Optimizing Job Portals: A Study on Integrated Job Recommendation Systems for Enhanced User Experience

Kavish Punjabi
Computer Engineering Department
V.E.S. Institute of Technology, Chembur, Mumbai, India
2021.kavish.punjabi@ves.ac.in

Harsh Chandiramani
Computer Engineering Department
V.E.S. Institute of Technology,Chembur, Mumbai, India
2021.chandiramani.harsh@ves.ac.in

Aryan Hinduja
Computer Engineering Department
V.E.S. Institute of Technology, Chembur, Mumbai, India
2021.aryan.hinduja@ves.ac.in

Akshita Bathija
Computer Engineering Department
V.E.S. Institute of Technology, Chembur, Mumbai, India
2021.akshita.bathija@ves.ac.in

Prof. Sunita Suralkar
Computer Engineering Department
V.E.S. Institute of Technology, Chembur, Mumbai, India
sunita.suralkar@ves.ac.in

***Abstract* - Job Grove represents a transformative advancement in the realm of job portals, showcasing sophistication through its cutting-edge features. The platform streamlines the resume creation process with an intuitive Resume Builder, empowering candidates to construct compelling professional profiles effortlessly. Through the integration of machine learning, Job Grove pioneers a personalized job recommendation system, optimizing the alignment between candidates and relevant employment opportunities. Emphasizing trust and authenticity, the platform implements a rigorous company registration and verification process to admit only legitimate entities. This commitment to integrity is further reinforced by an admin - controlled verification mechanism. Beyond its core functionalities, Job Grove enhances the candidate experience by providing an opportunity for anonymous blog reading, fostering knowledge sharing and community building. Job Grove stands as a comprehensive and advanced solution, transcending traditional job portals and cultivating a dynamic and supportive ecosystem for both candidates and companies.**

Keywords-Advanced job portal, Resume Builder, Job recommendation system

Introduction

In the contemporary digital landscape of job-seeking portals, the platform under scrutiny emerges as a pioneering entity poised to revolutionize the employment-seeking process.This research paper aims to meticulously analyze the sophisticated features that distinguish this platform, encompassing its intuitive Resume Builder, personalized job recommendation system driven by machine learning algorithms, and meticulous company registration and verification process aimed at ensuring the integrity of participating entities.Furthermore, this study delves into the platform's steadfast commitment to fostering trust and transparency in facilitating connections between qualified candidates and reputable organizations.This research endeavors to provide a comprehensive exploration of the platform's architectural intricacies and its potential implications for the landscape of online job search portals. Additionally, it integrates cutting-edge technologies such as Optical Character Recognition (OCR) and Applicant Tracking System (ATS) to optimize the application and matching process, thereby enhancing the efficiency and efficacy of both candidates and employers' experiences on the portal.

Literature Survey

This research introduces a web portal tailored for students in computer science and information technology, aiming to bridge the gap between academic institutions and ICT sector organizations to support graduates in their job search. Through a survey-based approach, key challenges faced by students in employment-seeking are identified, informing the portal's development. Utilizing the Web Information Systems Development Methodology (WISDM), conventional methods are combined with web development technology to integrate the portal's requirements. While job portals offer benefits like comprehensive analysis and efficient resource utilization, challenges such as inefficient information distribution and limited user-specific customization need to be addressed for optimal functionality.[1]

The paper introduces a novel approach for job recommendation systems, merging content-based and collaborative filtering techniques using a cost-sensitive Statistical Relational Learning (SRL) method. Noteworthy is its capability to handle varying costs for false positives and negatives, enhancing its applicability across recommendation systems. Key innovations include the utilization of probabilistic logic models for large-scale job recommendation systems, integration of domain-specific requirements, and validation of statistical relational learning's effectiveness with real-world data. The method

involves data preprocessing to extract features and construct a user - item matrix , followed by content - based and collaborative filtering steps. Employing a cost-sensitive learning approach, the authors integrate statistical relational learning to create a hybrid recommendation system. Evaluation metrics such as precision, recall, and AUC-ROC demonstrate the hybrid model's superiority over individual filtering methods, particularly in recall and AUC-ROC. While the model effectively addresses challenges in combining filtering methods and offers insights into job recommendation system dynamics, concerns include increased complexity, scalability issues with larger datasets, and a lack of detailed implementation information. These aspects highlight the model's strengths and suggest areas for further optimization and exploration.[2]

The application of Natural Language Processing (NLP) and Long Short-Term Memory (LSTM) techniques in the realm of resume screening, as discussed in the paper titled "Resume Screening using NLP and LSTM." The paper proposes a software system aimed at categorizing resumes into various job options based on the skills they contain, ultimately assisting job seekers in understanding their qualifications for different positions. Through processes such as data cleaning, tokenization, vectorization, model training, and prediction, the system efficiently screens resumes and presents the top job categories along with confidence percentages. Emphasizing the significance of time-saving and managing large candidate volumes, the study demonstrates the system's functionality using a dataset of 962 resumes and 25 job categories. It also explores existing approaches to resume screening, such as application tracking systems, dictionaries with critical terms, linear SVC, and K-nearest neighbor classifiers, while suggesting future upgrades like extracting skills from LinkedIn and GitHub profiles and handling resumes in diverse formats. Overall, the paper underscores how NLP and LSTM technologies can enhance the efficiency and accuracy of resume screening processes, aligning with my research focus on leveraging advanced technologies for optimized recruitment procedures.[11]
Exploring the integration of natural language processing (NLP) and machine learning (ML) techniques within resume screening, insights from a systematic review paper are drawn. This evaluation covers diverse approaches utilized in both open-source and commercial resume parsers, with a focus on enhancing information extraction and contextual understanding through semantic search. Despite notable advancements, persistent challenges like writing style variations and syntactical complexities prompt future research directions. The paper underscores the continuous evolution of resume parsing and the necessity for ongoing improvements, particularly in context-based searching, to optimize candidate evaluation processes.[12]

The article [4] describes a proposed system for generating resumes and suggesting jobs for users based on their skills and interests. The system is a combination of a resume builder and a job recommendation system. The proposed system overcomes the limitations of existing resume builders, such as being paid and generating resumes that are either too long or too short. The system uses recommendation algorithms such as Content-based filtering and Collaborative filtering to suggest jobs to users. The proposed system is ready to be used by working professionals and freshers in the industry. However, the web app does not allow users to edit their resumes, and currently, only a standard template is used for the generation of resumes. The article suggests future scope for the system, such as providing different templates, allowing subscriptions for different packages of templates, and showing some of the best resumes that got selected in top MNCs.[4]

Feature	**Naukri.com**	**Indeed**	**LinkedIn**	**Intershala**	**apna.co**	**Proposed System**
Resume builder	Easily build a professional resume by inputting your skills and experiences.	A web-based tool supporting job seekers in creating and modifying resumes conveniently. .	Users create resumes using an easy template on the platform and receive keyword suggestions.[5]	Guides students in creating professional resumes with templates, prompts, and formatting.	-	Customizable templates for job seekers to create professional resumes with personalization options.

Referral tools	Recommend friends for jobs and keep track of the results with simple tools.	Tools facilitating companies in attracting and recruiting top talent through employee networks.	Various options on the platform encourage active users to refer new connections.[5]	Offers personalized referral links, enabling users to earn bonuses for successful hires.	A referral tool enables users to recommend friends for jobs, creating opportunities with desired companies.	Integrate a referral program for job seekers to enhance chances through network connections.
Job Recommendation System	Receive personalized job suggestions based on your preferences through an intelligent system.	A system suggesting relevant job opportunities to users based on their profiles and preferences.	The platform utilizes machine learning algorithms to analyze a user's address, education, and interests, providing job suggestions.	AI suggests relevant jobs based on user preferences, skills, and historical data.[10]	A job recommender utilizes user data to offer personalized and relevant up-to-date job suggestions.	Create a ML feature for precise job recommendations by analyzing profiles and aligning skills semantically.
Application Tracking	Stay organized by tracking the progress of your job applications effortlessly.	An applicant tracking system aiding employers in organizing job applications and making hiring decisions.	An Applicant Tracking System named Talent Hub is provided for tracking job and internship applications.	Monitors internship application progress with real-time updates and required actions.	An Applicant Tracking System (ATS) empowers job seekers to monitor the status of their job applications in one place.	A feature for job seekers: ATS integration to organize and track job applications effortlessly in one place.
Job Search	Find relevant job opportunities more easily using straightforward search features.	An extensive job search engine allowing users to find positions using various criteria.	Job seekers leverage tools and resources for efficient job searching.[5]	Efficiently explores job vacancies in India using advanced search filters.	A job search engine allows users to find employment opportunities based on keywords, location, and other criteria.	Proposed: Job recommendation system using user clustering, optimized for search engines to attract top talent.
Company Profile reviews & its Rating	Share and discover workplace experiences through firsthand reviews and ratings.	A platform enabling users to assess and review companies, providing valuable insights for decision-making.	Users can share their views and experiences about a company or a specific domain within an industry.[5]	Provides insights into company profiles and ratings for informed job decisions.[10]	-	Implement a comprehensive company profile and rating system for informed decision-making.

Recruiter Dashboard	Provide recruiters with tools to simplify talent discovery, screening, and hiring processes.	A comprehensive dashboard empowering employers to manage job postings, candidates, and the hiring process efficiently.	Recruiters are equipped with tools to manage job postings, view profiles, access analytics, and streamline recruitment	Streamlines hiring with an efficient dashboard for identifying and screening talent.	A Recruiter Dashboard efficiently manages job postings, candidates, and hiring tasks.	Recruiter dashboard streamlines hiring, boosts efficiency, and offers insightful ATS tools.

METHODOLOGY

The development of a comprehensive job portal website integrating job search, application functionality, personalized job recommendations leveraging machine learning, and a built-in resume builder follows a systematic methodology. Initiation begins with a meticulous analysis of requirements, outlining user stories, and defining functional specifications. The subsequent phase involves frontend development, encompassing the creation of an intuitive user interface with pages for job search, job details, and user profiles. Concurrently, backend development involves establishing secure user authentication, constructing and managing databases for job listings and user data storage, and integrating machine learning models to generate job recommendations based on user location, skills. Simultaneously, a dedicated team focuses on developing the machine learning algorithms, acquiring and preprocessing job data, and rigorously testing recommendation algorithms for accuracy and relevance. The implementation of a robust resume builder involves a user-friendly interface for data input and template generation based on user-provided information.

1. Collaborative Filtering

Our initial approach involved the implementation of Collaborative Filtering, a widely used recommendation system technique. Collaborative Filtering relies on user-item interaction data to identify patterns and similarities between users. This method aims to recommend items to a user based on the preferences and behaviors of similar users. We employed collaborative filtering algorithms to generate job recommendations for users.

2. Hybrid Filtering

To enhance the recommendation system's performance, we progressed towards Hybrid Filtering. Hybrid Filtering combines the strengths of both Collaborative Filtering and Content-Based Filtering. This approach aims to overcome the limitations of individual techniques by merging them, providing more accurate and personalized recommendations. Our system utilized a combination of collaborative and content-based methods to further refine job suggestions.

3. Count Vectorizer and TFIDF Vectorizer

In the subsequent stage, we integrated text-based features into our recommendation system using Count Vectorizer and TFIDF Vectorizer. [9]

These techniques convert job descriptions into numerical vectors, enabling the system to consider the textual content of job postings. Count Vectorizer represents the frequency of terms, while TFIDF (Term Frequency-Inverse Document Frequency) Vectorizer considers the importance of terms in relation to the entire dataset. This addition aimed to capture the semantic meaning of job listings for more context-aware recommendations.[10]

4. Cosine Similarity

To measure the similarity between user profiles and job listings, we employed Cosine Similarity. This metric calculates the cosine of the angle between two vectors, providing a measure of their similarity. In our recommendation system, Cosine Similarity was utilized to identify the relevance of jobs to a user's preferences, considering both collaborative filtering and vectorized content features.[8]

5. Transition and Evaluation

Throughout the transition from collaborative filtering to hybrid filtering and the incorporation of vectorization techniques, we continuously evaluated the performance of the recommendation system. Evaluation metrics such as precision, recall, and F1 score were employed to assess the accuracy and effectiveness of the recommendations. This iterative process allowed us to refine and optimize the system for enhanced job suggestions. By integrating these methodologies, our research aimed to create a robust job recommendation system that considers both collaborative and content-based approaches, leveraging text analysis techniques for improved relevance and personalization in job recommendations.

IMPLEMENTATION

1. Introduction

This implementation presents a Python-based approach for automating the extraction and matching of skills from resumes to job descriptions (JDs). Leveraging various libraries such as spaCy, PyPDF2, and scikit-learn, the system aims to enhance the efficiency of the recruitment process by automating the initial skill matching step.

2. Required Libraries

Prior to executing the code, it is imperative to have the following Python libraries installed: spaCy, PyPDF2, pandas, ftfy, pyresparser, docx, python-docx, import-ipynb, nltk, and scikit-learn. These libraries collectively provide the necessary tools for natural language processing, PDF text extraction, data manipulation, and machine learning functionalities. In order to support extended functionalities, the implementation recommends the installation of additional libraries, including `ftfy`, `pyresparser`, `docx`, `python-docx`.

3. SpaCy Skill Matching

The implementation begins by initializing spaCy for skill extraction. A CSV file containing a predefined set of skills is loaded, and spaCy Matcher patterns are created to identify these skills within text data. This ensures a robust and context-aware skill extraction process.

4. PDF Text Extraction

To accommodate the common resume format in PDFs, a dedicated function utilizes PyPDF2 for extracting text from PDF files. This text extraction step sets the foundation for subsequent skill extraction from resumes.

```
# Function to extract skills from text
def extract_skills(text):
    doc = nlp(text)
    matches = matcher(doc)
    skills = set()
    for match_id, start, end in matches:
        skill = doc[start:end].text
        skills.add(skill)
    return skills
```

```
[ ] def extract_text_from_pdf(file_path:str):
        with open(file_path, 'rb') as f:
            pdf_reader = PyPDF2.PdfReader(f)
            text = ''
            for page in pdf_reader.pages:
                text += page.extract_text()
        return text
```

Fig(3.1) Code to extract skills from pdf

5. Skill Extraction from Resumes

The `skills_extractor` function seamlessly combines PDF text extraction and spaCy skill extraction, providing a comprehensive list of skills extracted from a given resume file. This step is crucial for understanding the qualifications and expertise of potential candidates.

```
def skills_extractor(file_path):
        # Extract text from PDF
        path=r'/content/'
        full_file_path = os.path.join(path, file_path)
        resume_text = extract_text_from_pdf(full_file_path)

        # Extract skills from resume text
        skills = list(extract_skills(resume_text))
        return skills
```

```
file_name = 'HITESH CV.pdf'
extracted_skills = skills_extractor(file_name)

# Print the extracted skills
print("Extracted skills from CV.pdf:", extracted_skills)
```

```
Extracted skills from CV.pdf: ['DJANGO', 'testing', 'p', 'HTM
L', 'System', 'JavaScript', 'Certification', 'CSS', 'Engineerin
g', 'SQL', 'database', 'GITHUB', 'JAVA', 'C', 'Python', 'PYTHO
N', 'communication', 'responses', 'debugging', 'system', 'r',
'JAVASCRIPT', 'administration', 'outreach', 'process', 'PHP']
```

Fig(3.2) Output of Skill Extractor

6. JD Data Loading

Job description data is loaded from a CSV file (`concatenated_structured_data.csv`) into a pandas DataFrame. This structured data serves as the reference for matching skills extracted from resumes with the required qualifications outlined in job descriptions.

```
[ ] nlp = spacy.load('en_core_web_sm')
```

```
[ ] file_path=r'/content/skills.csv'
    with open(file_path, 'r') as file:
        csv_reader = csv.reader(file)
        skills = [row for row in csv_reader]
```

```
[ ] matcher = Matcher(nlp.vocab)
```

```
# Set a confidence threshold
confidence_threshold = 0.9  # Adjust this threshold as needed

# Filter matches based on confidence threshold
predicted_matches = jd_df[jd_df['match'] > confidence_threshold]

# Assuming you have a list of predicted company names in the 'match' column
predicted_companies = predicted_matches['Processed_JD'].astype(str).tolist()

# Tokenize skills into individual words
tokenized_skills = set(word.lower() for skill in skills for word in skill.split())

# Tokenize job descriptions into individual words
tokenized_predicted_companies = set(word.lower() for company in predicted_companies
                                    for word in str(company).split())

# Calculate the overall accuracy based on confidence
correct_predictions = len(tokenized_skills.intersection(tokenized_predicted_companies))
accuracy = correct_predictions / total_resumes
```

Fig(3.3) Code to extract accuracy

7. N-gram Vectorization and Nearest Neighbors Matching

Skills are vectorized using n-grams, and the scikit-learn `NearestNeighbors` class is employed to calculate distances between skill vectors. This process facilitates the matching of skills from resumes with those specified in job descriptions, yielding a match confidence score.

8. Results and Output

The final output is encapsulated in a DataFrame (`jd_df`), providing a detailed overview of job descriptions alongside corresponding match confidence scores. Continuous refinement and optimization of the implementation can be performed based on specific organizational requirements and feedback from the recruitment process. This approach contributes to streamlining and enhancing the efficiency of the talent acquisition process.

9. Named Entity Recognition and Optical Character Recognition for Resume

In the pursuit of augmenting the versatility of our computational framework, this study endeavors to integrate a robust mechanism adept at assimilating and processing resumes encapsulated in the Portable Document Format (PDF). Given the heterogeneous nature of resume submissions, encompassing a myriad of file formats, our focus lies in the seamless extraction of pertinent information for subsequent analytical scrutiny. To this end, we introduce a meticulously devised two-step process, entailing PDF scanning and text extraction, culminating in the sophisticated application of entity recognition.

10. PDF Scanning and Text Extraction

Mitigating the intrinsic challenges associated with extracting textual information from PDFs demands a nuanced approach. Our methodology embraces a harmonious amalgamation of the PyPDF2 and textract libraries. The former facilitates the extraction of textual content from PDF documents through a methodical iteration across individual pages, while the latter offers an alternative, ensuring a comprehensive retrieval.

Illustratively, the `extract_text_from_pdf` function exemplifies the extraction of textual content from a PDF file. The resultant text undergoes a preparatory phase, rendering it amenable to in-depth analysis and subsequent application of entity recognition.

11. Integration with Entity Recognition

The extracted textual corpus undergoes a rigorous process of entity recognition, aimed at discerning specific elements such as job categories and skills. Drawing upon the formidable natural language processing capabilities of spaCy, our system meticulously identifies entities embedded within the resume text. This recognition process is informed by predefined patterns extracted from the Jobzilla skills dataset and job categories derived from the expansive resume dataset.

The `extract_entities_from_resume` function serves as an eloquent demonstration of the seamless integration of entity recognition with the extracted text. The resultant entities, spanning job categories and skills, furnish invaluable insights for subsequent analytical endeavors.

12. Entity Recognition Visualization for PDF Extracted Text

In the visual representation of entity recognition outcomes, we harness the capabilities of spaCy's `displacy` module. This visualization, an exquisite manifestation of sophistication, showcases a sample resume extracted from a PDF. The graphical presentation offers a nuanced understanding of how entities are meticulously recognized and annotated within the expanse of the resume text.

Conclusion

In conclusion, Job Grove represents a significant evolution in the digital employment platform landscape. Through its advanced features such as the user-friendly Resume Builder, personalized job recommendations, and stringent company verification process, it enhances the reliability and effectiveness of the job-seeking process. Job Grove's commitment to fostering trust and integrity in connecting job seekers with reputable organizations sets a new standard for online job portals. Moreover, the inclusion of anonymous blog reading promotes knowledge-sharing and community engagement, further enriching the platform's ecosystem. As a comprehensive and innovative solution, Job Grove closes the gap between qualified candidates and credible opportunities, thus shaping the trajectory of digital employment platforms in today's era.

References

[1] M. Mansourvar and N. B. M. Yasin, "Development of a Job Web Portal for the Faculty of Computer Science and Information Technology, University of Malaya," in International Journal of Computer Theory and Engineering, vol. 6, no. 1, pp. 44-49, Feb. 2014, doi: 10.7763/IJCTE.2014.V6.773.

[2] S. Yang and M. Korayem, "Combining content-based and collaborative filtering for job recommendation system: A cost-sensitive Statistical Relational Learning approach," Knowledge-Based Systems, vol. 136, pp. 37-45, 2017.

[3] M. Pinjari, N. De, R. Kokne, A. Siddiqui, and D. Chitre, "Online Job Portal," International Research Journal of Engineering and Technology, vol. 06, no. 04, pp. 156-159, Apr. 2019.

[4] Aslan Mulla, A., Kunjal Agrawal, A., Ayushi Khare, A., Atman Naik, A. (2023). Resume Builder Application with Automated Job Prediction. NeuroQuantology, 21(1), 128-134.

[5] R. Dukarić, "LinkedIn: Online community research," University of Madeira, Social Web Course 2009/2010.

[6] P. Jain, "Basics of CountVectorizer," Towards Data Science, May 24, 2021. https://towardsdatascience.com/basics-of-countvectorizer-e26677900f9c

[7] D. Kaplan, "Machine Learning 101: CountVectorizer Vs TFIDFVectorizer," December 24, 2023. https://enjoymachinelearning.com/blog/countvectorizer-vs-tfidfvectorizer/

[8] S. Prabhakaran, "Cosine Similarity – Understanding the math and how it works (with python codes)," https://www.machinelearningplus.com/nlp/cosine-similarity/

[9] amhjf, alexbuckgit, msakande, Blackmist, sdgilley, QueryCoordinator, SnehaGunda, "Create, evaluate, and score a recommendation system in Microsoft Fabric," Article, Nov. 16, 2023.https://learn.microsoft.com/en-us/fabric/data-science/retail-recommend-model

[10] V. Gupta, "Case Study: Internshala," Medium Article, Jul. 12, 2021. [Online].Available:https://designerblood23.medium.com/case-study-internshala-a367b515a013

[11] S. Bharadwaj, R. Varun, P. S. Aditya, M. Nikhil, and G. C. Babu, "Resume Screening using NLP and LSTM," in Proceedings of the Fifth International Conference on Inventive Computation Technologies (ICICT 2022), 2022, pp. 238-241, doi: 10.1109/ICICT54344.2022.9850889.

[12] A. K. Sinha, M. A. Khusru Akhtar, and A. Kumar wrote the paper "Resume Screening Using Natural Language Processing and Machine Learning: A Systematic Review," which was published in the Advances in Intelligent Systems and Computing series by Springer in 2021.

[13] Irfan Ali, Nimra Mughal, Zahid Hussain Khand, Javed Ahmed, and Ghulam Mujtaba wrote the paper "Resume Classification System using Natural Language Processing and Machine Learning Techniques," which was published in the Mehran University Research Journal of Engineering and Technology in January 2022.