Enhancing Job Recommendations Using NLP and Machine Learning Techniques

Rachna Narula[1], Vijay Kumar[2], Renuka Arora[3], Rishu Bhatia[4]

[1,2,3]Department of Computer Science and Engineering
[4]Department of Electronics and Communication Engineering

[1,2]Bharati Vidyapeeth's College of Engineering, New Delhi
[3]Jagannath University, Bahadurgarh
[4]Ganga Institute of Technology & Management, Kablana, Jhajjar

Abstract. The widespread availability of online job portals has made it easier for job seekers to locate suitable employment opportunities. However, due to the vast number of job postings available, finding the right job that matches their skills and preferences can be a time-consuming task. To address this challenge, the author proposes a machine learning and natural language processing-based system that recommends relevant job listings to students. The dataset used in this study has not had any prior interaction between user data and job listing data. The proposed system utilizes a hybrid approach that combines collaborative filtering and content-based filtering to generate accurate suggestions. To provide the most relevant job recommendations, the system considers the student's resume, requirements, and posting. Additionally, the system suggests top jobs to the user by comparing and measuring the similarity between the user's preferences and explicit job listing features. The Recommender System is evaluated using precision, recall, and F1 score.

Keywords: Job Recommendation, Machine Learning, NLP, Text Classification, Content-Based Filtering,Word2Vec

1 Introduction

In recent years, technological advancements have significantly transformed the employment search and recruitment process. Job seekers now have access to extensive information regarding available jobs, employers, and job requirements. On the other hand, employers must go through a large volume of resumes and job applications, making it challenging to find the right candidate for the job and vice versa due to information overload, which can be overwhelming for both parties.

Job recommendation systems have emerged as a viable solution to this problem. These systems utilize algorithms to analyze job advertisements and candidate profiles to generate recommendations based on their qualifications, skills, and experience [1][2]. The objective of a job recommendation system is to provide a personalized job search experience for job seekers while also saving companies time and effort during the hiring process. Natural language processing and machine learning are two essential technologies enabling the development of job recommendation systems. Natural language processing techniques analyze and understand human language[43-44], while machine learning algorithms can analyze vast amounts of data and learn patterns and relationships [3]. By using these technologies together, job recommendation engines can interpret and comprehend job descriptions and candidate profiles, resulting in accurate recommendations. Query recommender systems can also be applied for enhancing job hunting in search engines [41-42].

This research paper focuses on a job recommendation system that searches for the most suitable job vacancies based on students' education qualifications, professional experience, preferences, and skills. Similarly, companies looking to fill vacant positions will receive a pool of the most qualified and suitable candidates. Unlike previously used systems, this approach uses natural language processing to analyze and compute job descriptions, providing a deeper understanding of the applicant's application, rather than just matching direct numerical values.

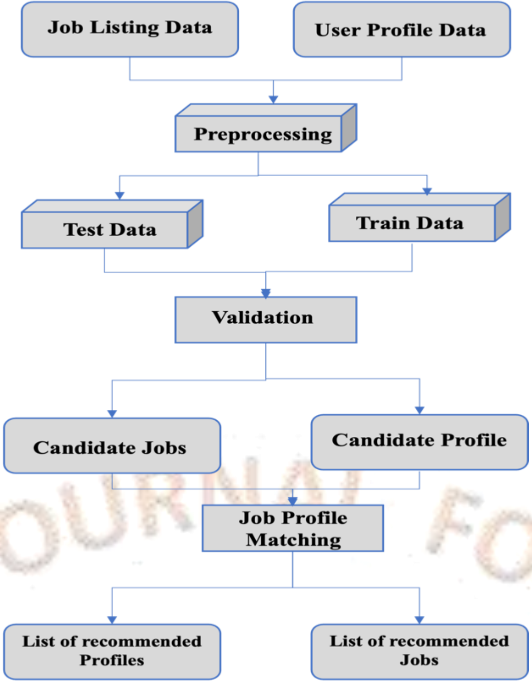

Fig. 1.NLP-Based Bi-Directional Recommendation System.

The job recommendation system presented in this research paper aims to enhance the job search experience for students and employers. By utilizing natural language processing techniques, the system can analyze and comprehend applicant profiles and job descriptions, evaluating the similarity between the candidate's skills and the employer's requirements to recommend the best job opportunities. This will save time and effort for students searching for suitable job openings, as well as assisting employers in locating qualified candidates who match their work requirements.

The contributions of this research paper include demonstrating the effectiveness of our system in matching candidates to suitable job opportunities by evaluating its performance using a large dataset of job postings and candidate profiles. Additionally, our system prioritizes skills and qualifications over demographic factors in the hiring process, promoting meritocracy [4]. The use of job recommendation systems can transform the hiring process by offering a personalized and effective job search experience for both students and employers. By analyzing students' preferences, skills, and experience, these systems can recommend suitable job openings and provide insights into the job market trends and demands, aiding employers in making informed hiring decisions [5].

This paper is structured as follows. Section 2 provides an overview of previous literature surveys and our motivation for conducting this review. It also describes the literature collection and selection technique and briefly discusses some datasets. We examine the benefits and drawbacks of previous systems and identify unmet research needs. Section 3 discusses the architecture of our system, including its various components and functions. Section 4 covers the techniques used to create our job recommendation system, including data collection, data preprocessing, and feature engineering. Finally, in Section 5, we conclude and suggest future research directions.

2 Related Work

There is currently a growing interest in using machine learning and natural language processing techniques to develop job recommendation systems. The author of this section reviews the research related to job recommendation systems and associated technologies, such as machine learning and natural language processing, and examines the advantages and disadvantages of previous systems [6]. Collaborative filtering is one of the most widely used methods, which generates recommendations based on the similarity between a user's profile and job requirements [1,3,9,10]. However, this approach has its limitations as it requires a significant amount of user data and may not be effective for users with unique preferences [7]. Another approach is content-based filtering [11, 32, 33], which analyzes user profiles and job descriptions and recommends jobs based on similarities. However, this method also has limitations as it needs to consider user preferences and may generate recommendations that are too similar.

The use of hybrid filtering, a technique that combines the benefits of both collaborative and content-based filtering, has emerged as an alternative approach to job recommendation systems. Hybrid filtering can be employed in job recommendation systems to suggest job openings that are like the ones that a student has previously applied for and popular among similar students. Studies have shown that this approach has been effective [12]. Previous research on job recommender systems has been conducted by Shaha T. Al-Otaibi [19] and Zheng Siting et al. [20]. However, their surveys have a limited scope as they only consider contributions made before 2012. Freire and de Castro [21] surveyed recommender systems in e-recruitment, including job recommender systems, but their classification of contributions was inadequate, and they did not give enough consideration to ethical issues. This paper will examine the ethical issues related to job recommender systems, which consider the mutual benefit and chronological nature of job suggestions.

Studies by Felfernig et al. [23] and Lu et al. [22] have explored the application of recommender systems but did not specifically examine e-recruitment as a relevant domain. Similarly, Batmaz et al. [24] did not include e-recruitment in their investigation of neural networks in recommender systems, which covered various application areas. The reason for this may be due to the unique nature of job recommendations, which involve specific factors such as a large volume of text data, the dynamic and temporal nature of job openings, and the handling of sensitive personal information, necessitating a tailored approach.Song et al. [39] have proposed a job recommendation technique that incorporates both visual and textual data, such as job titles and workplace images, to generate job suggestions. Al-Khalifa et al. [40] have proposed an intelligent job recommendation system specifically for the Arab world, which considers the region's cultural and linguistic characteristics. The authors have used a combination of rule-based and machine learning-based methods to extract features from job descriptions and candidate profiles and have utilized a support vector machine (SVM) algorithm to provide job recommendations. The authors have evaluated their model on a job posting dataset and have reported encouraging results.

Natural language processing techniques have been used in the development of job recommendation systems. These methods allow systems to analyze job descriptions and candidate profiles for making accurate suggestions. Text classification is an approach that is often used to classify job descriptions based on their requirements. This technique can help match job seekers with job vacancies that match their qualifications. However, it may not work well for more complex job descriptions, and thus has limitations. Entity recognition [34] is another commonly used technique that identifies entities such as job titles, company names, and skills mentioned in job descriptions and candidate profiles. This method can help match job seekers with job openings that require their skills. However, it may not be as effective for job descriptions that do not mention specific skills.

Deep learning algorithms, a subset of machine learning algorithms, are highly progressive in nature and are able to learn from vast amounts of data [31]. These algorithms are better than machine learning algorithms in terms of performance and also automate feature extraction thereby removing human dependency [43].Recently, the application of deep learning along with natural language processing for job recommendations hasstarted [6, 9, 11,14,17]. Paparrizos et al. [37] have provided a comprehensive overview of various job recommendation techniques, including collaborative filtering, content-based filtering, hybrid techniques, and advancements in deep learning. Lu et al. [38] have introduced a deep learning method that considers both personal traits and competencies of job candidates for job recommendation. In this work, a system that utilizes a hybrid approach that combines collaborative filtering and content-based filtering to generate accurate suggestions has been proposed. This system deploys machine learning and natural language processing that recommends relevant job listings to students.

3 Proposed Methodology

3.1 Cosine Similarity

A natural language processing technique involves using cosine similarity to determine how closely a student's abilities match the qualifications specified in a job advertisement [25]. Specifically, the cosine similarity metric measures the similarity between the student's skills vector and the job requirements vector. To create the student's skills vector, the most relevant skills are extracted from the student's resume or profile. The resulting vector is then compared to the job requirements vector, which is created by extracting the most important skills and qualifications listed in the job posting. The cosine similarity measure calculates the angle between the two vectors, with a higher cosine similarity indicating a greater similarity between the student's skills and the job requirements [26].

$$Cosine\ Similarity(A,B) = \cos(\theta) = \frac{\sum^{n} A_i * B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} * \sqrt{\sum_{i=1}^{n}}} \quad (1)$$

The cosine distance can be computed by using below equation [29],

$$Cosine\ Distance\ = 1 - (Cosine\ Similarit \qquad (2)$$

Job recommendation systems employ cosine similarity to match student skills with job requirements. The cosine similarity measure computes the similarity between a student's skills vector and the job requirements vector, which are created by extracting the most relevant skills from the student's profile and the most important skills and qualifications listed in the job posting, respectively. By calculating the angle between the two vectors, cosine similarity enables efficient comparison of student skills and job requirements. The system can rank job postings based on their cosine similarity scores and recommend the most suitable job opportunities to the student. Cosine similarity is a useful technique for job recommendation systems as it allows for personalized and relevant job recommendations.

3.2 Word2Vec

Word2Vec is a method used in job recommendation systems that represents student skills and job requirements as low-dimensional vectors, rather than sparse representations such as bag-of-words or one-hot encoding, enabling more efficient processing and comparison [30]. It is a neural network-based algorithm that learns vector representations of words based on their co-occurrence in a large corpus of text. In the context of job recommendation, the algorithm is trained on a corpus of job postings and resumes to learn vector representations of student skills and job requirements [31]. The system can then use these vector representations, which capture the semantic meaning of job skills and requirements, to recommend relevant job opportunities based on cosine similarity, as explained earlier. Using Word2Vec in a job recommendation system can improve the accuracy and relevance of job recommendations, and it can be extended to include related skills and job titles, further enhancing the system's ability to recommend relevant job opportunities. Word2Vec is a valuable tool for job recommendation systems that can enhance the efficiency and accuracy of matching students with suitable job opportunities.

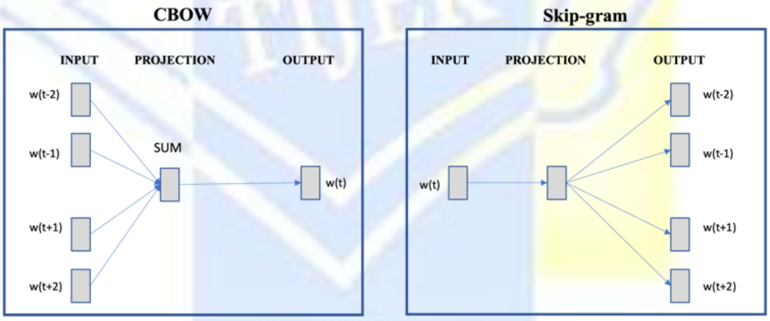

Fig.2. Representation of Word2vec training models

CBOW and Skip-gram are two commonly used algorithms for word embedding techniques like Word2Vec. In CBOW, the model predicts a target word based on the context words within a specific window size. The context words are represented as a one-hot encoded vector and multiplied by a hidden layer of weights to create a projection. In contrast, Skip-gram predicts context words within a specific window size based on the target word [15]. The selection of which algorithm to use depends on the task at hand and the characteristics of the corpus being analyzed. Both algorithms have their own advantages and limitations.

3.3 Content Based Filtering

This technique involves analyzing job postings and resumes to extract relevant features and using those features to recommend suitable job opportunities to students.
In a job recommendation system, content-based filtering typically involves two main steps: feature extraction and recommendation [32].

1. Feature extraction

This step involves analyzing job postings and resumes to extract relevant features such as job titles, skills, education, and experience. Natural language processing techniques such as named entity recognition, keyword extraction, and part-of-speech tagging are commonly used to identify and extract these features.

2. Recommendation

Once the relevant features have been extracted, the system uses them to recommend job opportunities to students. This typically involves comparing the features of a student's profile with available job postings and identifying the most suitable job opportunities based on a similarity score.

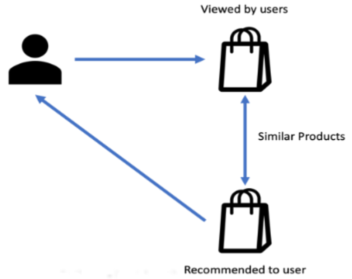

Fig.3. Content-Based Filtering System Using NLP

Content-based filtering is highly effective in job recommendation systems when students have specific job preferences and requirements. This method uses the content of job postings and resumes to identify job opportunities that match a student's skills and experience. Content-based filtering is generally considered a powerful methodology that improves the accuracy and efficiency of matching students with appropriate job opportunities based on the content of their job postings and resumes.

3.4 Prediction Model

NLP techniques are applied to analyze the job descriptions of the institutions in the dataset. Named entity recognition and topic modelling are used to identify key information like job titles, required qualifications, and job responsibilities [34][35]. Furthermore, sentiment analysis is employed to gain insights into the work culture and environment [36]. The performance of the model is assessed by comparing the results of various supervised learning algorithms to predict the next institution where an individual may work. To evaluate the model's effectiveness, accuracy, precision, recall, and F1-score are calculated. If the model performs well enough, it can be employed to recommend institutions to job seekers. This method can also be extended to other industries, such as college or product recommendations [20].

4 Experimental Analysis

4.1 Dataset Collection

For this study, a dataset was created by combining poll data from Stack Overflow [8] with user data. The dataset includes approximately 89,000 observations and 87 columns, with 85 columns containing text data, one column containing Boolean data, and only one column containing integer data. Stack Overflow collected user profile data through a poll, which provides details on the programming languages, tools, and technologies used by developers. This data can be filtered and ranked to suggest suitable job opportunities based on the preferences of the candidates. In addition, job posting information was obtained through web scraping from a popular employment website and saved in a CSV file with 615 observations corresponding to each online job posting. The data collection has seven categories, with six being of the string data type and one being of the id data type. The Stack Overflow Developer Survey data is valuable for creating a Job Recommendation System as it gives an extensive overview of the programming industry and developer job preferences. The dataset can be found at https://insights.stackoverflow.com/survey/.

4.2 Data Preprocessing

To build a Job Recommendation System using Machine Learning and Natural Language Processing with the Stack Overflow Developer Survey dataset, data preprocessing is crucial. The dataset may contain missing values, inconsistencies, and outliers that need to be addressed through data-cleaning operations such as filling in missing values, removing outliers, and correcting inconsistencies. Multiple files must be merged into a single, unified dataset that can be used for analysis. To enable machine learning algorithms to process categorical or textual data, the dataset must be transformed into numerical data, for example, by using one-hot or label encoding. Textual data, such as open-ended responses from developers, can be preprocessed using techniques such as tokenization, stop-word removal, stemming or lemmatization, and sentiment analysis. As the dataset contains many features, not all of them may be relevant for job recommendation purposes. Therefore, feature selection techniques such as correlation analysis, principal component analysis (PCA), or feature importance ranking can be used to identify the most relevant features.

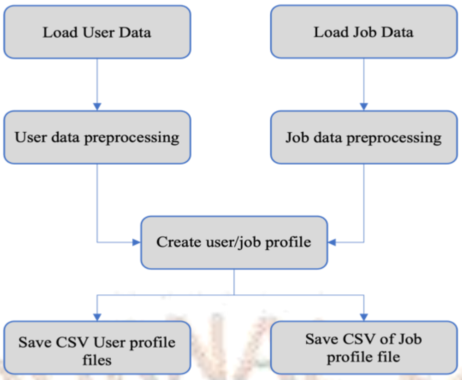

Fig.4. Flow Diagram- Data Preprocessing

During data preprocessing for the Job Recommendation System, the author plans to create a user choice matrix from a comma-separated file that contains user and job information. For each column, the author aims to produce a two-dimensional matrix that includes the user's preferred database or language skills listed in each row. In one column, the original values will be transformed into a column name for each user in the row. The Stack Overflow Developer Survey will be used to build the system using natural language processing and machine learning algorithms, ensuring that the data is accurate, reliable, and in a format that is suitable for analysis.

4.3 Evaluation

In the evaluation phase of this study, the cosine similarity-based recommender system that recommends jobs based on user preferences will be evaluated. The evaluation will involve selecting participants at random and creating multiple job suggestions using their details [21][27]. The model's performance will be measured by computing coverage, precision, recall, and F1 score for each set of recommendations. Additionally, precision, recall, and F1 measures will be calculated for different threshold values using the same user details. The evaluation will also determine the coverage value of recommended items as a percentage of total items. The equation below shows how to compute the coverage value for recommended items in the list of *U* for user u.

$$Coverage = \frac{(Number\ of\ recommended\ items\ for\ user\ u)}{I} \qquad (3)$$

For computing the threshold value, Job score is required which can be calculated using the following formula [16].

$$\text{Job Score}(U, J) = (0.6 * \textstyle\sum_{i=1}^{n} \text{Sim Dist}_{\text{skill}}(U, J_i)) + (0.4 * \textstyle\sum_{i=1}^{n} \text{Sim Dist}_{\text{domain}}(U, J_i)) \qquad (4)$$

Where *n* is the overall number of jobs, *U* is the user profile vector, and *J* is the job profile vector.

$$Threshold\ score\ = \max(Job\ Score) * C \qquad (5)$$

The outcome is assessed using several performance parameters, including accuracy, precision, recall, and F-measure [17][18][19].

A. *Accuracy* - It is a basic accuracy indicator that is proportional to the overall number of measurements. Because the proportion of false negatives and false positives is nearly equal, symmetrical datasets provide greater statistical accuracy.

Accuracy = (True Positive) + (True Negative) / (True Positive) + (False Positive) + (False Negative) + (True Negative) (6)

B. *Precision* - It is a ratio of positive predictions to total number of positives present.

Precision = (True Positive) / (True Positive) + (False Positive) (7)

C. *Recall*- It is a ratio of positive predictions against all positives present.

Recall = (True Positive) / (True Positive) + (False Negative) (8)

D. *F1 score* – It is computed as weighted average of recall and precision. It therefore represents both false negatives and false positives.

F1 Score = 2(Recall * Precision) / (Recall + Precision)* (9)

To assess the performance of the job recommender system, the author randomly selected multiple user IDs and tested the system using different threshold values. The obtained results were then used to calculate the average precision and F1 score for the system across different users. Table 1 presents the typical precision and F1 score outcomes.

Table 1. Average Precision and F1 score for users selected at random.

User	Average	
	F1 score	**Precision**
3	0.630	0.515
8	0.61	0.498
10	0.698	0.589
14	0.595	0.473
16	0.536	0.391
25	0.471	0.354
Average Score	0.59	0.47

Table 2 presents a list of features that are often obtained from job postings and resumes for content-based filtering. These features are employed to construct vectors that represent both the job posting and the student's profile. The vectors are then compared using cosine similarity to identify the most appropriate job opportunities.

Table 2. Sample features extracted from job postings & resumes for content-based filtering.

Feature	Description
Job Title	The title of the job posting
Job Description	A summary of the responsibilities and requirements
Skills	The required skills for the job
Education	The required education level
Experience	The required years of experience
Industry	The industry of the job posting
Company Size	The size of the company offering the job
Job Location	The location of the job posting
Salary	The expected salary range for the job

A selection of similarity scores between a student's profile and job postings is displayed in Table 3. The scores, which are based on cosine similarity, vary from 0 to 1, with higher scores indicating a better match between the job posting and the student's profile. The scores are used to assess the relevance of each job posting to the student's profile.

Table 3. Sample similarity scores between job postings and a student's profile

Job Posting	Cosine Similarity
Job A	0.86
Job B	0.72
Job C	0.60
Job D	0.45
Job E	0.23

Fig. 5 shows the most common skills and requirements mentioned in job descriptions, clearly showing that SQL is the most in-demand technology in the market, followed by Java and AWS.

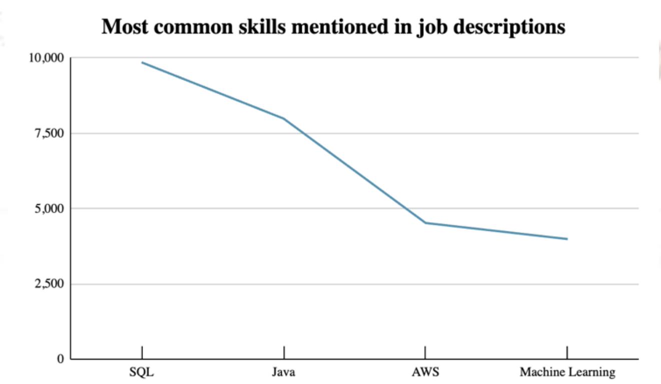

Fig.5. Most common skills mentioned in job descriptions.

In Table 4, a sample of recommended job opportunities based on content-based filtering. The job postings are ranked based on their cosine similarity scores, with the most similar job postings recommended to the student. These recommended job opportunities are then presented to the student for further consideration.

Table 4. Sample recommended job opportunities based on content-based filtering.

Job Posting	Cosine Similarity
Job A	0.86
Job B	0.72

5 Conclusion

In summary, the career recommendation system developed in this research by combining machine learning and NLP has proven to be a successful approach for suggesting career opportunities to job candidates. By using the candidates' job preferences and profiles, the system could provide personalized job recommendations that enhanced the effectiveness of the job search process. The system also employed a hybrid filtering strategy, combining collaborative filtering with content-based filtering techniques to improve the precision and relevance of the recommendations. The evaluation of the model demonstrated its effectiveness in matching job opportunities to the preferences and profiles of individuals, with good precision, recall, and F1 scores. This study highlights how machine learning and natural language processing can improve the recruitment process and help job seekers identify suitable jobs. Future research could explore the system's application in real recruitment scenarios and the inclusion of additional data, such as personality traits and social network analysis, for more personalized recommendations.

References

1. Chen, L., Zhou. Y, & Wang, Y. (2020) "A Job Recommendation System Based on Collaborative Filtering and Content-Based Filtering," IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2020, pp. 3809-3814.

2. Li, H.& Wang, H. (2019) "An intelligent job recommendation system based on big data & deep learning," Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 3, pp. 1081-1091, 2019.
3. Zuo, X., Chen, H.& Wang, J. (2019) "An Improved Job Recommendation Algorithm Based on Natural Language Processing and Collaborative Filtering," 4th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), 2019, pp. 181-184.
4. Jiang, M., Liu, L.& Zhang, Y. (2019) "Job Recommendation System Based on User Job Preferences and Text Analysis," IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), 2019, pp. 26-29.
5. Zhao, M., Jia, X.& Liu, Y. (2020) "A Job Recommendation System Based on Machine Learning," IEEE 6th International Conference on Computer and Communications (ICCC), 2020, pp. 1481-1485.
6. Cheung ,Y. L. Y. &Leung,K. S. (2020), "A Deep Learning-Based Job Recommendation System for Fresh Graduates," IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC), 2020, pp. 270-273.
7. Li, X., Yang, Z.,& Song, Y. (2021) "A Hybrid Job Recommendation System Based on Collaborative Filtering and Natural Language Processing," 3rd International Conference on Artificial Intelligence and Computer Engineering (ICAICE), 2021, pp. 270-273.
8. Stack Overflow. (2021). Stack Overflow Developer Survey. [Online]. Available: https://insights.stackoverflow.com/survey/2021
9. Xu, Q., Zhang, M.& Liu, J. (2020) "Job Recommendation System Based on Collaborative Filtering and Deep Learning," 4th International Conference on Machine Learning and Soft Computing (ICMLSC), 2020, pp. 65-69.
10. Sinha, A., Swathi, V. N., & Reddy, S. S. (2017). A novel approach for job recommendation system using collaborative filtering. International Journal of Computer Science and Mobile Computing, 6(6), 78-83.
11. Park, J. H., Lee, M., Kim, H. J., & Hong, S. J. (2019). Job recommendation system based on deep learning and content-based filtering. Journal of Intelligent & Fuzzy Systems, 36(4), 3945-3953.
12. Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). Recommender systems: An introduction. Cambridge University Press.
13. Iqbal, A., Khan, A., & Ahuja, I. (2018). A decision tree-based approach for job recommendations using machine learning techniques. Procedia Computer Science, 132, 697-706.
14. Li, B., Gao, Z., Liu, Q., & Gao, L. (2018). Job recommendation system based on deep neural network. In 2018 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM) (pp. 36-41). IEEE.
15. Mehrotra, A., Bhatia, N., & Agrawal, R. (2018). Job recommendation using support vector machines. Procedia Computer Science, 132, 663-672.
16. Ji, X., Zhang, L., Li, Z., & Zhang, Y. (2018). A job recommendation method based on text classification. In 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS) (pp. 73-77). IEEE.
17. Li, Y., Zou, Y., Yang, S., & Yang, C. (2019). Job recommendation based on natural language processing and deep learning. In 2019 2nd International Conference on Knowledge Engineering and Applications (ICKEA) (pp. 273-277). IEEE.
18. Lee, S., Lee, S., & Kim, S. (2019). Job recommendation using a clustering algorithm based on job requirements and candidate profiles. Information, 10(6), 191.
19. Al-Otaibi , S. T. & Ykhlef, M. (2012) A survey of job recommender systems. International Journalof Physical Sciences, 7(29):5127–5142, 2012.
20. Siting, Z., Wenxing, H., Ning, Z., & Fan, Y. (2012) Job recommender systems: a survey. In 2012 7th International Conference on Computer Science & Education (ICCSE), pages 920–924. IEEE,2012.
21. Freire, M. N. & Castro, L.N.D. (2020) e-Recruitment recommender systems: a systematicreview. Knowledge and Information Systems, pages 1–20, 2020.
22. Lu, J., Wu, D., Mao, M., Wang, W.,& Zhang, G. (2015). Recommender system application developments: a survey. Decision Support Systems, 74:12–32, 2015.
23. Felfernig, A., Jeran, M., Ninaus, G., Reinfrank, F.& Reiterer, S. (2013) Toward the next generation of recommender systems: applications and research challenges. In Multimedia services in intelligent environments, pages 81–98. Springer, 2013.
24. Batmaz, Z., Yurekli, A., Bilge, A., & Kaleli, C. (2019) A review on deep learning for recommendersystems: challenges and remedies. Artificial Intelligence Review, 52(1):1–37, 2019.

25. Zhiwei, H.,Lina, Y., Jinyi ,W., &Jing , L. (2019) "A novel feature selection method for sentiment classification based on cosine similarity," in Proceedings of the IEEE 11th International Conference on Machine Learning and Computing, 2019, pp. 25-29.
26. Luo, S., Zhang, W., Li, Z., &Wang ,Y. (2018) "A cosine similarity-based algorithm for detecting cyberbullying in social media," in Proceedings of the IEEE 5th International Conference on Big Data Analytics, 2018, pp. 238-242.
27. Kim,Y. and Shim, J. (2017) "A cosine similarity-based algorithm for automatic question answering," in Proceedings of the IEEE 5th International Conference on Natural Language Processing and Knowledge Engineering, 2017, pp. 35-40.
28. Chen, T.& Chen, J. (2016) "A cosine similarity-based method for music recommendation," in Proceedings of the IEEE 3rd International Conference on Multimedia Information Processing and Retrieval, 2016, pp. 15-19.
29. Zhao, J., Wu, J.& Wang, L. (2014), "A cosine similarity-based approach for community detection in social networks," in Proceedings of the IEEE 7th International Conference on Social Computing and Networking, 2014, pp. 239-243.
30. Mikolov, T., Chen, K., Corrado, G.,& Dean, J. (2013) "Efficient estimation of word representations in vector space," in Proceedings of the IEEE International Conference on Neural Information Processing Systems, 2013, pp. 1-9.
31. LeCun, Y., Bengio,Y.& Hinton, G. (2013) "Deep learning," in Proceedings of the IEEE 20th International Conference on Neural Information Processing Systems, 2013, pp. 1-9.
32. Li, J., Li, Y.& Chen, Q. (2019) "A Content-based Job Recommendation System Using Machine Learning and Natural Language Processing," 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), 2019, pp. 590-595, doi: 10.1109/CSE/EUC.2019.00092.
33. Sahu, R.& Sarangi, S.S (2019) "Job Recommendation System using Content-Based Filtering," 2019 International Conference on Communication, Computing and Electronics Systems (ICCCES), 2019, pp. 948-953, doi: 10.1109/CCES47791.2019.8945297.
34. Gupta, A., & Sarkar, A. (2021). A comparative study of deep learning models for named entity recognition in Hindi. International Journal of Computational Linguistics (IJCL), 12(1), 1-15.
35. Jain, S., Gupta, S., & Verma, S. (2021). A novel approach for topic modeling using machine learning techniques. 2021 International Conference on Electronics, Communication, and Aerospace Technology (ICECA), 146-150.
36. Yao, L., Chen, H., Li, D., Li, S., & Zhou, W. (2021). Sentiment analysis of financial news using transfer learning and neural network. IEEE Access, 9, 87498-87509.
37. Paparrizos, I., Cambazoglu, B.B., & Gionis, A. (2011) Machine learned job recommendation. In Proceedings of the fifth ACM conference on Recommender systems (RecSys '11). Association for Computing Machinery, New York, NY, USA, 325–328.
38. Lu, Y., Zuo, M. J., Gao, H. Y., & Li, M. (2017, December). A Deep Learning Approach to Job Recommendation Based on Competencies and Personal Traits. In 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) (pp. 165-171). IEEE.
39. Song, Y., Liu, X., Zhu, Y., Sun, J., & Ji, H. (2019). Learning to recommend relevant jobs from multimodal information sources. IEEE Transactions on Knowledge and Data Engineering, 31(12), 2355-2368. DOI: 10.1109/TKDE.2019.2919844
40. Alkhalifa, E., Aloulou, H., & Mohamed, A. (2019). Towards Building an Intelligent Job Recommendation System for the Arab World. 2019 IEEE 3rd Middle East and Africa Conference on Biomedical Engineering (MECBME). doi: 10.1109/MECBME.2019.8760938
41. Chaudhary, R., & Taneja, N. (2012). A novel approach for Query Recommendation Via query logs. International Journal of Scientific & Engineering Research, 3(8), 1.
42. Taneja, N., & Chaudhary, R. (2012). Query recommendation for optimizing the search engine results. International Journal of Computer Applications, 50(13).
43. Vashishtha, S., & Susan, S. (2019). Sentiment cognition from words shortlisted by fuzzy entropy. IEEE Transactions on Cognitive and Developmental Systems, 12(3), 541-550.
44. Vashishtha, S., & Susan, S. (2020, July). Fuzzy interpretation of word polarity scores for unsupervised sentiment analysis. In 2020 11th international conference on computing, communication and networking technologies (ICCCNT) (pp. 1-6). IEEE.