Interactive Personalized Job Recommender

CSE 6242 Final Report

Liqing Jing, Min Cai, Yiwen Wang, Bohan Ye, Shunyu Wang,

Introduction

In the digital age, the emergence of digital technology and the widespread availability of online platforms have introduced both challenges and opportunities for job seekers. As upcoming job seekers, we are motivated by the desire to create a tool that can streamline the job search process, hoping it would be useful for our own job hunting. Hence, we strived to develop an interactive job recommendation system that leverages the power of analytics, machine learning algorithms, and interactive interfaces to deliver personalized job recommendations.

Problem Definition

Given the over-abundance of job listings, sifting through countless postings to uncover positions that align with one's skill set is often a disheartening endeavor. Despite the presence of existing job recommendation systems, which do provide personalized recommendations, they frequently overlook the significance of data visualization in effectively conveying recommendation results. Thus, the problem we try to solve is to develop an end-to-end solution that seamlessly integrates the capabilities of a recommendation system with interactive data visualization, empowering job seekers to efficiently find their dream jobs.

Literature Survey

The team surveyed and synthesized findings from 18 papers across the following fields to understand the current job recommendation practices and associated limitations.

<u>Domain-Specific Recommendation Algorithms:</u> Alotaibi [1] and Siting et al. [2] produced overviews of the domain and explored recommender systems' role in recruitment. Zhao et al. [3] proposed a two-staged embedding-based system with retrieval and rerank stages. The system uses data from CareerBuilder.com and offers insights into evaluation methodologies applicable to our project. Almalis et al. [4] introduced FoDRA, a novel content-based algorithm. While offering valuable insights to our project's model refinement, it overlooks implicit semantic relations. Shalaby et al. [5] introduced a graph-based job recommendation approach for online platforms, overcoming scalability and relevance challenges. Despite its value for our algorithm, enhancements are needed in parameter settings, language support, and behavioral pattern integration. Yi et al. [6] proposed an item-based collaborative filtering algorithm integrating position descriptions and resumes for an improved accuracy, though it lacks the latest collaborative filtering advancements. S. K. A. S et al. [7] introduced SVD methods for ranking of similar documents for job recommendation, though concerns existed for computational resources. Lu et al. [8] introduced a hybrid recommender utilizing content-based and interaction-based relations, providing insights into recommendation cases useful for our project.

Kumalasari and others [9] presented an IT job recommendation system based on collaborative filtering and LinkedIn skills endorsement data, offering insights applicable to our project. Yadalam et al. [10] developed an ML-based system to assist new graduates in finding job opportunities, while highlighting challenges like cold start and scalability. Aligning with our project's goals, pre-registration user preference collection could enhance our system's usability.

<u>Data Extraction & Search Frameworks:</u> Karakolis [11] and Gottipati [12] et al. applied NLP to extract skills from job postings, providing insights to our project. However, limited scope of research and static presentation call for a larger dataset and interactive dashboard in our implementation. For more advanced implementations, Kang et al. [13] developed SkillGPT based on LLM for extracting and standardizing job skills, and Bhola et al. [14] predicted required

skills from job descriptions using Extreme Multi-label Classification. Improving recognition of implicit skills is crucial for accuracy while addressing randomness is crucial for system reliability. On the topic of search systems, Wang et al. [15] proposed a framework for semantic search. Zamfir et al. [16] used Elasticsearch for efficient system monitoring and big data analysis. Their work is relevant to our project of processing and analyzing large amounts of user data related to job skills. Kononenko et al. [17] highlighted capabilities of Elasticsearch in software analytics and real-time search, which can enhance the accuracy and relevance of our recommendation system. Minnie and Srinivasan [18] presented text-based search engine algorithms, with room for improvement in handling multiple keywords and phrases. All of the papers are relevant to developing the search functionalities of our system. Leveraging LLMs could enhance semantic understanding and recommendation accuracy, while optimization strategies will be essential for efficient search functionality.

Proposed method

- **Innovations**

The project aims to develop an interactive job recommendation system. This system will accept user-provided data, generate a curated list of relevant jobs, and provide interactive visualization options to users. The system compositions are shown below.

The proposed solution for the interactive job recommendation system offers several advancements over the state of the art:
<u>Integration of React and Flask Framework:</u> By leveraging React and Flask integration, the system ensures scalability and responsiveness, offering users a seamless experience.
<u>Data Source, Data Processing and SQLite Database Buildup:</u> We enable advanced search capabilities, querying features to streamline job search processes and improve user satisfaction.
<u>Utilization of LLMs</u>: We employ LLMs such as ChatGPT to extract explicit job skills and uncover implicit requirements with high skill matching and recommendation accuracies.
<u>Tableau Visualization for Job Recommendation</u>: Utilizing Tableau, we create interactive visualizations that empower users to explore and understand job trends effortlessly.

- **Detailed approaches**

The following section walkthroughs the detailed approaches, expanding on each of the four innovation areas:

<u>React and Flask Framework Integration</u>

The presentation layer uses React to build the web application and uses Flask to create a front-end service which serves the requested data to fulfill the visualizations. We built the full stack and ran it locally with examples to prove the feasibility. In the frontend, we selected Google material design as styling library and its relevant component library so as to build elegant UI controls. Furthermore, we started with an open-source UI template whose design was very close to our presentation goals. We built the basic form of controls, such as inputs and dropdowns to input search words or select filters. The web application is running at *localhost:3000* upon start. The backend is a Flask server primarily with different APIs and set up necessary libraries in Pipfile. The python libraries are SQLite3, PySpark, and Flask. The root URL of the service is *127.0.0.1:5000/api*, and we have added necessary APIs as the project moves forward. Each part has the README file that helps users to set up the environment and run successfully at the local environment. We also used Git as a version control tool and collaborated on Github [19] for source management.

<u>Data Source, Data Processing and SQLite Database Buildup</u>

To enhance project innovations, we leverage insights from existing literature and LinkedIn's comprehensive data source of 1.3 million job listings in 2024 [20]. The data source consists of three files: *linkedin_job_postings.csv*, *job_skills.csv* and *job_summary.csv*. Data cleaning and processing were performed using Python Pandas and SQLite packages. For example, rows with missing values were deleted and unimportant columns were filtered out. In *linkedin_job_postings.csv*, the job_location column was split into 2 columns of city and state for future integration into an interactive dashboard. In *job_skills.csv* and *job_summary.csv*, text was standardized by converting to lowercase, removing special characters and handling abbreviations. Then, the three files/tables were merged through a join operation on the key of job_link. In the merged dataset, the columns are job_link, job_skills, job_summary, job_title, company, city, state and first_seen.

In the consolidated table, the job_skills column was further splitted. The resultant file of *job_skills_detail.csv* consisted of job_link and job_skills_new columns. Exploratory Data Analysis (EDA) showed that the dataset had 26,924,758 rows and 273,9208 distinct job skills. Initially, the NLP stemming method by NLTK package was employed to refine similar job skills, aiming to reduce redundancy. Despite an 8.62% decrease in the number of distinct job skills post-stemming, the method did not perform as efficiently as expected. Due to its limited effectiveness and complexities introduced to the future LLM utilization, NLP stemming was not considered. The unstemmed analysis revealed that the top 3 job skills were “communication”, “custom service” and “teamwork”. In addition, the top 2000 job skills accounted for 49.75% of the total, with an average of 21 job skills per job_link. These top 2000 job skills will serve as the foundation for future job search studies.

To facilitate the front-end search input, the columns of state and job_title underwent preprocessing. EDA displayed that 81.87% of the dataset comprised entries with 50 US state abbreviations while the top 100 job titles represented 149,990 data points. Subsequently, the data was filtered and stored as data_cleaning_title_state.csv to serve as a dropdown list for search functionality.

<u>Utilization of LLMs</u>

LLM of OpenAI "GPT-3.5-turbo-instruct" serves as a crucial component for extracting explicit job skills and facilitating skill matching. Due to the maximum input token limitation of 4096, we utilized the LLM to categorize top 2000 job skills into approximately 423 distinct skill categories. Subsequently, we established a mapping between the 2000 job skills and categorized skills by calling the LLM API, then saved that information into a *job_skills_cat* table linked to the *job_skills_detail* table within the SQLite database.

For the job search, the frontend enabled the input of skill, state and job_title. Leveraging OpenAI "GPT-3.5-turbo-instruct" LLM, which efficiently extracted pertinent information from free input skill and assigned them to the corresponding skill categories. Following this extraction process, the SQLite database facilitated a search for posted jobs that closely aligned with the skills extracted by LLM and information of state and job_title from the front end. The retrieved job information is then stored in a CSV file to enable Tableau Visualization for further analysis and insights.

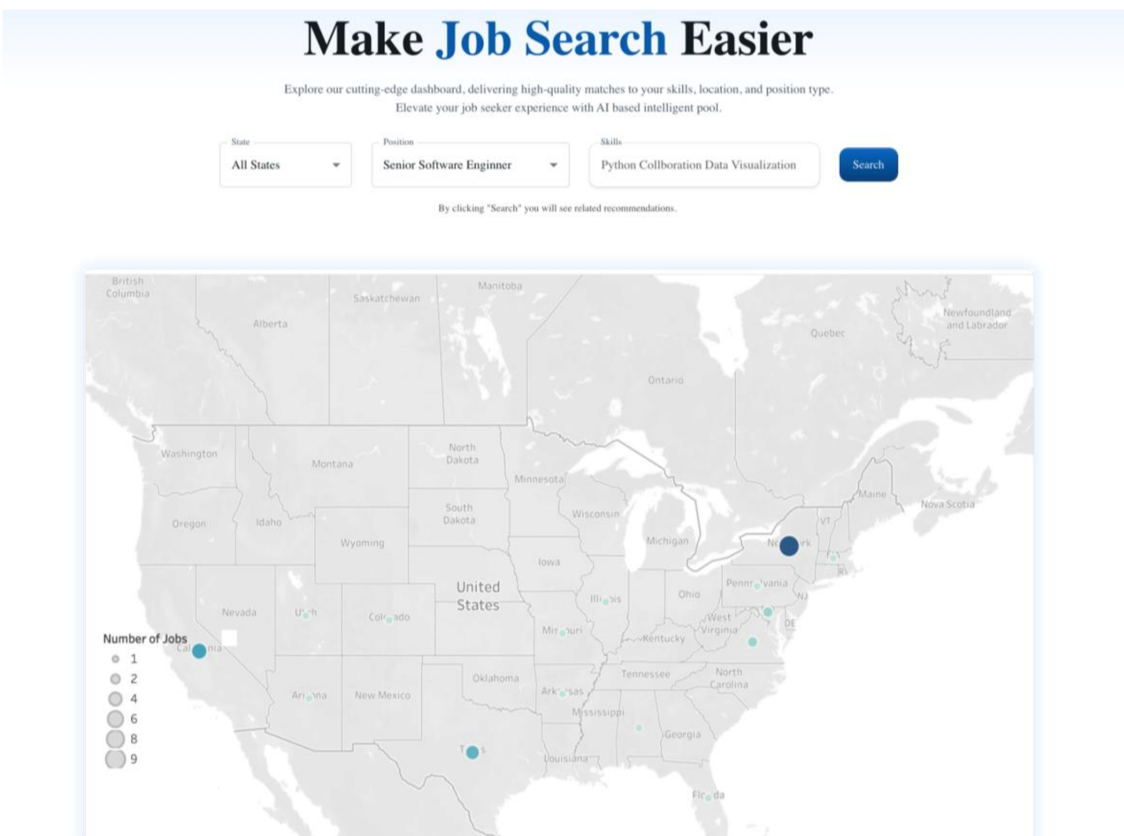
<u>Tableau Visualization for Job Recommendation</u>

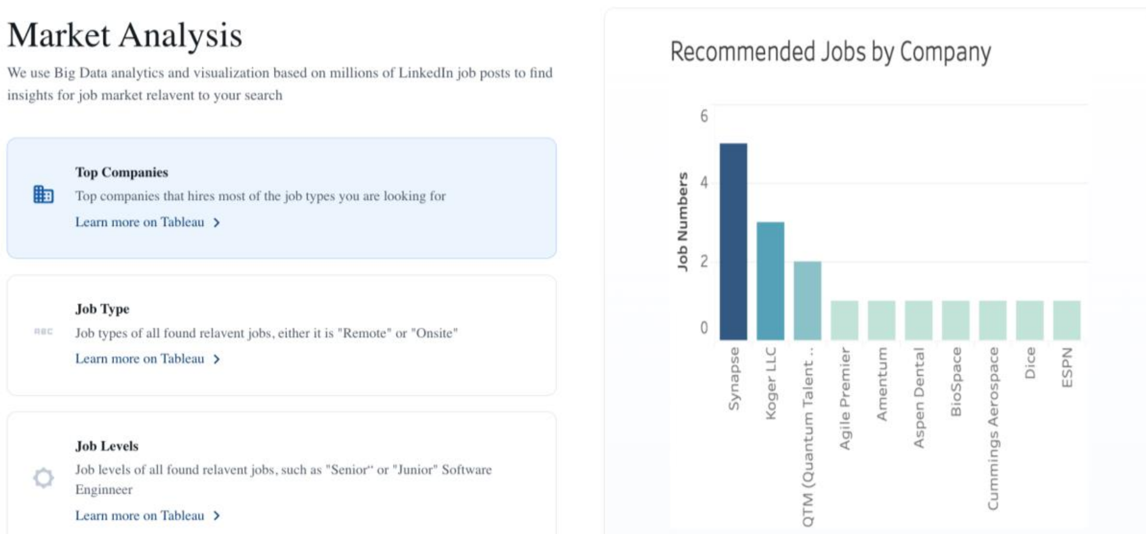
We utilized a Tableau student license to access Tableau Desktop, which served as the primary tool for developing interactive dashboards. While our preference would have been to utilize Tableau Server for real-time data refreshes, given budget constraints, we opted to prioritize cost-effectiveness and developed a MVP solution that would not incur any cost. Our approach involved creating a template dashboard equipped with multiple pre-made views and clear instructions to facilitate users to download the template dashboard and link it to the output data to refresh the views.

The template dashboard offers a range of interactive features, including filtering options by Job Type, Job Level, Company, State, and the ability to adjust the number of displayed jobs. Hovering the mouse over a specific job triggers a tooltip displaying its associated job description, enhancing user engagement and ensuring ease of use.

Additionally, we incorporated multiple bar charts to illustrate the distribution of job counts by company, job type, and job level, providing users with valuable insights at a glance. Furthermore, a map view was integrated to visually represent the geographical distribution of jobs, with interactive functionality allowing users to click on a state for more detailed information. This detailed view includes available roles within the selected state and job counts by companies operating in that region.

Leveraging the above four innovative components, our representative recommendation system is shown below.





Experiments/ Evaluation

- **Testbed and experiment questions**

The evaluation of our job recommendation system was conducted using a subset of data from LinkedIn's extensive repository of 1.3 million job listings and skills from 2024. We employed a customized user testing and evaluation method, case studies, and an online survey to assess the system's performance and user satisfaction. The questions we attempted to answer include:

1. How effectively does the recommendation system perform in suggesting relevant job postings within familiar job domains, such as data science, data architecture, or data engineering?
2. What is the precision level of the recommendation system, as determined through manual examination and assessment of recommended job postings within the chosen job domains?
3. How does the system's accuracy compare with job data obtained from the report by the U.S. Bureau of Labor Statistics, particularly in terms of identifying top job titles and employment opportunities across selected states?
4. To what extent do the recommendations generated by the system align with national trends, as indicated by the comparison with the Bureau of Labor Statistics reports?
5. Are users able to easily navigate and utilize the system to find job recommendations that match their skills and requirements?

- **Experiments and observations**

Customized Evaluation: In our evaluation process, we initially focused on testing the recommendation system within familiar job domains, specifically targeting roles such as data scientist, data architect, and data engineer. By inputting relevant job skills and specifying desired states, we generated a curated list of top job postings recommended to the user. The tested keywords include: “data management”, “data analytics”, “data architecture”, “data modeling” and more.

Given the constraints of our project timeline, we have leveraged our expertise in these domains to efficiently assess the relevance and correctness of the results. This approach allowed us to swiftly identify any potential issues or discrepancies and iterate on system improvements early in the evaluation process. Our analysis revealed that the system achieved an impressive average precision rate of about 82% on these keywords, accurately matching user-provided data with relevant job postings within these domains. The evaluation also revealed no significant process breakdowns, demonstrating the robustness and reliability of the system in performing its intended functionality.

Case Study: Except for testing on accuracy of the results, we had a case study and compared it with table E of the report “States with statistically significant employment changes from February 2023 to February 2024” [21] from the U.S. Bureau of Labor Statistics. We have located the top 20 states that have the most employment opportunities for selected job titles and matched against the top 20 states listed in table E. The job titles we have chosen are the ones that are most common in the dataset. 14 out of 20 states in our result align with the findings presented in the report. Moreover, the top 4 states are the same as the top 4 states in the report.
One of our visualization features is to show the recommended jobs’ locations on the map. Thus the accuracy of the results on the map is very important to our job recommendation system. The precise mapping greatly enhances the user experiences and helps them to make informed decisions.

Online Survey: We also incorporated an online survey to gather direct feedback from users who interacted with our job recommendation system. The survey, which garnered participation from over 20 users (mostly friends and family members), was designed to focus on assessing user experience and the quality of recommendations provided by the system. Preliminary analysis indicates a high level of user satisfaction, with 85% of respondents rating the system's ease of use as excellent or very good. Additionally, 90% of participants expressed that the recommended jobs were closely aligned with their skills and requirements, highlighting the system's effectiveness in meeting user needs.

Conclusions and Discussion

Overall, our job recommendation system aims to help users find their dream job with greater efficiency and accuracy. The recommendation system leverages cutting-edge technologies, such as React, Flask, SQLite, LLM of ChatGPT and Tableau, alongside LinkedIn’s extensive dataset of 1.3 million job listings and skills from 2024. Through experiments including customized evaluation, case study and online survey, our system enhanced user interaction, notably with features like a map visualization for job post localization. Moving forward, we are committed to address the limitations and future extensions , including accommodating complex textual input and retrieval (such as resume or job summary), improving search accuracy, enabling real-time visualization, and scaling GPT API to eliminate the necessity for categorization.

Team members meet weekly online to discuss project progress and actively collaborate and share codings using GitHub platform [19]. Every team member has participated in the project report writing, recommendation system design and presentation. Specifically, Min is the team contact person, and has performed the data cleaning and data processing. Liqing has researched LLM Solutions, categorized job skills using LLM and tested search ability with LLM. Shunyu initialized GitHub for collaboration, and implemented fronted and service API. Yiwen has made the proposal and poster presentations, and performed data processing. Bohan has researched Tableau solutions, developed Tableau templates and tested Tableau flow. Thus, all team members have contributed a similar amount of effort.

References

[1] Al-Otaibi, S.T., & Ykhlef, M. (2012). Job Recommendation Systems for Enhancing E-recruitment Process. https://worldcomp-proceedings.com/proc/p2012/IKE7862.pdf

[2] Z. Siting, H. Wenxing, Z. Ning and Y. Fan, "Job recommender systems: A survey," *2012 7th International Conference on Computer Science & Education (ICCSE)*, Melbourne, VIC, Australia, 2012, pp. 920-924, doi: 10.1109/ICCSE.2012.6295216.
https://ieeexplore.ieee.org/document/6295216

[3] Zhao, J., Wang, J., Sigdel, M., Zhang, B., Hoang, P., Liu, M., & Korayem, M. (2021). Embedding-based Recommender System for Job to Candidate Matching on Scale. arXiv.Org. https://doi.org/10.48550/arxiv.2107.00221

[4] N. D. Almalis, G. A. Tsihrintzis, N. Karagiannis and A. D. Strati, "FoDRA — A new content-based job recommendation algorithm for job seeking and recruiting," 2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA), Corfu, Greece, 2015, pp. 1-7, doi: 10.1109/IISA.2015.7388018.
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7388018

[5] W. Shalaby et al., "Help me find a job: A graph-based approach for job recommendation at scale," 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 2017, pp. 1544-1553, doi: 10.1109/BigData.2017.8258088.
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8258088

[6] Peng Yi, C. Yang, Chen Li and Y. Zhang, "A job recommendation method optimized by position descriptions and resume information," 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, China, 2016, pp. 761-764, doi: 10.1109/IMCEC.2016.7867312.
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7867312

[7] S. K. A. S, S. S, A. S. M and S. K. S, "Machine Learning based Ideal Job Role Fit and Career Recommendation System," *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 2023, pp. 64-67, doi: 10.1109/ICCMC56507.2023.10084315.
https://ieeexplore.ieee.org/document/10084315

[8] Lu, Y., El Helou, S., & Gillet, D. (2013). A recommender system for job seeking and recruiting website. Proceedings of the 22nd International Conference on World Wide Web, 963–966. https://doi.org/10.1145/2487788.2488092

[9] Kumalasari, L. D., & Susanto, A. (2020). Recommendation System of Information Technology Jobs using Collaborative Filtering Method Based on LinkedIn Skills Endorsement. Sisforma, 6(2), 63–72. https://doi.org/10.24167/sisforma.v6i2.2240

[10] T. V. Yadalam, V. M. Gowda, V. S. Kumar, D. Girish and N. M., "Career Recommendation Systems using Content based Filtering," 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 660-665, doi:10.1109/ICCES48766.2020.9137992.

[11] Karakolis, E.; Kapsalis, P.; Skalidakis, S.; Kontzinos, C.; Kokkinakos, P.; Markaki, O.; Askounis, D. Bridging the Gap between Technological Education and Job Market Requirements through Data Analytics and Decision Support Services. Appl. Sci. 2022, 12, 7139. https://doi.org/10.3390/app12147139

[12] S. Gottipati, K. J. Shim and S. Sahoo, "Glassdoor Job Description Analytics – Analyzing Data Science Professional Roles and Skills," 2021 IEEE Global Engineering Education Conference (EDUCON), Vienna, Austria, 2021, pp. 1329-1336,
doi: 10.1109/EDUCON46332.2021.9453931.

[13] i, N., Kang, B., Bie, T.D.: Skillgpt: a restful api service for skill extraction and standardization using a large language model (2023). https://doi.org/10.48550/arXiv.2304.11060

[14] Bhola, A., Halder, K., Prasad, A., Kan, M.Y.: Retrieving skills from job descriptions: A language model based extreme multi-label classification framework. Proceedings of the 28th International Conference on Computational Linguistics. pp. 5832–5842 (2020). https://aclanthology.org/2020.coling-main.513.pdf

[15] Wang Wei, Payam M. Barnaghi, Andrzej Bargiela. “Search with Meanings: An Overview of Semantic Search Systems”. http://www.bargiela.com/papers/a35.pdf

[16] Vlad-Andrei Zamfir, Mihai Carabas, Costin Carabas, Nicolae Tapus. “Systems monitoring and big data analysis using the Elasticsearch system” 2019 22nd International Conference on Control Systems and Computer Science(CSCS). https://ieeexplore.ieee.org/abstract/document/8745151

[17] Oleksii Kononenko, Olga Baysal, Reid Holmes, and Michael W. Godfrey. Mining modern repositories with elasticsearch. Proceedings of the 11th Working Conference on Mining Software RepositoriesMay 2014 Pages 328–331. https://doi.org/10.1145/2597073.2597091

[18] D. Minnie and S. Srinivasan, "Intelligent Search Engine algorithms on indexing and searching of text documents using text representation," *2011 International Conference on Recent Trends in Information Systems*, Kolkata, India, 2011, pp. 121-125, doi: 10.1109/ReTIS.2011.6146852. https://ieeexplore.ieee.org/document/6146852

[19] https://github.gatech.edu/swang3001/InteractivePersonalizedJobRecommender

[20] https://www.kaggle.com/datasets/asaniczka/1-3m-linkedin-jobs-and-skills-2024

[21] Bureau of Labor Statistics, U.S. Department of Labor, *State Employment and Unemployment Summary*, “Table E. States with statistically significant employment changes from February 2023 to February 2024, seasonally adjusted”. Retrieved April 17, 2024, from https://www.bls.gov/news.release/laus.nr0.htm

README.txt - a concise, short README.txt file, corresponding to the "user guide".

DESCRIPTION - Describe the package in a few paragraphs

INSTALLATION - How to install and setup your code

EXECUTION - How to run a demo on your code

CODE - All your code should be added here. Make sure that your package includes only the absolutely necessary set of files.

Submits one zip file called teamXXXfinal.zip 10%

- **README.txt - a concise, short README.txt file, corresponding to the "user guide". This file should contain:**
    - **DESCRIPTION - Describe the package in a few paragraphs**
    - **INSTALLATION - How to install and setup your code**
    - **EXECUTION - How to run a demo on your code**
- **DOC - a folder called DOC (short for "documentation") containing:**
    - **teamXXXreport.pdf - Your report writeup in PDF format; can be created using any software, e.g., latex, Word.**
    - **teamXXXposter.pdf - Your final poster.**
- **CODE - All your code should be added here. Make sure that your package includes only the absolutely necessary set of files.**

CSE 6242 Progress Report

Team #052: Min Cai, Yiwen Wang, Bohan Ye, Shunyu Wang, Liqing Jing

Introduction

In the digital age, the emergence of digital technology and the widespread availability of online platforms have introduced both challenges and opportunities for job seekers. As upcoming job seekers, we are motivated by the desire to create a tool that can streamline the job search process, hoping it would be useful for our own job hunting. Hence, we strived to develop an interactive job recommendation system that leverages the power of analytics, machine learning algorithms, and interactive interfaces to deliver personalized job recommendations.

Problem Definition

Given the over-abundance of job listings, sifting through countless postings to uncover positions that align with one's skill set is often a disheartening endeavor. Despite the presence of existing job recommendation systems, which do provide personalized recommendations, they frequently overlook the significance of data visualization in effectively conveying recommendation results. Thus, the problem we try to solve is to develop an end-to-end solution that seamlessly integrates the capabilities of a recommendation system with interactive data visualization, empowering job seekers to efficiently find their dream jobs.

Literature Survey

The team surveyed and synthesized findings from 18 papers across the following fields to understand the current job recommendation practices and associated limitations.

Domain-Specific Recommendation Algorithms: Alotaibi [1] and Siting et al. [2] produced overviews of the domain and explored recommender systems' role in recruitment. Zhao et al. [3] proposed a two-staged embedding-based system with retrieval and rerank stages. The system uses data from CareerBuilder.com and offers insights into evaluation methodologies applicable to our project. Almalis et al. [4] introduced FoDRA, a novel content-based algorithm. While offering valuable insights to our project's model refinement, it overlooks implicit semantic relations. Shalaby et al. [5] introduced a graph-based job recommendation approach for online platforms, overcoming scalability and relevance challenges. Despite its value for our algorithm, enhancements are needed in parameter settings, language support, and behavioral pattern integration. Yi et al. [6] proposed an item-based collaborative filtering algorithm integrating position descriptions and resumes for an improved accuracy, though it lacks the latest collaborative filtering advancements. S. K. A. S et al. [7] introduced SVD methods for ranking of similar documents for job recommendation, though concerns existed for computational resources. Lu et al. [8] introduced a hybrid recommender utilizing content-based and interaction-based relations, providing insights into recommendation cases useful for our project.

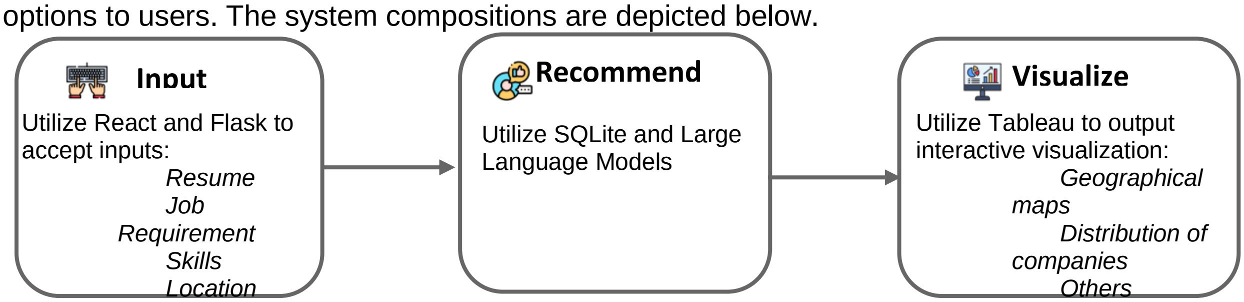
Kumalasari and others [9] presented an IT job recommendation system based on collaborative filtering and LinkedIn skills endorsement data, offering insights applicable to our project. Yadalam et al. [10] developed an ML-based system to assist new graduates in finding job opportunities, while highlighting challenges like cold start and scalability. Aligning with our project's goals, pre-registration user preference collection could enhance our system's usability.

Data Extraction & Search Frameworks: Karakolis [11] and Gottipati [12] et al. applied NLP to extract skills from job postings, providing insights to our project. However, limited scope of research and static presentation call for a larger dataset and interactive dashboard in our implementation. For more advanced implementations, Kang et al. [13] developed SkillGPT based on LLM for extracting and standardizing job skills, and Bhola et al. [14] predicted required skills from job descriptions using Extreme Multi-label Classification. Improving recognition of implicit skills is crucial for accuracy while addressing randomness is crucial for system reliability.

On the topic of search systems, Wang et al. [15] proposed a framework for semantic search. Zamfir et al. [16] used Elasticsearch for efficient system monitoring and big data analysis. Their work is relevant to our project of processing and analyzing large amounts of user data related to job skills. Kononenko et al. [17] highlighted capabilities of Elasticsearch in software analytics and real-time search, which can enhance the accuracy and relevance of our recommendation system. Minnie and Srinivasan [18] presented text-based search engine algorithms, with room for improvement in handling multiple keywords and phrases. All of the papers are relevant to developing the search functionalities of our system. Leveraging LLMs could enhance semantic understanding and recommendation accuracy, while optimization strategies will be essential for efficient search functionality.

Proposed method

The project aims to develop an interactive job recommendation system. This system will accept user-provided data, generate a curated list of relevant jobs, and provide interactive visualization options to users. The system compositions are depicted below.



<u>React and Flask Framework Integration:</u> We will create a scalable and responsive platform for seamless data handling and user interaction.

The presentation layer uses React to build the web application and uses Flask to create a front-end service which serves the requested data to fulfill the visualizations. We have already built the full stack and run it locally with examples to prove the feasibility. In the frontend, we selected Google material design as styling library and its relevant component library so as to build elegant UI controls. Furthermore, we started with an open-source UI template whose design was very close to our presentation goals. We built the basic form of controls, such as inputs and dropdowns to input search words or select filters. The web application is running at *localhost:3000* upon start. The backend is a Flask server primarily with different APIs and set up necessary libraries in Pipfile. The python libraries are SQLite3, PySpark, and Flask. The root URL of the service is *127.0.0.1:5000/api*, and we will add necessary APIs as the project moves forward. Each part has the README file that helps users to set up the environment and run successfully at the local environment. We also used Git as a version control tool and collaborated on Github [19] for source management.

<u>Data Source, Data Processing and SQLite Database Buildup:</u> We will enable advanced search capabilities, querying features to streamline job search processes and improve user satisfaction.

To enhance project innovations, we leverage insights from existing literature and LinkedIn's comprehensive data source of 1.3 million job listings in 2024 [20]. The data source consists of three files: *linkedin_job_postings.csv*, *job_skills.csv* and *job_summary.csv*. Data cleaning and processing have already been performed using Python Pandas and SQLite packages. For example, rows with missing values were deleted and unimportant columns were filtered. In *linkedin_job_postings.csv*, the job_location column was split into 2 columns of city and state for future use when creating the interactive dashboard. In *job_skills.csv* and *job_summary.csv*, text was standardized by converting to lowercase, removing special characters and handling abbreviations. The 3 files/tables were joined on the key of job_link and the columns were filtered as job_link, job_skills, job_summary, job_title, company, city, state and first_seen.

In the consolidated table, the job_skills column was splitted. The new table consisted of job_link and job_skills_new columns. From the Exploratory Data Analysis (EDA), the dataset has 26,924,758 rows and 273,9208 extinct job skills. Then, the NLP stemming method by NLTK package was used to filter the similar job skills. Stemming is a method to reduce words to their basic form or stem. After stemming, the number of extinct job skills reduced to 250,3112. The top 3 job skills are 'communic', 'custom servic' and 'teamwork'. In addition, the top 2000 job skills consist of 51.47% of total job skills and each job_link has an average of 21 job skills. The top 2000 job skills will be used for the future job search study.

Utilization of LLMs: We will employ LLMs such as ChatGPT to extract explicit job skills and uncover implicit requirements with high skill matching and recommendation accuracies.

LLM of OpenAI "GPT-3.5-turbo" serves as a crucial component for extracting explicit job skills and facilitating skill matching. Due to the maximum input token limitation of 4096, we utilized the LLM to categorize top 2000 job skills into approximately 200 distinct skill categories. Subsequently, we will establish a mapping between the 2000 job skills and categorized skills by calling the LLM API, then save that information into a skill_link table within the SQLite database.

For the future job search, the frontend enables the input of skill, resumes or job requirements. Leveraging OpenAI "GPT-3.5-turbo" LLM, which efficiently extracts pertinent information from resumes or job requirements and assigns them to the corresponding skill categories. Following this extraction process, the SQLite database facilitates a search for posted jobs that closely align with the skills extracted by LLM. The retrieved job information is then stored in a CSV file to enable Tableau Visualization for further analysis and insights.

Tableau Visualization for Job Recommendation: Utilizing Tableau, we will create interactive visualizations that empower users to explore and understand job trends effortlessly.

We leveraged the Tableau student license to access Tableau Desktop, which would be the primary tool to develop interactive dashboards. While, if given sufficient budget, we would elect to use Tableau Server to enable real-time data refreshes in an ideal scenario, we decided to create a MVP solution instead that would not incur any cost. We will develop a template dashboard that includes multiple pre-made views: a list of top matched jobs returned from the database, a list of top companies hiring for the aforementioned jobs, and a map view showcasing the states in which these jobs are located with interactive components enabling users to click on a state and see details.

We would add instructions guiding users to download the template dashboard and the CSV file storing the data extracted from the previous step, link the dashboard to the output data, refresh the views, and publish it to Tableau Public to be able to explore the visualization.

Experiments/ Evaluation

The experiments and evaluation for our job recommendation system will use a subset of data from 1.3M Linkedin Jobs & Skills (2024). Focusing on the subset of data will allow us to access the results faster, thus improving the recommendation system more efficiently. We will be able to check the precision through manually examining our results, case study and online survey.

Customized Evaluation: First of all, we are going to test the recommendation system using the job domain that we are familiar with, for example, data scientist, data architect or data engineer. By putting in job skills in this field and the states needed, we will get a list of top job postings recommended to the user. Because of the limited time for this project, we can use our expertise to assess the relevance and correctness of the results more efficiently. In the early stage of the evaluation, it will also let us find the problem quickly and refine the system.

Case Study: Except for testing on accuracy of the results, we will have two case studies and compare them with two reports from the U.S. Bureau of Labor Statistics. For selected states, we will discover the top 10 job titles with the highest number of job listings. The analysis will be done by comparing the results with "Top 10 fastest growing occupations from U.S. Bureau of Labor Statistics". Moreover, we will locate the top 5 states that have most employment opportunities for selected job titles and match against the states listed in table E of the report "States with statistically significant employment changes from February 2023 to February 2024". These can help with validating the results of our job recommendation system as well as checking if the recommendations match national trends.

Online Survey: We will also have an online survey to gather user's feedback directly from using our job recommendation system. Our survey will focus on user experience towards the system and the quality of the recommendations. For example, questions regarding if the system is easy to use and if the recommended jobs align with the user's skills and requirements. More details and results of this survey will be shared in the final report.

Conclusions and Discussion

Overall, our job recommendation system aims to help users find their dream job with greater efficiency and accuracy. Our system has enhanced user interaction. The map with the final list of job posts will better help the users to visually locate the job opportunities. More details about the results will be discussed in the final report.

Team members meet weekly online to discuss project progress and actively collaborate and share codings using GitHub platform [19]. Project activities that every team member has participated in are listed below in the old and revised plans. All team members have contributed a similar amount of effort.

Old Plan	**Project Activity**	**Start**	**End**	**Due**	**Member**
	Proposal Writing	Feb 12	Feb 27	Mar 1	All
	Proposal Video	Feb 27	Feb 29	Mar 1	All
	Milestone 1 - MVP	Mar 3	Mar 22	Mar 24	All
	Milestone 2 - Goal	Mar 25	Apr 7	Apr 8	All
	Final Report	Apr 8	Apr 18	Apr 19	All

Revised Plan:

Member	**Project Activity**	**Start**	**End**
All	Proposal Writing and presentation	Feb. 12	Feb. 27
	Progress report	Mar. 11	Mar. 29
	Project and recommendation system design	Feb. 12	Apr. 18
Min Cai	1. Team contact person and proposal proofread	Feb. 12	Feb. 27
	2. Data cleaning and processing	Mar. 18	Mar. 29
Liqing Jing	1. Research LLM Solutions	Mar. 29	Apr. 05
	2. Categorize job skills using LLM	Apr. 05	Apr. 18
	3. Test search ability with LLM	Apr. 05	Apr. 18
Shunyu Wang	1. Initialize Github for collaboration	Mar. 18	Mar 20
	2. Implement frontend	Mar. 18	Apr. 16
	3. Implement Service API	Mar. 18	Apr. 16
Yiwen Wang	1. Proposal presentation	Feb. 26	Feb. 29
	2. Data cleaning and processing	Mar. 18	Mar. 29
Bohan Ye	1. Research Tableau Solution	Mar. 18	Mar. 29
	2. Develop Tableau Template	Mar. 29	Apr. 12
	3. Test Tableau Flow	Apr. 12	Apr. 18

References

[1] Al-Otaibi, S.T., & Ykhlef, M. (2012). Job Recommendation Systems for Enhancing E-recruitment Process. https://worldcomp-proceedings.com/proc/p2012/IKE7862.pdf

[2] Z. Siting, H. Wenxing, Z. Ning and Y. Fan, "Job recommender systems: A survey," *2012 7th International Conference on Computer Science & Education (ICCSE)*, Melbourne, VIC, Australia, 2012, pp. 920-924, doi: 10.1109/ICCSE.2012.6295216. https://ieeexplore.ieee.org/document/6295216

[3] Zhao, J., Wang, J., Sigdel, M., Zhang, B., Hoang, P., Liu, M., & Korayem, M. (2021). Embedding-based Recommender System for Job to Candidate Matching on Scale. arXiv.Org. https://doi.org/10.48550/arxiv.2107.00221

[4] N. D. Almalis, G. A. Tsihrintzis, N. Karagiannis and A. D. Strati, "FoDRA — A new content-based job recommendation algorithm for job seeking and recruiting," 2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA), Corfu, Greece, 2015, pp. 1-7, doi: 10.1109/IISA.2015.7388018. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7388018

[5] W. Shalaby et al., "Help me find a job: A graph-based approach for job recommendation at scale," 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 2017, pp. 1544-1553, doi: 10.1109/BigData.2017.8258088. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8258088

[6] Peng Yi, C. Yang, Chen Li and Y. Zhang, "A job recommendation method optimized by position descriptions and resume information," 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, China, 2016, pp. 761-764, doi: 10.1109/IMCEC.2016.7867312. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7867312

[7] S. K. A. S, S. S, A. S. M and S. K. S, "Machine Learning based Ideal Job Role Fit and Career Recommendation System," *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 2023, pp. 64-67, doi: 10.1109/ICCMC56507.2023.10084315. https://ieeexplore.ieee.org/document/10084315

[8] Lu, Y., El Helou, S., & Gillet, D. (2013). A recommender system for job seeking and recruiting website. Proceedings of the 22nd International Conference on World Wide Web, 963–966. https://doi.org/10.1145/2487788.2488092

[9] Kumalasari, L. D., & Susanto, A. (2020). Recommendation System of Information Technology Jobs using Collaborative Filtering Method Based on LinkedIn Skills Endorsement. Sisforma, 6(2), 63–72. https://doi.org/10.24167/sisforma.v6i2.2240

[10] T. V. Yadalam, V. M. Gowda, V. S. Kumar, D. Girish and N. M., "Career Recommendation Systems using Content based Filtering," 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 660-665, doi:10.1109/ICCES48766.2020.9137992.

[11] Karakolis, E.; Kapsalis, P.; Skalidakis, S.; Kontzinos, C.; Kokkinakos, P.; Markaki, O.; Askounis, D. Bridging the Gap between Technological Education and Job Market Requirements through Data Analytics and Decision Support Services. Appl. Sci. 2022, 12, 7139. https://doi.org/10.3390/app12147139

[12] S. Gottipati, K. J. Shim and S. Sahoo, "Glassdoor Job Description Analytics – Analyzing Data Science Professional Roles and Skills," 2021 IEEE Global Engineering Education Conference (EDUCON), Vienna, Austria, 2021, pp. 1329-1336,
doi: 10.1109/EDUCON46332.2021.9453931.

[13] i, N., Kang, B., Bie, T.D.: Skillgpt: a restful api service for skill extraction and standardization using a large language model (2023). https://doi.org/10.48550/arXiv.2304.11060

[14] Bhola, A., Halder, K., Prasad, A., Kan, M.Y.: Retrieving skills from job descriptions: A language model based extreme multi-label classification framework. Proceedings of the 28th International Conference on Computational Linguistics. pp. 5832–5842 (2020). https://aclanthology.org/2020.coling-main.513.pdf

[15] Wang Wei, Payam M. Barnaghi, Andrzej Bargiela. "Search with Meanings: An Overview of Semantic Search Systems". http://www.bargiela.com/papers/a35.pdf

[16] Vlad-Andrei Zamfir, Mihai Carabas, Costin Carabas, Nicolae Tapus. "Systems monitoring and big data analysis using the Elasticsearch system" 2019 22nd International Conference on Control Systems and Computer Science(CSCS). https://ieeexplore.ieee.org/abstract/document/8745151

[17] Oleksii Kononenko, Olga Baysal, Reid Holmes, and Michael W. Godfrey. Mining modern repositories with elasticsearch. Proceedings of the 11th Working Conference on Mining Software RepositoriesMay 2014 Pages 328–331. https://doi.org/10.1145/2597073.2597091

[18] D. Minnie and S. Srinivasan, "Intelligent Search Engine algorithms on indexing and searching of text documents using text representation," *2011 International Conference on Recent Trends in Information Systems*, Kolkata, India, 2011, pp. 121-125, doi: 10.1109/ReTIS.2011.6146852. https://ieeexplore.ieee.org/document/6146852

[19] https://github.gatech.edu/swang3001/InteractivePersonalizedJobRecommender

[20] https://www.kaggle.com/datasets/asaniczka/1-3m-linkedin-jobs-and-skills-2024

1. Introduction — motivation
2. Problem definition — provide a precise formal problem definition, in addition to the jargon-free version (for Heilmeier question #1).
3. Literature survey (example from ed)
4. Proposed method
    a. Intuition — why should it be better than the state of the art?
    b. Detailed description of your approaches: algorithms, user interfaces, etc.
5. Experiments/ Evaluation
    a. Description of your testbed; list of questions your experiments are designed to answer
    b. Detailed description of the experiments; observations (as many as you can!)
6. Conclusions and discussion
    a. Depending on the topic of your project, this section may clearly summarize the key points of the project, such as the main ideas, results, impacts, significances, etc., and discuss issues such as limitations, implications, potential future extensions, etc.
    b. Provide a statement that summarizes the distribution of team members' effort. The summary statement can be as simple as "all team members have contributed a similar amount of effort". If effort distribution is too uneven, we may assign higher scores to members who have contributed more.

Data cleaning - Min Cai, Yiwen Wang (6242 group project)
Front end UI design - Jack
LLM - Liqing Jing
Tableau - Bohan Ye

Recommendation implementation

[70%] Proposed method - 1.5 Pages
Provide a clear list of innovations: give a list of the best 2-4 ideas that your approach exhibits.
This section should be almost finished. That is, we expect your team to have made substantial progress in the design and implementation of BOTH the computation and visualization components.

Intuition — why should it be better than the state of the art?
Detailed description of your approaches: algorithms, user interfaces, etc.

- What are they?
- How do they work?
- Why do you think they can effectively solve your problem (i.e., what is the intuition behind your approaches)?
- What is new in your approaches?
- How did you get the data?
- What are its characteristics?

[5%] Conclusions and discussion - ⅓ Page

Plan of activities (same requirements as described in “Proposal” section above)
Provide the old plan and the revised plan
[To be filled]

Depending on the topic of your project, this section may clearly summarize the key points of the project, such as the main ideas, results, impacts, significance, etc., and discuss issues such as limitations, implications, potential future extensions, etc.

[-5% if not included] Provide a statement that summarizes the distribution of team members’ effort. The summary statement can be as simple as "all team members have contributed a similar amount of effort". If effort distribution is too uneven, we may assign higher scores to members who have contributed more.