**The Transactions of the Korea Information Processing Society**
**TKIPS** Vol.13, No.9, pp.460~467

eISSN: 3022-7011
https://doi.org/10.3745/TKIPS.2024.13.9.460

# Enhancing Career Development Utilizing LLM for Targeted Learning Pathway

Mahisha Patel† · Vishakha Tyagi† · Isabel Hyo Jung Song††

## ABSTRACT

Targeted career development is critical for student success but is often lacking for underrepresented students at many public higher-education institutions due to insufficient career counseling resources. We propose an innovative career development tool leveraging Large Language Models (LLMs) to enhance student career prospects through three steps: (1) identifying relevant jobs by analyzing resumes, (2) pinpointing skill gaps using external resources such as classroom assignments, in addition to resumes, and (3) suggesting customized learning paths. Our tool accurately matches jobs in real-world settings, identifies true skill gaps while reducing false positives, and provides learning paths that receive high satisfaction scores from faculty. Future research will enhance the solution's capabilities by incorporating diverse external resources and leveraging advancements in LLM technology to better support early-stage career seekers.

Keywords : Career Development, Job Matching, Skill Gap Identification, Customized Learning Paths, LLM

# 경력 개발 증진을 위한 LLM 기반 맞춤형 학습 경로 개발

Mahisha Patel† · Vishakha Tyagi† · 송 효 정††

## 요 약

개인 맞춤형 경력 개발은 학생 성공에 필수적이지만, 많은 공립 고등 교육 기관에서 자원 부족으로 인해 소외된 학생들에게 충분한 진로 상담이 제공되지 않는 경우가 많다. 본 논문에서는 언어 모델(LLMs) 및 prompt engineering 기법, agent 기법을 활용하여 학생들의 미래 진로 개발을 위한 솔루션을 제안한다: (1) 이력서를 분석하여 후보 직업을 식별하고, (2) 이력서 외에 학생들이 기 제출한 학습 과제와 같은 외부 자원을 사용하여 채용공고 대비 학생의 실제 기술 격차 (skill gap)를 정확히 찾아내며, (3) 기술격차에 기반한 맞춤형 학습 경로를 제안한다. 이 도구는 실제 환경에서 채용공고를 정확하게 매칭하고, 거짓 긍정(false positive) 줄이면서 학생의 이력서와 채용공고와의 기술 격차를 정확히 식별하며, 교수진으로부터 높은 만족도를 받는 개인별 학습 경로를 제공한다. 향후 연구에서는 다양한 외부 자원을 추가로 활용하고 더 진화된 prompt engineering 및 agent 기법을 적용하는 솔루션 개발을 목표로 하고 있다.

키워드 : 경력 개발, 직무 매칭, 기술 격차 식별, 맞춤형 학습 경로, LLM

## 1. Introduction

Guiding students from disadvantaged backgrounds toward establishing their career paths presents significant challenges. Unfortunately, many higher-education institutions in the U.S. and in Korea, which serve a substantial number of these students, lack the necessary resources to provide targeted career services for them. These institutions often fail to provide adequate career counseling, a critical element for professional development. For instance, San Francisco State University (SFSU), a part of the extensive California State University (CSU) system, provides only a few career counselors for its 20,000 students. This scarcity of support can result in students lacking clear guidance in their initial career decisions, leading to insufficient preparation for their future. Our career development solution steps in to bridge this gap between students and the scarce career counseling support, providing a much-needed boost, particularly to underrepresented groups. The solution could potentially enhance their educational prospects and foster a marked improvement in their socieon tal standing.

Previous research online career development tools has focused on strategies related to resume building, job searching, and interview preparation. However, these studies have not fully addressed key student needs, particularly

† 비 회 원 : 미국 샌프란시스코 주립대학 학생
†† 정 회 원 : 미국 샌프란시스코 주립대학 조교수
Manuscript Received : July 22, 2024
Accepted : August 14, 2024
* Corresponding Author : Isabel Hyo Jung Song(hjsong@sfsu.edu)

the identification and development of missing skills essential for employment. Our solution fills this gap by utilizing large language models (LLM) to analyze student resumes to identify pertinent job opportunities and performing a comprehensive assessment of the skills students are lacking. This assessment leverages data from resumes, classroom assignments, and GitHub, which hosts software portfolios for computer science (CS) students, effectively addressing the common issue of students inaccurately representing their skills on their resumes. From this detailed analysis, our solution creates a personalized learning pathways using LLM to equip students with the necessary competencies.

The primary design challenges for our solution revolve around ensuring the accuracy, efficiency, and adaptability. To secure precise responses from the LLM, we utilize advanced prompt engineering techniques. To enhance the efficiency of our solution when handling extensive volumes of resumes and job descriptions (JDs), we have optimized the frequency of LLM calls, significantly reducing associated costs and latency. Furthermore, to keep pace with ongoing advancements in LLM technology, we have adopted a modular software architecture that facilitates easy updates and the seamless integration of emerging LLMs.

The remainder of this paper is structured as follows: Section 2 reviews previous research on online career development tools and the ML (Machine Learning) techniques applied to essential components such as skill extraction and job matching. Section 3 details the architecture of our solution and describes our techniques to leverage LLM, such as prompt engineering. Section 4 evaluates the effectiveness of our solution. Finally, Section 5 concludes the paper by discussing future research directions to extend this work in aiding early stage career seekers in real environments.

## 2. Related Work

### 2.1 Career Development Solutions

A variety of online career development tools are readily accessible to assist students and individuals at the early stages of their careers. These platforms offer services such as resume building, resume assessment, job searches, and interview practice. Tools like MyPerfectResume[1], Kickresume[2] facilitate the resume creation process by providing content recommendations based on user inputs, simplifying job application preparations for candidates.

Additionally, platforms like EasyResume[3], and VMOCK [4] offer in-depth resume feedback, aiding candidates in refining their resumes. LinkedIn, Indeed, MyPerfectResume, and Zety assist job seekers by customizing job searches based on individual profiles and preferences, while Kickresume offers practice interview questions to users. However, these commercial solutions typically do not disclose their internal mechanisms. In addition, there remains a significant shortage of tools that can thoroughly identify skill gaps based on a user's comprehensive background, including previous courses and projects, and develop customized learning pathways tailored to individual needs.

### 2.2 ML for Career Development Solutions

With the advent of ML technology, many research studies have explored its application in key areas of career development such as skill extraction and job matching. Jivtode et al. [5] developed an NLP (Natural Language Processing)-based system utilizing the Natural Language Toolkit (NLTK) [6] for extracting key information from resumes, which is then assessed using Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) classifiers against JDs. The system achieves an accuracy of 80% in resume analysis. Tejaswini [7] introduced a novel method for resume screening that uses TF-IDF vectorization [8] and cosine similarity [9] to match resumes with JDs, supplemented by KNN to select and rank the most relevant candidates. They report a text parsing accuracy of 85% and a ranking accuracy of 92%, indicating the system's effectiveness.

Prasad [10] employed a Graph Neural Network (GNN)-based domain adaptation approach combined with NLP for feature extraction and semantic understanding, achieving a 97% accuracy in job category prediction. NLP techniques such as Named Entity Recognition (NER) [11] and word embeddings [12] significantly enhance the matching of resumes to JDs.

With the advent of LLMs, semantic-based analysis has been proposed to discern the subtle nuances in resumes and JDs. Li [13] introduced SkillGPT, a RESTful API service for skill extraction from unstructured JDs by utilizing Llama [14] via summarization and vector similarity search. To classify job types, Clavié [15] examined the effectiveness of various text classification methods, including SVM and DeBERTa [16]. They also used LLM like GPT-3.5 [17], employing prompt engineering to enhance accuracy. A comparative analysis reveals that a GPT-3.5-turbo, when guided by well-crafted prompts, surpasses all other models, showing a 6% improvement in accuracy.

Prompt engineering is the process of designing and refining users' input prompts to optimize the performance and output quality of LLM. Extensive research has been conducted in this area to ensure effective task completion. Techniques include the Chain of Thought (CoT) [18] method, Socratic prompting, and reiteration [19]. The CoT method facilitates the LLM in delineating its thought processes in a clear and structured manner, enhancing the transparency of its decisions. The Socratic technique stimulates deeper exploration, through providing examples, and nudging to draw the conclusion. Reiteration helps the model to remember key information.

Despite these technological advances, many job matching and skill extraction tools still face significant practical limitations, especially for students early in their careers in resource-constrained settings. Existing solutions are primarily designed to identify skills listed in resumes to facilitate job searches, to streamline the application process itself. However, for students in the early stages of their academic journeys, a crucial concern is identifying and developing the skills they lack throughout their education. In this paper, we propose a solution leveraging state-of-the-art LLMs and relevant technologies to address this gap.

## 3. Proposed Methodology

To address the shortcomings of existing career development solutions, we employ several critical strategies to enhance both job matching and skill development: (Step 1) identifying relevant JDs, which outline the responsibilities, qualifications, and skills required for specific roles, by analyzing students' resumes in real-world conditions, (Step 2) pinpointing skill gaps by leveraging external resources such as GitHub [20] and class assignments, and (Step 3) suggesting customized learning paths based on accurately identified skill gaps to equip students with the necessary skills. In Section 3.1, we explore the design challenges and propose a software architecture to overcome these challenges. In Section 3.2, we elaborate on how we utilize LLMs, including prompt engineering and the agent concept, which are crucial to the effectiveness of our tool.

### 3.1 ML for Career Development Solutions

To develop the system, we address three primary design challenges: accuracy, efficiency, and adaptability. We aim to deliver high accuracy, optimize operations, and maintain flexibility to accommodate advances in LLMs. Fig. 1 of the overall architecture demonstrates how our solution addresses these challenges.

In Step 1 of job matching, we ensure high accuracy in real-world settings by considering factors such as seniority levels and visa status, which are particularly crucial for early-stage career seekers. To enhance efficiency, we implement a preprocessing step that initially filters out jobs based on years of experience and citizenship requirements. This approach reduces unnecessary data processing in later steps and decreases the need for frequent LLM queries, thereby lowering costs and reducing latency.

Given the matched job in Step 1, in Step 2, we first calculate the missing skills based on the resume. To enhance the accuracy of this identification, we utilize external tools such as Canvas [21], a widely used Learning Management System (LMS), and GitHub portfolios particularly for computer science students, to access students' past assignments and projects. It is because that students' resumes might not fully reflect all relevant skills often due to their inexperience in resume building. Our thorough approach allows us to accurately pinpoint skill gaps. In Step 3, we develop tailored learning pathways based on the identified missing skills. Throughout Steps 1 to 3, we apply sophisticated LLM techniques to ensure precise responses, which will be detailed in Section 3.2.

In this paper, we use GPT-4o for the best performance after testing other state-of-the-art LLMs, including Google Gemini [22]. Aware of the rapid advancements in LLM technology, we have implemented an LLM interface within our architecture to ensure our system remains adaptable

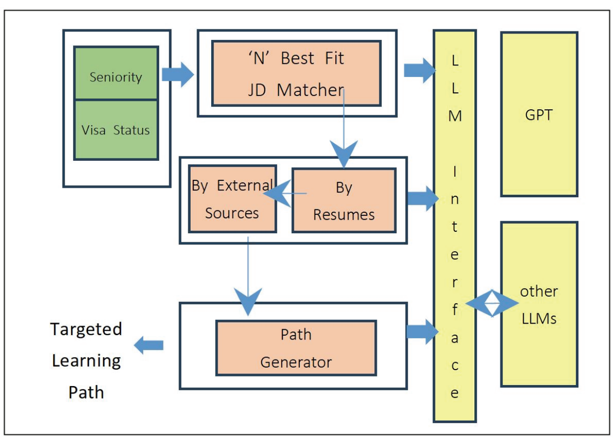



Fig. 1. Overall architecture

to future LLMs with minimal disruption. This interface allows the rest of the software to consistently interact with the LLM, encapsulating any changes to future LLMs. This approach ensures the adaptability of our software.

### 3.2 ML for Career Development Solutions

Throughout Steps 1 to 3, we employ advanced prompt engineering techniques to guarantee accurate responses from the LLM. To improve the LLM reasoning capability, our prompt engineering is based on the CoT method, the Socratic prompting technique, and reiteration. Table 1 gives the prompt used in Step 1 for finding matched JDs. It gives a step by step prompt using CoT by detailing each step, provides examples and reiterates a key message.

In Step 2, to accurately identify missing skills, we utilized LangChain [23] to configure a workflow involving an LLM and two custom-built tools. The LLM, using a CoT-style prompt, initially identifies missing skills by comparing the resume with the JDs from Step 1. For any identified missing skills, LangChain then calls the Canvas retrieval tool to find relevant skills from past class assignments, filtering out false positive missing skills. If missing skills still remain, LangChain employs the second tool to query GitHub, further filtering out any missing skills present in the GitHub portfolio project. This comprehensive approach ensures the precise identification of true missing skills.

Once the true missing skills are identified, we use reflection to generate a customized learning path. Table 2 shows the prompts used for STEP 3. Initially, we generate the LLM response for the learning path (initial generation) and ask the LLM to provide a feedback score based on our criteria. We then regenerate the LLM response, incorporating the feedback, until each criterion receives a score of at least 8 out of 10, with 10 being the highest score. Table 3 shows the final response for a customized learning path to learn C++ based on the student's prior Python skill. During the two iterations, the tool received self-feedback indicating low engagement, suggesting improvements such as incorporating engaging activities and online resources with interactive elements for hands-on practice. The final response is presented in Table 3, with the improvements highlighted in grey to reflect the incorporated feedback.

Table 1. Prompts for Step 1

| Prompt | Used Method |
|---|---|
| You are provided with a JD (Job Description) of {json.dumps(job)} and resume of {json.dumps(resume)}. You will generate a score by performing a semantic matching of the entire content of resume with the JD. Generate a numerical score out of 10 by matching the skills, title, experience, academics and qualifications listed in JD with candidate's resume (10 if most of the skills are matched). Return only a numerical score as output. | CoT |
| Assess and match the job title to determine the job domain. | CoT |
| For example, the job title 'Front-end Software Engineer' requires a background in front-end web development, including skills in HTML, CSS, JavaScript, and related technologies. | Socratic method (example) |
| Another important factor is that the skills of the candidate should significantly influence their score. If the candidate does not have at least half of the skills listed in the JD, that candidate should receive a low score. | CoT |
| For example, if the JD lists skills such as Python, Django, Flask, SQL, a resume having Python, Flask and SQL should receive a high score. | Socratic method (example) |
| Think step by step but remember to return only a numerical score in answer. | reiteration |

Table 2. Prompts for Step 3

| Prompt | Method |
|---|---|
| You have the following missing skills: {json.missing-skills}, and you already possess these skills: {json.current-skills}. Please identify a targeted learning path to master each missing skill. The generated response should be: 1) relevant: The learning path should leverage your current skills, 2) informative: Provide detailed steps with specific sources for reference, and 3) engaging: Include interactive elements such as hands-on projects. | initial generation |
| Evaluate your response based on the following criteria: 1) Relevance, 2) Informativeness, 3) Engagement. Please provide a score for each out of 10 (10 being the highest). | feedback |
| Given the feedback, please regenerate a targeted learning path to master each missing skill. | regeneration |

Table 3. Output of Step 3

| Final Response |
| --- |
| Mastering C++: A Targeted Path for Programmers (Enhanced Engagement)<br>Building upon your Python skills, mastering C++ becomes an exciting opportunity to expand your programming prowess. Let's leverage your existing knowledge and create an engaging learning path that incorporates fun, hands-on projects with interactive elements!<br><br>Step 1: Building the Bridge (1-2 weeks)<br>Familiarity Boost: While Python and C++ have different syntaxes, the core programming concepts (variables, loops, conditionals, functions) remain similar.<br><br>Take a crash course specifically designed for programmers transitioning from Python to C++. Websites like Udacity or Codecademy offer great options. Focus on recognizing the C++ equivalents of familiar Python constructs. This will give you a head start and establish a foundation for the deeper concepts to come.<br><br>Step 2:DivingDeeper(4-6weeks)<br>Solidify the Basics: Now, delve into core C++ concepts that might be new to you:<br>Pointers and Memory Management: This is crucial in C++.<br><br>Utilize online resources like tutorialspoint (https://www.tutorialspoint.com/cplusplus/cpp_pointers.htm) to grasp the concepts.<br>References: Understand how references provide an alternative for passing arguments to functions.<br><br>Classes and Objects: Object-Oriented Programming (OOP) is a fundamental paradigm in C++. There are many great online resources like LearnCpp with detailed explanations and examples.<br><br>Hands-on Project 1: Building a Choose-Your-Own-Adventure Game with Online Challenges: This project allows you to practice core concepts like variables, functions, user input, and conditional statements in a fun and engaging way. Find online tutorials that guide you through creating a text-based adventure game in C++.<br><br>Bonus: Spice it up by adding multiple endings based on player choices or incorporating basic inventory management!<br><br>Step 3: Power Up with C++ Features (6-8 weeks)<br>Templates: Learn how templates enable generic programming for functions and classes, increasing code reusability. Explore resources like the C++ Reference (https://en.cppreference.com/w/cpp/language/templates) for detailed explanations.<br><br>Standard Template Library (STL): The STL provides powerful containers (like vectors, lists) and algorithms (sorting, searching) for efficient data manipulation. Utilize resources like CPPlusPlus to explore the STL in detail.<br>Hands-on Project 2: Building a Data Analyzer with Interactive Visualization:<br><br>This project helps you practice using STL containers (vectors) to store and manipulate data. You can build a program that reads data from a file (CSV format) and performs calculations (e.g., finding averages).<br>Step 4: Elevate Your Skills (ongoing)<br>... |

## 4. Results

To evaluate the effectiveness of our solution, we assess each step outlined in Section 3. This detailed analysis ensures that all components perform as intended and meet students' needs, helping identify areas for improvement.

### 4.1 Job Matching in Step 1

To evaluate Step 1, we conducted two types of analysis for job matching using the resumes of five CS students from SFSU and 25 JDs collected via open-source RAPID job APIs [24]: 1) verification against human labeling and 2) comparison with other tools like LinkedIn and Indeed. Verification against human labeling ensures our tool meets the reliability of human judgment, while comparison with other tools assesses our tool's effectiveness in real-world settings.

For each of the 25 job descriptions, we conducted scoring by randomly selecting a resume from five students, with three human labelers possessing professional CS expertise and by our solution. Fig. 2 illustrates the comparison of scores, showing the average ratings given by the human labelers and those generated by our solution on a scale of 1 to 10, with 10 being the best match in terms of skills between the JD and the resume. As shown in Fig. 2, we observe a small difference between the scores from the human labelers and ours, with the largest difference being 0.67 for JD 7 (human labeling average of 6.83 and tool score of 7.5). Across the 25 JDs, the average difference between the two is 0.23 on a 10-point scale. This low difference indicates that our job matching closely aligns with human expectations.

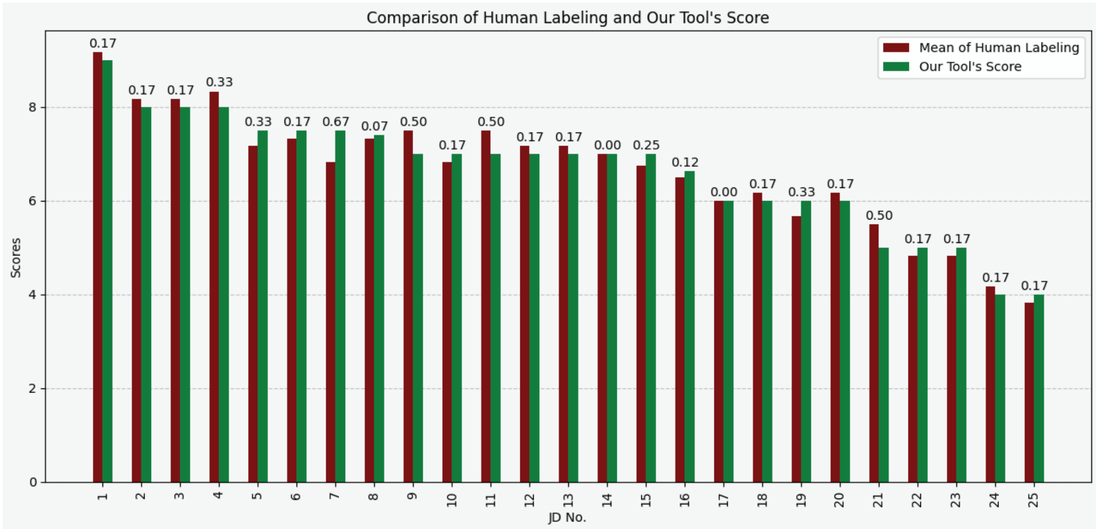



Fig. 2. Comparison of Job Matching Scores Between Human Labeling and Our Solution, with Each Data Label Indicating the Differences Between the Two.

Job matching in real-world settings requires consideration of factors beyond skills, such as seniority level and visa status. To assess our solution's effectiveness in such settings, we compared it with other commercial tools. Using the same set of 5 resumes, we searched for JDs for "software engineer" with varying required years of experience (from 0 to 11 years) and visa status (US citizenship required or not), and analyzed the top 10 match results from each tool. Comparing 50 results from LinkedIn and Indeed with 46 results from our solution (limited by the job pool from RAPID APIs), Table 4 demonstrates that ours provides better-matched jobs in terms of skills, years of experience, and visa status. Considering skills matched if 80% of the skills align between the JD and the resume, ours found 37 jobs matching in skills. Additionally, it identified 46 matches in seniority level and 46 matches in visa status. Each of these metrics outperforms LinkedIn and Indeed, highlighting the accuracy of our job matching in real-world settings, even with a limited pool of job descriptions.

Table 4. The Number of Matching Results Across Different Solutions

| Matching Criteria | LinkedIn | Indeed | Our Solution |
|---|---|---|---|
| Skills | 18 | 31 | 37 |
| Years of Experience | 34 | 33 | 46 |
| Visa status | 45 | 31 | 46 |

Table 5. Missing Skill Identification

| A combination of each resume, JD, github and assignments | Bench-mark | Our Solution (Basic Mode) | Our Solution (Advanced Mode) |
|---|---|---|---|
| Data #1 | 25 | 25 | 4 |
| Data #2 | 8 | 8 | 2 |
| Data #3 | 11 | 11 | 4 |
| Data #4 | 10 | 10 | 4 |
| Data #5 | 16 | 10 | 6 |
| Data #6 | 8 | 8 | 3 |
| Data #7 | 8 | 7 | 5 |
| Data #8 | 16 | 16 | 10 |
| Data #9 | 12 | 12 | 6 |
| Data #10 | 8 | 7 | 3 |

### 4.2 Missing Skill Identification in Step 2

To evaluate the accuracy of identifying skill gaps, we used the same set of JDs and resumes from Section 4.1, along with preselected GitHub project collections and class assignments. We analyzed our solution using ten combinations of these elements. Table 5 provides a detailed breakdown for each combination, with the "Benchmark" representing the actual number of missing skills for each resume compared to the given JD. "Our Solution (Basic Mode)" shows the number of missing skills detected by our solution based on each resume for the given JD, while "Our Solution (Advanced Mode)" reveals the truly missing skills identified after consulting external sources of GitHub and class assignments. This method demonstrates a significant reduction of up to 84% in "Data #1", decreasing the missing skills from 25 (including falsely identified missing skills) to 4 true missing skills. The identification of true missing skills enables students to focus on their genuine skill gaps. Given that students often struggle to fully list all their skills on their resumes, our solution is particularly valuable.

### 4.3 Targeted Learning Path Generation in Step 3

For the 47 true missing skills identified in Table 5 (the sum of all the missing skills identified by “our Solution (Advanced mode)”), we generated customized learning paths as detailed in Section 3.2, leveraging the student’s existing skills.

We found the generated learning paths to be highly effective. They not only facilitated transitioning from one programming language (e.g., Python) to another new skill (e.g., C++) but also supported broader skill development. For example, a student's learning path to master React, given his/her existing skills in Python, CSS, JavaScript, and HTML, suggested transitioning from JavaScript to JSX (React's JavaScript component). For mastering rendering technology, which might seem unrelated to his/her current skill set, the generated learning path recommended leveraging existing web development skills(HTML, CSS, JavaScript) and experimenting with WebGL and Three.js (well-known web-based graphics libraries). To learn the missing skill of trigonometry, the learning path suggested starting with the Python math library and Python Matplotlib to build familiarity. Each response consisted of step-by-step guidance, similar to Table 3, specifying particular resources such as websites, tutorials, and libraries. To engage students, most of generated learning paths included references to hands-on projects (e.g., "interactive 3D scene") and online communities (e.g., HackerRank, Reddit). Due to space constraints, we cannot include the each output here but it follows a format similar to Table 3.

To solidify our analysis, we involved three CS faculty members who manually reviewed the responses, evaluating them based on the criteria of Relevance, Informativeness, and Engagement, as defined in Section 3. Each faculty member graded the responses on a scale of 1 to 10 (with 10 being the best). Table 6 shows the average scores for the generated learning path for each of the 47 missing skills across each criterion. Both faculty members were satisfied with the Relevance, Informativeness and Engagement of the responses, each giving scores above 8.0.

Table 6. Faculty Evaluation for Personalized Career Path Identification

| Evaluation Criteria | Faculty #1 | Faculty #2 | Faculty #3 |
|---|---|---|---|
| Relevance | 9.1 | 9.0 | 9.1 |
| Informativeness | 9.1 | 9.1 | 8.9 |
| Engagement | 8.0 | 9.0 | 8.5 |

## 5. Conclusion

This study introduced a career development solution that leverages LLMs to enhance student career prospects through three steps: (1) identifying relevant jobs by analyzing students' resumes in real-world conditions, (2) pinpointing skill gaps by leveraging external resources, and (3) suggesting customized learning paths that build on their existing skills. This is particularly helpful for students from disadvantaged backgrounds, addressing the gap between limited career counseling resources and the need for personalized guidance.

We demonstrated that our job matching solution accurately identifies true skill gaps, significantly reducing false positives and generating customized learning paths. These paths were positively evaluated by CS faculty members for relevance, informativeness, and engagement. Our findings highlight the potential of using LLMs, prompt engineering, and agent workflows for personalized career development. To our knowledge, this is the first paper to utilize prompt engineering and agents for career development solutions using external tools.

Future research will enhance our solution’s capabilities in several key areas. Although initially implemented for CS students, it can be easily expanded into other majors using different tools. We will ensure our solution evolves with advancements in LLMs, prompting technologies, and agent technologies. Additionally, we plan to assess our tool’s efficiency in real-world scenarios with a large volume of job descriptions and resumes, making necessary improvements.

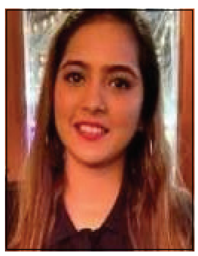
We will explore incorporating real-time labor market data to provide students with up-to-date information on in-demand skills and emerging job opportunities, making the solution's guidance more dynamic and responsive. We will also conduct longitudinal studies to assess the long-term impact of the solution on students' career outcomes, examining metrics such as job placement rates, career progression, and overall satisfaction. By continually refining and expanding our solution, we aim to provide comprehensive support to early-stage career seekers, helping them navigate the job market and achieve their professional goals.

## References

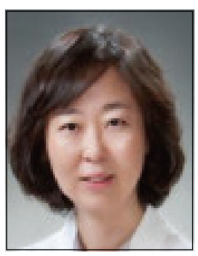
[1] Myperfectresume, “Myperfectresume,” [Internet], https://www.myperfectresume.com/.

[2] Kickresume, “Kickresume,” [Internet], https://www.kickresume.com/.
[3] EasyResume, “Easy Resume: Professional Resume Builder,” [Internet], https://www.easyresume.io/.
[4] Vmock, “Vmock Dashboard,” [Internet], https://www.vmock.com/.
[5] A. Jivtode, et al., “Resume Analysis Using Machine Learning and Natural Language Processing.”
[6] E. Loper, and S. Bird, “Nltk: The natural language toolkit,” *arXiv preprint cs/0205028*. 2002.
[7] K., Tejaswini et al., “Design and development of machine learning based resume ranking system,” *Global Transitions Proceedings*, Vol.3, No.2, pp.371-375, 2022.
[8] T. Roelleke and J. Wang, “TF-IDF uncovered: a study of theories and probabilities,” *ACM SIGIR Conference on Research and Development in Information Retrieval*, 2008,
[9] N. Dehak, et al., “Cosine similarity scoring without score normalization techniques,” In *Odyssey* (Vol. 15). 2010,
[10] B. L. Prasad et al., “An Advanced Real-Time Job Recommendation System and Resume Analyser,” *International Conference on Self Sustainable Artificial Intelligence Systems*, 2023,
[11] J. Li et al., “A survey on deep learning for named entity recognition,” *IEEE Transactions on Knowledge and Data Engineering*, Vol.34, No.1, pp.50-70, 2020.
[12] A. Dekhtyar and V. Fong, “Re data challenge: Requirements identification with word2vec and tensorflow,” IEEE (RE), 2017,
[13] N. Li, B. Kang, and T. De Bie, “SkillGPT: a RESTful API service for skill extraction and standardization using a Large Language Model,” *arXiv preprint arXiv:2304.11060*, 2023.
[14] Llama, “llama”, [Internet], https://llama.meta.com/.
[15] B. Clavié, et al., “Large language models in the workplace: A case study on prompt engineering for job type classification,” In *International Conference on Applications of Natural Language to Information Systems* (pp. 3-17). 2023,
[16] P., He, X., Liu, J. Gao, and W., Chen, “Deberta: Decoding-enhanced bert with disentangled attention,” *arXiv preprint arXiv:2006.03654*, 2020.
[17] OpenAI, openAI’s GPT3.5, [Internet], https://platform.openai.com
[18] J. Wei et al., “Chain-of-thought prompting elicits reasoning in large language models,” *Neural Information Processing Systems*, 2022.
[19] E. Chang, “Prompting large language models with the socratic method,” *Computing and Communication Workshop and Conference*, 2023.
[20] Github, “Code With GitHub,” [Internet], https://github.com/
[21] Canvas, “Canvas-Infrastrucutre”, [Internet], https://www.instructure.com/canvas
[22] Google, Gemini, [Internet], https://gemini.google.com/app
[23] LangChain, “LangChain”, [Internet], https://python.langchain.com/
[24] RapidAPI, “Rapid API Hub,” [Internet], https://rapidapi.com.

Mahisha Patel
https://orcid.org/0009-0003-5159-4355
e-mail : mpatel17@sfsu.edu
She received an M.S. degree from San Francisco State University (SFSU) in May 2024. Her research interests include Machine Learning.



Vishakha Tyagi
https://orcid.org/0009-0002-9847-8450
e-mail : vtyagi@mail.sfsu.edu
She received an M.S. degree from SFSU in May 2024. Her research interests include Machine Learning and Data Mining.



Isabel Hyo Jung Song
https://orcid.org/0000-0001-6758-8599
e-mail : hjsong@sfsu.edu
She has been an assistant professor at SFSU since 2019. Her research interests include Machine Learning and education.