BioGecko
Vol 12 Issue 03 2023
ISSN NO: 2230-5807

A GRAPH-BASED APPROACH FOR PERSONALIZED JOB RECOMMENDATION

B.Subbulakshmi[1], M.Nirmaladevi[2], A.M.Rajeswari[3] and M.Sivakumar[4]
[1] Assistant Professor, Department of Computer Science and Engineering, Thiagarajar College of Engineering, Madurai, India.
bscse@tce.edu
[2] Assistant Professor, Department of Computer Science and Engineering, Thiagarajar College of Engineering, Madurai, India.
nirmaladevi2004@gmail.com
[3] Associate Professor, Department of Computer Science and Engineering, Velammal College of Engineering and Technology, Madurai, India
amrtce@gmail.com
[4] Associate Professor, Department of Computer Science and Engineering, Thiagarajar College of Engineering, Madurai, India.
mskcse@tce.edu

Abstract
In today's society, finding a job is more challenging than getting one. On-campus placements make it easier for students to access work at the best universities. But when students go into mainstream colleges, they face challenges. These students have to go through a plethora of web pages, which is a time-consuming process. Thus, a graph based recommendation engine on user competence and location is built. The proposed work facilitates the referral process so that students can obtain information about jobs that interest them with just a click. This approach aims to address gaps in previous literature, such as cold start issues, safety concerns, and scalability, in order to provide a more effective recommendation system. As a result, by providing a career that matches their interests, the recommendation system can significantly aid recent graduates and job seekers in realising their goals.

Keywords: Natural Language Processing, graph-based recommendation, similarity matrix.

1. Introduction

There is a huge group of skilled workers in India that is growing the fastest. With the internet, a plethora of opportunities is available. People spend the majority of their time looking for employment, while working employees who seek to change careers may not have enough opportunity to locate the perfect job. There is indeed a plethora of info on numerous websites. However, for beginners or job searchers, this activity is highly arduous because they must browse through dozens of websites to explore the ideal employment. Nowadays, recommendation systems are more popular. However, depending on the domain in which it is used, the type of recommendations supplied may differ.

Customized employment recommendations are more applicable as part of the employment referral system. A system which recommends jobs to users based on their past research has been designed to help job seekers find their ideal place of work. There are plenty of job search sites that present jobs that have been listed by recruiters. Numerous job search websites exist, such as Indeed, Internshala, LinkedIn and others. These sites may be used to search for jobs and apply with the help of the given links. These websites ask for personal information as well as educational background. These websites do include a filter option that can save time. However, the major issue is that recruiters can only post to a single website, which is not always available. Therefore, a fantastic opportunity for the users is lost. As well, searching all available websites takes a long time. Disclosure of personal information on several websites is risky too. Data leaks are very widespread nowadays. Based on the job recommendation system, several articles have been proposed.

2. Related Work

Author [1] proposed a referral system in which graduates are divided into groups depending on their academic performance and family economic circumstances. In addition, graduates' similarity scores and customized preferences are merged to recommend some suitable occupations. The disadvantage of this work

[1] is academic achievement and family background does not appear to be excellent features in grouping graduates. Miao Jiang et al., [2] proposed a referral system through which job applicants subscribe to mail alerts for new job postings that fit their professional interests. In this referral system, the set of features that reflect the click-through behavior of individual applicants was considered to group users. The drawback here is the gray sheep problem which means that there are always people whose taste does not correspond to anyone.

The various employment suggestion systems and their methods were analysed by Ravita Mishra and Sheetal Rathi [3]. Additionally, they highlighted the drawbacks of each recommendation system, including scalability, security, and cold start issues.

Qingyu Guo et al., [4] proposed a system based on interactions within social networks, and a knowledge graph is constructed from there. The downside here is that the knowledge chart is user-based and the scalability issue will arise. A hybrid referral system that uses a web crawler to obfuscate data was proposed by Zhenqi Dong et al.,[5]. Additionally, they built a group-based algorithm for content filtering and recommendations. The drawback in this situation is that web crawlers will scan and retrieve all the data from a website, whereas web scraping has a much more focused approach and objective. Ravita Mishra and Sheetal Rathi [10] proposed a recommendation system that uses a Deep Semantic Structure Modelling (DSSM) system that uses the semantic representation of sparse data and represents the job description and skill entities in character trigram format, increasing the system's efficacy. Here the DSSM system is a combination of CNN, RNN and GNN. The drawback is that since it is a combination of three neural networks, it is more complicated which may lead to more memory and time.

3. Existing Methods

3.1 Job Search Websites

Jobs placed by recruiters are listed on job search websites. There are many websites for job searching, including Linkedin, Internshala, and Indeed. By using the offered links, one can use these websites to look for employment and apply for it. We are required to enter details about our academic record and personal lives on these websites. Additionally, they feature a number of filter options that can be utilised to speed up search results. Despite its positive attributes, it is not always appropriate.

For example in LinkedIn, a new account was created and only user skills and location were added. When the jobs page opens, it is seen that there is no recommendation. From this point, one can search for the required jobs but no recommendation was provided. This is a major problem called the cold start problem. When the user gets some history, this system gives good recommendations.

In Indeed, a new user is created as usual, and only skills are added. Then the preferred location was selected. Based on these data, recommendations are suggested, but the recommendations provided were not up to the specific requirements as given by the user. In Internshala website, only preferred job titles can be selected in Fields of interest. This site doesn’t have a great recommendation engine.

3.1.1 Drawbacks of Job Search Websites

- Hiring managers may not be able to publish on all job-searching platforms. Certain individuals may miss out on a wonderful offer, and researching all available websites is a time-consuming approach.
- It's risky to disclose all of your personal information on multiple websites. Data leaks are very wide spread nowadays.

3.2 Recommendation Engines

A job recommendation engine is a data filtering tool that employs machine learning algorithms to recommend the most relevant jobs to a particular user. Various algorithms with merits and downsides have been proposed in the literature.

3.2.1 Drawbacks of Recommendation Engines

- Cold Start problem: When a new item or user is added there won't be enough information for a recommendation.
- Security issue: User information is given which is not safe
- Scalability issue: The system is not scalable because it can’t handle when the number of users or jobs goes over a limit.

4. Proposed Approach

To recommend jobs to the job seekers based on different parameters like job details which includes job description, required skills, job location and user data such as user skills, user-preferred location. In the proposed approach, memory-based content filtering recommendation algorithm is used. This algorithm requires the new characteristic or description of an item and user characteristics to recommend.

Collaborative filtering is not suitable for our approach as it recommends solutions based on other users' likes and dislikes. The job recommendation system is not dependent on other users' recommendations. The users have their own likes and dislikes. The gray sheep issue is a well-known issue that is occurring in this technique. The issue arises for a new unique user who does not fall into any set or group of like-minded users.

The proposed approach has the following objectives:

1. To scrape data from online to create an offline job dataset.
2. To construct a recommender model that can address the cold start issues and scalability issues.
3. To design an algorithm that is well suited for recommending jobs to job seekers based on skills, location and job description.

Figure 1 demonstrates the design of the proposed system. There are four layers to the system: an input layer, a database layer, a recommendation layer, and a user interface layer. The input layer gathers all pertinent data, including user information and job openings, from various websites using a web scraper. The data is saved in a secure and easy-to-recover container by the database layer, which is the second layer. At the recommendation layer, a recommendation engine is developed with all of this data to generate top recommendations. The UI layer is a user interface with three screens for listing recommendations, searching for relevant jobs, and seeing a job's detailed description.

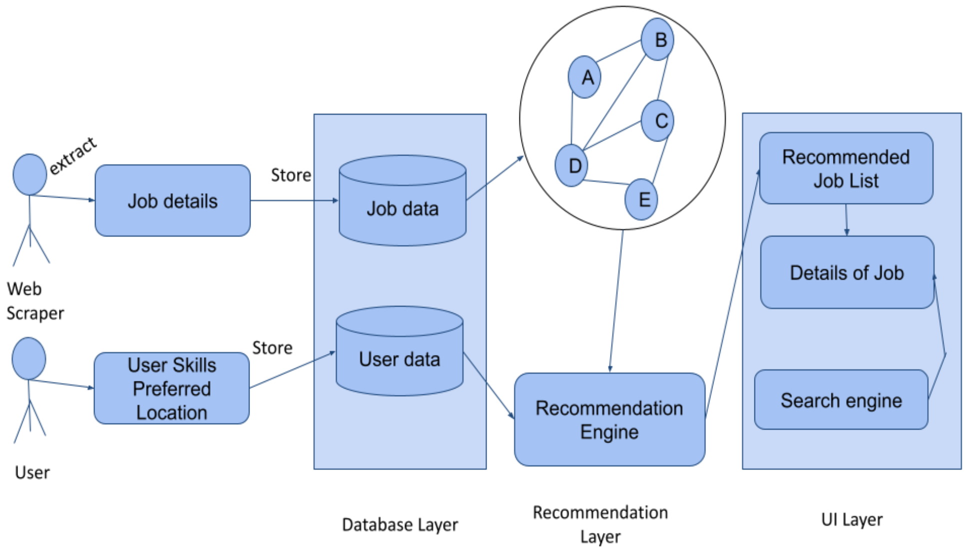


Figure 1. Design of Proposed System

Figure 2 shows the workflow of the recommendation algorithm and an explanation is provided below.

- Various job data are scraped from multiple websites, such as job descriptions, required skills, and job locations.
- The skills and preferred location of the user are acquired. Using these two inputs, the source is identified, which is a job in scraped data.
- Three different score matrices are created for the job description, skills, and location.
- Based on the different contribution scores, a single score matrix is constructed from the three different score matrices.
- On the single score matrix, a graph-based technique is used to get the best recommendations.

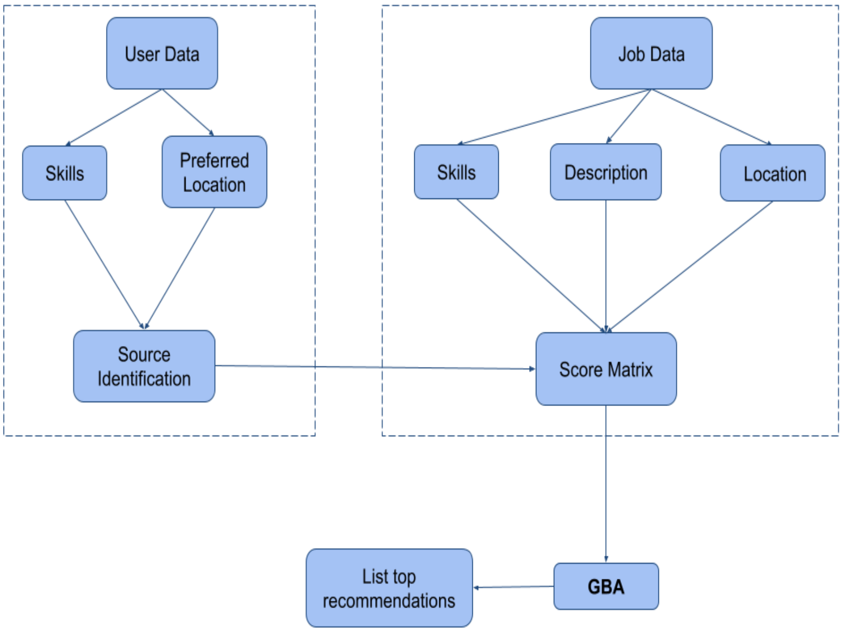


Figure 2. Workflow of Recommendation System

In the proposed approach, there is no breach of privacy because just the user's skills and preferred location are known. The user is not asked for any personal information. This approach provides a good level of security and privacy.

There is no scalability issue because only a job to job matrix is created that will contain only a finite number of job postings, and obsolete jobs are removed from the website itself and the job to job matrix is formed from real-time job postings.

There is also no difficulty with a cold start because if a new user joins, he or she will receive recommendations because the user-to-user matrix is not taken into account here. If a new job is posted, there is also no cold start issue because the new job will be scrapped and moved to the score matrix construction process. Hence it is taken into consideration in the recommendation list.

4.1 Extracting Job Details

Information about the project is obtained from the website "Internshala". Information is extracted from websites and online material using a process called web scraping. It is a cost-free way to obtain datasets and extract information. Additional analysis is done using this dataset.

Requests and Beautifulsoup are the two used packages. In order to swiftly extract DOM Elements, Beautiful Soup parses HTML into a machine-readable tree representation. It enables the extraction of certain table and paragraph elements with a specific HTML ID or Class. The HTML element would be obtained by requests from the URL and used as the input for BS to parse.

4.2 Preprocessing Extracted Job Details

Natural Language Processing or NLP is a field of Artificial Intelligence that gives machines the ability to read, understand and make sense of human languages. All basic preprocessing steps like removing stop words, lemmatization, converting all texts to lowercase, etc., were done. The package used is NLTK. NLTK, or Natural Language Toolkit, is a Python package you can use for NLP. Before analyzing the data, it must be preprocessed for better results.

4.3 Recommendation Engine - Graph-based Recommendation

4.3.1 Cosine Similarity:

The first step of this algorithm involves constructing three matrixes of similarity. Cosine similarity is a measure of how similar two linguistic documents are. Text documents are converted into an n-dimensional vector form so they can be used to find similarity. The mathematical equation for the cosine similarity of two nonnull vectors is as per equation (1)

$$Similarity = A.B/||A|||B|| \tag{1}$$

The text is vectored using the vector TF-IDf (term inverse frequency of the document). 'TF' represents the number of times a term appears in a certain document. 'IDF'" is a measure of the frequency or infrequency of a term appearing in the corpus of a document. The 'TF-IDf' is the value of a term within a document is the output of its 'TF' and 'IDF'. The higher the value, the more relevant are the words used in the document.

Skills to skills similarity matrix, job description to job description similarity matrix, and location to location similarity matrix are three similarity matrices. All of this information was gathered from the previously stated source. Skills are the skills that recruiters have mentioned on the website. Similarly, the website was used to obtain the job description and location. Figure 3 depicts an example of a skill-to-skill similarity matrix:

	c, python, java	reactjs, css, html, js
c , python, java	1	0.0000003
reactjs, css, html, js	0.0000003	1

Figure 3 Skill-to-Skill Similarity Matrix

A single matrix is created by combining these three matrices. The skills to skills similarity matrix accounts for 60% of the score. Similarly, the job description to job description matrix and the location to location matrix each contributed 30% and 10%, respectively.

4.3.2 User Input

The user's skill and the user's desired work location are taken as input.

4.3.3 Source identification

Based on the user's expertise, a source is identified. The source is nothing but a job from the scraped data. User skills and job skills are analyzed and the most relevant job is taken as the source.

4.3.4 Dijkstra Algorithm

There is a single similarity matrix that represents the similarity value. The distance matrix is formed by taking the difference. The mathematical equation for distance is given in equation (2).

$$distance = 1 - similarity \tag{2}$$

This distance graph has several nodes where every node is linked to several other nodes. If the minimum distance between each node from every other node is known, then it's easy to pick the top recommendations.

This need can be fulfilled by the Dijkstra algorithm. Negative edges are not needed, so remove them from the graph. Use this graph and source to implement the Dijkstra algorithm that provides the best recommendations.

5. Results and Discussions

5.1 Dataset:

Numerous job details were obtained from various sources, as well as user skills and preferred location. A dataset having test user's history is created. Userid and Job_Titles are two features in the user history. It is obtained from the user through a Google form. The features in the job dataset include jobid, job_title, job description, and job location. Table 1 shows the job data fields extracted from multiple websites. Figure 4 represents sample data that belongs to various jobs. The first column is job id . Table 2 shows user data fields collected from users. Figure 5 represents a sample data entered by users through Google form.

Table 1. Job Data Fields

Field	**Type**
Job_id	int
Job Title	String
Job Description	String
Skills Required	String
Location	String

Table 2. User Data Fields

Field	**Type**
Timestamp	String
Enter your skills	String
Enter your location	String
Job_title_1	String
Job_title_2	String
Job_title_3	String
Job_title_4	String
Job_title_5	String

A	B	C	D	E
101	A-IT Sales Executive (B2B)	Key responsibilities: 1. Prospect for potential new clients and turn this into increased business 2. Work on cold calling as appropriate within your market or geographic area to ensure a robust pipeline of opportunities 3. Meet potential clients by growing, maintaining, and leveraging the network 4. Identify potential clients and the decision-makers within the client organization 5. Research and build relationships with new clients 6. Set up meetings between client decision-makers and the company's practice leaders/principles 7. Plan approaches and pitches 8. Work with the team to develop proposals that speak to the client's needs, concerns, and objectives 9. Participate in pricing the solution/service 10. Handle objections by clarifying, emphasizing agreements, and working through differences to a positive conclusion 11. Use a variety of styles to persuade or negotiate appropriately 12. Present an image that mirrors that of the client 20	Client Relationship	Gurgaon,Bangalore

Figure 4 Sample Data of Jobs

Vol 12 Issue 03 2023
ISSN NO: 2230-5807

Timestamp	Enter your skills	Enter your location	Job_title	Job_title	Job_title	Job_title	Job_title
2022/04/09 12:30:53 PM GMT+5:30	accounting excel office tally	bangalore	Accountant	Finance Executive	Associate Accountant	Finance Executive	Junior Accountant

Figure 5 Sample Data of User History

5.2 Sample Input and Output

User 1:

User 1 is expertise in both software testing and manual testing and his/her preferred location is Bangalore. The top recommendations are software development engineer in testing, software tester, QA tester, QA engineer and junior full stack developer. Hence the recommendations are mostly related to testing and QA and location for all those jobs also mostly Bangalore which are acceptable. Figure 6 represents users 1's input. Figure 7 represents recommendations for this user.

```
Enter your skills:manual testing software testing

Enter your location:bangalore
```

Figure 6 User 1's Inputs

predicted_title - List (20 elements)

Index	Type	Size	Value
0	str	1	Software Development Engineer In Testing (SDET)
1	str	1	Software Tester
2	str	1	QA Tester
3	str	1	QA Engineer
4	str	1	Junior Full Stack Developer (Python + JavaScript)

predicted_location - List (20 elements)

Index	Type	Size	Value
0	str	1	bangalore
1	str	1	bangalore
2	str	1	remote
3	str	1	bangalore
4	str	1	bangalore

Figure 7 User 1's Recommendation List

User 2:

User 2 enters the respective skills and location as shown in Figure 8. Graphic designer, video editor, video editor, multimedia content analyst, and assistant video editor are the top recommendations provided. Since the skills are adobe related, the recommendations are also related to video editor and graphic designer. Figure 9 represents recommendations for this user.

```
Enter your skills:Adobe After Effects Adobe illustrator Adobe
photoshop Adobe Primere Pro Final Cut Pro Video Editing Video Making

Enter your location:Delhi
```

Figure 8 User 2's Inputs

predicted_title - List (20 elements)

Index	Type	Size	Value
0	str	1	Graphic Designer
1	str	1	Video Editor
2	str	1	Video Editor
3	str	1	Analyst - Multimedia Content
4	str	1	Associate Video Editor

predicted_location - List (20 elements)

Index	Type	Size	Value
0	str	1	gurgaon
1	str	1	gurgaon
2	str	1	mohali
3	str	1	kolkata
4	str	1	delhi

Figure 9 User 2's Recommendation List

User 3:

User 3 is expertise in Dart and Flutter. The top recommendations are flutter developer, flutter app developer, junior flutter developer, junior full stack developer and flutter developer. As skills are related to flutter, the recommendations produced are also related to Flutter development. Figure 10 represents user 3's input. Figure 11 represents recommendations for this user.

```
Enter your skills:Dart Flutter

Enter your location:Bangalore
```

Figure 10 User 3's Inputs

predicted_title - List (20 elements)

Index	Type	Size	Value
0	str	1	Flutter Developer
1	str	1	Flutter App Developer
2	str	1	Junior Flutter Developer
3	str	1	Junior Full Stack Developer
4	str	1	Flutter Developer

predicted_location - List (20 elements)

Index	Type	Size	Value
0	str	1	noida
1	str	1	remote
2	str	1	remote
3	str	1	remote
4	str	1	bangalore

Figure 11 User 3's Recommendation List

5.3 Performance Metrics

For the tests, user history from some users was used. The user history for a set of six users was collected and depicted in the Figure 12 and calculated based on recommendation results and user history, accuracy, recall and average accuracy. The algorithm had a pretty good recommendation.

A	B	C	D	E	F	G	H
Timestamp	Enter your skills	Enter your location	Job_title	Job_title	Job_title	Job_title	Job_title
2022/04/09 12:30:53 P	accounting excel office	bangalore	Accountant	Finance Executive	Associate Accountant	Finance Executive	Junior Accountant
2022/04/09 12:32:29 P	creative writing digital	delhi	Digital Marketing Spec	Digital Marketer	Digital Marketing Asso	Content Writer	Digital Marketing Executive
2022/04/09 12:55:58 P	adobe effect , adobe c	mumbai	Jr. Executive Graphic [	Graphic Designer And	Graphic Designer	Associate Graphic Des	Junior Graphic Designer
2022/04/09 12:55:58 P	manual testing softwar	bangalore	Software Development	Software Tester	QA Tester	QA Engineer	Quality Assurance
2022/04/09 12:55:58 P	android flutter io kotlin	bangalore	Mobile App Developer	Flutter Developer	React Native Develope	Mobile App Developer	Flutter/Android Developer
2022/04/09 12:55:58 P	english proficiency spo	hyderabad	Associate Recruiter	HR Recruiter	Junior Recruitment Ex	HR Generalist	Junior Recruitment Executive

Figure 12 User History Data

When the "k" break occurs, the precision and recall are employed. Five was chosen for the k. Only the subset of rank 1 to k recommendations is used to derive precision and recall to cutoff k, P@k, and r@k. At cutoff k, precision and recall are employed. Five was the chosen k. P@k and r@k are the precision and recall calculated using only the subset of suggestions from rank 1 to k.

The Mean Average Precision (MAP) and Hit Rate (HR) metrics are utilised in this study to evaluate the effectiveness of the recommendation system. The mean of the average precision across all users is known as MAP. You receive awards from Average Precision for making sensible recommendations. HR is calculated by dividing the number of hits by the total number of occurrences.

Average Precision AP@N is defined as (3) if there is a need to propose N things and there are m relevant items in the entire space of items. Single data points, such as a single user, are covered by AP. Mean Average Precision MAP@N takes it a step farther by averaging all users' AP. The acronym MAP@N is calculated as in (4). The proposed recommendation approach gives the Mean Average Precision as 88.78%. Hit rate can be defined as in equation (5). The hit rate for the recommender system is 51%.

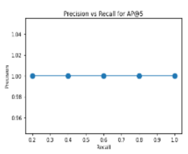
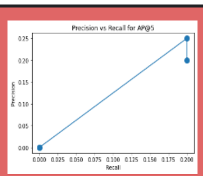
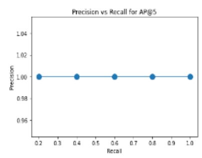
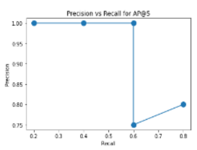
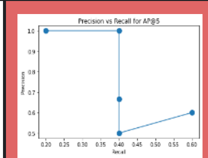
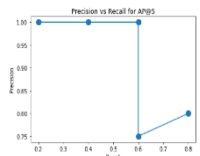
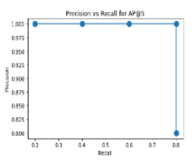
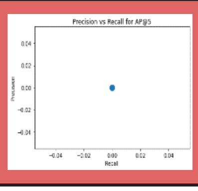
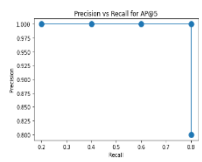
$$AP@N = \left(\frac{1}{m}\right)\sum_{k=1}^{N}\left(Precision@k \text{ if } k^{th} \text{ item is relevant}\right) \quad (3)$$

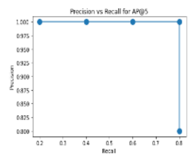
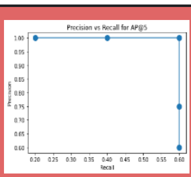
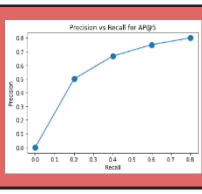
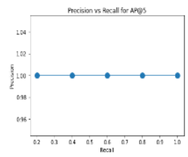
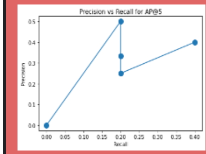
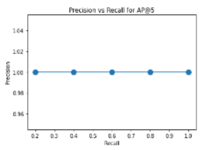
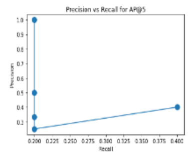
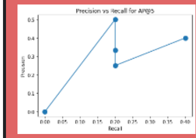
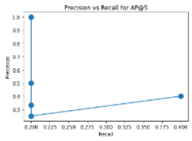
$$MAP@N = \left(\frac{1}{|U|}\right)\sum_{u=1}^{|U|}(AP@N)_u \quad (4)$$

$$Hit\ Rate(HR) = \frac{number\ of\ hits}{number\ of\ occurences} \quad (5)$$

Graphical representations of Precision@k vs Recall@k for the randomly chosen user chosen for the presented approach and other typical approaches are shown in Table 3. In Table 3, each line represents the precision plot against a user recall produced by various approaches, as mentioned in the header.

Table 3. Graphical representation of precision vs recall at k = 5 for GBA, content based - skills and content based - description approaches

User IDs	The Proposed Approach	Content based - Skills	Content based - Description
1.			
2.			
3.			

4.			
5.			
6.			

Note: This colour represents precision vs recall graph that shows variations from our approach

5.3.1 Analysis of graphs represented in Table 3

User 1: Both the proposed approach and the content-based approach (Description) produce a good graphical representation, with the precision of 1 at every k and recall steadily improving. The maximum recall attained is 1. In the content-based approach (skills) the maximum precision is 0.25 and again gradually decreases. The recall also follows the same. Here both our presented approach and content-based approach (Description) win since they give the best performance.

User 2: In the proposed approach and in content-based approach (Description) the precision starts at 1 but gradually decreases to 0.8 and recall gradually increases to 0.8. Precision starts at 1 and falls to 0.6 in a content-based approach (skills), whereas recall steadily climbs to 0.6. Here also both the presented approach and content-based approach (Description) win because of their better performance.

User 3: The precision of both the proposed approach and the content-based approach (Description) is 1 and recall progressively grows to 0.8. Content-based approach (skills) shows both precision and recall as 0. It is clearly seen that both the presented approach and content-based approach (Description) win.

User 4: In the proposed approach, precision is 1 in all cases, and recall gradually increases to 0.8. Content-based approach (skills) has a precision of 1 at all k and recall gradually increases to 0.6. In the content-based approach (Description), both recall and precision increases to 0.8. Here the presented approach wins.

User 5: Both the proposed approach and content-based approach (Description) have precision 1 at all k and the same follows for recall. The precision for this user inthe Content-based approach (skills) fluctuates and when k = 5 it is 0.4 which is low. The recall gradually increases to 0.4 which is also considered low. It is clearly seen that both the presented approach and content-based approach (Description) win.

User 6: All three approaches have some similar fluctuations in their graph. In this case, all the approaches' performances are mostly similar.

Table 4 gives a clear picture that the proposed approach has a maximum number of wins. Various users get the best result in different approaches but the maximum wins are present in the proposed approach and it is to be noted that there is no loss in the proposed approach.

Table 4. Comparison among Graph Based Approach (proposed), content-based -skills and content-based - description approaches

User IDs	The Proposed Approach	Content based - Skills	Content based - Description
1.	win	loss	win
2.	win	loss	win
3.	win	loss	win
4.	win	loss	loss
5.	win	loss	win
6.	draw	draw	draw

Tables 3 and 4 indicate that, when taking into account the top 5 suggestions, the suggested technique provides good precision and recall scores. Draw means that all the approaches give consistent performance for user 6. Table 5 clearly shows that the offered approach outperforms prior approaches.

Table 5. Comparison of content based approaches and proposed approach

Evaluation Metric	The Proposed Approach	Content-Based - Skills	Content- Based - Description
Mean Average Precision	88%	38%	81%
Hit Rate	51%	22%	48%

Table 6. Comparison of Existing Approaches

S.no	Existing approach	Drawback	How does this approach overcome these drawbacks?
1.	LinkedIn	Cold start problem	The approach presented in this paper does not have a cold start issue since the user skills and locations are the main attribute considered.
2.	Internshala	No matching recommendations	This Graph-based approach has a natural language-based recommendation engine that can recommend jobs based on user skills as mentioned earlier in the paper.
3.	Indeed	Extracts the job titles the user gives	Since this approach has recommendations based on skills that would do a better job than just matching the Title.

5.4 Why no to Collaborative filtering?

Collaborative filtering does not always lead to the desired outcomes. Here there is a possibility of a grey sheep problem. When collaborative filtering on the data that was collected from different users through google forms, the result was not satisfactory. Consider that there exists one user whose skills are flutter and dart; jobs that the user accessed is only mobile app developer. When a new user with the following skills android, flutter, kotlin, react native arrive, he only gets recommendations based on the already available user who has a history only on Mobile App developer. When the same user uses the proposed approach, he gets various recommendations related to a Mobile App Developer, a Flutter Developer, a React Native Developer, and a Java App Developer. Hence for job recommendation content-based is more suitable than a collaborative approach.

6. Conclusion

Nowadays, freshers and job seekers find it tough to select the most suitable profession on a variety of websites. This type of seeking wastes time and makes it harder to grow in one's job. Personalized recommendations are made to help people overcome such hurdles. To achieve this goal, job information is gathered from a variety of sources, and jobs are recommended based on work location, required skills, job description, user skills, and user's preferred location. As a result, personalized job recommendations are provided.

Comparisons on various approaches have been made. It was seen that there was a significant difference in the precision vs recall graph which was produced by various approaches. However, it was evident that this proposed approach gives more appropriate results than other approaches. By collaborating with businesses to publish job opportunities in an app and expanding the app to include course recommendations for the user's career path growth, this idea can be improved in the future. This algorithm can be used as a backend and the frontend can be developed so that users can see recommendations on the flow. This will be done so that this algorithm can be utilized and the application can be published for people to be used.

5. References

[1] Qing Zhou, Fenglu Liao, Chao Chen, Liang Ge, "*Job recommendation algorithm for graduates based on personalized preference*", CCF Transactions on Pervasive Computing and Interaction, vol. 1, pp. 260–274,2019.

[2] Miao Jiang, Yi Fang, HuangmingXie, Jike Chong &MengMeng, "*User click prediction for personalized job recommendation*", World wide web, vol. 22, pp. 325–345,2018.

[3] Ravita Mishra, SheetalRathi, "*Efficient and Scalable Job Recommender System Using Collaborative Filtering*", Lecture Notes in Electrical Engineering book series, pp, 845–856, 2020.

[4] Qingyu Guo; Fuzhen Zhuang; Chuan Qin; Hengshu Zhu; Xing Xie; HuiXiong; Qing He,"*A Survey on Knowledge Graph-Based Recommender Systems*", IEEE Transactions on Knowledge and Data Engineering, pp. 1-1,2020.

[5] Zhenqi Dong, ChunxiaLeng, Hong Zheng, "*Employment Service System Based on Hybrid Recommendation Algorithm*", Advances in Intelligent Systems and Computing book series, vol. 1, pp. 368–375,2020.

[6] D.Mhamdi, R.Moulouki, M.Y. El Ghoumari, M.Azzouazi, L.Moussaid, "*Job Recommendation based on Job Profile Clustering and Job Seeker Behavior*", The International Workshop on Artificial Intelligence and Smart City Applications (IWAISCA), vol. 175, pp. 695–699, 2020.

[7] TanviTayade, RutujaAkarte, GayatreeSorte, RohitTayade, PritiKhodke, "*Data Mining Approach to Job Recommendation Systems*", EAI/Springer Innovations in Communication and Computing book series (EAISICC), vol. 1, pp. 503–509,2020.

[8] Pradeep Kumar Roy, Sarabjeet SinghChowdhary, Rocky Bhatia, "*A Machine Learning approach for automation of Resume Recommendation systems*", International Conference on Computational Intelligence and Data Science (ICCIDS), vol. 161, pp. 2318–2327,2019.

[9] HariniPriya B R, Kannimalar K, Saranya S B and Subbulakshmi B, "A Machine Learning Approach for Personalized Job Recommendation System", In: Rahul Srivastava and Aditya Kr. Singh Pundir (eds), New Frontiers in Communication and Intelligent Systems, SCRS, India, 2022, pp. 399-404. https://doi.org/10.52458/978-81-95502-00-4-42.

[10] Mishra, R., & Rathi, S. (2022). Enhanced DSSM (deep semantic structure modelling) technique for Job recommendation. Journal of King Saud University-Computer and Information Sciences, 34(9), 7790-7802.