International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 04 | Apr 2024 www.irjet.net p-ISSN: 2395-0072

JOB AND SKILLS RECOMMENDATION SYSTEM FOR JOB SEEKERS AND RECRUITERS

Bhavani R[1], Harikrishna P[2], Sabarinathan S[3], Yaswanth K[4]

[1234]*Dept. of Computer Science and Engineering, Government College of Engineering Srirangam, Tamil Nadu, India*

---***---

Abstract - *In today's dynamic job market, both job seekers and recruiters face challenges in matching skills to job requirements efficiently. The Job and Skills Recommendation System (JSRS) aims to address this by providing personalized recommendations to job seekers and recruiters based on their respective needs and preferences. For job seekers, JSRS offers a user-friendly interface where they can input their skills, qualifications, and career preferences. Leveraging advanced algorithms and machine learning techniques, JSRS analyzes this information along with historical job data to generate tailored job recommendations. These recommendations consider factors such as job relevance, career growth opportunities, and geographical preferences, enabling job seekers to discover suitable job openings quickly and easily. Ultimately, JSRS aims to bridge the gap between job seekers and recruiters, facilitating mutually beneficial connections that drive success in today's competitive job market.*

Key Words: *Machine Learning, Natural Language Processing, Job and Skills Recommendation System, K Nearest Neighbor, Term Frequency-Inverse Document Frequency.*

1.INTRODUCTION

In today's rapidly evolving job market, the process of matching job seekers with suitable employment opportunities and recruiters with qualified candidates has become increasingly complex. With the abundance of job postings and resumes available online, both job seekers and recruiters face the daunting task of sifting through vast amounts of information to find the perfect match. Traditional methods of recruitment often rely on manual sorting and keyword matching, which can be time-consuming, inefficient, and prone to biases.

To address these challenges, advanced technologies such as Artificial Intelligence (AI) and Machine Learning (ML) are being leveraged to develop innovative solutions that streamline the job search and recruitment process. One such solution is the development of JSRS, which aim to revolutionize the way job seekers find employment opportunities and recruiters identify top talent. By utilizing sophisticated algorithms and data analytics techniques, JSRS offers personalized recommendations based on the skills, preferences, and requirements of both job seekers and recruiters.

2. RELATED WORKS

This section aims to review the existing t techniques for job and skill recommendation. Alsaif et al. [1] developed a bi-directional recommendation system using NLP techniques to match job seekers with job recruiters. The system includes web scraping, data pre-processing, NLP model training, and bi-directional matching steps, utilizing sa.indeed.com data. Named entities detection and validation using precision metrics were conducted, with Word2vec employed for term retrieval to enhance matching efficiency. Mahalakshmi et al. [2] developed a Job Recommendation System focusing on skill sets to aid college graduates in finding fitting employment. It analyzes resumes to suggest tailored job opportunities and recommend skill enhancements. Employing preprocessing techniques and cosine similarity, it offers hierarchical job recommendations, aiming to reduce unemployment and foster career growth. Desai et al. [3] developed an Automated Job Recommendation System using Collaborative Filtering for personalized job suggestions. The system integrates user-based and item-based algorithms, utilizing student resumes and recruitment details for tailored recommendations. It encompasses data preprocessing, collaborative filtering, and evaluation phases, offering a promising solution to traditional job search inefficiencies. Mhamdi et al. [4] developed a Job Recommendation System using Profile Clustering and Job Seeker Behavior analysis, offering personalized suggestions by clustering job attributes and aligning with user behavior. Enhance accuracy by integrating Word2vec and k-means clusterings. Punitavathi et al. [5] proposed a three-tier architecture for an online job and candidate recommendation system, utilizing PHP Standards Recommendation (PSR) and text field filtering for efficient data management. The model integrates recommender system technology and web services to provide personalized recommendations, ensuring privacy through encryption. Their system effectively manages data flow, tackling information overload in online recruiting for robust and personalized recruitment processes.

3. PROPOSED SYSTEM

3.1 System Architecture

Figure 1 depicts the overall system architecture of the proposed system. The system architecture comprises three phases: User Login, Admin Login, and Recruiters Login. Users access the system using their credentials to upload resumes for the Job Seekers Recommendation System. Resumes, typically in PDF or DOC format, are processed to extract and convert them into Term Frequency-Inverse Document Frequency (TF-IDF) vectors. Simultaneously, job postings are scraped from LinkedIn or retrieved from the system's database. Job descriptions are parsed to extract skills, transformed into TF- IDF vectors, facilitating recommendation of the top K jobs to users using a and K-Nearest Neighbors (KNN) algorithm. Recruiters can upload job descriptions upon logging in. These descriptions are converted into TF-IDF vectors after extraction. Resumes stored in the database are analyzed to extract skills, also transformed into TF-IDF vectors. These components streamline the recommendation of the top K resumes to recruiters, aiding in efficient candidate selection. Admin login credentials provide access for editing and maintaining the database within the system for smooth operation and management.

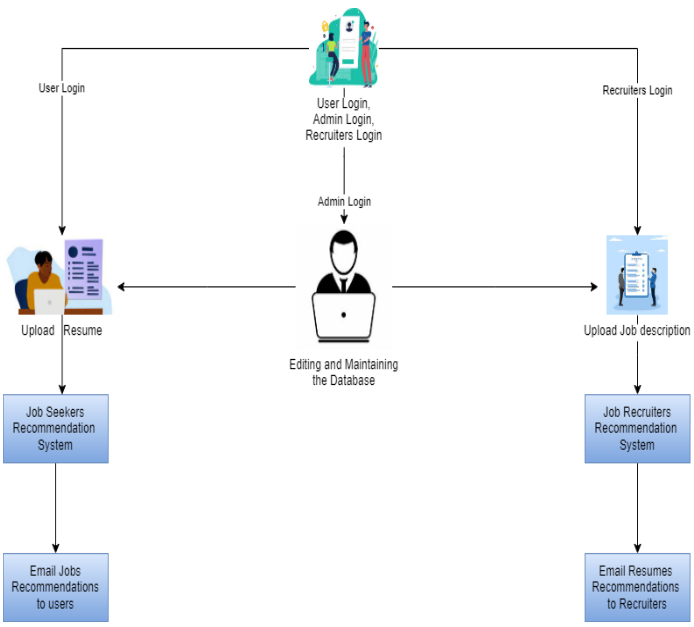

Fig. 1 System Architecture

3.2 Recommendation Engine

In recommendation engine that works on user side as well as recruiters side uses TF-IDF vectorization, KNN algorithm.

3.2.1 Term Frequency - Inverse Document Frequency

Textual data in the resumes and job descriptions are converted to numerical vectors using TF-IDF. TF-IDF assigns Determine the weights of words by considering their frequencies in a document relative to their frequency across all documents, capturing the importance of each word as given (1).

$$TF-IDF = TF * IDF \rightarrow (1)$$

Where $TF(i,j) = \frac{Term\, i\, frequency\, of\, the\, document\, j}{Total\, words\, in\, document\, j}$

$$TF(i,j) = \frac{Term\, i\, frequency\, of\, the\, document\, j}{Total\, words\, in\, document\, j}$$

$$IDF(i) = \log_2\left(\frac{Total\, documents}{Documents\, with\, term\, i}\right)$$

$$IDF(i) = \log_2\left(\frac{Total\, documents}{Documents\, with\, term\, i}\right)$$

3.2.2 K-Nearest Neighbour (KNN)

K-nearest neighbors (KNN) algorithm serves as a crucial component for recommending top job opportunities to users. KNN works by calculating the similarity between data points based on their features. By identifying the K nearest neighbors in the feature space, the algorithm recommends the most relevant job opportunities to users. The steps followed in K-NN algorithm is given below.

Input: X: training data, Y: class labels of X, K: number of nearest neighbors.

Output: Class of test sample x.

Start

Classify (X, Y, x)

1.for each sample x do Calculate the Distance:

d (x, X) = $\sqrt{\sum_{i=1}^{n}(x_i - X_i)^2}\sqrt{\sum_{i=1}^{n}(x_i - X_i)^2}$

end for

2.Classify x in the majority class:

C $(x_i) = argmax_K \sum_{x_j \in KNN} C(X_j, Y_K)$

$(x_i) = argmax_K \sum_{x_j \in KNN} C(X_j, Y_K)$

End

3.3 Web Scraping

Web scraping serves as a crucial component in gathering job postings from LinkedIn, enriching the system's database with a diverse array of opportunities. Python, a popular programming language renowned for its robust web scraping capabilities, is employed to implement these functionalities. Leveraging Python libraries such as Beautiful Soup and Scrapy, the system programmatically accesses and extracts data from LinkedIn's, Indeed etc. web pages.

These libraries offer powerful tools for parsing HTML content, navigating web elements, and extracting relevant job information efficiently. Admin login functionalities oversee the integration of Python-based web scraping methods into the system, ensuring seamless data acquisition and management. The extracted job postings are subsequently converted into a Comma-Separated Values (CSV) file format, enabling easy storage, retrieval, and manipulation within the system's database. Overall, Python's versatility and the utilization of web scraping techniques enhance the system's capability to provide users with up-to-date and relevant job opportunities, contributing to a more efficient and effective job matching platform.

3.4 Job Seekers Recommendation System

Figure 2 depicts the system architecture component of the Job Seekers Recommendation System. The job seekers recommendation system facilitates resume uploads, initiating a process where resumes are converted into text format for analysis. Advanced text processing techniques are then applied to extract relevant information, subsequently transformed into TF-IDF vectors, representing term significance for numerical analysis. Concurrently, job postings are scraped from LinkedIn, storing detailed descriptions in a CSV file, including titles and required skills. These job descriptions undergo text feature extraction, followed by conversion into TF-IDF vectors akin to resumes, enabling numerical representation. Upon TF-IDF conversion of both resumes and job descriptions, the KNN algorithm is deployed. This algorithm computes similarity between TF-IDF vectors of job seekers' skills and job requirements, identifying K nearest job positions aligning with the user's profile. Recommended job positions are then displayed to users, offering personalized opportunities. The process integrates text processing, web scraping, TF-IDF vectorization, and KNN algorithm, optimizing job recommendation for enhanced job seeking experiences.

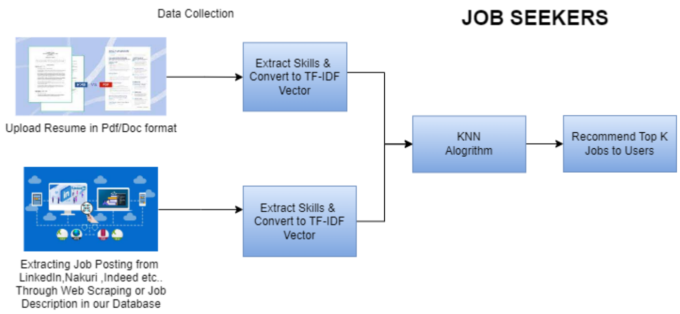

Fig. 2 Job Seekers Recommendation System

3.5 Job Recruiters Recommendation System

Figure 3 depicts the system architecture component of the Job Recruiters Recommendation System. The job recruiters recommendation system begins with the upload of job descriptions in text format, where text extraction isolates pertinent details such as titles, descriptions, and requisite skills. Extracted text undergoes conversion into TF-IDF vectors, weighting terms based on their frequency relative to other descriptions in the database, facilitating efficient storage and comparison. Concurrently, user resumes stored in the database are subject to skill extraction via Natural Language Processing (NLP) techniques, identifying key qualifications and skills. These are then converted into TF-IDF vectors, weighted based on frequency across all resumes in the system. Both job descriptions and resumes are represented as TF-IDF vectors, enabling the application of the KNN algorithm for recommendation. KNN calculates the distance between TF-IDF vectors to identify the K nearest resumes for each job description, facilitating effective matching. The recommended resumes, identified by the KNN algorithm, are then displayed to job recruiters for review through the system's interface. Recruiters can assess these recommendations to identify candidates closely aligned with the job requirements specified in the descriptions. This process streamlines candidate selection for job recruiters by leveraging TF-IDF vectorization and the KNN algorithm, ensuring efficient identification of suitable candidates for their job openings.

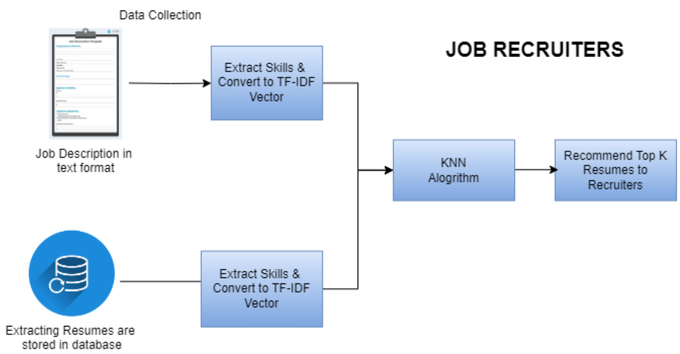

Fig. 3 Job Recruiters Recommendation System

4. EXPERIMENTAL RESULTS AND OUTPUTS

The experimental evaluation of the JSRS was aimed at assessing its performance across various components and functionalities. By conducting rigorous experiments, the effectiveness, efficiency, and usability of the system in facilitating job matching and talent recommendation processes were sought to be evaluated. In this section, the findings and outputs obtained from the experimental analysis are presented, starting with an overview of the system architecture and methodology performance.

4.1 User Login

New users begin the user login process by registering through a form, providing email, password, and confirming password. They select between candidate registration, input necessary details, and upon successful registration, are directed to the login page. Here, users access their profiles and associated functionalities by entering their email and password as given in figure 4.

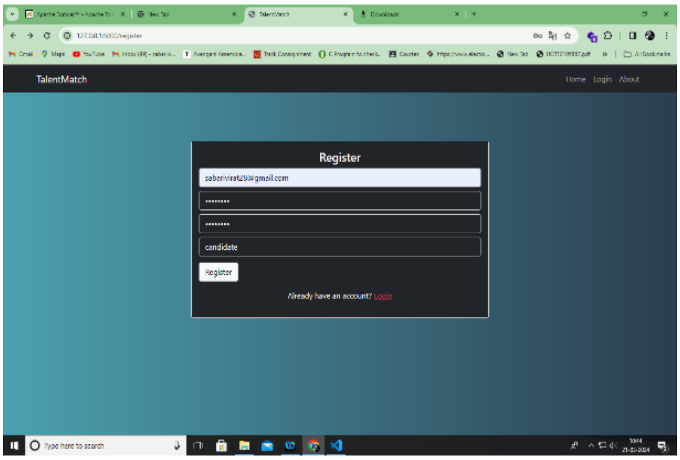

Fig. 4 User Login

4.2 Admin Login

Admin login directly accesses the database, bypassing the registration form. Upon successful authentication, the admin gains immediate access to the database for editing and maintenance purposes. This streamlined process enhances efficiency in managing system data given in figure 5.

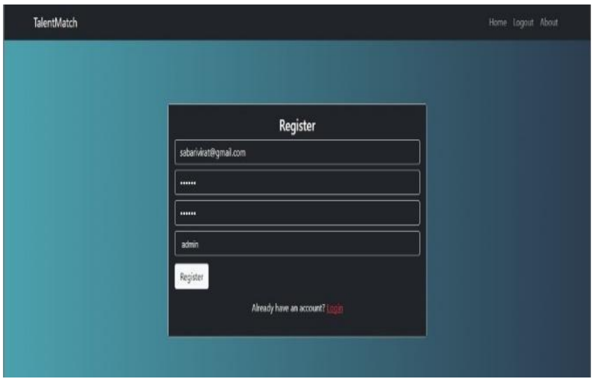

Fig. 5 Admin Login

4.3 Recruiters Login

New recruiters begin the user login process by registering through a form, providing email, password, and confirming password. They select between company registration, input necessary details, and upon successful registration, are directed to the login page. Here, users access their profiles and associated functionalities by entering their email and password as given in figure 6.

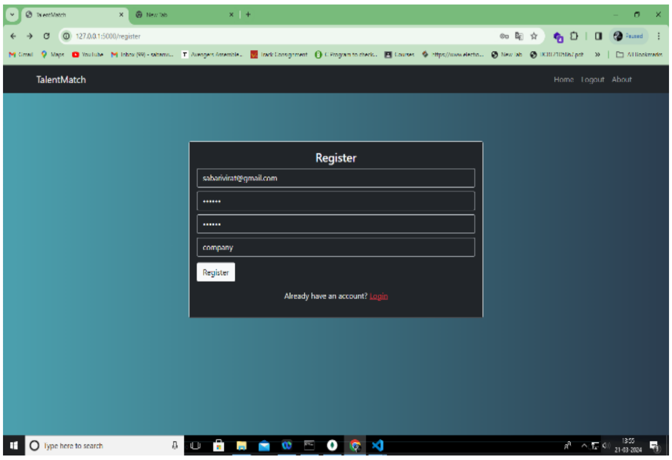

Fig. 6 Recruiters Login

4.4 Performance Evaluation

Performance metrics introduced, such as precision, recall, F1 score, and accuracy, are pivotal in assessing the effectiveness of our job recommendation system. Analyzing these metrics ensures accurate and impactful recommendations, elevating the user experience. Continuous optimization based on these evaluations underscores our commitment to delivering reliable solutions for job seekers and recruiters. Table 1 displays the job domain categories utilized in the confusion matrix.

Table 1: Job Domain Class

S.NO	JOB DOMAIN
01	Data Scientist
02	Software Engineer
03	Business Analyst
04	Web Developer
05	Product Analyst
06	Backend Engineer
07	PHP Developer

The table 2 below summarizes the performance metrics for various job roles in our recommendation system. These metrics help us evaluate how accurately the system recommends relevant job roles to users.

Table 2: Performance Metrics Table

Job Role	Precision	Recall	F1 Score	Accuracy
Data Scientist	0.85	0.80	0.82	88.33%
Software Engineer	0.78	0.75	0.76	81.66%
Business Analyst	0.82	0.79	0.80	88.33%
Web Developer	0.75	0.70	0.72	85.66%
Product Analyst	0.88	0.85	0.86	93.66%
Backend Engineer	0.80	0.76	0.78	86.33%
PHP Developer	0.83	0.81	0.82	87.66%
Macro Average	0.81	0.78	0.79	87.37%

Precision: Indicates how accurately the system recommends a specific job role. It is calculated as $\frac{TP}{TP+FP}$ $\frac{TP}{TP+FP}$, where TP is the number of True Positives and FP is the number of False Positives.

Recall: Measures the system's ability to capture all relevant instances of a job role. It is calculated as $\frac{TP}{TP+FN}$ $\frac{TP}{TP+FN}$, where FN is the number of False Negatives.

F1 Score: Balances precision and recall, providing an overall measure of the system's effectiveness. It is calculated as $\frac{2\times precision\times Recall \quad 2\times precision\times Recall}{Precision+Recall \quad Precision+Recall}$.

Accuracy: The accuracy of a classification model is calculated as the ratio of the sum of True Positives (TP) and True Negatives (TN) to the total number of predictions made, which includes True Positives, True Negatives, False Positives (FP), and False Negatives (FN). Mathematically, it can be represented as: $\frac{TP+TN}{TP+TN+FP+FN}$ $\frac{TP+TN}{TP+TN+FP+FN}$.

The performance metrics of the JSRS project, including precision, recall, F1 score, and accuracy, offer comprehensive insights into the effectiveness of our job recommendation system. Through rigorous experiments, it is observed that, the JSRS recommends jobs to the job seekers and recruiters with an accuracy of 87.37%. It recommends jobs that are most relevant to the skill sets specified in the resume and job description.

5. CONCLUSION & FUTURE WORK

JSRS revolutionizes job search and recruitment using NLP, ML, and web scraping, simplifying job seeking and candidate screening. It employs web scraping from platforms like LinkedIn, Indeed, and Naukri for accurate job postings, NLP for TF-IDF vectorization, KNN for personalized recommendations. JSRS offers tailor-made dashboards for job seekers, enhancing matching and bridging the gap between candidates and recruiters in the competitive job market. Future enhancements for JSRS include refining recommendation algorithms, integrating diverse job data sources, and exploring advanced technologies for optimization while ensuring privacy, security, and regulatory compliance. JSRS's evolution hinges on its adaptability to market dynamics and user requirements, empowering talent acquisition and career advancement for individuals and organizations.

REFERENCES

[1] Alsaif S. A., Sassi Hidri M ,Ferjani I , Eleraky H. A., & Hidri A (2022). NLP-based bi-directional recommendation system: Towards recommending jobs to job seekers and resumes to recruiters. Big Data and Cognitive Computing, 6(4), 147.

[2] Mahalakshmi G, Arun Kumar A, Senthilnayaki B, Duraimurugan J ,(2022). Job recommendation system based on skill sets: International journal of creative research thoughts (IJCRT), 10, ISSN: 2320-2882.

[3] Desai, V., Bahl, D., Vibhandik, S., & Fatma, I. (2017). Implementation of an automated job recommendation system based on candidate profiles. *Int. Res. J. Eng. Technol*, *4*(5), 1018-1021.

[4] Mhamdi, D., Moulouki, R., El Ghoumari, M. Y., Azzouazi, M., & Moussaid, L. (2020). Job recommendation based on job profile clustering and job seeker behavior. Procedia Computer Science, 175, 695-699.

[5] Punitavathi, D., Shinu, V., Kumar, S., & Sp, V. P. (2019). Online job and candidate recommendation system. International Research Journal of Multidisciplinary Technovation, 1(3), 84-89.