International Journal of Trend in Research and Development, Volume 11(5), ISSN: 2394-9333
www.ijtrd.com

Design and Realization of Cloud-based Job-matching Website Based on Collaborative Filtering Recommendation Algorithm

[1]Zeng Qiaozhi and [2]Zhang Hong,
[1,2]School of Electronics and Information Engineering, Liaoning University of Science and Technology, Benxi, Liaoning, China

Abstract: With the rapid development of internet technology, job search websites have become an important part of the modern recruitment market. This paper introduces a job search website called Cloud Job Reach, which is based on the Spring Cloud microservice architecture and integrates instant messaging, video interviews, personalized recommendations, and AI assistants. The goal is to improve the efficiency of interaction between job seekers and recruiters and enhance the precision of matching. The paper focuses on the design of personalized job recommendations based on the user collaborative filtering recommendation algorithm (UBCF), which recommends jobs by analyzing the similarity of user behavior. The system is developed using Java, with MySQL and Redis as the database, and the front-end uses the Vue framework. The system demonstrates good extensibility and stability through the use of Spring Cloud Alibaba's Nacos and Spring Gateway. Cloud Job Reach also integrates AI technology to improve the website's intelligence level. The paper describes the system design, development environment, technical architecture, database design, function implementation, and system testing in detail, verifying its effectiveness in terms of functionality, performance, and user experience.

Keywords: *Microservice Architecture; Job Search Website; Instant Messaging; Video Interviews; Collaborative Filtering Recommendation Algorithm; Artificial Intelligence Assistant.*

I. INTRODUCTION

A. Research Background and Significance

In the digital age, job search websites have become increasingly important as bridges connecting job seekers with employers. However, existing job platforms face issues such as service fees, advertisement interference, and privacy leaks, necessitating the development of a free, efficient, and secure job platform. The design and implementation of Cloud Jobs job search website aims to address these issues and provide a new online job solution. In particular, this study introduces a collaborative filtering recommendation algorithm to enhance the accuracy and personalization level of job recommendations.

B. Research Status at Home and Abroad

Research on job search websites at home and abroad generally focuses on key areas such as platform functions, service quality, and user interaction experience. In the study of recommendation systems, collaborative filtering algorithms have gained popularity for their excellent recommendation effects.

Job Seeker End: With the continuous growth of college graduates in China, the total number of graduates in 2024 has exceeded 11.79 million, making the job market more competitive. This generation of graduates is accustomed to instant messaging and prefers to use vertical recruitment platforms for job hunting. A survey shows that about 64% of recent graduates recognize the value of professional job counseling and are willing to seek support for it.

Employer End: Related surveys show that more than half of the companies can complete their recruitment tasks within the stipulated time, but about one-third of them face significant challenges in recruiting senior and technical personnel.

This study builds on existing research results and considers various factors, introducing a user-based collaborative filtering recommendation algorithm and combining it with microservice architecture. This innovation not only enhances the technical advancedness of the job search website but also significantly improves the user experience.

II. SYSTEM DESIGN AND IMPLEMENTATION

A. System Design Goals

The design and implementation of the Cloudzhi Career Job Search Website aims to build a highly scalable, secure and user-friendly online recruitment platform. The specific design goals of this system include the following aspects:

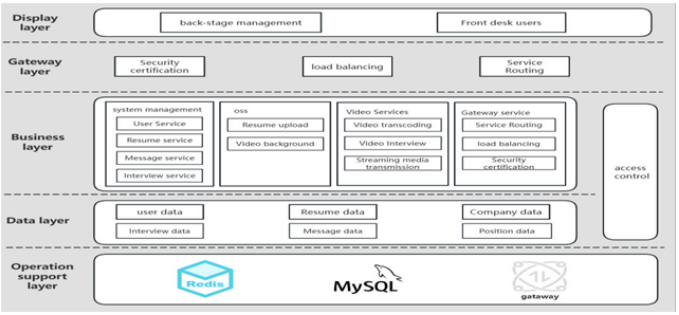
- Efficiency: The system should be able to quickly respond to user requests, provide instant job search, resume submission, and communication services, ensuring that job seekers and recruiters can quickly obtain the information they need.
- User Experience: Provide intuitive and user-friendly user interfaces to ensure that users can operate easily and quickly, and improve user satisfaction through personalized services.
- Precise Recommendation: Utilize advanced collaborative filtering recommendation algorithms to analyze user behavior and preferences, and provide precise job matching and recommendations to improve job search efficiency and increase recruitment success rate.

B. System Development Environment

The system uses Java JDK 1.8 as the development language and Spring Cloud microservice architecture, coupled with Nacos as the service registration and configuration center to manage the entire microservice cluster. The front-end interface is based on the Vue.js framework, and the load balancing is implemented through NGINX. The main database for the back-end is MySQL, the cache database is Redis, and the storage is placed in the cloud-based OSS.

C. Technical Architecture

The system uses Spring Cloud Alibaba's Nacos as the service registration and discovery center, Spring Gateway as the API gateway, and Shiro for security verification. WebSocket is used to implement real-time chat and video interview functions.



D. Database Design

In the database design of Cloudzhi Career Job Search Website, we adopted the relational database MySQL as the main data storage solution, and supplemented Redis as the cache database to improve data access speed. The core goal of the database design is to ensure data integrity, consistency, and efficient access. Here are the detailed database design contents:

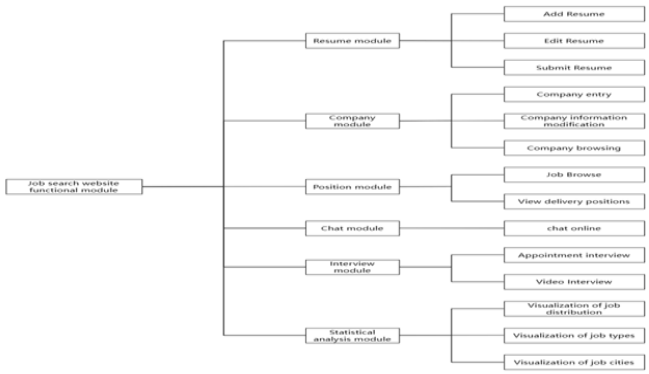
Table design: A total of 20 tables were designed in the database, covering data storage needs in various aspects such as user information, job information, company information, resume information, chat records, and interview arrangements.

1. User table (t_user): stores basic information about users, including but not limited to usernames, passwords, email addresses, and mobile numbers.
2. Job table (t_job): records detailed information about job positions, such as job title, requirements, salary range, and work location.
3. Company table (t_company): includes basic information about companies, such as company name, industry, size, and brief introduction.

In addition, there are tables supporting the recommendation system, such as the user behavior table and the rating table, as well as Redis key-value storage for caching.

- Security design: sensitive information such as user passwords is encrypted for storage, ensuring the security of user data.
- Sharding strategy: a sharding strategy is designed for tables that may generate large amounts of data, such as the user table and the job table, to avoid performance issues caused by a single table with too much data.

E. Function Realization



The system has implemented six major functional modules: user registration, resume management, job search, online chat, video interview, etc. Each module interacts with microservices to form a complete system. In particular, this paper provides a detailed description of the implementation process of the collaborative filtering recommendation algorithm based on user behavior, including the collection of user behavior data, similarity calculation, and the generation of recommendation lists.

F. Implementation of System Business Processes

The overview of the system's business processes clearly outlines the operation flow of the recruitment system. The system is set up with two roles, job seekers and recruiters, and users must log in before accessing the system. The system will then allocate corresponding rights based on the user's identity. Recruiters can manage job information, while job seekers can manage their resumes.

After browsing job postings and selecting a job of interest, job seekers can initiate contact with recruiters and send their resumes. Once recruiters receive a resume and express interest, they can schedule a video interview. Job seekers can view and participate in the video interview in their submission records. After the interview, recruiters will decide whether to approve or reject the job seeker's application based on the interview results.

The entire process provides a highly efficient and interactive recruitment platform that ensures smooth communication between job seekers and recruiters, while simplifying the process of scheduling and participating in interviews, thereby improving recruitment efficiency.

III. CO-FILTERING RECOMMENDATION ALGORITHM EXPLAINED

Co-filtering recommendation algorithm is the core mechanism for personalized job recommendation on cloud-based job search websites. This section will provide a detailed explanation of the design thinking, implementation steps, and optimization strategies of the co-filtering algorithm based on the Pearson correlation coefficient to calculate similarity.

A. Algorithm Principle

The co-filtering algorithm is based on user behavior data and predicts the preference of the target user for unknown jobs by discovering the similarity between users. In user-based co-filtering, the algorithm first identifies other users with similar behavior patterns as the target user, and then recommends jobs to the target user based on the preferences of these similar users.

B. Data Preprocessing

Before implementing co-filtering, it is necessary to preprocess user behavior data to ensure the effectiveness of the algorithm. Data preprocessing includes the following steps:

- Data cleaning: Remove anomalous values, duplicate records, and incomplete user behavior data.
- Normalization: Normalize the user's rating data to eliminate the influence of different scales.
- Behavior encoding: Convert the user's behavior (resume ratings) towards jobs into numerical data for easy calculation.

C. Similarity Measurement

Similarity measurement is the core of collaborative filtering algorithms. In this paper, the Pearson Correlation Coefficient (PCC) is used as the standard to measure the similarity between users. The PCC measures the similarity between two users by calculating the ratio of the covariance and standard deviation of their ratings for all positions. The formula is as follows:

$$\text{Pearson Correlation}(U_i, U_j) = \frac{\sum_{k=1}^{n}(r_{ik}-\bar{r_i})(r_{jk}-\bar{r_j})}{\sqrt{\sum_{k=1}^{n}(r_{ik}-\bar{r_i})^2}\cdot\sqrt{\sum_{k=1}^{n}(r_{jk}-\bar{r_j})^2}}$$

1. Collect user behavior data: First, we collect user behavior data on work and companies through the front-end rating system.
2. Build a rating matrix: Create a user-project rating matrix, where the rows of the matrix represent users, the columns represent projects, and the elements of the matrix represent the user's rating of the project. If a user has not rated or interacted with a particular project, the value at that position is 0.
3. Calculate the similarity coefficient: For target user u and other user v, we need to calculate the Pearson similarity coefficient r_{uv} between them.

This can be done by the following steps:

1) For each project i, calculate the difference in ratings between users u and v for project i, $X_{ui} - \bar{X}_u$, and$X_{vi} - \bar{X}_v$, where X_{ui} and X_{vi} are the ratings of users u and v for project i, respectively, and $\bar{X}_u$ and $\bar{X}_v$ are the average ratings of users u and v.
2) Calculate the sum of the product of the ratings difference for all projects, i.e. $\sum_{i=1}^{n} = (X_{ui} - \bar{X}_u)(X_{vi} - \bar{X}_v)$。
3) Calculate the sum of the squares of the difference between each user's rating and the average rating, i.e.$\sum_{i=1}^{n} = (X_{ui} - \bar{X}_u)^2$和 $\sum_{i=1}^{n} = (X_{vi} - \bar{X}_v)^2$。
4) Use Pearson's correlation coefficient formula to calculate the correlation coefficient between users u and v.r_{uv}.

D. Neighbor Selection

After calculating the similarity between all user pairs, a set of the most similar users is selected for each target user as neighbors. The user behaviors of these neighbors will be used to generate the target user's recommendation list. Neighbor selection can be achieved by setting a threshold or selecting the top K most similar users.

E. Recommendation List Generation

The recommendation list for the target user is generated by using the behavior data of the selected neighbor users through a weighted average method. For each position, the average rating of all neighboring positions is calculated, and this is used as the predicted rating for the target user's position. The recommendation list is sorted in descending order of the predicted rating, and the position with the highest predicted rating is selected as the recommendation.

IV. AI ASSISTANT

In our cloud-based job search platform, we have innovatively introduced an AI assistant called "Career Partner" that is customized based on the StarFire large-scale language model. This assistant is designed to provide personalized job support to job seekers. It has been specifically trained by our team and focuses on the areas of job recommendation and mock interview to meet the job-related needs of job seekers.

- Job Recommendation: Using advanced dialogue systems, Career Partner is able to provide continuous dialogue based on context, intelligently recommending suitable job opportunities based on the user's personal information and career preferences. This personalized recommendation mechanism helps job seekers find job opportunities that align with their career goals more efficiently.
- Mock Interview: For job seekers who have successfully secured interview opportunities, Career Partner provides mock interview services. By simulating actual interview scenarios, Career Partner helps job seekers prepare in advance, enhance their interview skills, and increase the probability of interview success.

The introduction of Career Partner not only improves the efficiency of job seekers but also greatly enhances the quality of job interview preparation. As a reliable assistant for job seekers on their career path, it brings value to users through the following ways:

- Personalized job matching: The AI assistant analyzes the user's resume, work experience, educational background, skills and expertise, and personal interests, and uses machine learning algorithms to accurately recommend job positions that match the user's career development.
- Career Path Planning: The AI assistant provides users with career development advice, helping them plan their future career paths.

With these features, ZhiTuoBanLv provides users with a more personalized, efficient, and secure job-seeking experience, becoming a reliable partner on their career path.

V. METHODS OF SYSTEM TESTING

The purpose of system testing is to verify the functionality, performance, and reliability of the system through a series of testing activities. By conducting system testing, we can ensure that the system meets the needs and expectations of users and improve the quality and usability of the system.

To comprehensively evaluate the key functions of the system, including but not limited to resume query, editing, verification code login, user registration, job and city search, company search, HR information management, position management, candidate management, interview appointment, online chat, video interview, and back-end management, we adopted diverse testing tools and methods. The specific methods are as follows:

- Performance testing: Using JMeter and LoadRunner pressure testing tools, simulate high concurrency scenarios to ensure the stability and response speed of the system under extreme load.
- Function testing: Using Selenium and Appium automation testing tools, perform comprehensive testing on the system's various functions, and use Mockito and other tools to simulate data and dependencies to ensure the accuracy and robustness of the system.

- PI Test: Using API testing tools such as Postman, we conduct comprehensive testing on system interfaces to ensure seamless integration with external systems and services.
- Unit and Integration Testing: Given that the system employs Spring Cloud microservice architecture and SpringBoot framework, we utilize Spring Test and Vue Test Utils testing frameworks for in-depth unit testing and integration testing to enhance test coverage and efficiency.

Through these comprehensive testing methods, we can not only ensure that every aspect of the system has been rigorously tested, but also prove the system's outstanding performance in terms of design and implementation. This not only provides confidence to the end-users, but also lays a solid foundation for continuous improvement and optimization of the system.

CONCLUSION AND PROSPECTS

This paper successfully designed and implemented a cloud-based job matching website based on the Spring Cloud microservice architecture, which integrates core functions such as real-time chat, video interview, personalized recommendation, and artificial intelligence assistant. The goal is to provide a high-efficiency, convenient, and secure online recruitment platform for job seekers and recruiters. By adopting Java JDK1.8, MySQL, Redis, and Vue.js technologies, the system has excellent performance in terms of development efficiency, data processing capabilities, and user experience.

In particular, this study focuses on the application of the User-based Collaborative Filtering (UBCF) recommendation algorithm, which analyzes user behavior data to calculate the similarity between users and recommends job positions to target users based on the preferences of similar users. Experimental results show that the UBCF algorithm significantly improves the accuracy and personalization level of job recommendations, thereby enhancing the job-seeking experience of users. The system test results show that the system has achieved its expected goals in terms of function implementation, performance, and user satisfaction.

Future work will continue to conduct in-depth research and optimization in the following areas:

Algorithm Optimization: Explore more efficient methods for calculating similarity to reduce the computational complexity of the recommendation system and improve response speed. At the same time, consider introducing deep learning technology to capture non-linear features in user behavior data and further improve the accuracy of the recommendation algorithm.

User Experience: Continuously collect user feedback and optimize the user interface design to provide a more intuitive and friendly operation flow. At the same time, increase personalization options to allow users to customize their job search experience based on their needs.

AI Assistant: Integrated with specially-tuned AI assistants to provide personalized job guidance and career development advice. Utilizing a dataset specifically tailored to job recruitment, this AI assistant is able to provide more relevant answers to user questions, as well as customized job recommendations and career path planning.

Through continuous technological innovation and service optimization, Cloud Job Reach aims to become the preferred online recruitment platform for job seekers and hiring parties, providing both sides with more efficient, intelligent, and convenient recruitment solutions.

References

[1] Lu Wei. Research on Job Seeker Needs Analysis and Redesign of Job Recruitment Website Based on Kano Model [D]. Zhejiang University of Technology, 2020.
[2] Zhu Zhihui, CaiJie. Design of Campus Lost and Found System Based on SpringBoot+Vue+Uni-app Framework [J]. Electronic Technology and Software Engineering, 2022(17): 62-65.
[3] Jing Cheng, Xie Jun. Optimization Algorithm and Realization of Load Balancing Based on Microservice Architecture [J]. Journal of Nanchang University, 2022, 46(04): 399-404+ 415. DOI: 10.137