International Journal of Research Publication and Reviews, Vol 5, no 4, pp 2353-2358 April 2024

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Enhanced Job Suggestion: Resume Ranking and Personalized Job Recommendations Using NLP and Machine Learning

Karthik R [a], Kiruthika S [b], Dharun M P[c], Hemashree V [d], Hithesh K [e]

[a] Department of Computer Science and Business Systems , Professor , Knowledge Institute of Technology ,Salem ,Tamil Nadu ,India
[b,c,d,e] Department of Computer Science and Business Systems , Student , Knowledge Institute of Technology ,Salem ,Tamil Nadu ,India

A B S T R A C T

In today's highly competitive job market, both job seekers and employers face significant challenges. For job seekers, the process of finding suitable employment opportunities can be daunting and time-consuming. They often have to sift through numerous job listings, many of which may not be relevant to their skills, experience, or career goals. This can lead to frustration and inefficiencies in the job search process.

In this context, the need for a job recommendation system becomes evident. Such a system leverages data analytics and machine learning algorithms such as NLP to match job seekers with relevant job opportunities based on their skills, experience, preferences, and other relevancy factors. By providing personalized job recommendations, these systems streamline the job search process for candidates while helping employers identify and connect with the most suitable candidates ace efficiently. This not only saves time and efforts for both job seekers and recruiters, but also improves the overall qualities of matches between job seekers and employers, leading to higher satisfactions and better outcomes for all stakeholders involved in the recruitment process.

Keywords: Machine learning algorithms, NLP, Personalized job recommendations, Efficiency, Time-saving, Preferences, Match quality

INTRODUCTION

The goal of recommender system technology is to assist users in locating products that match with their personal interests. It has been successfully applied in e-commerce applications to effectively address issues connected to information overloads. Numerous recommender system techniques have been put forth to enhance the functionalities of e-recruiting. An overview of the electronic recruiting process and current recommended strategies for creating candidate/job matching recommender systems will be providing in this article. The recommender system approaches are classified into the following main four categories: Collaborative filtrating, Contend-based filtering, Knowledge-based and Hybrid approaches. The following paragraphs gives thorough explanation of various techniques.

COLLABRATIVE FILTERING APPROACH

When developing recommenders' systems, one of the most effectively methods is collaborative filtrating (CF). It bases the unknown preferences for new users on the known preferences of a group of users. For CF to work, it is necessaries that users x and y rate n objects similarly or exhibit similar behaviors. As a result, they will providing comparable ratings to other objects (Su and Khoshgoftaar, 2009). According to Breese et al. (1998), ratings can be either explicit or implicit. Explicit ratings involve a user stating their choices for an item using a numerical scale, such as 1–5. Implicit ratings, on the other hands, refers to inferring the user's behaviors or choosing in order to give the user preferences. CF techniques can function in fields when the contents of the objects are hardly to get or cannot be done.

CF approaches can be classifying into two main types: Memory-based and Model-based methods. Memory-based CF methods used a sample of user-item databases to producing predictions, while Model-based methods used user-based and item-based correlation/similarity measures.

CHARACTERISTICS AND CHALLENGES OF COLLABORATIVE FILTERING (CF)

One of the primary characteristics of CF methods is their independence from machine-readable representations of recommended objects. Unlike content-based approaches that rely on explicit features of items, CF relies solely on user interactions and preferences. This characteristic makes CF particularly suitable for recommending complex objects such as sounds and movies, where variations in taste heavily influence preferences. A major challenge in CF arises from the scarcity of historical data, especially during the initial operational stages of a recommender system. Sparse user data, such as viewing, purchasing, or rating history, can lead to unreliable similarity measures or nearly zero similarities between users. Even when there is a substantial user base with known preferences, new users face difficulties in receiving accurate recommendations until a sufficient amount of item ratings has been collected. This ramp-up period hampers the effectiveness of CF for new users. As the number of users and items in a system increases significantly, CF techniques encounter scalability problems. Computational resources required for processing user-item interactions can surpass practical or acceptable levels, leading to performance degradation.

CONTENT BASED FILTERING

Content-based filtering proves highly effective in job recommendation systems, particularly when students have specific job preferences and requirements. This method leverages the content of job postings and resumes to identify job opportunities that align with a student's qualifications and career objectives. This approach typically involves two main steps: feature extraction and recommendation. Feature extraction entails analyzing job postings and resumes to identify relevant attributes such as job titles, skills, education, and experience. Natural language processing (NLP) techniques such as named entity recognition, keyword extraction, and part-of-speech tagging are commonly employed to extract these features.

By extracting key features from both job postings and resumes, the system gains insights into the requirements of job roles and the qualifications of potential candidates. Once the relevant features have been extracted, the system utilizes them to recommend job opportunities to students. This process involve comparing the features of a student's profile with those of available job postings and identifying the most suitable opportunities based on a job postings and identifying the most suitable opportunities based on a similarity score. The system prioritizes job opportunities that closely match the skills, experience, and preferences outlined in a student's resume

ADVANTAGES OF CONTENT-BASED FILTERING

By focusing on the content of job postings and resumes, content-based filtering enhances the precision of job recommendations. By matching students with relevant job opportunities based on their skills and experience, content-based filtering streamlines the job search process.

Content-based filtering enables the system to tailor recommendations to the individual preferences and requirements of students. Overall it serves as a powerful methodology for improving the accuracy and efficiency of job recommendation systems. By leveraging NLP techniques to extract relevant features from job postings and resumes, this approach facilitates personalized matching between students and job opportunities, ultimately enhancing the effectiveness of the job search process.

KNOWLEDGE BASED APPROACH

Knowledge-based approach to recommender systems focuses on leveraging explicit knowledge about products or services and user preferences to make recommendations. Instead of relying solely on historical user behaviour or preferences, knowledge-based recommenders use rules, patterns, and deep domain knowledge to suggest items that best match a user's needs and preferences.

ADVANTAGES OF KNOWLEDGE BASED APPROACH

Knowledge-based recommenders perform reasoning to determine which products best meet the user's requirements. They may employ techniques such as quantitative decision support tools to assist in this process. Unlike collaborative filtering methods, knowledge-based recommenders do not require collecting information about specific users' tastes. Instead, they make judgments based on the inherent characteristics of the products and the user's stated preferences.

Since knowledge-based recommendations are not based on user ratings or historical data, they do not face the "ramp-up problem" commonly encountered by collaborative filtering systems. This makes them suitable for new users or users with sparse interaction data.

CHALLENGES OF KNOWLEDGE RECOMMENDER SYSTEMS

Building and maintaining the knowledge base requires acquiring relevant information about products, user preferences, and their relationships. This process can be time-consuming and resource-intensive. Developing and managing the knowledge base involves knowledge engineering, which encompasses designing, implementing, and maintaining the rules, patterns, and logic used by the recommender system. This can be challenging due to the complexity of the domain and the need for domain expertise.

HYBRID RECOMMENDER SYSTEMS

Hybrid recommender systems combine different recommendation approaches to improve performance and overcome the challenges inherent in individual methods. Burke (2002, 2007) proposed several types of hybrid recommender systems, each with its own approach to integrating collaborative filtering, content-based filtering, and knowledge-based techniques.

Weighted Hybrid Recommender approach calculates the score of item recommendations by combining the results of all available recommendation techniques in the system. Each technique's contribution is weighted, and the final recommendation score is a combination of these weighted scores.

Switching Hybrid Recommender approach, the system uses a measure to switch between recommendation techniques based on certain criteria or conditions. For example, it may switch between collaborative filtering and content-based filtering depending on the availability of user data or the nature of the items being recommended.

OBJECTIVES

The job recommendation system developed in this research aims to address two key challenges in the recruitment process: reducing the workload of the recruiting team and ensuring fairness to job aspirants. By leveraging automated machine learning techniques, the system attempts to efficiently identify the most suitable candidates from a vast pool of resumes while ensuring thorough consideration of each applicant's qualifications and skills. The motivation behind these endeavours stems from the recognition of the significant burden placed on recruiting teams to sift through numerous resumes to find the best-fit candidates, often resulting in inefficiencies and potential biases.

Additionally, job aspirants deserve a fair chance to be evaluated based on their merits and qualifications rather than being overlooked due to manual screening limitations. To achieve these objectives, the proposed model operates in two distinct phases.

Firstly, the preprocessing phase encompasses several essential tasks to prepare the data for analysis. This includes cleaning the resumes to remove any irrelevant or redundant information, pickling to serialize objects for efficient storage and retrieval, tokenization to break down text into individual words or phrases, preparation of the data for further analysis, and feature extraction using techniques such as NLP to identify important keywords or phrases. Additionally, feature mapping is employed to map extracted features to a numerical representation that can be processed by machine learning algorithms. This preprocessing phase lays the groundwork for the subsequent analysis and recommendation steps.

METHODOLOGY

The deployment and inference phase involve the application of machine learning algorithms to classify, compute similarities, rank, and ultimately suggest the most suitable resumes based on predefined criteria. A classifier is utilized to categorize resumes based on their relevance to specific job roles or skill sets, enabling recruiters to focus their attention on the most promising candidates. Mathematical computations are employed to quantify the similarity between job aspirants' resumes and job requirements, facilitating more accurate matching. Furthermore, ranking mechanisms are utilized to prioritize resumes based on their alignment with the desired qualifications and experiences. The final output of the system consists of a curated list of recommended resumes that best match the job criteria, providing recruiters with a streamlined and efficient means of identifying potential candidates.

DATA COLLECTION

The dataset utilized in this study is a meticulously curated compilation of poll data from Stack Overflow, supplemented by user-specific information, creating a robust resource for analysing developer preferences and job trends. With approximately 89,000 observations and 87 columns, the dataset predominantly comprises textual data representing programming languages, tools, and technologies utilized by developers.

This comprehensive dataset offers insights into the diverse skill sets and interests within the developer community, with 85 columns dedicated to textual data, one for Boolean values, and one for integer data. The Stack Overflow Developer Survey serves as a primary data source, providing extensive coverage of the programming industry and developer job preferences. Through polls conducted by Stack Overflow, developers contribute detailed information about their skills, experiences, and job aspirations, enabling a nuanced understanding of the job market dynamics.

This data serves as a foundational element for the development of a sophisticated Job Recommendation System capable of suggesting tailored job opportunities based on candidate preferences. Supplementary data was collected through web scraping from a prominent employment website, further enriching the dataset with real-time insights into the current job market landscape. Structured into seven categories, with six containing string data and one designated for unique identifiers, the dataset facilitates efficient data manipulation and analysis.

Researchers and practitioners interested in exploring the dataset can access it via the provided link on the Stack Overflow website. In summary, this curated dataset offers valuable insights into developer preferences and job trends, laying the groundwork for advanced job recommendation systems and contributing to a deeper understanding of the dynamics of the developer job market.

DATA PREPROCESSING

Data preprocessing is a fundamental aspect in constructing a Job Recommendation System leveraging Machine Learning and Natural Language Processing with the Stack Overflow Developer Survey dataset. This phase entails rectifying missing values, discrepancies, and outliers through data-cleaning procedures, such as imputation, outlier removal, and error correction. Integration of multiple files into a cohesive dataset is imperative for subsequent analysis. To facilitate machine learning algorithms' handling of categorical or textual data, conversion of the dataset into numerical format is necessary, typically achieved via techniques like one-hot or label encoding. Textual data undergoes preprocessing steps, including tokenization, stop-word removal, stemming or lemmatization, and sentiment analysis.

Given the dataset's breadth of features, feature selection methods like correlation analysis, principal component analysis (PCA), or feature importance ranking are employed to identify the most pertinent features for job recommendation tasks.

The creation of a user choice matrix from a comma-separated file containing user and job information. Each column is intended to yield a two-dimensional matrix incorporating the user's preferred database or language skills listed in each row, with original values transformed into column names for each user in the row. Utilizing the Stack Overflow Developer Survey dataset as the basis for system development ensures data accuracy, reliability, and suitability for analysis, facilitated by natural language processing and machine learning algorithms.

This meticulous preprocessing regimen guarantees that the resulting recommendation system can effectively align job opportunities with candidate preferences, offering invaluable insights for both job seekers and recruiters.

Evaluation

In the evaluation phase, the cosine similarity-based recommender system will be assessed by recommending jobs based on user preferences. Participants will be randomly selected, and multiple job suggestions will be generated using their details. Performance metrics such as coverage, precision, recall, and F1 score will be computed for each set of recommendations.

Furthermore, precision, recall, and F1 measures will be calculated for various threshold values using the same user details. The evaluation will also determine the coverage value of recommended items as a percentage of total items, calculated using the equation provided.

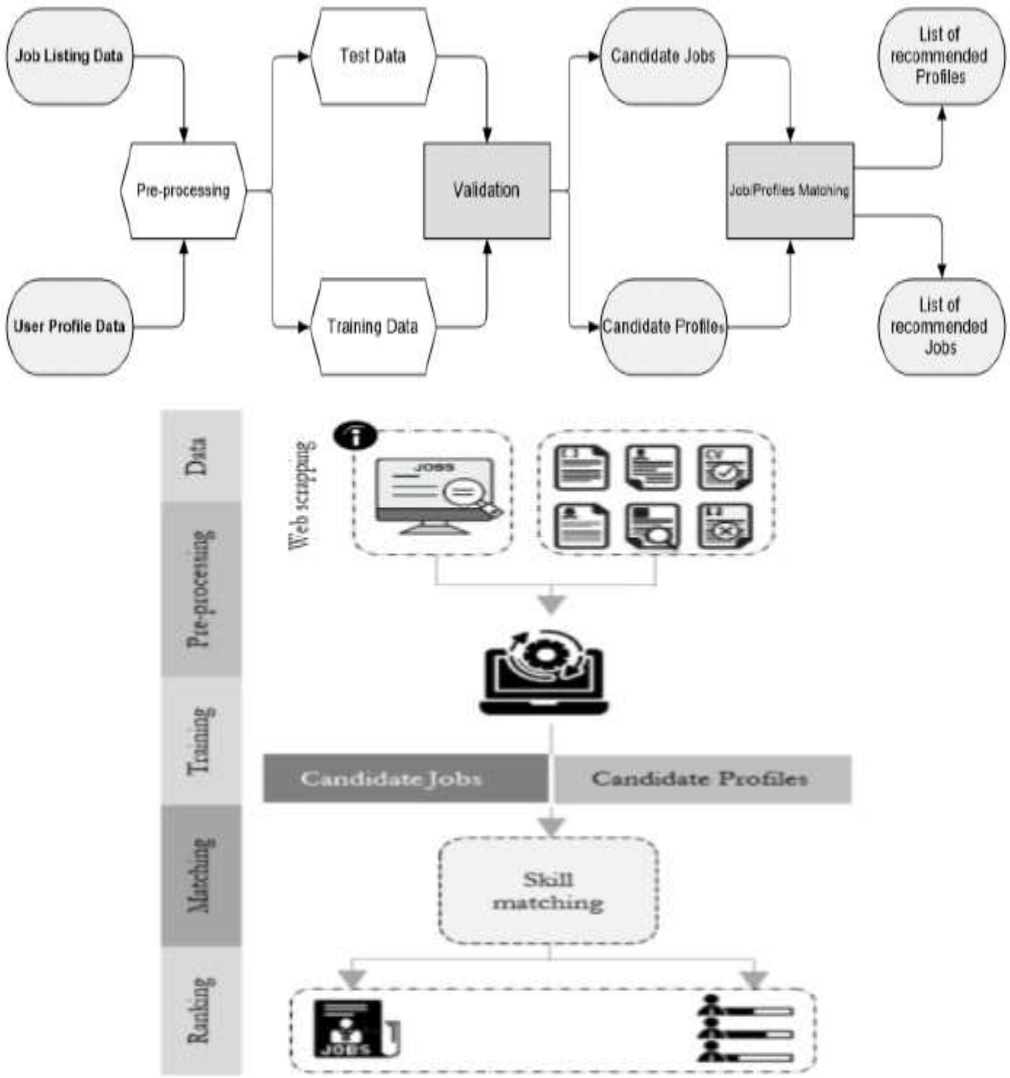

Fig. 1 - (a) flowchart of bi-directional recommender system; (b) job seeker summary

RESULTS AND DISCUSSION

The career recommendation system developed in this research integrates machine learning and NLP to offer personalized job suggestions based on candidate preferences. Utilizing a hybrid filtering strategy, it enhances recommendation precision and relevance. Evaluation results confirm its effectiveness in matching job opportunities to individual profiles, emphasizing the transformative potential of ML and NLP in optimizing recruitment processes for job seekers. Future research could explore real-world application scenarios and the incorporation of additional data dimensions such as personality traits and social network analysis for further customization of recommendations.

CONCLUSION

In summary, this study shows how machine learning and natural language processing may be successfully combined to create a system that recommends careers. Through the utilization of job candidates' characteristics and interests, the system provides tailored job recommendations, consequently improving the effectiveness of the job search procedure. By employing a hybrid filtering approach that blends cooperative Using content-based filtering methods, the system improves the accuracy and pertinence of its suggestions. Positive precision, recall, and F1 scores in the evaluation findings demonstrate to the model's efficacy in correctly matching job prospects with individual preferences. This study highlights how machine learning and natural language processing (NLP) may completely change the hiring process and help job seekers choose the right career path.

REFERENCES

1. Pradeep Kumar Roy, Sarabjeet Singh Chowdhary, Rocky Bhatia, “A Machine Learning approach for automation of Resume Recommendation system” 2020 and IEEE, Procedia Computer Science, Volume 167, 2020.
2. Tejaswini K, Umadevi V, Shashank M Kadiwal, Sanjay Revanna, “Design and Development of Machine Learning based Resume Ranking System” 2021 and IEEE.
3. W. Chen, X. Zhang, H. Wang and H. Xu, "Hybrid deep collaborative filtering for job recommendation," 2020 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA).
4. H. Jain and M. Kakkar, "Job Recommendation System based on Machine Learning and Data Mining Techniques using RESTful API and Android IDE," 2019 9th International Conference on Cloud Computing.
5. K. Appadoo, M. B. Soonnoo and Z. Mungloo-Dilmohamud, "Job Recommendation System, Machine Learning, Regression, Classification, Natural Language Processing," 2020 IEEE Asia-Pacific 1] Shaha T Al-Otaibi and Mourad Ykhlef [2012] - A Survey Of Job Recommender Systems.International Journal of Physical Sciences, Vol. 7, 29 (2012).
6. Stack Overflow. (2021). Stack Overflow Developer Survey. [Online]. Available: https://insights.stackoverflow.com/survey/
7. Orestis Nalmpantis & Christos Tjortjis [2017] - The 50/50 Recommender: A Method Incorporating Personality into Movie Recommender System.
8. Koen Bothmer and Tim Schlippe [2022] - Skill Scanner: Connecting and Supporting Employers, Job Seekers and
9. Educational Institutions with an AI-based Recommendation System.
10. Naresh Kumar, Manish Gupta, Deepak Sharma and Isaac Ofori [2022] - Technical Job Recommendation System Using APIs and Web Crawling.
11. Ying Sun, Fuzhen Zhuang, Hengshu Zhu, Qi Zhang, Qing He & Hui Xiong [2021] - Market-Oriented Job Skill Valuation with Cooperative Composition Neural Network.
12. D Bañeres Besora, J Conesa Caralt [2017] - A Life-long Learning Recommender System to Promote Employability.
13. Juhi Dhameliya; Nikita Desai [2019] - Job Recommender Systems: A Survey.
14. Deepjyoti Roy & Mala Dutta [2022] - A systematic review and research perspective on recommender systems.
15. Christin Katharina Kreutz & Ralf Schenkel [2022] - Scientific paper recommendation systems: a literature review of recent publications.
16. Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua [2017] - Neural Collaborative Filtering: Recommender Systems via Collaborative Filtering.
17. Chen Ma, Xiaolong Jin, and Yuanfang Guo [2022] - Social Collaborative Filtering by Trust.
18. Jeong Hyeok Park, Dongjin Song, and Seungjin Choi [2021] - Preference-based k-Nearest Neighbor Collaborative
19. Filtering. [7] D Bañeres Besora, J Conesa Caralt [2017] - A LifConference on Computer Science and Data Engineering (CSDE)
20. Hao Wang, Naiyan Wang, and Dit-Yan Yeung [2020] - Jointly Learning to Recommend and Personalize from Implicit Feedback.

21. Qianqian Shan, Yifan Chen, Huizhi Liang, and Yue Gao [2018] - Trust-Aware Collaborative Filtering for Recommender Systems.
22. Qingwen Liu, Shoubin Kong, and Hui Tian [2020] - Collaborative Filtering for Recommendation with Multiple Interaction Types.
23. Yuan Yao, Hua Wang, Jingtao Ding, and Hongyuan Zha [2018] - Transfer Collaborative Filtering for Cross-Domain Recommendation.
24. Zheng Zhang and Shaoda Zhang [2021] - Hybrid Algorithm Based on Content and Collaborative Filtering in Recommendation System Optimization and Simulation