5.4. Chatbot Integration Limitations

While the LLM-based chatbot effectively handles general queries, it cannot access real-time application data without additional database integration. This limitation restricts its ability to provide specific, personalized status updates.

7.5.5. Bias Propagation Considerations

Although the matching algorithm is mathematically objective, biases present in job descriptions or resume language may propagate through the matching process. Careful monitoring and mitigation strategies are necessary to ensure fair evaluation.

7.6 Comparison with Related Work

Compared to traditional ATS systems, RecruitAI's hybrid approach directly addresses the vocabulary mismatch problem inherent in keyword-only systems while reducing administrative burden more effectively through comprehensive automation.

Compared to deep learning approaches, while neural networks may achieve marginally higher accuracy given sufficient labeled training data, RecruitAI's TF-IDF approach offers critical advantages: zero training data requirement, superior interpretability for debugging and validation, and significantly faster processing (1.6s versus 5–10s for BERT-based models).

Compared to commercial solutions, RecruitAI delivers comparable or superior matching accuracy at zero cost through its open-source model. Integration of modern LLM-based chatbot capabilities provides competitive advantages over rule-based conversational systems.

7.7 Practical Implications

For organizations, particularly small to medium enterprises, RecruitAI enables deployment of cost-effective recruitment automation without substantial licensing costs. Reduced time-to-hire improves competitive advantage in talent acquisition markets. Data-driven candidate evaluation enhances overall hiring quality and reduces subjective decision-making.

For candidates, transparent matching scores provide actionable feedback on application strength and profile completeness. Automated status updates significantly improve candidate experience by reducing uncertainty and communication delays. Profile completeness guidance helps candidates present themselves more effectively to potential employers.

For HR professionals, the system facilitates a strategic shift from time-consuming administrative tasks to high-value activities including strategic talent management, employer branding, and candidate relationship building. Bulk actions and analytics dashboards enable data-driven decision making. Automated communications ensure consistent and professional candidate engagement throughout the recruitment lifecycle.

8 Future Work

8.1 Short-term Enhancements

Planned short-term improvements include development of advanced analytics dashboard featuring time-to-hire metrics, source-of-hire tracking, diversity and inclusion analytics, and predictive hiring trend analysis. Native mobile applications for iOS and Android platforms with push notification support and mobile-optimized application processes are prioritized. Enhanced chatbot capabilities including integration with application database for real-time status queries, multi-language support for international recruitment, and voice interaction capabilities will improve user experience. Integration of video interview scheduling, AI-powered interview analysis for sentiment and keyword detection, and recording functionality will provide comprehensive recruitment workflow support.

8.2 Long-term Research Directions

Long-term research will explore deep learning integration utilizing BERT or GPT-based models for semantic matching, named entity recognition for automated skill extraction, and automated skill taxonomy generation. Bias detection and mitigation through fairness-aware matching algorithms, bias auditing tools for job descriptions, and anonymized screening options will address ethical considerations in AI-powered recruitment. Predictive analytics including candidate success prediction based on historical hiring outcomes, attrition risk modeling, and skill gap analysis with training recommendations will enhance strategic workforce planning. Blockchain integration for verifiable credential storage, immutable application history, and decentralized candidate profiles represents an innovative direction. Federated learning approaches enabling privacy-preserving collaborative model training across organizations, shared skill taxonomies without data sharing, and industry-wide matching benchmarks will advance the field.

8.3 Scalability Improvements

Future architectural improvements include migration to microservices architecture with separate services for matching, notifications, and chatbot functionality, enabling independent scaling based on component-specific load and improved fault tolerance. Advanced caching strategies utilizing Redis for frequently accessed data, TF-IDF vector caching for job descriptions, and optimized session management will enhance performance. Database optimization through comprehensive indexing strategies for large-scale queries, database sharding for multi-tenant deployments, and read replicas for analytics workloads will support enterprise-scale deployments.

9 Conclusion

This paper presented Recruit AI, an intelligent recruitment management system leveraging artificial intelligence to automate and optimize hiring processes. The system's hybrid resumes matching algorithm, combining TF-IDF vectorization with keyword matching, achieves 85% accuracy in candidate-job matching while maintaining computational efficiency with 1.6s average processing time.

The research contributes a novel hybrid matching algorithm with weighted combination of keyword and semantic similarity tailored to the recruitment domain, integrated AI chatbot utilizing SambaNova's Llama 3.1 model for real-time user assistance, automated workflow system with event-driven email notifications reducing manual intervention by 82%, and comprehensive dual-role web platform serving both candidates and recruiters with modern user experience design.

Experimental evaluation demonstrates significant efficiency gains, with recruiters saving an average of 13.9 hours per 100 applications processed. User satisfaction scores of 4.3–4.4/5.0 across both user groups indicate strong system acceptance and usability.

The system addresses critical challenges in modern recruitment including scalability limitations of manual screening, subjectivity and bias reduction through algorithmic evaluation, time efficiency improvements enabling strategic focus, and enhanced candidate experience through transparency and automation. By providing an open-source, cost-effective solution, RecruitAI democratizes access to advanced recruitment technology for organizations of all sizes.

Future research directions include deep learning integration for enhanced semantic understanding, comprehensive bias mitigation frameworks, predictive analytics for workforce planning, and scalability improvements through microservices architecture. The modular system design and well-documented codebase facilitate community contributions and continued academic research.

RecruitAI represents a significant advancement in HR technology, demonstrating that intelligent automation enhances rather than replaces human decision-making in recruitment processes. As organizations continue navigating increasingly competitive talent markets, systems like RecruitAI will play vital roles in efficiently and fairly identifying, engaging, and hiring top talent while enabling HR professionals to focus on strategic talent management activities.

Compliance with ethical standards

Acknowledgements

The authors would like to thank the Department of Information Technology, Sanjivani Pratisthan Institute of Technology (S.P.I.T.) Polytechnic for providing the necessary resources and support for this research.

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] P. Cappelli, “Your Approach to Hiring Is All Wrong,” Harvard Business Review, vol. 97, no. 3, pp. 48–58, 2019.

[2] The Ladders, “Eye-Tracking Study: Recruiters Spend an Average of 7.4 Seconds Reviewing a Resume,” The Ladders Research Report, 2018.

[3] A. P. Carnevale and I. Hatak, “The Pandemic and the Future of Work,” Georgetown University Center on Education and the Workforce, 2020.

[4] LinkedIn Talent Solutions, "Global Recruiting Trends 2020," LinkedIn Corporation, 2020. Available: https://englishbulletin.adapt.it/wp-content/uploads/2020/01/linkedin-2020-global-talent-trends-report.pdf

[5] D. A. Keim et al., “Visual Analytics for Resume Screening and Candidate Ranking,” IEEE Trans. Vis. Comput. Graphics, vol. 20, no. 12, pp. 1726–1735, 2014.

[6] E. Faliagka et al., “Application of Machine Learning Algorithms to an Online Recruitment System,” in Proc. Int. Conf. Internet Web Appl. Services, 2012, pp. 215–220.

[7] N. Schmitt and D. Chan, Personnel Selection: A Theoretical Approach. Thousand Oaks, CA: Sage Publications, 1998. ISBN: 0-7619-0813-4.

[8] K. Yu et al., “Resume Information Extraction with Cascaded Hybrid Model,” in Proc. ACL, 2005, pp. 499–506.

[9] R. Naik and S. Ragothaman, “Automated Resume Screening System Using Machine Learning,” Int. J. Comput. Appl., vol. 164, no. 7, pp. 17–21, 2017.

[10] J. Devlin et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in Proc. NAACL-HLT, 2019, pp. 4171–4186.

[11] C. Qin et al., “A Survey on Automatic Resume Analysis,” ACM Comput. Surv., vol. 53, no. 6, pp. 1–35, 2020.

[12] A. K. Upadhyay and K. Khandelwal, “Applying Artificial Intelligence: Implications for Recruitment,” Strategic HR Rev., vol. 17, no. 5, pp. 255–258, 2018.

[13] H. Touvron et al., “Llama 2: Open Foundation and Fine-Tuned Chat Models,” arXiv preprint arXiv:2307.09288, 2023.