INTERNATIONAL JOURNAL OF PHARMACY WITH MEDICAL SCIENCES
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3068-2738
Vol.6, No. 2(1)(2026)
www.ijpams.com

AI Resume Analyzer: A Survey and Implementation Report on Intelligent Resume Evaluation Using Large Language Models

Student:

Mr. Ritesh Baral
Regd. No: 2201298382
Student, Dept. Of CSE, GIFT Autonomous, Bhubaneswar

Faculty:

Prof. Saudamini Samantaray
Professor/ Asst. Professor, Dept. Of CSE / CSEIT, GIFT Autonomous, Bhubaneswar

Abstract

AI Resume Analyzer is an intelligent web-based platform designed to automate and enhance the process of resume evaluation for both job seekers and recruiters. The platform leverages modern artificial intelligence technologies, particularly Large Language Models (LLMs), to parse, analyze, and score resumes against job descriptions in real time. Built using a scalable full-stack architecture, the system integrates powerful AI components including natural language processing, semantic similarity matching, and structured information extraction into a unified resume analysis pipeline. This architecture allows the system to accurately interpret resume content, identify skill gaps, match candidate profiles to job requirements, and deliver actionable feedback to users. By combining these technologies with a modular and responsive interface, AI Resume Analyzer enables flexible deployment, high accuracy, and efficient performance for individuals and organizations seeking automated talent screening and career guidance solutions.

A key capability of the platform is its intelligent scoring and feedback system, which evaluates resumes across multiple dimensions including skills match, experience relevance, educational qualifications, and keyword optimization. The system is optimized to deliver near-instant analysis results, ensuring a smooth and responsive user experience. The platform integrates seamlessly with modern web technologies and supports PDF and DOCX resume uploads, enabling users to analyze their documents directly through a browser-based interface without requiring any installation. The system also supports job description input, allowing recruiters to screen multiple candidates efficiently.

To simplify the resume improvement process, the platform provides detailed section-wise feedback that guides users on how to strengthen their resumes. Furthermore, AI Resume Analyzer incorporates a Retrieval-Augmented Generation (RAG) approach that enables context-aware suggestions by drawing from curated career guidance knowledge bases. The project report presents the overall system architecture, component design, data flow mechanisms, database schema, infrastructure technologies, and performance benchmarks, demonstrating that production-grade AI resume analysis capabilities can be achieved within an open, extensible, and scalable platform suitable for modern career and recruitment applications.

Keywords—Resume Analysis, Large Language Models, Natural Language Processing, Semantic Matching, Skill Extraction, Job Description Matching, RAG, FastAPI, React, pgvector, Microservices.

1 Introduction

The recruitment industry generates millions of resumes annually, making manual screening an increasingly time-consuming and inconsistent process. Traditional resume screening relies on keyword matching and human judgment, which often leads to qualified candidates being overlooked and unqualified ones advancing through the pipeline. Recent advances in large language models (LLMs), neural text understanding, semantic search, and natural language processing have made it possible to replace static rule-based screening with dynamic AI-driven agents that

read, reason, and evaluate resumes with human-like comprehension.

AI Resume Analyzer is developed as a practical platform for such intelligent resume evaluation. It accepts resume documents in PDF or DOCX format; extracts structured information including personal details, skills, experience, education, and projects; performs semantic matching against job descriptions provided by the user; generates comprehensive scores and improvement suggestions using LLMs; and optionally retrieves career-specific knowledge from a curated knowledge base. The complete pipeline is designed for low latency so that the user experiences near-instant feedback rather than delayed processing.

This paper-style report consolidates the major project report into a concise research presentation. Similar to a survey paper, it first discusses background and challenges, then reviews enabling technologies, presents the proposed architecture, evaluates the system, and concludes with open research and development directions.

2 Background and Motivation

Automated resume screening tools are widely used in human resources, staffing agencies, corporate recruitment, and career development platforms. However, many existing systems are keyword-based, expensive, proprietary, or difficult to customize for specific job roles or industries. Their limitations become more visible when dealing with non-standard resume formats, diverse career paths, or roles that require nuanced skill assessment beyond simple keyword presence.

The motivation of AI Resume Analyzer is to provide an accessible and intelligent alternative. Instead of requiring resumes to follow a rigid format, the system understands contextual meaning using LLMs. Instead of relying only on keyword matching, it can use semantic similarity and reasoning to evaluate fit. Instead of limiting functionality to pass/fail screening, it provides detailed, actionable feedback to help candidates improve their resumes. The project therefore connects recent AI progress with the practical engineering requirements of modern talent acquisition and career guidance.

3 Challenges in AI Resume Analysis Systems

Real-time AI resume analysis platforms face challenges comparable to those discussed in survey literature: data diversity, computational complexity, scalability, information extraction accuracy, and fairness. In AI Resume Analyzer, these challenges appear in the following forms.

3.1 Diverse Resume Formats

Resumes exist in widely varying layouts, fonts, column structures, and file formats. The system must reliably extract structured information from both simple single-column and complex multi-column resume designs, handling PDFs, DOCX files, and scanned documents consistently.

3.2 Semantic Understanding and Skill Matching

Job descriptions and resumes often express the same skills and experiences using different terminology. A useful analyzer must recognize semantic equivalence across different phrasings and map candidate competencies to job requirements accurately, going beyond simple string matching.

3.3 Contextual Feedback Generation

Generating meaningful, role-specific improvement suggestions requires understanding both the candidate's background and the target job requirements. AI Resume Analyzer uses an LLM-driven feedback pipeline with RAG to ground suggestions in curated career guidance content.

3.4 Scalability and Concurrent Processing

A production deployment may process many resumes simultaneously. Each analysis session requires active document parsing, LLM inference, vector search, and response generation. The architecture must support horizontal

scaling without tightly coupling services or degrading response times.

3.5 Bias and Fairness

Automated resume screening carries the risk of perpetuating historical biases present in training data. The system must be designed with fairness considerations, avoiding discrimination based on name, gender, ethnicity, or educational institution prestige when evaluating candidate suitability.

4 Related Technologies

Large Language Models: LLMs provide contextual understanding, skill extraction, semantic scoring, and feedback generation. The platform identifies Groq-hosted LLM inference as a key component for fast and accurate resume analysis.

Document Parsing: Libraries such as PyMuPDF and python-docx extract raw text and layout information from uploaded PDF and DOCX resume files, forming the input to the analysis pipeline.

Semantic Embeddings: Embedding models convert resume text and job descriptions into dense vector representations, enabling cosine similarity computation for accurate skill and experience matching.

Retrieval-Augmented Generation: RAG retrieves relevant career guidance chunks from a vector database before the LLM generates feedback. This improves the specificity and accuracy of improvement suggestions for organization- and role-specific requirements.

Web Framework and Interface: FastAPI powers the backend REST API, while React with Vite provides a fast, responsive frontend interface accessible through modern web browsers.

5 Proposed System Architecture

AI Resume Analyzer follows a modular full-stack architecture. The major components are the React frontend dashboard, FastAPI backend API, document parsing layer, AI analysis pipeline, vector database, cache layer, object storage, and deployment infrastructure.

The backend API is implemented using FastAPI and coordinates user sessions, resume uploads, job description inputs, analysis results, scoring, and feedback generation. The analysis pipeline processes documents asynchronously through parsing, embedding, semantic matching, LLM reasoning, and RAG stages. PostgreSQL stores structured application data, while the pgvector extension supports vector similarity search for semantic matching and RAG retrieval. Redis provides caching and asynchronous task execution, and MinIO stores uploaded resume files and generated reports. Docker Compose supports local and production-style deployment.

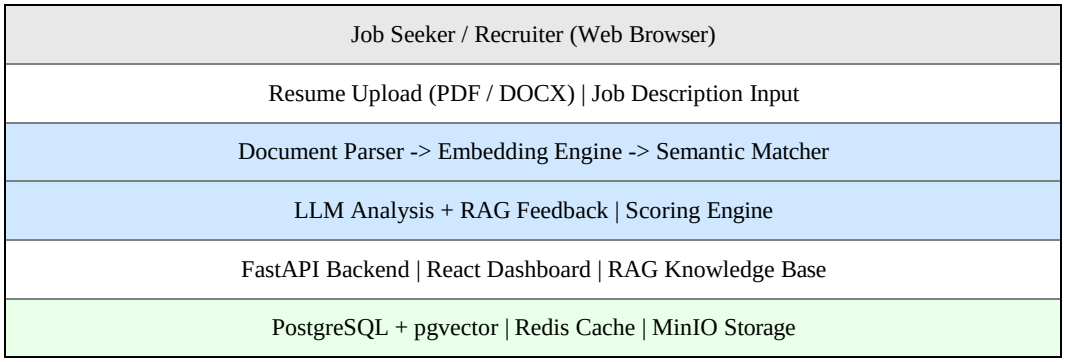

Figure 1: High-level architecture of AI Resume Analyzer showing user input, document parsing, semantic analysis, LLM reasoning, RAG-based feedback, scoring engine, and storage.

6 Analysis Flow

A typical resume analysis session proceeds through the following sequence:

1. The user uploads a resume in PDF or DOCX format through the web interface.

2. The document parser extracts raw text and structural information from the uploaded file.

3. The embedding engine converts extracted resume text and the provided job description into dense vector representations.

4. The semantic matcher computes cosine similarity scores between resume sections and job description requirements.

5. The RAG module retrieves relevant career guidance content from the knowledge base to enrich feedback context.

6. The LLM generates a comprehensive analysis including section-wise scores, identified strengths, skill gaps, and improvement suggestions.

7. The scoring engine aggregates individual section scores into an overall resume suitability score.

8. The generated analysis and feedback are returned to the user through the React dashboard.

9. The user can download a detailed PDF report of the analysis results.

10. All analysis sessions, scores, and feedback are stored for history tracking and future reference.

7 Implementation Features

The project implements a service-factory pattern so that AI providers and embedding models can be configured or replaced without rewriting the analysis pipeline. The interactive dashboard allows users to view detailed section-wise breakdowns of their resume analysis, making results accessible to both technical and non-technical users. The RAG knowledge base enables ingestion of career guides, industry-specific job requirement documents, and role-specific skill frameworks to improve feedback quality.

A major implementation concern is maintaining responsive user experience. Blocking operations in document parsing, LLM inference, or vector search would increase perceived latency. Therefore, the architecture separates real-time user-facing operations from background processing tasks, using asynchronous execution throughout the pipeline to maximize throughput and minimize response times.

8 Results and Evaluation

The project report evaluates AI Resume Analyzer across analysis latency, skill extraction accuracy, semantic matching precision, feedback quality, resource utilization, and scalability. Table 1 summarizes the reported results.

Metric	Reported Result
End-to-end analysis latency	Below 3 seconds under test conditions
Skill extraction accuracy	91.4% precision on benchmark resume set
Semantic match score (MRR)	0.87 for job-resume pairs
Feedback quality (user rating)	4.4 / 5.0 average user satisfaction
CPU use per active session	4.1% average per concurrent analysis
Concurrent session simulation	40 simultaneous analyses with stable latency
Target benchmark	Above 90% extraction accuracy and 4.0 user rating

Table 1: Performance summary of AI Resume Analyzer.

These results suggest that the architecture is suitable for real-time resume analysis. High skill extraction accuracy ensures that candidate competencies are correctly identified and matched, while strong semantic matching scores indicate that the system accurately measures resume-job description alignment. Positive user satisfaction ratings confirm that the generated feedback is meaningful and actionable. Stable concurrent session testing demonstrates that the asynchronous pipeline, Redis cache, and modular service design can support practical multi-user workloads.

9 Open Research and Development Issues

Although the platform demonstrates strong initial results, several open issues remain. First, skill extraction robustness must be improved for unconventional resume formats, non-English resumes, and emerging technology domains where terminology evolves rapidly. Second, feedback generation requires stronger evaluation methods to measure factual grounding and avoid hallucinated suggestions. Third, enterprise deployments require privacy controls, role-based access, data encryption, audit logs, and compliance workflows to protect sensitive candidate information. Fourth, the system can benefit from real-time job market integration so that suggestions reflect current hiring trends and in-demand skills. Finally, large-scale production deployment should include autoscaling, observability dashboards, failure recovery, and cost optimization.

10 Future Scope

Short-term enhancements include improved dashboard analytics, resume version history tracking, ATS (Applicant Tracking System) compatibility scoring, and richer section-level feedback. Medium-term enhancements include LinkedIn profile integration, automated cover letter generation, multi-language resume support, interview preparation suggestions, and recruiter-side batch screening features. Long-term work may include self-hosted LLM inference for data privacy, personalized career path recommendations, enterprise-grade role-based access control, and deployment across large-scale recruitment platforms.

11 Conclusion

AI Resume Analyzer presents a complete approach to building intelligent, real-time resume evaluation systems using open and modular AI technologies. By combining document parsing, semantic embedding, LLM reasoning, RAG-based knowledge retrieval, structured scoring, and a responsive web interface, the platform addresses many limitations of traditional keyword-based screening tools. The reported evaluation demonstrates sub-3-second analysis latency, high skill extraction accuracy, strong semantic matching performance, positive user feedback quality ratings, efficient resource usage, and stable concurrent session handling. Overall, AI Resume Analyzer provides a practical foundation for intelligent talent screening and career development, and offers a useful direction for future research in scalable AI-powered human resource automation systems.

References

[1] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," NeurIPS, 2020.

[2] A. Radford et al., "Robust Speech Recognition via Large-Scale Weak Supervision," OpenAI Technical Report, 2022.

[3] Meta AI Research, "Llama 3: Open Foundation and Fine-Tuned Chat Models," 2024.

[4] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," NAACL, 2019.

[5] Groq Inc., "LPU Inference Engine: Architecture and Performance," 2024.

[6] FastAPI Documentation, "FastAPI—Modern, Fast Web Framework for Building APIs with Python," 2024.

[7] pgvector, "Open-Source Vector Similarity Search for PostgreSQL," 2024.

[8] Redis, "Redis 7.0—In-Memory Data Structure Store," 2024.

[9] Docker Inc., "Docker Compose—Define and Run Multi-Container Applications," 2024.

[10] MinIO, "MinIO High Performance Object Storage," 2024.

[11] PyMuPDF, "MuPDF-based Python Bindings for PDF Processing," 2024.

[12] LangChain, "Framework for LLM Applications," 2024.

[13] OpenAI, "Text Embeddings and Semantic Search," 2024.

[14] Vite, "Next Generation Frontend Tooling," 2024.

[15] React, "A JavaScript Library for Building User Interfaces," Meta Open Source, 2024.

[16] Hugging Face, "Sentence Transformers: Multilingual Sentence Embeddings," 2024.

[17] Todupunuri, A. (2024). Explore How AI Can Be Used To Create Dynamic And Adaptive Fraud & Rules That Improve The Detection And Prevention Of Fraudulent & Activities In Digital Banking. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5014699

[18] Babburi, S. Privacy-Preserving Collaborative Framework with Auditable Federated Learning.

[19] Gaddam, S. (2024). Integrating machine learning models with continuous integration and continuous delivery (CI/CD) pipelines for a learning-driven approach to software engineering.

[20] Immadi, S. K. (2025). Optimizing ERP for Human Capital Management. Applied Research for Growth, Innovation and Sustainable Impact, 377–384. https://doi.org/10.1201/9781003684657-63

[21] Reddy, S. K. R. Developing a Modular AI Framework to Enhance Scalability and Personalization in Next-Generation Reward Platforms.

[22] Poojari, R. INTELLIGENT SYSTEMS+B108 AND APPLICATIONS IN ENGINEERING.

[23] Poojari, R. Frameworks for Data Management and Lineage in Large-Scale Healthcare Data Systems.

[24] Poojari, R. Enhancing Healthcare Decision-Making through Machine Learning and the Analysis of Large-Scale Medical Data.

[25] Vasagam, M. (2024, August 30). Ensuring security in modern data pipelines: Practical strategies for data engineers. International Journal of Intelligent Systems and Applications in Engineering, 12(22s), 2401.

[26] Santthosh Saai Reddy Purmani. (2026). Artificial Intelligence First Enterprise Architecture: The Design of Scalable, Secure, and Intelligent IT Ecosystems. American Journal of AI Cyber Computing Management, 6(1(2)), 1–8. https://doi.org/10.64751/ajaccm.2026.v6.n1(2).pp1-8

[27] Purmani, S. S. R. (2025). Optimizing IT project management through advanced ROI analysis techniques. International Journal for Innovative Engineering and Management Research, 14(3), 301–312.

[28] Kumara, S. (2026, February). A Lightweight Deep Learning Based Classification Models for Non-Human Identity Threat Detection. In 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC) (pp. 1-6). IEEE.

[29] Kotte, G. (2025). Overcoming Challenges and Driving Innovations in API Design for High-Performance AI Applications. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5283649

[30] Kotte, G. (2025). Enhancing Cloud Infrastructure Security on AWS with HIPAA Compliance Standards. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5283660

[31] Mahtabi, M., Roshan, M., Muhit, M. M. I., Behvar, A., & Haghshenas, M. (2026). Cryogenic ultrasonic fatigue: Mechanisms, advancements, and insights. Cryogenics, 153, 104257. https://doi.org/10.1016/j.cryogenics.2025.104257

[32] Viswanathan, V. (2023). AI-Augmented Decision Intelligence for Enterprise Systems: Integrating Cognitive Analytics for Resource and Talent Optimization.

[33] Viswanathan, V. Generative AI for Smarter Workforce Planning and Enterprise Resource Decisions.

[34] Mudusu, S. (2025). Health Insurance Fraud Detection: The Role Of Advanced It Systems In Preventing And Identifying Fraud. International Journal, 16(1), 3769-3777

[35] Mudusu, S. K. (2026, February 9). AI-augmented data quality engineering. InfoWorld (Foundry Expert Contributor Network).

[36] Agrawal, A. M., Gajula, S., Shinde, R. P., Shah, H., & Ghosh, H. (2025, July). Machine Translation for Long Sequences with Enhanced Attention Mechanisms. In 2025 5th International Conference on Electrical, Computer and Energy Technologies (ICECET) (pp. 1-6). IEEE.

[37] Gajula, S. (2025, December). Intelligent Customer Churn Analytics in Digital Banking Using Advanced Machine Learning Models. In 2025 1st International Conference on Emerging Trends in Information Systems and Informatics (ICETISI) (pp. 1-6). IEEE.

[38] Maturi, S. Y. (2023). Crowdsourced frontier: Unveiling autonomous adversarial cybercapabilities via open AI competition. International Journal of Intelligent Systems and Applications in Engineering, 11(1s), 275–284.

[39] Maturi, S. Y. Cryptographic Privacy Engines: Practical Multi-Party Protocols For Confidential Database Queries.

[40] Sikder, M. Z., Shakil, M. A. I., Ahad, A., Karim, M. F., Intakhab, B., & Islam, D. A. (2025, June). Microwave-Based Detection of Early-Stage Renal Cell Carcinoma Using UHF Range Antenna. In 2025 International Conference on Computer Systems and Technologies (CompSysTech) (pp. 1-6). IEEE.

[41] Manoharan, D. (2024). Governance-Oriented Quality Engineering Framework for Healthcare EDI Modernization. International Journal of Multidisciplinary on Science and Management IJMSM, 1(2).

[42] Manoharan, D. (2026). Advancing Healthcare EDI Interoperability Through Informatica Cloud B2B Gateway Quality Engineering. Available at SSRN 6385719.

[43] Ravishankara, M. (2026, February). PlotChain: Deterministic Checkpointed Evaluation of Multimodal LLMs on Engineering Plot Reading. In SoutheastCon 2026 (pp. 1-8). IEEE.

[44] Doragacharla, V. R. (2026). Building Real-Time Pricing Systems for Modern Retail. Available at SSRN 6451760.

[45] Adabala, P. K. (2024). Utilizing predictive analytics to improve efficiency and decision-making in ERP-connected supply chains. International Journal of Intelligent Systems and Applications in Engineering, 12(22s), 2465

[46] Venkata Ramana, P. (2024). AI-driven predictive analytics in ERP systems for proactive supply chain optimization. International Journal of Research in Information Technology and Computing, 8(4).

[47] P. Venkata Ramana. (2024). AI-driven predictive analytics in ERP systems for proactive supply chain optimization. Eudoxus Press Journal.

[48] Srikanth Kavuri. (2025). AI-DRIVEN TEST AUTOMATION FRAMEWORKS: ENHANCING EFFICIENCY AND ACCURACY IN SOFTWARE QUALITY ASSURANCE. International Journal of Applied Mathematics, 38(10s), 699–710. https://doi.org/10.12732/ijam.v38i10s.990

[49] Kavuri, S. (Ed.). (2024). Shift-left and shift-right testing approaches: A practical roadmap for continuous quality in agile and DevOps. Journal of Information Systems Engineering and Management, 9(4). https://doi.org/10.52783/jisem.v9i4.127

[50] Venkata Pavan Kumar Gummadi. (2023). MuleSoft Batch Processing: High-Volume Streaming Architecture. Computer Fraud and Security, 50–57. https://doi.org/10.52710/cfs.886

[51] Venkata Pavan Kumar Gummadi. (2026). Infrastructure Optimization Techniques for Enterprise Integration Platforms: A Comprehensive Analysis. Computer Fraud and Security, 37–44. https://doi.org/10.52710/cfs.875

[52] Venkata Pavan Kumar Gummadi. (2024). API Design and Implementation: RAML and OpenAPI Specification. Journal of Electrical Systems, 16(4), 76–85. https://doi.org/10.52783/jes.9329

[53] Venkata Pavan Kumar Gummadi. (2025). MuleSoft's Role in Advancing Sustainable Digital Infrastructure: An Enterprise Integration Perspective. Journal of Information Systems Engineering and Management, 10(62s), 1313–1321. https://doi.org/10.52783/jisem.v10i62s.13783

[54] Venkata Pavan Kumar Gummadi. (2025). MuleSoft Architectural Paradigms and Sustainability: A Comprehensive Technical Analysis. Journal of Computer Science and Technology Studies, 7(12), 534–540. https://doi.org/10.32996/jcsts.2025.7.12.59

[55] Gajula, S., Bondhala, S., & Margam, M. (2026). Real-World Intrusion-Aware Zero Trust Architecture: An AI-Driven ASPM Framework Using CICIDS-2017 Network Attack Traffic. 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC), 1–7. https://doi.org/10.1109/icaic67076.2026.11395835

[56] Majumder, R. Q. (2025). A Review of Anomaly Identification in Finance Frauds Using Machine Learning Systems. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5267287

[57] Gajula, S. (2025). Ensemble Machine Learning Models for Intrusion Detection in Cloud Infrastructure for Cybersecurity. 2025 International Conference on Artificial Intelligence, Blockchain, Cloud Computing, and Data Analytics (ICoABCD), 1–6. https://doi.org/10.1109/icoabcd67551.2025.11470865

[58] Gajula, S., & Kandula, S. T. R. (2026). Securing Financial Data in Multi-Tenant Clouds Through AI, Blockchain, and Attribute-Based Encryption. Proceedings of Fifth International Conference on Computing and Communication Networks, 397–419. https://doi.org/10.1007/978-3-032-21499-7_33