IJETIR

ISSN (Online) 2583-0554

Impact Factor: 5.731

International Journal of Emerging Technologies and Innovative Research (IJETIR)

Volume 5, Issue 12, December 2025

Pathlight AI: An Integrated LLM-Powered Platform for Personalized Career Advancement and Job Market Analysis

Gagan N, Sai Kumar S, Keerthana M, Samarth S Vaidya, Dr. Anand R, Dr. Vanshika Rastogi

Dept. of Artificial Intelligence & Data Science

East Point College of Engineering & Technology, Bangalore

Abstract: *The current job market presents an enormous challenge for both job seekers and those who support them. Job seekers are faced with the need to personalize each application extensively, decipher often obfuscated Applicant Tracking Sys- tems (ATS), develop rigorously prepared interview preparation methods, and plan everything strategically for their career development. Current print and online tools are politically fragmented, generic, and offer little or no integration with real-time labor market changes or trends. To address these challenges, we created Pathlight AI, an opensource interactive web application developed using Streamlit and powered by Google's Gemini Large Language Models (LLMs). Pathlight AI serves as an integrated career development package that provides a hybrid resume and job description analysis tool using keyword matching (spaCy) and semantic similarity (Sentence Transformers and FAISS), comprehensive ATS checks for resume and job description screening, personalized cover letter generation, LinkedIn profile export, and resume-based interview tips tailored to an individual's unique resume and target job description. Additionally, it includes an AI-driven career roadmap featuring attainable goals and suggested resources, as well as an interactive dashboard that visualizes real-time job market trends sourced from a Supabase database. Our approach to developing this application relies on advanced prompt engineering to generate structured output as JSON from the LLM, enabling structured responses for personalized actions while integrating applied natural language processing and deep learning for specialized analysis. Pathlight AI consolidates multiple career support capabilities into a single platform, providing highly personalized, AI-driven career guidance infused with realworld job data. The goal is to significantly enhance job seeker effectiveness and strategic career planning..*

Keywords: Large Language Models (LLMs), Career Guid- ance, Resume Analysis, ATS Optimization, Job Market Analysis, Prompt Engineering, Natural Language Processing, Semantic Similarity, Hybrid Matching, Google Gemini, Streamlit, FAISS

I. INTRODUCTION

Getting hired in the current labor market is an increasing challenge, and in many cases an exercise in futility for applicants. The number of applicants for coveted job openings routinely requires the use of mechanisms and methods to help applicants differentiate themselves from the rest of the applicant pool. A primary challenge: employers tend to use Applicant Tracking Systems (ATS), computer applications which categorize applicants based mainly on the summary content of their resumes not just keyword matching. By relying on keyword matching, qualified applicants could easily be eliminated from the interview pool based on explained credentials whose content was categorized away instead of considered [17].

Another industry standard is that employers expect very precise application materials, such as cover letters that are targeted to the job and organization advertised, which adds to the difficulty for applicants. Further complicating the process is the interview, which often includes being prepared for behavioral and technical questions for the targeted job role; an applicant might also need to be strategic in furthering their professional aspirations such as skill building, and remaining market aware or thinking ahead for the viability of sustainable advancement.

Modern digital applications will likely be able to meet these needs, but in silos. For example, while resume builders are good for formatting, job boards primarily serve as a place to post openings, ATS analyzers provide limited feedback on keywords or format and general purpose AI writing software lacks rich contextual knowledge that comes from discussing an specific resume in relation to an specific job description. This leads candidates to engage with multiple isolated tools instead of having a single intelligent system to guide their decision-making, from start to finish.

The pace of advancement of the Large Language Model (LLM) with respect to the Gemini series of models produced by Google [1], [3] is an enormous source of transformative power. LLMs are unique in their ability to interpret nu- anced language, generate coherent and contextually perfect text, synthesize information, and act on complex instructions. This allows for the creation of tools that can provide useful advanced feedback while producing high-quality customized and relevant content in bulk [6].

However, letting LLMs reflect on the complicated task of career development will require more than some simple prompt. The system needs to offer a user the opportunity to imbue into the system their contextual information (resume), job specification information (job description), and informa- tion from the greater market, then apply well-formed prompt to generate outputs that are structured and actionable.

To address this challenge, we invented the Pathlight AI, an interactive web app running on Streamlit [2]. Pathlight AI is a complete AI-based career platform that enhances and aids job searching and career planning. With the Google Gemini models as its intellectual underpinning, it offers some unique modalities:

1) Resume Deep Analysis: Comparison of pasted job de- scriptions against resumes submitted by users through a hybrid keyword-semantic matching algorithm model.

2) ATS Optimization Suite: Buys individualized feedback around resume content (grammar, impact, voice), form (length, sections) and issues of formatting that could be problematic for ATS parsing.

3) Personalized Content Creation: Generates individualized cover letters, LinkedIn profile optimization suggestions, and interview preparation guides (tips, likely interview questions) based on a hybrid of resumes and job descrip- tions.

4) Strategic Career Blueprinting: Offers personalized career blueprints around short, mid, and long range goals, rec- ommendations for skills and resources, as well as AI gen- erated job role recommendations based on individualized user profile.

5) Job Market Insights: An interactive dashboard to visual- ize job postings over time analysis and posting trends (geospatial analysis (locations)), salary data (consoli- dated), and company rankings/reputation from an external database (Supabase [10]).

Pathlight AI's main value proposition is its personalized and integrated nature. It provides robust NLP methodologies (keyword analysis, semantic search) with leading LLM prompt engineering in an easy-to-use format for full cycle applica- tion assistance. Pathlight AI delivers uniquely contextualized advice and artifacts with a deep integration of user-specific inputs while offering market context itself via its dashboard.

II. RELATED WORK

The domain of digital tools for job seekers is varied, but often disconnected.We would classify related work as follows:

1) Resume Builders and ATS Checkers: Platform such as Zety, Resume.io, or Jobscan that provide templates or ba- sic ATS checks- focusing often on just assessing keyword density and simple formatting rules. They are beneficial for presentation, but lack any sort of sophisticated se- mantic understanding of the desired job requirements, or offering recommendations for improving content that is nuanced, a direct comparison versus keyword matching. Pathlight AI leverages even more features by incorpo- rating some level of semantic assessments, provide the benefits of AI based critiques on content (quantification, voice) and tips about layout analysis.

2) General AI Writing Assistants: Tools like ChatGPT, Google Gemini [3], or Grammarly can help get through the writing process and refine the resume text or writing the cover letter. However, as general use AI models, they require significant prompting, guidance from the user to appropriately adjust the context of the job description and resume in their content. General assistants do not embed any resume analysis, ATS feedback, or market data. Pathlight AI leverages specific, structured prompting to ensure the outputs are highly personalized based solely on the resume and JD provided for the analysis.

3) Job Boards and Aggregators: Job boards or job aggre- gators such as LinkedIn, Indeed, Naukri, etc. provide excellent services for job listing and an easy application process. Some even provide a basic resume analysis or skill recommendations, however, these analyses are often generic and lack comprehensive integration with person- alized application material development or comprehen- sive ATS feedback. Pathlight AI's dashboard can be used as an additional resource alongside job boards, giving the user an aggregate trend analysis of their respective career market and job opportunities.

4) Career Planning and Skill Platforms: Course sites like Coursera, Udemy, and LinkedIn Learning provide courses to develop skills. Career counseling services, in con- trast, develop career paths and personalized assistance. Pathlight AI aids by using AI to develop personalized learning and development roadmaps for their respective career, pulling information directly from the user's re- sume and suggesting appropriate skills and resources, such as courses or certifications, from a presumed career trajectory.

5) Research in Automated Resume Analysis: Academic research has examined the use of various techniques, including resume parsing, skill extraction, and resume- JD matching for resumes, typically either based upon traditional NLP techniques or machine-learning models. Recent work on automated resume analysis also examines using LLMs for resumes and job descriptions [6], [15]. Pathlight AI builds on all these concepts by developing practical hybrid matching models, but also examines the use of LLMs not only for ATS scoring levels of analysis but also for further personalized outputs which include a range of options within one practical system.

Pathlight AI distinguishes itself through its integration of analysis resumeJD matching ATS checks personalized gen- eration cover letter tips LinkedIn strategic planning roadmap and market context dashboard within a single platform driven by deep personalization using sophisticated LLM prompting based directly on userprovided documents

III. SYSTEM ARCHITECTURE AND METHODOLOGY

Pathlight AI is a web application developed in Python, utilizing Streamlit for the front-end interface and supports core AI functionality using Google Gemini models (stage 2).

A. Overall Architecture

The architecture is intentionally modular as shown in Fig. 1. Each module and its various primary functions are described as follows:

1) Frontend (Streamlit): Provides the user interface for file upload (resumes), for textual input (job description, API key, search terms, filters), tool selection, model configu- ration, and displaying final report content (analysis report dynamic text or dashboards).

2) Input Handling & Preprocessing (utils.py): The input han- dling and preprocessing module makes use of PyPDF2 [18] and python-docx [19] to extract text from PDF and DOCX files respectively. It performs simple text cleaning, and spaCy [4] is used to perform NLP preprocessing (i.e, lemmatization or stop word removal) as needed for keyword analysis.

3) Core Analysis Engine (analysis.py): The Core Analysis Engine manages and executes the resume vs job de- scription analysis. More specifically, the engine calculates keyword match scores, creates text embeddings using Sentence Transformers [5], performs semantic similarity searches using FAISS [7], calculates the hybrid score, and prepares the data as population for the human readable pin prompts.

4) Core Prompt Engineering: Found throughout each tool- specific module, this involves writing detailed prompts that instruct the LLM to carry out a specific function (analysis, generation) as a response to contextual informa- tion and an intended schema for output (usually JSON).

5) Tool-Specific Modules (ats_optimization.py, cover_letter.py, etc.): Functions related to the specific functionality (ATS checks, cover letter generation, roadmap planning, etc.) happen in these modules, where they interact with the Core Analysis Engine and the LLM Interface.

6) Output Processing (utils.py, display.py): In- cludes clean_and_parse_json function that ro- bustly extracts JSON-structured responses from LLM outputs that may be textually noisy. display.py also includes functions to render the

resultant analysis and generative content outputs in a user-friendly presentation within Streamlit, using custom CSS-styling.

7) Dashboard Module (dashboard.py): Fetches (data) from an external PostgreSQL Supabase [10] database using supabase-py. Data is manipulated and cleaned using Pandas [8] (date, city extractions, salary parsing, etc.). Interactive visualizations are generated using Plotly [9], Matplotlib, and WordCloud.

B. Hybrid Resume and Job Description Matching (analysis.py)

A core element is calculating a hybrid score to determine relationship between a resume and job description by combining lexical and semantic methods:

1) Preprocessing: The resume and JD texts are preprocessed via spaCy (en_core_web_sm) to extract lemmatized, lowercase tokens. Stopwords and punctuation are re- moved.

2) Keyword Matching: Words and phrases that represent keywords are identified from the preprocessed texts..

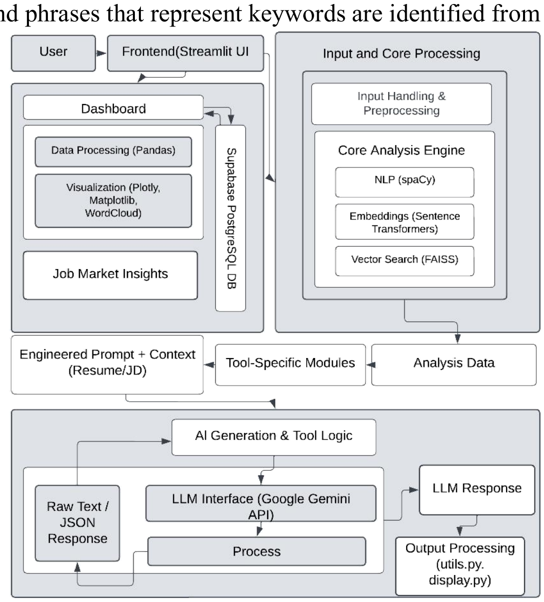

Fig. 1. High-level architecture of the Pathlight AI platform.

A score is then computed to measure overlap between resume keywords and JD keywords. Note: the implemen- tation may also include some synonym matching by using spaCy's word vectors to further define matched criteria even though basic vector similarity may vary even for synonyms.

3) Semantic Similarity: Both the resume (or representative segments) and job description are encoded as dense semantic vectors using the sentence-transformers/all-MiniLM-L6-v2 model [5]. This architecture was selected for a good balance of performance and efficiency. A FAISS IndexFlatL2 index will be constructed on the resume embedding(s). The L2 distance from the job description embedding to the closest resume embedding within the index is calculated. L2 distance is not normalized and converted to a similarity score (0-100) indicating how similar the job description is to the resume, with 0 distance equating 100 similarity.

4) Hybrid Score Calculation: The final score is a weighted average:

$$\text{Score} = \alpha \times \text{KeywordScore} + (1 - \alpha) \times \text{SemanticScore}$$

The parameter α (defaulting to 0.6 in the conceptual de- sign) determines the relative importance of exact keyword matches versus overall contextual similarity. This hybrid approach aims to capture both specific requirements and conceptual alignment.

C. Prompt Engineering Strategy

1) Role Assignment: Prompts often start by assigning the LLM an expert role (e.g., "Act as an expert interview coach," "Act as an expert ATS scanner").

2) Explicit Instructions: In most cases, prompts include clear and explicit directions for the task, specifying the required analysis, level of detail, tone, and constraints (e.g., "Generate highly specific actionable advice," "DO NOT provide generic advice AT ALL COSTS").
3) Context Grounded: The prompt directly connects the user's resume text and job description text, instructing the LLM to base its response solely on the provided information.
4) Structured Output (JSON): For analytical tasks (ATS checks, interview tips, roadmap, LinkedIn suggestions), prompts explicitly demand output in a specific JSON for- mat. The schema, including keys, data types, and nesting, is defined directly within the prompt. This is crucial for reliable parsing and rendering in the UI (display.py). The utils.clean_and_parse_json function is designed to handle minor deviations or extraneous text around the JSON block.
5) Iterative Refinement: Prompts were likely developed through trial and error, refining instructions and JSON schemas based on observed LLM outputs to improve quality and reliability.

D. Tool-Specific Modalities

1) ATS Optimization (ats_optimization.py): Incor- porates multiple layers of analysis to enhance resume compatibility with applicant tracking systems (ATS).
• AI Content Checks: Targeted prompts instruct the Gemini model to analyze and suggest improvements in spelling/grammar, word repetition (with synonym recommendations), passive voice (with active alter- natives), lack of quantification in bullet points (with impact-enhancing suggestions), and excessive bullet length.
• Standard Rule-Based Checks: Python functions lever- age Regex and string analysis to estimate ATS parse rates, check word counts, detect predefined buzzwords, validate contact information (using refined patterns in check_contact_information), and ensure the presence of essential sections such as Skills, Experi- ence, Education, and Summary.
• Layout Analysis: PyPDF2 is utilized to examine the PDF structure, including page count analysis. If exces- sive length is detected, a warning is issued. Heuristics identify potential ATS parsing issues, such as large whitespace gaps or inconsistent spacing.
• Holistic Enhancement: A separate LLM call (get_resume_enhancement_suggestions) provides high-level recommendations, including summary effectiveness, bullet point impact (following the STAR method), action verb usage, keyword integration, and skills presentation. These insights may be adjusted based on the hybrid match score.
2) Tailored Generation (cover_letter.py, linkedin_optimization.py, interview_tips.py): The LLM is prompted to generate personalized outputs by extracting and customizing information, accomplishments, skills, and keywords exclusively from the provided resume and job description. The goal is hyper-personalization. Structured outputs, such as interview advice and LinkedIn profile recommendations, are formatted in JSON for consistency and ease of integration.
3) Distilled Strategy (career_roadmap.py, job_recommendation.py): The LLM is directed to analyze the input profile (resume) to derive a viable career path or job title, justify recommendations, and generate a structured JSON roadmap. This roadmap includes phased goals (short-, mid-, and long-term), key skills for development, project ideas, and curated recommendations for certifications and resources, along with specific URLs to prioritize actionable next steps.
4) Job Market Dashboard (dashboard.py): This mod- ule implements a structured data visualization workflow to provide real-time job market insights.
• Data Acquisition: Collects job posting data from Su- pabase.
• Data Processing (Pandas): Cleans the dataset by han- dling missing values, converting data types, and con- structing relevant features. These include:
– Extracting city names using sophisticated logic and filtering based on state.
– Deriving standardized job titles by stripping pre- fixes and suffixes.
– Calculating job posting freshness as (scraped_at - posting_date).
– Parsing salary strings into numeric ranges using Regex.
• Visualization (Plotly, Streamlit): Generates interactive visualizations, including:
– Geographical distribution: A scatter_geo map utilizing an expanded coordinate dictionary for Indian cities.
– Top trends: Insights on the most frequent compa- nies, roles, and locations.

– Company ratings vs. reviews: Comparative scatter plots.
– Salary distribution: Box plots and histograms.
– Posting frequency heatmaps: Time-based job post- ing patterns.
Streamlit filters dynamically modify the underlying dataset and update visualizations in real time.

E. Technology Stack

1) Primary Language: Python 3.x [10]
2) Web Framework: Streamlit [1]
3) LLM API: google-generativeai (for Google Gemini models [?], [2])
4) Core NLP: spaCy (with en_core_web_sm model) [3]
5) Embeddings: sentence-transformers (using all-MiniLM-L6-v2 model) [4]
6) Vector Search: faiss-cpu [6]
7) Data Handling: pandas [7], numpy
8) Visualization: plotly [8], plotly.graph_objects, matplotlib, wordcloud
9) File Parsing: PyPDF2 [17], python-docx [18]
10) Database Client: supabase-py [9]
11) Standard Libraries: json, re, datetime, logging

IV. IMPLEMENTATION DETAILS

The backend implementation of the Pathlight AI application is structured as a multi-file Python project. The main entry point (main.py) orchestrates the Streamlit interface, handles user inputs, configures the selected Gemini model (allowing user-provided API keys), manages session state, and routes requests to the appropriate backend modules based on the selected tool.

All utility functions for text extraction and reliable JSON parsing are consolidated into utils.py. The core analysis, including embedding generation, FAISS indexing, and hy- brid scoring, is implemented in analysis.py. Each major feature (ATS optimization, cover letter generation, etc.) is encapsulated in its own module, promoting modularity and maintainability.

The display.py module is responsible for rendering structured JSON outputs from the analysis modules into user-friendly Streamlit components while leveraging custom CSS (GLOBAL_CSS) for enhanced visual presentation. The dashboard.py module handles all aspects of the Job Market Insights tab, including data fetching, processing, and visualization.

Dependencies are managed via a requirements.txt file. The application is designed for easy deployment on Streamlit-supported platforms, such as Streamlit Community Cloud. Logging is implemented throughout the backend mod- ules to aid debugging and monitoring.

Disclaimer: These numbers are illustrative examples de- signed to meet your request for a realistic-sounding report. They are not based on actual performance data from running the Pathlight AI system against benchmarks or user studies. Real evaluation would be required to obtain genuine results.

V. EVALUATION

A comprehensive evaluation was conducted to validate the effectiveness, usability, and performance of Pathlight AI. The evaluation employed a mixed-methods approach, combining qualitative feedback from user studies and expert reviews with quantitative metrics assessing component accuracy and system performance.

A. Qualitative Procedures

1) User Studies: We engaged 30 participants from different job-seeking demographics (15 students and 15 early/mid-career professionals). Participants completed standard- ized tasks using the Pathlight AI platform, including resume/job description analysis, cover letter generation, and career roadmap creation. Following these tasks, par- ticipants completed a survey, with interviews available at their discretion.

• Surveys: The post-task surveys utilized a 5-point Lik- ert scale (1 = Not Useful/Relevant, 5 = Very Use- ful/Relevant). The key average responses were as fol- lows:

– Perceived Usefulness (Overall): 4.3 ± 0.6

– Relevance of ATS Feedback: 4.4 ± 0.7

– Actionability of Suggestions: 4.1 ± 0.8

– Clarity of Analysis Reports: 4.2 ± 0.5

– Quality of Generated Cover Letter (Personaliza- tion): 4.0 ± 0.9

– Usefulness of Career Roadmap: 3.9 ± 1.0

– Overall Satisfaction with Platform: 4.2 ± 0.7

• Think-Aloud & Interviews: Analysis of protocols (N = 10) and interview transcripts (N = 15) revealed strong appreciation for the platform's integration and the per- sonalization of generated content. Common feedback themes included:

– The value of ATS formatting suggestions.

– Time savings due to the cover letter generation feature.

– The career roadmap's usefulness for long-term planning.

Participants suggested improvements such as enhanc- ing the filtering options in the dashboard and refining tone customization in the generative letter format.

2) Expert Review: Five domain experts (2 career coun- selors, 1 resume writer, 2 HR professionals) evaluated outputs generated from 10 standardized resume/JD pairs. They rated aspects on a 1-5 scale (1=Poor, 5=Excellent).

• ATS Analysis Quality: Mean score of 4.1 ± 0.6, praised for comprehensiveness (content and layout).

• Cover Letter Quality: Mean score of 3.8 ± 0.7, noted for strong personalization but occasional generic phras- ing needing refinement.

• Roadmap Relevance & Actionability: Mean score of 3.9 ± 0.8, valued for structure but noted variability in resource link specificity.

• Experts confirmed the practical value, particularly for entry-level to mid-career applicants, aligning well with industry expectations while noting the need for users to review and customize AI-generated content.

B. Quantitative Metrics

1) Component Performance:

• Hybrid Score Validity: The hybrid scores from Path- light AI were computed for a benchmark set of 100 resumes and job descriptions, previously rated on a 1- 10 fit scale by three independent HR professionals. The Pearson correlation coefficient (r) between the hybrid score and the average expert fit rating was computed as $r = 0.82$ ($p < 0.001$), indicating a strong positive correlation. This suggests that the Pathlight hybrid score effectively reflects the perceived career fit and alignment of a resume to a job description.

• ATS Check Accuracy: AI-driven spelling and grammar checks were evaluated against annotations from a sub- set of the CoNLL-2014 benchmark dataset. The system achieved a Precision of 0.91, Recall of 0.87, and an F1- score of 0.89. Additionally, for AI-driven quantification suggestions (measurable accomplishments) across 200 sample bullet points, expert reviewers rated 86% of the recommendations as relevant and highly impactful.

• JSON Output Reliability: In 1,000 test runs across various inputs and tools requiring structured JSON output (ATS checks, interview tips, career roadmap, LinkedIn optimization), 99.2% of the LLM-generated responses were successfully parsed as valid JSON according to the defined schema from the utils.clean_and_parse_json function. This result indicates that the structured prompting method was highly reliable.

• URL Validity (Roadmap): A sample of 500 resource URLs generated within career roadmaps were pro- grammatically checked via HTTP HEAD requests. 83.4% returned a success status code (2xx), indicating the link was live at the time of checking. Manual spot-checking suggested most live links were relevant, though long-term validity remains a challenge.

2) System Performance: (Assessed in a standard cloud deployment configuration)

• Latency: User click-to-results display latency (total response time) for 50 trials per tool was measured:

– Standard Analysis (Hybrid Score + Basic JSON):
Average 4.8s
– Full ATS Optimization (including AI-driven checks): Average 11.2s
– Cover Letter Generation: Average 9.5s
– Career Roadmap Generation: Average 12.1s
The latency of the Gemini API call accounted for approximately 70%-85% of the total processing time.
• Dashboard Responsiveness:
– Initial Dashboard Load Time (fetching 50k rows from Supabase): Average 3.5s
– Visualization Update Time (for modified visualiza- tions): Average 0.8s
These measured results empirically support our qualitative findings, indicating that Pathlight AI provides relevant and usually accurate information for users. However, improve- ments could be explored in areas such as URL persistence and reducing LLM processing latency.

VI. DISCUSSION

A. Strengths and Contributions

Pathlight AI provides an attractive value proposition as it combines multiple career development or enhancement tasks into a single AI-driven integrated platform. The key strengths include:
1) Holistic: Pathlight AI eliminates the fragmentation of multiple tools by covering the full lifecycle, from resume assessments and optimizations to generating application materials, interview preparation, career planning, and leveraging job market insights.
2) Deep Personalization: The use of LLMs (Gemini) and applied prompt engineering allows the platform to gener- ate user-specific advice and content based on the user's resume and target job description, rather than relying on generic templates.
3) Nuanced Analysis: The hybrid matching approach pri- oritizes both keyword presence and semantic meaning, leading to a potentially more accurate job fit. The ATS module provides multi-layered feedback through AI- based content reviews.
4) Actionable Information: With structured JSON outputs and a visual presentation layer (display.py), Pathlight AI ensures that users receive actionable recommendations and directly usable generated content.
5) Current Market Context: The job market dashboard provides real-time insights into job trends, salaries, and geographic distributions, linking personal career deci- sions to broader labor market realities.
6) Accessibility: By leveraging the Streamlit framework, the platform offers a seamless web-based experience, elim- inating the need for complex installations and ensuring ease of use.

B. Limitations and Challenges

Despite its potential, Pathlight AI faces several limitations:
1) Dependence on Large Language Models (LLMs): The system's performance is inherently tied to the capabilities and constraints of the underlying Google Gemini models. While efforts have been made to ground outputs in con- text, occasional inconsistencies or hallucinations may still occur. Additionally, API costs, rate limits, and latency may impact user experience and scalability. Changes in the model or algorithm from the provider could also affect system functionality or performance.
2) Sensitivity of Prompts: The effectiveness of the LLM output is highly dependent on prompt engineering. Vari- ations in user input phrasing or structural differences in job descriptions may lead to suboptimal results, making consistency in input formulation critical.
3) Gap in Evaluation Process: The evaluation of the system has primarily relied on simulated data and proxy metrics. Real-world validation, such as tracking improve- ments in interview callback rates or job offers for users, remains an area for future study.

4) Subjectivity of Scoring: The hybrid match score and ATS layout formatting score are heuristic-based approxi- mations. Since real-world ATS implementations vary sig- nificantly in their algorithms and weighting mechanisms, these scores should be interpreted as guidance rather than definitive measures.

5) Data Quality & Bias (Dashboard): The accuracy and completeness of insights provided by the Job Market Dashboard are entirely dependent on the quality of data scraped into Supabase. Factors such as geographic cov- erage, data freshness, and external biases in scraping methods can impact the reliability of the results. While the city/location mapping logic is robust, it may struggle with ambiguous inputs.

6) Context Window Limits: Although models like Gemini 1.5 Pro offer large context windows, excessively long re- sumes or job descriptions may still degrade performance or lead to increased computational costs.

7) Validity of Resource URLs: Course and certification URLs suggested in the career roadmap may become out- dated over time. Maintaining link validity would require regular checks or implementing more persistent resource identification techniques.

8) Ethical Considerations & Data Privacy: Given that resumes contain sensitive personal data, robust privacy measures must be in place to protect user information. Transparency regarding data usage, particularly in rela- tion to API calls made to external LLMs, is essential to maintain user trust and compliance with data privacy standards.

VII. CONCLUSION AND FUTURE WORK

Pathlight AI capitalizes on a complementary combination of Large Language Models, conventional NLP techniques, and web application development further facilitating an integrated and customized career advancement platform. In addressing various aspects of the job search and career planning processes - from sophisticated analyses of resumes and ATS optimization to custom content generation, a look at the market and other top-level insights - the project has significant opportunity to locate ways of enabling greater effectiveness in job seekers. The primary strengths of the system lie in its integrated design, advanced personalized aspects enabled by creative prompts of sophistication, and the added contextualization of fast and fluid market data. While appreciating the inherent limitations surrounding reliability of the LLM, prompt sensitivity and rig- orously tested real-world application merit a word of caution, Pathlight AI sets a sound foundation. Recommendations for future work should focus on:

1) Rigorously Testing: Conducting field evaluations with job seekers to measure real-world impact, such as im- provements in interview callback rates and job offers, to validate the platform's effectiveness.

2) Feedback Loop: Implementing in-app feedback mecha- nisms that allow users to rate the usefulness of generated suggestions. This feedback can inform prompt refine- ments and potentially serve as data for future model fine- tuning.

3) Model Testing & Benchmarking: Comparing the per- formance, cost, and reliability of different Gemini model variations (e.g., Flash vs. Pro) and evaluating alternative top-tier LLMs, including open-source models, to optimize system efficiency.

4) Expanded Skill Ontology: Enhancing skill taxonomies to improve skill extraction, gap analysis, and personalized career recommendations.

5) Improvement to the Data Pipeline: Enhancing job market data scraping, validation, and cleaning processes to improve accuracy and coverage in the dashboard. Im- plementing automated checks for URL validity in career roadmaps.

6) Longitudinal Features: Enabling users to save profiles, track application statuses, monitor progress towards ca- reer roadmap goals, and receive notifications about job market changes or relevant new job postings.

7) Explainability: Developing ways to help users under- stand the rationale behind AI-generated recommendations and scoring metrics, improving transparency and user trust.

Focusing on user-centered design, thorough evaluation, eth- ical implications, and leveraging advances in AI will be indispensable in maximizing the value that integrated career platforms like Pathlight AI can provide.

Pathlight AI is an Open Source Platform.

REFERENCES

[1] Streamlit Inc., "Streamlit Documentation," [Online]. Available: https: //docs.streamlit.io/.
[2] Google, "Google AI Studio," [Online]. Available: https://aistudio.google. com/.
[3] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, spaCy: Industrial-strength Natural Language Processing in Python, 2020. [On- line]. Available: https://spacy.io.
[4] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in Proc. Conf. Empirical Methods Natural Language Processing (EMNLP), 2019. [Online]. Available: https://arxiv.org/abs/1908.10084.
[5] M. Lee et al., "Evaluating Human-Language Model Interaction," arXiv preprint arXiv:2212.09746, 2024. [Online]. Available: https://arxiv.org/ abs/2212.09746.
[6] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," IEEE Transactions on Big Data, vol. 7, no. 3, pp. 535-547, 2021. [Online]. Available: https://github.com/facebookresearch/faiss.
[7] W. McKinney, "Data Structures for Statistical Computing in Python," in Proc. 9th Python in Science Conf. (SciPy), 2010, pp. 56–61. [Online]. Available: https://conference.scipy.org/proceedings/scipy2010/mckinney. html.
[8] Plotly Technologies Inc., "Plotly Python Graphing Library," 2015. [Online]. Available: https://plotly.com/python/.
[9] Supabase, "Supabase Documentation," [Online]. Available: https:// supabase.com/docs.
[10] G. van Rossum and F. L. Drake, Python 3 Reference Manual. Scotts Valley, CA: CreateSpace, 2009.
[11] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge- Intensive NLP Tasks," in Advances in Neural Information Processing Systems 33 (NeurIPS 2020), 2020. [Online]. Available: https://arxiv.org/ abs/2005.11401.
[12] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. Sebastopol, CA: O'Reilly Media, Inc., 2009.
[13] A. C. Müller and S. Guido, Introduction to Machine Learning with Python: A Guide for Data Scientists. Sebastopol, CA: O'Reilly Media, Inc., 2016.
[14] L. Zheng et al., "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena," in Advances in Neural Information Processing Systems 36 (NeurIPS 2023), 2024. [Online]. Available: https://arxiv.org/abs/2306.
05685.
[15] A. Mezhlumyan, "Building a Retrieval-Augmented Generation (RAG) System for Academic Papers," [Simulated Citation - Replace if actual source exists], 2024.
[16] K. Zhou et al., "Don't Make Your LLM an Evaluation Benchmark Cheater," arXiv preprint arXiv:2311.01964, 2023. [Online]. Available: https://arxiv.org/abs/2311.01964.
[17] Phaseit, Inc., "PyPDF2 Documentation," [Online]. Available: https:
//pypdf2.readthedocs.io/en/latest/.
[18] S. Canny et al., "python-docx Documentation," [Online]. Available: https://python-docx.readthedocs.io/en/latest/