An intelligent system for job recommendation based on semantic analysis of candidate's resume

Hardik Jain[1], Aparna Joshi[1], Deepali Naik[2], Rupali Gangarde[3], Ranjit Koragoankar[1], Yash Khapke[1], Varad Kulkarni[1]
[1]Department of Computer Engineering, PCET's Pimpri Chinchwad College of Engineering, Pune, India.
[2]Symbiosis Centre for Management and Human Resource Development, Symbiosis International (Deemed University), Pune, India
[3]Department of Computer Engineering, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, India

Article Info

Article history:

Received Nov 28, 2024
Revised Apr 9, 2025
Accepted Jul 4, 2025

Keywords:

Job recommendation system
ML
NLP
Resume analysis
Semantic analysis

ABSTRACT

The contemporary job market presents significant obstacles to effectively aligning proficient candidates with pertinent employment prospects. The conventional methods of resume screening and job matching frequently require substantial manual effort and are susceptible to subjective biases, resulting in recruiting decisions that are frequently suboptimal. The present study proposes the development of an intelligent job recommendation system that utilises semantic analysis of candidates' resumes and job descriptions sourced from several job portals. The objective of the proposed intelligent system is to enhance and streamline the recruiting process through the automated extraction and analysis of pertinent skills from resumes and job descriptions, utilising natural language processing (NLP) and machine learning (ML) techniques. In addition, web scraping techniques were used to collect job advertisements from several job portals. The developed model exhibits the ability to recommend the most suitable job prospects by computing similarity metrics, such as Euclidean distance, between skill clusters identified in a job advertisement and a specified candidate's resume. The implemented model achieves an accuracy rate of 98.92%. It is anticipated that the integration of an intelligent job recommendation system will augment the recruitment procedure for both job seekers and employers.

Corresponding Author:

Deepali Naik
Symbiosis Centre for Management and Human Resource Development
Symbiosis International (Deemed University)
Pune, India
Email: deepali.bhanage@gmail.com

1. INTRODUCTION

In this competitive job market, highly skilled professionals are needed constantly, necessitating recruitment strategies that are efficient and effective. Yet, with the high volume and complex nature of resumes and job advertisements, traditional strategies sometimes are not able to effectively match job applicants with the appropriate jobs. The overall objective of this study was to create an efficient job recommendation system through the application of semantic analysis through natural language processing (NLP) and the use of classification techniques from machine learning (ML). The objective of the proposed system was to improve the efficiency of the recruitment process for employers and applicants by automating qualification extraction and alignment, which would result in enhanced matches and decreasing time-to-hire. Furthermore, the proposed system avoids the drawbacks of keyword matching and also offers suggestions for investigating temporal analysis and advanced filtering techniques.

The GLAD framework [1], which focuses on its strengths over conventional approaches in optimising work performance and career development alignment, highlights the significance of transformation-based models in the personalised recommendations. Similarly, academic research in recommendation systems [2] highlights the significant need for fairness recommendations, referencing the need for fairness standards and methods to alleviate long-standing adverse effects and unfair recommendations. Moreover, the use of techniques such as web scraping [3]–[6] facilitates personalised filters and improved analytics, which provide significant insights for the enhancement of job search engines. Furthermore, advancements in fuzzy cosine similarity retrieval techniques [7] and text classification [8] improve the methods used in enhancing the accuracy and efficiency of recommendation systems. The use of deep learning (DL) methods in recommendation systems [9]–[11], as well as the use of trained vectors extracted from domain corpora [12], illustrates that recommendation methods consistently improve performance and the ability to serve a broader user base. Through a comprehensive review of methodologies, challenges, and future directions, the primary objective of this research is to provide researchers and practitioners with valuable insights that can inspire innovation and improve the effectiveness of recruitment processes. The contributions of this research are summarised as follows:

- Web scraping was applied to job sites for the extraction of job-related information, specifically job descriptions, using the Apify API tool.
- The collection of skill sets involved the utilisation of a skills dataset encompassing candidate resumes and job descriptions obtained from job sites.
- The NLP technique Word2Vec was employed to extract semantic vectors from candidate resumes and job descriptions.
- The technique of agglomerative clustering was utilised to arrange job description skills into contiguous clusters.
- The model was trained using different classifiers, including support vector machines (SVM), Naïve Bayes, random forest, and k-nearest neighbors (KNN). The features used were job description skills, while the labels were cluster numbers.
- A ranking method that utilises semantic vectors and Euclidean similarity indexing algorithms was developed to provide personalised recommendations for the top 10 most relevant positions to candidates, considering their skill sets as indicated in their resumes.

The remaining part of the paper is structured in the following format: Section II shows the literature review. Section III explains the methodology and the overall framework of the system, as well as the data acquisition, preprocessing, and classification models. Section IV shows the experiments and the calculations with the evaluation metrics and testing results. Section V shows the results and analysis of different classification models, and eventually, the future directions and concluding remarks are expressed in Sections VI and VII, respectively.

2. LITERATURE REVIEW

There are some problems with adding good learning strategies to business learning management systems (LMS). This is because how things are done now tends to put user click preferences ahead of career development alignment, leading to poor results [1]. The GLAD framework is revolutionary because it uses a transformer-based model with a performance predictor and a rationality discriminator to tailor learning plans to each student. It has been demonstrated that reinforcement learning RL approaches outperform current methods and endeavour to enhance the work performance of individuals [1]. Concerns regarding unethical recommendations and long-term damage have also increased consciousness regarding the need for impartial recommendation systems [2]. The academic literature in this field underscores the criticality of incorporating fairness indicators and strategies such as consistency and group fairness to mitigate negative implications. Future research should focus on developing explicit definitions, consistent assessment standards, various algorithm designs, and transparent justifications for inequitable results [2], [10].

Web browsing emerges as a critical information extraction technique, particularly in the context of employment and job hunting. Numerous techniques to increase the efficacy of data extraction are the subject of research, including reading based on regular expressions and novel approaches such as UzunExt [3]–[6]. But there are still problems with matching people with jobs and sorting them into groups. This shows how important it is to use more advanced methods that use standardised entity data and NPL [12]-[15]. Deep learning techniques, specifically graph neural networks (GNNs), are gaining popularity in the field of recommendation systems [9]–[11] due to their ability to comprehend complex graph topologies and high-order connectivity. These methodologies facilitate a more comprehensive comprehension of user-item interactions and contribute to the optimisation of recommendation outcomes. In recommendation scenarios, RL signifies a fundamental change from supervised learning to providing more precise recommendations;

this modification also signifies a paradigm shift in thought [16]. Recent improvements in text classification methods, like cosine similarity-based mechanisms that make many classifiers work better and be more accurate, show that recommendation systems are still being improved [8]. Case-based reasoning also shows that fuzzy cosine similarity retrieval methods work and suggests ways to make retrieval faster and users happier [7]. As the sector evolves, recommendation systems are gaining increasing significance in the processes of matching individuals with employment and locating suitable personnel.

With the convergence of findings from different areas of study, the current study aims to contribute to the development of a comprehensive employment counselling system that adequately addresses the complexity of challenges brought about by talent management and talent development. The main aim of the current study is to present actionable knowledge to practitioners and academics alike in the field of human resource management and talent acquisition through the general examination of research methodology, challenges encountered, and areas for improvement.

3. DESIGN OF THE PROPOSED SYSTEM

This section provides a comprehensive overview of the proposed system, focusing on the overall architecture, data-gathering methods, preprocessing techniques, skill collection from a candidate's resume and job description, vector generation, skill-based cluster creation and job ranking techniques employed in its implementation. Figure 1 illustrates the approach employed to implement an intelligent job recommendation system utilising NLP and ML techniques. The system's architecture comprises five main phases: data collection, preprocessing, vectorisation, clustering, and job ranking. Initially, data was collected with the help of web scraping from sources such as the Apify API [17], focusing on platforms like Indeed [18] and potentially LinkedIn [19]. This data was extracted to generate a job database. In the preprocessing phase, NLP techniques such as lowercasing, punctuation removal, and tokenisation were applied to extract skills from both candidates' resumes and job descriptions. Word embedding method Word2Vec was utilised to represent extracted skills in the form of numerical vectors.

Furthermore, agglomerative clustering algorithms were employed to group similar job description skills into clusters, where typically around five clusters were formed based on user-defined parameters. Subsequently, classification algorithms SVM, Naive Bayes, KNN, and Random Forest were trained using job description skills as features and cluster numbers as labels. To demonstrate the applicability of the implemented model, the system was tested by providing the candidate's resume as input, and the result was delivered by predicting the appropriate cluster to which it belongs. Finally, Euclidean similarity measures were calculated to rank and recommend the top 10 jobs within the predicted cluster to the user to facilitate efficient job matching.

3.1. Resume input and parsing

A primary phase in the recommendation system was gathering resumes from users or outside sources. A variety of methods, for example, direct uploads via the system's interface or interaction with already existing databases that contain resumes, can be used to gather resumes. The system supports PDF files for handling resumes from diverse sources. After acquiring them, a parsing operation was performed to extract pertinent data, including the candidate's contact details, education, experiences, and abilities. Parsing was carried out with the PyPDF2 [20] specific Python libraries that can process textual data efficiently. PyPDF2 aids in precisely locating and extracting particular resume portions. An extension of skills is vital, particularly in the later stages of the recommendation process.

3.2. Web scraping job sites

Web scraping job sites: Indeed, LinkedIn and Glassdoor are useful for collecting diverse job descriptions, which serve as the basis for matching candidates with suitable job opportunities. Web scraping involves the programmatic extraction of data from web pages, typically using specialised APIs like Apify. These tools provide functionalities to navigate through web pages, locate specific elements containing job listings, and extract relevant information such as job titles, descriptions, and requirements. However, web scraping poses various challenges, including handling dynamic content, managing rate limits, and ensuring data integrity. To address these challenges, the scraping process should be carefully designed and optimised to minimise the risk of errors and disruptions.

3.3. Preprocessing using NLP techniques

To improve the quality and coherence of the data, preprocessing textual data from job descriptions and candidate's resumes was essential. In the preprocessing phase, various steps were present, such as lowercasing all text to maintain consistency and deleting special letters and punctuation to reduce noise.

Furthermore, common, uninformative terms that don't add anything to the text's meaning were filtered away using stopword removal. By standardising the textual data, these preprocessing procedures enable it to be used for additional analysis and modelling. Moreover, to enable further analysis at the word level, the text was divided into individual words, or tokens, using tokenisation. Words can be reduced to their most basic versions by using stemming operations.

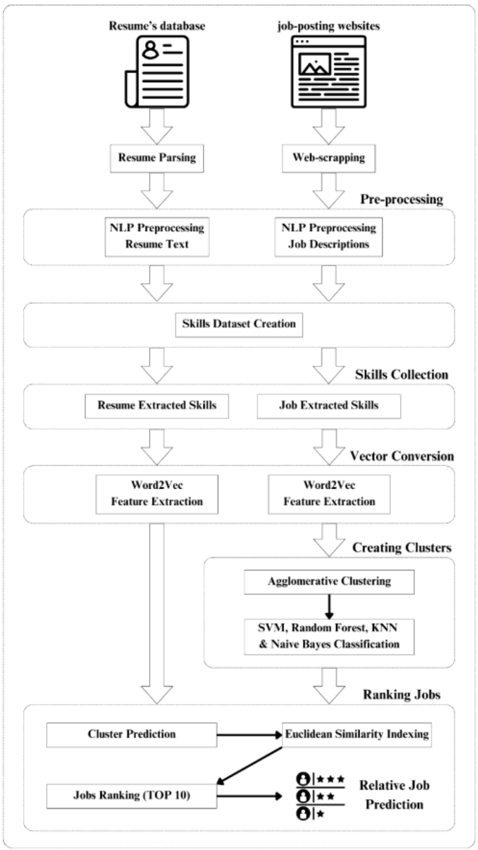

Figure 1. Overview of the proposed system

3.4. Vectorisation using Word2Vec

Vectorisation was done to transform preprocessed text data into numerical forms that ML algorithms can utilise. This was done to facilitate the application of computational and statistical techniques to text data. The literature provides a variety of vectorisation methods [21], [22], of which word embedding is especially prevalent. Word embedding method Word2Vec generates dense vector representations of words by examining their contextual usage in a large text corpus [23], [24]. By capturing semantic relationships between words, these word embeddings enhance the efficacy of text data representation.

3.5. Vector comparison and similarity computation

Vector comparison and similarity computation were carried out to calculate the fit of a candidate's skills to a particular job post. A similarity score was obtained by comparing job posting skill vector representations with candidate resumes. The similarity between vectors can be stated in terms of diverse distance measures, including the cosine similarity and Euclidean distance. These measures provide a numeric estimate of similarity between a pair of sets of attributes by quantifying the amount of overlap between the two. Furthermore, individual similarity scores can be combined in different ways to generate an overall similarity measure, which allows for a generic measurement of candidate fit for job postings.

3.6. Ranking and job recommendation

The last step of the proposed system included ranking the job opportunities based on the similarity scores that were obtained from the candidate and job vectors. To give meaningful job recommendations, the job opportunities were ranked based on the Euclidean similarity between the candidates' skills and the job requirements. The job opportunities that best fit were given a higher rank in the job listings, which were ranked in the order of decreasing similarity scores. Furthermore, ranking methods could be employed to highlight specific features or factors in ranking activities, such as thresholds or weighting systems. The highest-ranked job recommendations were presented to the users in reports or interfaces with supporting information, such as application links, job descriptions, and company profiles. The process improves the users' ability to search for corresponding job opportunities in order to take meaningful action and make effective decisions.

4. EXPERIMENTS

4.1. Data description

The database holds 1,550 job postings collected from three of the most common websites: Indeed, LinkedIn, and Glassdoor. Indeed, it holds the most with 900 postings, followed by LinkedIn with 350 and Glassdoor with 300. The websites individually show varying levels of data, the highest being Indeed at 5.1 MB, followed by LinkedIn at 1.8 MB and Glassdoor at 1.7 MB. Apify was among the web scraping tools used to scrape the records shown in Table 1.

Taking into account the demands of several industries and organisations, the numerous internet platforms present varying perspectives on today's employment market. Indeed, offers a comprehensive list of jobs for all experience levels and industries. As a site focused on professional connections, LinkedIn probably lists more niche roles tailored to specific domains or capability sets. Glassdoor, widely known for its emphasis on worker fulfilment and company evaluations, likely conveys information about typical job responsibilities as well as perspectives on corporate culture and workplace environment.

Table 1. Dataset description

Website	Number of jobs extracted	Size	Source/Resource
https://indeed.com/	900	5.1MB	https://console.apify.com/actors/hMvNSpz3JnHgl5jkh/console
https://linkedin.com/	350	1.8MB	https://console.apify.com/actors/BHzefUZlZRKWxkTck/console
https://glassdoor.com/	300	1.7MB	https://console.apify.com/actors/t2FNNV3J6mvckgV2g/console

4.2. Evaluation matrices

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \tag{1}$$

Accuracy is one of the most common evaluation metrics in classification problems. It calculates the percentage of accurate forecasts among all the forecasts made.

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

Precision assesses how well the classifier made its positive prediction.

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

Recall is sometimes referred to as a true positive rate or sensitivity. It is the ratio of the dataset's actual positive instances to the number of accurate positive predictions.

$$F1\ Score = \frac{2*Recall*Precision}{Recall + Precision} \tag{4}$$

Taking the harmonic mean of recall and precision, one can calculate the F1 score. It offers a single metric that balances the tradeoff between precision and recall. When there is an uneven distribution of classes, it is invaluable.

4.3. Software stack

The suggested intelligent job recommendation system based on semantic analysis of the resume is titled the "EasyJob" project. The selection of frontend, backend, and development tools is a crucial factor in the development of "EasyJob" as it directly influences the success of project goals and the delivery of a seamless user experience. The use of HTML, CSS, and Python in frontend development allows for the delivery of a visually pleasant and user-friendly interface that responds to the needs of the job seeker and employer. HTML is the basic framework of web pages, controlling their layout and content. CSS provides visual appeal by imposing styles, colour schemes, and layouts, thus offering an interactive user experience. Python, being highly versatile and simple to use, provides dynamic functionality for the front end, allowing interactive features and seamless user interactions throughout the site.

Flask was selected as the backend web framework to enhance the frontend technology. Flask offers a robust and scalable solution for server-side logic and data processing. Flask's thin and light layer architecture makes it especially well-suited for web application development on the internet, with optimised routing, request handling, and data processing mechanisms. When integrated with Python properly, Flask facilitates simple application of business logic, user authentication, and data processing, thus ensuring the effective functioning of the "EasyJob" platform.

Furthermore, the project employs GitHub as a version control platform and enhanced collaboration between development teams. This facilitates proper code management, change tracking, and seamless integration of new features. The collaborative tools provided by GitHub, including pull requests and code reviews, enhance team productivity and code quality, thus timely delivery of high-quality software.

The project utilises the Apify API, a robust web scraper and automation tool, in job listing retrieval and data processing. Through the Apify API, the project facilitates the gathering of job data from different websites and job sites, thus creating a massive database for employers and job seekers. The project's main objective is to make employers and job seekers more efficient and effective with the platform through the automation of the job listing retrieval process, thus ensuring the delivery of updated and complete job listings.

The "EasyJob" project is based on the strategic use of tools like Python, Flask, HTML, CSS, GitHub, Apify API, and Jupyter Notebook. Such a base allows the development of a platform that is known for usability, scalability, and high features in favour of job seekers and employers. The use of such tools and technology allows the project to enhance the job search experience, create interactions between job seekers and appropriate opportunities, and empower employers to find appropriate candidates quickly. The success of "EasyJob" as a valuable resource in the job market is attributed to its careful planning, development, and integration phases. The platform facilitates smooth interaction between employers and job seekers, thus enhancing the overall efficiency and effectiveness of the recruitment process.

5. RESULTS AND ANALYSIS

This section presents the derived results in two forms: i) research-based, where the performance of the trained models was evaluated and ii) realtime execution through a website. Futher the discussion is conducted based on the prominent observations of authors. The developed system is available for use on GitHub, thus other researchers can configure it to understand the process.

5.1. Models performance evaluation

The ability to improve the job search experience and productivity is beneficial with the employment of a job suggestion system integrating resume parsing, web scraping, and ML classifiers. The employment recommendation system improves the user experience significantly by offering employment recommendations based on the specific skills and interests of every user. The easy accessibility of suitable employment opportunities leads to a rise in the overall satisfaction levels of the employment seekers, while at the same time helping to identify the right employment opportunities accurately.

In addition, the inclusion of automated procedures in the system, including resume parsing and skill matching, would significantly improve operating efficiency for organisations and job applicants. Automating the candidate-job matching process substantially reduces the time-consuming and cumbersome aspects involved with manual screening and evaluation, thereby accelerating the overall recruitment process. Furthermore, information collected by the job recommendation system, including user activity, reviews, and job application trends, can provide valuable information to employers and job seekers alike. By examining such information, stakeholders can better understand market trends, talent needs, and recruitment tendencies. This allows them to make informed decisions and pursue continuous improvement initiatives.

Finally, the institution of the job recommendation system brings more than convenience benefits. It provides significant advantages, including a richer job search experience, increased operating efficiency, and increased interaction with stakeholders in the labour market. The histogram in Figure 2 shows clusters along the x-axis, labelled 0 through 5, which represent different cluster groups developed using the skill sets obtained from the job descriptions. The y-axis is utilised to represent the frequency of occurrence of each skill set. Cluster 0 is present with a frequency of around 100, cluster 1 with around 10 times, cluster 2 with around 110 times, and cluster 3 with the maximum frequency of around 310 times. Cluster 4 is present around 200 times, and Cluster 5 is present around 190 times. The histogram clearly represents the distribution of data among the different clusters to identify trends or outliers in the dataset.

The evaluation of the job recommendation system's efficacy, performance, and dependability heavily relies on the outcomes derived from its implementation. This section will thoroughly analyse the outcomes produced by various ML classifiers. The accuracy, precision, recall, and F1 score will be assessed and validated using the data supplied in Table 2.

The SVM classifier demonstrates a notable accuracy rate of 98.92%, signifying a substantial proportion of accurate predictions in relation to the overall number of occurrences. The accuracy of the Random Forest classifier is 94.11%, indicating its proficiency in generating precise predictions using the given dataset. The Naive Bayes classifier demonstrates a notable accuracy rate of 90.44%, signifying a substantial degree of accurate predictions derived from the given dataset. The KNN classifier demonstrates a commendable accuracy rate of 83.82%, suggesting a satisfactory degree of accurate predictions obtained from the given dataset.

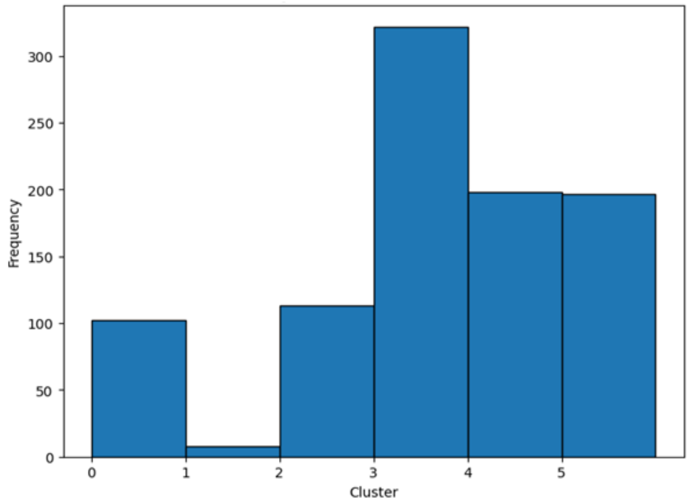

Figure 2. Histogram of job clusters based on the required skillsets

Table 2. Performance evaluation of various classifiers

Evaluation measures	SVM	Random forest	Naive bayes	KNN
Accuracy	98.92	94.11	90.44	83.82
Precision	99.64	94.07	90.84	78.31
Recall	98.92	94.11	90.4	83.82
F1-Score	99.18	93.94	90.5	80.97

5.2. Realtime system testing

A simple, user-friendly interface was designed to assess the performance of the trained models in realtime. A simple web-based platform, EasyJob, was developed to allow users to upload their resumes for a quick test of how well they can recommend relevant job applications. The EasyJob site is a simplified interactive interface that uses the developed intelligent recommendation system.

The project uses a systematic approach:

– Resume Upload: The user has to log on to the EasyJob portal to upload their current resume, shown in Figure 3.
– Keyword Extraction: After uploading the resume, the system extracts the user's skillset. The developed, trained models run in the background and identify relevant skills.
– Job Recommendation: Matching job offers to the skillset of the job seeker from an up-to-date job database built through web scraping. A job name and the "Apply" button can be seen in Figure 4; thus, the user is given a chance to apply directly.

this would mean we have an efficient, automated, intelligent recommendation system for jobs, connecting job seekers with prospective employers. This project is made available on GitHub [25]

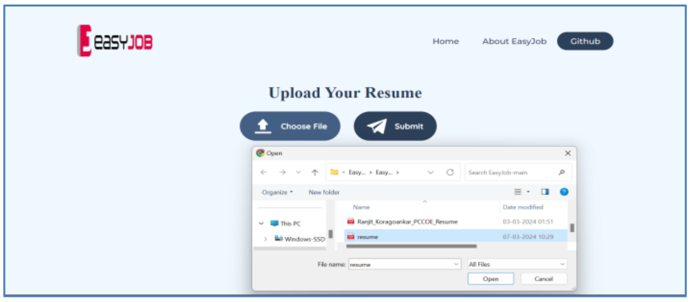

Figure 3. Resume input/upload function

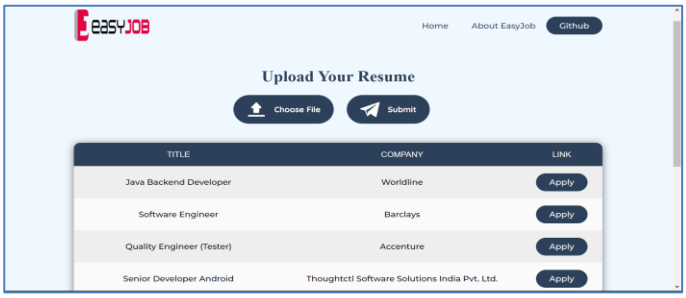

Figure 4. Recommendations generated with developed model

6. FUTURE WORK

The job recommendation system is evolving to improve user experience and cater to the diverse needs of job seekers and employers. Future enhancements include integrating an interactive chatbot for realtime assistance, offering skill development and training resources, and enhancing analytics and reporting capabilities. The chatbot can answer queries, provide guidance on resume writing and interview preparation, and facilitate communication between job seekers and employers. Additionally, the system can offer access to online courses, tutorials, webinars, and certification programs to enhance skill sets and qualifications, increasing competitiveness in the job market. Advanced analytics and insights can help users track progress, identify areas for improvement, and make informed decisions about their job search strategies. In the future, advanced NLP techniques like pre-trained models can be utilised to conduct the contextual analysis of the data. Further performance of the Word2Vec and other (BERT and GPT) models can be compared.

7. CONCLUSION

The contemporary job market poses several hurdles to matching effective candidates with good job opportunities. The conventional methods of resume screening and job matching require great manual work, which can easily suffer from bias since subjective judgments are being made that lead to unflavoured recruitment decisions. This paper proposes the introduction of an intelligent job recommendation system to help semantic analyse candidates' resumes and job descriptions coming from some job portals. The aim of this developed intelligent system will be to help recruit in the process by automatically extracting and evaluating the essential skills required from job descriptions, applying the Word2Vec NLP technique, SVM, Naive Bayes, Random Forest, and KNN Classifiers.

The developed model is capable of recommending the most suitable job openings by computing similarity metrics, like Euclidean distance, between skill clusters in a job advert and a certain candidate's resume. With an excellent accuracy of 98. 92%, SVM was the top performer among the classifiers. In close second place came Random Forest with 94.11% accuracy. Statistics of the results also prove that the system could effectively relate the resume skills of candidates to their job requirements and recommend the right job to users.

In addition to the quantitative measures, the impact of the system spans widely. The ability to get direct access to personalised job recommendations has greatly enriched user experience, leading to a high level of overall satisfaction among job searchers. The database utilised for this research consists of 1,550 job postings collected from three major websites: Indeed, LinkedIn, and Glassdoor. Among the web scraping tools used to collect these records was Apify. The evaluation metrics include accuracy, precision, recall, and F1 score, helping to address the tradeoff between precision and recall.

This is where a recommendation system based on the job could really enhance the user experience, providing relevant employment opportunities according to specified skills and interests and thereby assuring the user's satisfaction in their search for a job. These are some automated procedures such as resume parsing and skill matching to improve performance for both organisations and job applicants. The system collects information and offers valuable input to employers and job seekers in decision-making and continuous improvements.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Hardik Jain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Aparna Joshi				✓	✓					✓		✓		
Deepali Naik	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Rupali Gangarde		✓		✓						✓	✓	✓		
Ranjit Koragoankar		✓	✓			✓	✓	✓	✓					
Yash Khapke	✓		✓	✓		✓	✓	✓	✓		✓			
Varad Kulkarni	✓	✓	✓	✓			✓	✓	✓		✓			

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The job opening and description records were extracted from 3 job portel website using webscrapers.

– Website: https://indeed.com/, Extracted using:
  https://console.apify.com/actors/hMvNSpz3JnHgl5jkh/console
– Website: https://linkedin.com/, Extracted using:
  https://console.apify.com/actors/BHzefUZlZRKWxkTck/console
– Website: https://glassdoor.com/, Extracted using:
  https://console.apify.com/actors/t2FNNV3J6mvckgV2g/console

REFERENCES

[1] Z. Zheng, Y. Sun, X. Song, H. Zhu, and H. Xiong, “Generative learning plan recommendation for employees: a performance-aware reinforcement learning approach,” in *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023*, New York, NY, USA: ACM, Sep. 2023, pp. 443–454. doi: 10.1145/3604915.3608795.

[2] Y. Wang, W. Ma, M. Zhang, Y. Liu, and S. Ma, “A survey on the fairness of recommender systems,” *ACM Transactions on Information Systems*, vol. 41, no. 3, pp. 1–43, Jul. 2023, doi: 10.1145/3547333.

[3] F. A. Riski, N. Selviandro, and M. Adrian, “Implementation of web scraping on job vacancy sites using regular expression method,” in *2022 1st International Conference on Software Engineering and Information Technology, ICoSEIT 2022*, IEEE, Nov. 2022, pp. 204–209. doi: 10.1109/ICoSEIT55604.2022.10029964.

[4] C. Bhatt, A. Bisht, R. Chauhan, A. Vishvakarma, M. Kumar, and S. Sharma, “Web scraping techniques and its applications: a review,” in *Proceedings - 2023 3rd International Conference on Innovative Sustainable Computational Technologies, CISCT 2023*, IEEE, Sep. 2023, pp. 1–8. doi: 10.1109/CISCT57197.2023.10351298.

[5] E. Uzun, “A novel web scraping approach using the additional information obtained from web pages,” *IEEE Access*, vol. 8, pp. 61726–61740, 2020, doi: 10.1109/ACCESS.2020.2984503.

[6] R. Muthyala, S. Wood, Y. Jin, Y. Qin, H. Gao, and A. Rai, “Data-driven job search engine using skills and company attribute filters,” in *IEEE International Conference on Data Mining Workshops, ICDMW*, IEEE, Nov. 2017, pp. 199–206. doi: 10.1109/ICDMW.2017.33.

[7] K. S. Lin, "A case-based reasoning system for interior design using a new cosine similarity retrieval algorithm," *Journal of Information and Telecommunication*, vol. 4, no. 1, pp. 91–104, Jan. 2020, doi: 10.1080/24751839.2019.1700338.
[8] K. Park, J. S. Hong, and W. Kim, "A methodology combining cosine similarity with classifier for text classification," *Applied Artificial Intelligence*, vol. 34, no. 5, pp. 396–411, Apr. 2020, doi: 10.1080/08839514.2020.1723868.
[9] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, "Graph neural networks in recommender systems: a survey," *ACM Computing Surveys*, vol. 55, no. 5, pp. 1–37, May 2023, doi: 10.1145/3535101.
[10] A. Da'u and N. Salim, "Recommendation system based on deep learning methods: a systematic review and new directions," *Artificial Intelligence Review*, vol. 53, no. 4, pp. 2709–2748, Apr. 2020, doi: 10.1007/s10462-019-09744-1.
[11] S. Singh, M. Lohakare, K. Sayar, and S. Sharma, "RecNN: A deep neural network based recommendation system," in *Proceedings - 2021 1st IEEE International Conference on Artificial Intelligence and Machine Vision, AIMV 2021*, IEEE, Sep. 2021, pp. 1–5. doi: 10.1109/AIMV53313.2021.9670990.
[12] F. C. Fernández-Reyes and S. Shinde, "CV retrieval system based on job description matching using hybrid word embeddings," *Computer Speech and Language*, vol. 56, pp. 73–79, Jul. 2019, doi: 10.1016/j.csl.2019.01.003.
[13] S. Chala, S. Harrison, and M. Fathi, "Knowledge extraction from online vacancies for effective job matching," in *Canadian Conference on Electrical and Computer Engineering*, IEEE, Apr. 2017, pp. 1–4. doi: 10.1109/CCECE.2017.7946793.
[14] J. Li, D. Arya, V. Ha-Thuc, and S. Sinha, "How to get them a dream job?: Entity-aware features for personalized job search ranking," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM, Aug. 2016, pp. 501–510. doi: 10.1145/2939672.2939721.
[15] J. Ren *et al.*, "Matching algorithms: fundamentals, applications and challenges," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 3, pp. 332–350, Jun. 2021, doi: 10.1109/TETCI.2021.3067655.
[16] C. Gao *et al.*, "A Survey of graph neural networks for recommender systems: challenges, methods, and directions," *ACM Transactions on Recommender Systems*, vol. 1, no. 1, pp. 1–51, Mar. 2023, doi: 10.1145/3568022.
[17] Apify website, "Apify: full-stack web scraping and data extraction platform." Accessed: May 01, 2024. [Online]. Available: https://apify.com/?fpr=jgcmc&gad_source=1&gclid=CjwKCAjwuJ2xBhA3EiwAMVjkVOzfcbIXoh-krqTQyvSNDUbr8jOpFFNNM-lLlnqnSIGEeSRVvgA0ixoCLYkQAvD_BwE
[18] "Job Search India | Indeed." Accessed: May 01, 2024. [Online]. Available: https://in.indeed.com/
[19] "LinkedIn: Log In or Sign Up." Accessed: May 01, 2024. [Online]. Available: https://www.linkedin.com/home
[20] "PYPDF2 Tutorial - Working with PDF in Python | Nanonets." Accessed: May 01, 2024. [Online]. Available: https://nanonets.com/blog/pypdf2-library-working-with-pdf-files-in-python/
[21] D. A. Bhanage, A. V. Pawar, and K. Kotecha, "IT infrastructure anomaly detection and failure handling: a systematic literature review focusing on datasets, log preprocessing, machine deep learning approaches and automated tool," *IEEE Access*, vol. 9, pp. 156392–156421, 2021, doi: 10.1109/ACCESS.2021.3128283.
[22] D. A. Bhanage, A. V. Pawar, K. Kotecha, and A. Abraham, "failure detection using semantic analysis and attention-based classifier model for IT infrastructure log data," *IEEE Access*, vol. 11, pp. 108178–108197, 2023, doi: 10.1109/ACCESS.2023.3319438.
[23] D. A. Bhanage and A. V. Pawar, "Robust Analysis of IT Infrastructure's log data with BERT language model," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 6, pp. 705–714, 2023, doi: 10.14569/IJACSA.2023.0140675.
[24] R. Koragoankar, V. Kulkarni, and D. Naik, "Search engine using NLP text processing techniques to extract most relevant search results," in *2023 14th International Conference on Computing Communication and Networking Technologies, ICCCNT 2023*, IEEE, Jul. 2023, pp. 1–7. doi: 10.1109/ICCCNT56998.2023.10307392.
[25] Github, "gitranjit / EasyJob," github.com. Accessed: Aug. 11, 2025. [Online]. Available: https://github.com/gitranjit/EasyJob

BIOGRAPHIES OF AUTHORS

Hardik Jain is pursuing his master's in Computer engineering from Indraprastha Institute of Information Technology, Delhi. He has Completed his graduation as a Computer Engineering, at Pimpri Chinchwad College of Engineering, Pune. His research interests include cyber security, malware analysis, machine learning, natural language processing, computer graphics and pattern recognition. He can be contacted at email: hardik24037@iiitd.ac.in

Dr. Aparna Joshi received a Ph.D. degree from Department of Computer Science & Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology Chennai, India. Author working as an Assistant Professor at Department of computer engineering department at PCET's Pimpri Chinchwad College of Engineering, Pune. Authors area of interests is cloud computing. She can be contacted at email: aparna.joshi@pccoepune.org.

Dr. Deepali Naik received master's in computer engineering from Sinhgad Institute of Technology, University of Pune. She has received a Ph.D. from the Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune. She is employed as an Assistant Professor with PCET's Pimpri Chinchwad College of Engineering, Pune. Her research interests include IT infrastructure monitoring, machine learning, deep learning, and natural language processing. She can be contacted at email: Deepali.naik@pccoepune.org.

Dr Rupali Gangarde received master's in computer engineering from University of Pune. She has received a PhD from the Lovely Professional university. She is employed as an Assistant Professor with Symbiosis Institute of Technology, Pune. Her research interests include data privacy, machine learning, deep learning, and natural language processing. She can be contacted at email: Rupali.gangarde@sitpune.edu.in.

Ranjit Koragoankar is working as an Associate Software Engineer, at Dassault Systèmes Global Services. He has Completed his graduation as a Computer Engineering, at Pimpri Chinchwad College of Engineering, Pune. His research interests include machine learning, natural language processing, graphic design, pattern recognition. He can be contacted at email: ranjit.koragoankar20@pccoepune.org.

Yash Khapke is working as an Associate Software Engineer, at Dassault Systèmes Global Services. He has Completed his graduation as a Computer Engineering, at Pimpri Chinchwad College of Engineering, Pune. His research interests include machine learning, natural language processing, graphic design, pattern recognition. He can be contacted at email: yash.khapke20@pccoepune.org

Varad Kulkarni is working as an Associate Software Analyst, at PTC India. He has Completed his graduation as a Computer Engineering, at Pimpri Chinchwad College of Engineering, Pune. His research interests include machine learning, natural language processing, graphic design, and pattern recognition. He can be contacted at email: varad.kulkarni20@pccoepune.org.