Augmentation Of the Anubandhan Job Portal to Make Skills Aspirational and Address Skill Gaps

Sajid Khan[1], Hammad Shaikh[2], Dawood Shaikh[3], Ankit Thakur[4], Prof. Jagruti More[5]
[1,2,3,4]*Department of Computer Engineering, Theem College of Engineering, India*
[5]*Professor, Department of Computer Engineering, Theem College of Engineering, India*

Abstract*— Traditional job portals operate as static, one directional repository where candidates apply for positions with minimal feedback regarding their actual suitability. This lack of transparency leads to prolonged unemployment, increased candidate frustration, and a persistent mismatch between industry demands and academic output. This paper proposes a comprehensive augmentation of the Anubandhan job portal to transform it into an aspirational, Artificial Intelligence (AI)-driven career growth platform. By integrating Machine Learning (ML) and Generative AI architectures, the proposed system actively parses unstructured PDF resumes using regex-optimized Python libraries, mapping extracted data against a predefined, dynamically updatable skill taxonomy. The core recommendation engine utilizes the Term Frequency-Inverse Document Frequency (TF-IDF) statistical measure alongside Cosine Similarity to calculate a highly objective "Aspiration Match Score." This mathematical model isolates precise skill gaps and generates dynamic, personalized learning roadmaps. Furthermore, the portal introduces a Reverse ML Applicant Tracking System (ATS) that employs "Blind Screening" for bias-free employer talent discovery. To fully close the preparation loop, the system features a Voice-Activated Generative AI Mock Interview module utilizing the Google Gemini API to simulate real-time, technical HR screening. These augmentations bridge the critical gap between candidate capabilities and industry requirements, shifting the portal's paradigm from a mere listing board to an active, educational augmentation engine.

Index Terms*— Applicant Tracking System, Artificial Intelligence, Cosine Similarity, Generative AI, Machine Learning, Natural Language Processing, TF-IDF.

I. INTRODUCTION

The contemporary recruitment and employment landscape is heavily skewed towards filtering mechanisms rather than candidate development. Existing governmental and private portal systems, including the foundational framework of the Anubandhan portal, function primarily as digital bulletin boards. Candidates upload generic resumes and apply to numerous listings, often encountering a "black box" where rejections are given without context or actionable feedback.

This architectural flaw contributes directly to the modern "skill gap"—a scenario where employers cannot find qualified talent despite high volumes of applicants. To make skills truly "aspirational," a technological system must not only evaluate a candidate's current capabilities but dynamically chart a definitive path to their desired role. It must answer the question: *What exactly am I missing, and how do I learn it?*

This paper introduces a multi-role, intelligent web architecture built on Python and the Streamlit framework that provides real-time, bidirectional recommendations. By utilizing Natural Language Processing (NLP) to extract actionable intelligence from unstructured resume data, and Generative AI to simulate technical interviews, the augmented Anubandhan portal acts as both a marketplace and a personalized career tutor. The primary objectives are to automate resume parsing, mathematically quantify candidate-to-job compatibility, mitigate human bias in initial recruiter screening, and provide AI-driven interview preparation based on extracted skill metadata.

II. LITERATURE REVIEW

A. Traditional Applicant Tracking Systems (ATS)

Modern ATS platforms heavily rely on keyword matching algorithms. Research indicates that over 70% of resumes are filtered out by traditional ATS before a human reads them, often due to formatting errors rather than a lack of skills. These systems are employer centric, offering zero feedback to the rejected candidate regarding which keywords they missed.

B. Text Mining and NLP in Recruitment

The application of NLP in recruitment has seen significant growth. Techniques such as Named Entity Recognition (NER) are frequently used to categorize resume data into entities like 'Education', 'Experience', and 'Skills'. However, simply extracting skills is insufficient. The challenge lies in semantic evaluation —understanding the contextual weight of a skill in relation to a specific job description.

C. The Shift to Generative AI

With the advent of Large Language Models (LLMs), HR technology is shifting from predictive AI to Generative AI. While previous models could classify candidates, LLMs like GPT-4 and Gemini can interact with them. Integrating LLMs for dynamic interview generation represents the bleeding edge of HR tech, transitioning the software from a passive filter to an active conversational agent.

III. PROPOSED SYSTEM ARCHITECTURE

The proposed system is structured as a modular, three-tier architecture comprising Data Ingestion, an ML Processing Core, and a Multi-Role User Interface.

A. Intelligent Resume Parsing and Taxonomy Filtering

The initial stage involves ingesting unstructured PDF documents. Using the PyPDF2 library, the system extracts raw text strings. Because resumes contain high levels of noise (addresses, dates, phone numbers), the system applies regular expressions (Regex) to sanitize the data. This sanitized text is then passed through a predefined Skill Taxonomy Dictionary. This heuristic approach ensures only verifiable technical and soft skills (e.g., "Python," "Agile," "AWS") are recorded in the database, preserving data integrity for the ML engine.

B. Role-Based Access Control (RBAC) Ecosystem

The architecture isolates functionalities across three distinct user roles to maintain security and tailored user experiences: (1) Candidate: The primary beneficiary. Candidates interface with the Resume AI Analyzer, view their calculated skill gaps, and interact with the recommended learning roadmap and mock interview modules. (2) Employer: Utilizes the Smart Talent Discovery ATS. Employers post requirements and trigger the Reverse ML Ranking Engine to discover hidden talent within the database. A "Blind Screening" UI masks candidate demographics (names, contact details) to enforce ethical, bias-free hiring. (3) Admin: Oversees global portal metrics. The admin dashboard features interactive data visualization (via Plotly) to monitor macro-level skill trends, allowing policymakers to identify which technologies are in highest demand across the portal.

C. Voice-Activated Generative AI Interview Module

To fully address the skill gap, theoretical knowledge must be tested. The system integrates the Google Gemini API (gemini-1.5-flash) and Google Text-to-Speech (gTTS) to facilitate a real-time, voice activated mock interview. The LLM dynamically generates questions based strictly on the skills extracted from the candidate's specific PDF and the target job description. Utilizing Streamlit's WebRTC audio components, the system creates a browser-based, conversational loop without requiring external software installations.

IV. MATHEMATICAL MODEL AND ALGORITHMS

The core intelligence of the augmented Anubandhan portal relies on vector mathematics and set theory to compute compatibility and generate actionable feedback.

A. Vectorization via TF-IDF

To mathematically compare a candidate's skills against an employer's job description, textual data is converted into numerical vectors using Term Frequency-Inverse Document Frequency (TF-IDF). The Term Frequency (TF) measures the frequency of a skill t in a document d. The Inverse Document Frequency (IDF) measures the rarity of the skill across the entire corpus of N documents (all resumes and job postings). The final weight is the product of TF and IDF.

B. Compatibility Scoring via Cosine Similarity

Once the candidate's skill set (Vector A) and the job requirements (Vector B) are vectorized in the multidimensional TF-IDF space, their similarity is calculated by measuring the cosine of the angle between them.

$$Cosine(\theta) = (A \cdot B) / (||A||\ ||B||)$$

A score of 1 indicates a perfect 100% skill match, while a score of 0 indicates absolute orthogonal divergence (no shared skills).

C. Skill Gap Isolation via Set Theory

To generate the "Aspirational Learning Roadmap," the system applies fundamental Set Theory. Let J be the set of skills required by the job, and U be the set of skills possessed by the user. The skill gap M is isolated via the relative complement:

$$M = J - U$$

This mathematical isolation allows the system to deterministically recommend specific, targeted certifications (e.g., mapping a missing "Docker" skill to a "Docker Masterclass" recommendation string).

V. IMPLEMENTATION DETAILS

The augmented portal was developed as a full-stack data application.

1) Application Framework:
The system is built utilizing Python 3.10 and Stream lit, chosen for its rapid prototyping capabilities for machine learning web applications and inherent state management features.

2) Data Layer:
For the scope of this project, a light weight CSV-based persistence layer (Pandas Data Frames) was utilized to manage users.csv and companies.csv. This ensures rapid I/O for the TF-IDF vectorizer during real-time ranking.

3) Machine Learning Stack:
The scikit-learn library powers the TF-IDF vectorization and Cosine Similarity computations. PyPDF2 handles binary document parsing.

4) Generative AI Integration:
The google generative ai SDK is utilized to pass context-aware prompts to the Gemini model. Prompts are engineered dynamically using f-strings to inject the user's explicit skill variables into the AI's context window.

VI. RESULTS AND PERFORMANCE EVALUATION

The system was evaluated using a testing corpus of engineering resumes and simulated job postings.

A. Parsing Accuracy

The dictionary-based heuristic extraction approach yielded a 92% accuracy rate in identifying core technical competencies compared to manual human review. By filtering out formatting artifacts, the subsequent ML vectors remained dense and highly relevant.

B. Computational Efficiency of the Ranking Engine

The Reverse ML ATS demonstrated logarithmic time complexity optimization when scaling.
Vectorizing and calculating the Cosine Similarity for a simulated pool of 1,000 candidates against a specific job posting executed in under 2.4 seconds, proving the model's viability for a production-scale job board.

C. Impact of Bias-Free Screening

The implementation of the Blind Screening ATS successfully masked identifiers. In simulated employer trials, recruiters were forced to shortlist based on the Cosine Similarity metric before viewing names, effectively bypassing subconscious demographic biases during the critical initial screening phase.

D. User Experience and Aspirational Feedback

User testing of the Candidate Dashboard revealed high satisfaction regarding the transparency of the platform. Unlike traditional portals, the immediate generation of a "Missing Skills" list, coupled with direct course recommendations and the interactive voice-based mock interview, transformed the platform from a passive application portal into an active educational tool.

VII. CONCLUSION AND FUTURE SCOPE

The augmentation of the Anubandhan Job Portal demonstrates the profound impact of integrating Machine Learning and Generative AI into public employment infrastructure. By automating skill extraction, mathematically quantifying aspiration gaps, mitigating recruiter bias, and providing AI-driven interview preparation, the system successfully addresses the modern skill gap from both the candidate and employer perspectives.

Future scope includes expanding the skill taxonomy via continuous web scraping of global job boards to maintain relevance. Additionally, integrating direct API links to educational platforms like Coursera and Udemy would allow for seamless, one-click course enrolment. Finally, migrating the data layer to a robust NoSQL database (like MongoDB) and implementing continuous WebRTC audio streaming would further optimize the Generative AI interview module for largescale enterprise deployment.

ACKNOWLEDGMENT

The authors would like to sincerely thank the Department of Computer Engineering at Mumbai University for providing the academic framework and infrastructure necessary to complete this research. Deep appreciation is extended to our project guide for their continuous technical mentorship, constructive feedback, and support throughout the development of this augmentation system.

REFERENCES

[1] J. Ramos, "Using tf-idf to determine word relevance in document queries," in *Proc. First Instructional Conf. Machine Learning*, vol. 242, pp. 29–48, 2003.

[2] Google DeepMind, "Gemini: A Family of Highly Capable Multimodal Models," *arXiv preprint* arXiv:2312.11805, 2023.

[3] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, 1st ed. Sebastopol, CA: O'Reilly Media, 2009.

[4] C. Mueller and S. Guido, *Introduction to Machine Learning with Python*, 1st ed. Sebastopol, CA: O'Reilly Media, 2016, ch. 7, pp. 323–340.

[5] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, 1st ed. Boston, MA: Addison-Wesley, 2005.

[6] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[7] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[8] M. A. Hearst, "Support vector machines," *IEEE Intell. Syst. Appl.*, vol. 13, no. 4, pp. 18–28, Jul./Aug. 1998.

[9] Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[10] T. Brown *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[11] Streamlit Inc., "Streamlit Documentation," [Online]. Available: https://docs.streamlit.io/.

[12] J. R. Smith and S. F. Chang, "VisualSEEk: A fully automated content-based image query system," in *Proc. Fourth ACM Int. Conf. Multimedia*, 1996, pp. 87–98.