Chapter 1

Leveraging Knowledge Graph for Analysis and Recommendation of Jobs

Amit Patil, Muskan Jain, Neetu Sardana, Sanjoli Goyal

Department of Computer Science & Engineering and Information Technology, JIIT, Noida, India
amitgpatil215@gmail.com,jainmuskan5787@gmail.com,saradana.neetu@gmail.com,sanjoligo6mar@gmail.com

Abstract:

One of the most important aspects of the modern recruitment market is online job boards. With millions of job seekers pursuing job posts on a daily basis, the need for accurate, effective, meaningful, and transparent employment suggestions is greater than ever. While the Recommender System is making strides in a variety of online domains by adding social and economic value, the job recommendation sector is less explored. When dealing with the massive volume of recruiting information available on the Internet, a job seeker spends hours searching for the most useful one. We proposed a bipartite graph based job-hunting recommendation system. The system is efficient in comparison to existing collaborative filtering as it handles spareness and helps in quick decision making. The chapter also analyzed the resumes and job description separately using homogeneous graphs. The analysis has been carried out using the varied features like age, domain, skills and gender.

1.1 Introduction

The growing popularity of the Internet has increased the demand for online job searching. Job Sites enable job seekers in building and enhancing the network opportunities. According to Jobsite's report 2014, 68% of online job seekers are college graduates or postgraduates. But now scenario has changed

Post pandemic many companies have initiated sacking for instance recently Facebook parent Meta has laid-off 11,000 people, about 13% of its workforce. Twitter has fired about 50 per cent of its workforce which is around 3,800 employees. In addition, a new normal, post pandemic has led to the rise in demand for diverse jobs. Many people of varied backgrounds and experience are looking for the right job on online platforms[14].

The main issue in most of the job-searching websites is that they simply present recruitment information to website visitors. Job seekers manually gather the necessary information to locate and apply for jobs. They have to apply their wisdom to analyze the information present on the job sites. The entire process is time-consuming and inefficient. An efficient job recommender system is required that can help the job hunter in taking quick and timely decisions

Generally, past studies have used collaborative filtering (CF) approach or content-based (CB) approach for recommendations. To support the user during the decision-making process, the CF based recommender system exploits the user data in order to identify interesting items (jobs) for a given individual who may be similar to a set of those previous users, following the intuition that similar users like similar items. Conversely, CB recommender systems use user preferences (resumes) with items (jobs) textual content. Collaborative filtering methods rely solely on the past activity of users (e.g. ratings, purchases) for generating recommendations. The short-lived nature of the items (jobs) in the system and the rapid rate in which new users and jobs enter the system make the cold-start a serious problem hindering CF methods.

Graph-based models adopt link analysis methods from graph theory to address the shortcomings of CF-based approaches such as sparsity and improve the quality of the recommendations. Link analysis gives us the ability to calculate centrality measures—namely degree, betweenness, closeness, and eigenvector to see the connections on a link chart or link map. Graph-based recommendation systems are differentiated based on how they build the graph and traverse it for recommendations. We have used Heterogeneous graph-based models to build a bipartite graph of both resumes and Job Descriptions. They are connected with a common node which can be Skill, DevType, Age etc. Using a Bipartite graph, we have performed job recommendations using two techniques Adamic adar and Common neighbors. We have also investigated homogeneous graph models for resumes and Job Descriptions in isolation. Work in this chapter has two dimensions. Firstly, we have analyzedL the data present in the resume and Job description separately. We constructed homogeneous graph performed analysis on resumes and job description and addressed four questions:

a) What are the hot and cold skills in the available Resume and Job Description.
b) What is the distribution of Resumes gender wise corresponding to different ages.
c) What is the distribution of salaries corresponding to different age groups?
d) What is the correlation between domain and skills?
e) What are the top skills used in various domains in the resume?
f) Find all the Job Description having similar skills as a resume.

In the second dimension we have constructed a heterogeneous bipartite graph between resume and job description and performed the job recommendations.

Rest of the chapter is organized in section 2 presents the past studies based on recommendations, Section 3 describes proposed methodology, section 4 provides the experimental details and chapter ends with the conclusion.

1.2 Related Work

The various studies closely related to our work has been presented in this section.

A study has proposed a novel method for a scalable and robust job recommendation system for online recruiting services. This approach uses a multigraph of jobs connected by similarity edges based on user behavior and job content. It evaluates the recommendations generated by this system and finds that it achieves around 90% accuracy on average. Also shows that this system performs better than a classical collaborative filtering approach while requiring only a third of the number of emails sent.[5]

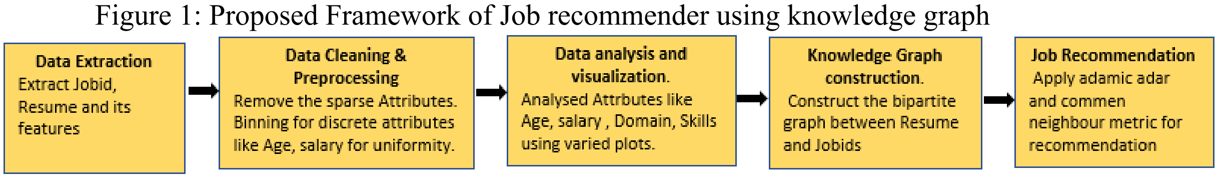
Different methods discussed in previous studies that discuss development phases of the recommender system First, deep learning models are widely used to extract latent features from item content and auxiliary information, and to improve recommendation accuracy and handle sparse data. Autoencoders and CNNs are the most commonly used models for this purpose. Second, it is important for a recommender system to incorporate data from multiple sources in order to obtain accurate user preferences and intentions. Third, the user profile is a critical component of the recommendation process. Vector-based profiles, which model users as vectors of numeric ratings, are the most commonly used. However, these profiles can contain sensitive information, so histogram-based profiles, which model user data as samples of predefined categories, are sometimes used to protect privacy. Fourth, the performance and quality of a recommender system can be evaluated using various metrics such as MSE, MAE, and ROC for prediction accuracy and top-n recommendations. Finally, techniques such as generalization and deep learning can be used as preprocessing steps to reduce the dimensionality of large datasets and improve the performance of a recommendation system.[13]

There are various works like GUapp is a platform that helps people find jobs in the Italian public administration. The platform uses a technology called Latent Dirichlet Allocation to recommend jobs to users based on their skills and preferences. It also has a chatbot that lets users communicate with the platform in natural language to refine their job search. The chatbot makes the job search process more interactive and allows users to add new requirements as they go along.[8]

1.3 Proposed Methodology

Figure 1 presents the overview of the proposed framework of Job Recommender using Knowledge Graph (JRKG). The various stages in the framework are described below:

Figure 1: Proposed Framework of Job recommender using knowledge graph



Data collection: For this study we have used two datasets. First dataset consists of resumes of the job hunters and another dataset contains the Job descriptions (Jds) of the available jobs. Resume dataset is extracted from the Kaggle website. We have considered 7 seven features of a particular resume. The Job description dataset has been extracted from USjobs on Dice.com.

Data Cleaning & Preprocessing: In this stage the features in both the datasets are cleaned and preprocessed. The Salary field values were either weekly, monthly or annually, we used the salary_type field to achieve the salary as a discrete value of depicting annual salary. Since the salary field had discrete values we performed Binning to represent it with uniformity. The four attributes LanguageWorkedWith, DatabaseWorkedWith, PlatformWorkedWith and FrameworkWorkedWith were eliminated from the table and a single attribute namely Skills has been created which collectively contained the data values of these four attributes. We removed the countries having 40-50 resumes.

Data analysis and visualization: The features in the resumes and JDs are analyzed in varied dimensions. Hot and cold skills have been investigated to analyze the skill set in demand. the chapter also analysed the Job seekers gender, their domain, age. We also analysed whether there is any variation in the salary slabs corresponding to age or gender. The various plots like Histogram, Bar graph, Heat Maps, Scatter plot have been used for data analysis.

Knowledge Graph construction: To construct knowledge graphs, we considered nodes of three types namely Resume, Skill and JDs_id. Each resume node has a link with its skills set and similarly JD_ids are linked to skills required for that job. To correlate and connect the Resume and JDs and represent it in the form of a knowledge graph, common Skills and domains are used.

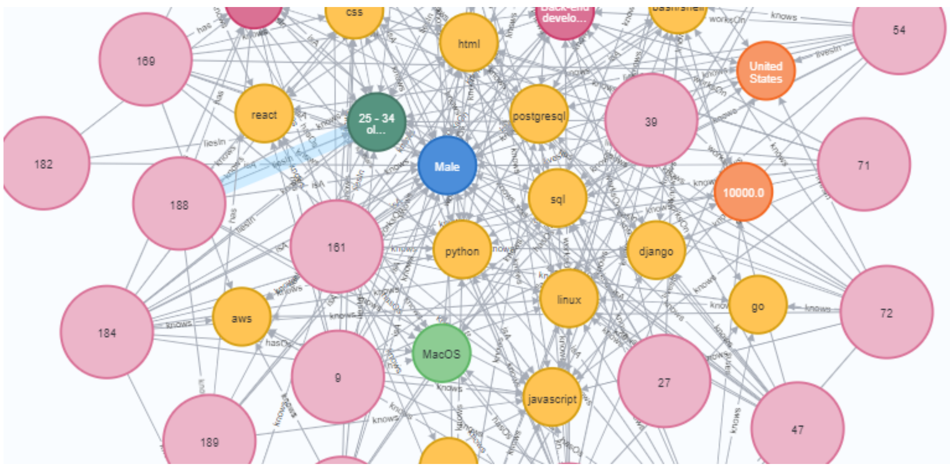


Figure 2: Homogenous Graph of Resume

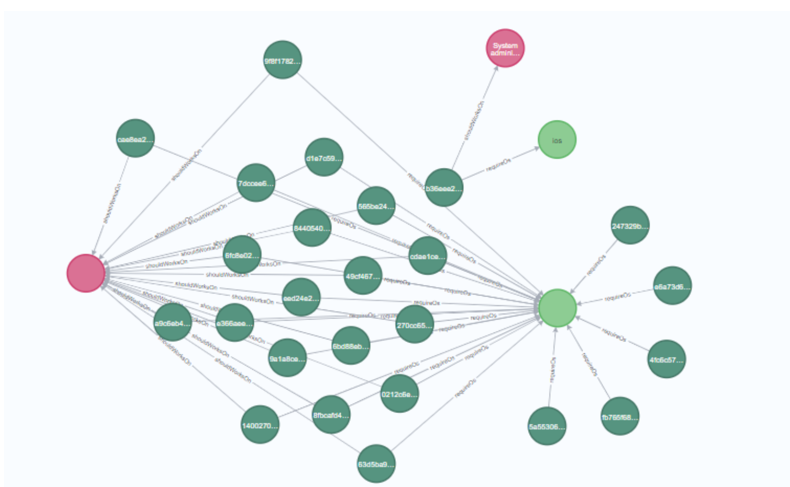


Figure 3: Homogenous Graph of Job Description

Job Recommendation: Knowledge Graph is used to predict the suitable Job (JD_id node) for the Job Seekers (Resume node). The following process is used for recommendation:

1. Matching skills and domains of a resumeId node with all the JobId nodes. That is, when we are given a resume, we derive the skills it has to offer and search those skills within our KG. And then those skills are matched with the JobIds already present in our database. So transitively a resume is connected to jobId via a common connecting link which are skills and domain.
2. Counting the number of matches between a resume and a JobId. That is, we now count how many connections a resume form with a particular JobId. Count here refers to the number of intermediary nodes between the resumeId and jobId. Note that, we always use the same resumeId with different JobIds because we are trying to recommend a job to a job seeker.
3. Calculating the common skill and common domain. Among the counts there are 2 types of connecting nodes: the skill property node and a domain property node. We count them separately as both of them might be forming an equal number of connections between resumeId and jobIb but the percentage significance that they hold when we seek a job are very different.
4. Choosing the resultant JobId with minimum of 10% skills in common with ResumeId. For further refining our recommendation we only take those jobIds to the next step which have at least 10% skills common with total skills offered by the resumeId. We remove those JobIds from our list of potential jobs that the job seeker would be interested in, if only less than 10% skills are all that they have in common. This also corresponds to real life situations where both a job seeker and the employer would like to have common interests in terms of what they have to work with.
5. We have taken the scores of Adamic Adar and Common neighbor between a particular resume and the jobs present in the database, if both suggest the same job then we take it to the next step.
6. Ordered by this score (in descending) we took up the first few recommendations and called this the JobRecommended List.

1.4 Experimental Results

1.4.1. Dataset used:

We have used two datasets. One dataset consists of resumes of users and second dataset consists of job descriptions for which jobs are available.

- The users resume dataset is provided by “stack overflow” on the “Kaggle” website in 2018. Stack Overflow did a survey in which they asked the developer community about everything from their favorite technologies to their job preferences to design their resume. There are 98,855 responses in this public data release [9]. Feature set present in the resume are Respondent, Country, CompanySize, Dependents, Type, Gender, JobSatisfaction, LastNewJob, Salary, SalaryType, LanguageWorkedWith, DatabaseWorkedWith, PlatformWorkedWith, FrameworkWorkedWith, OperatingSystem, Age.
- The job Description dataset was created by PromptCloud's in-house web-crawling service. This is a pre-crawled dataset, taken as a subset of a bigger dataset (more than 4.6 million job listings) that was created by extracting data from Dice, a prominent US-based technology job board in 2017. There are 22,000 job profiles in this public data release [10].

Feature set present in the Job Description are in jobid, skills, domain, platform, databases used.

1.4.2. Evaluation Parameters:

We used three metrics in this chapter for evaluation namely support for mining the skills in resumes and adamic adar for job recommendation.

To extract the frequent skills, *Support (Sp)* has been considered. Support reveals how often given skill is present in the resume. It is calculated as the total number of resumes having a particular skill, PS divided by the total resumes(n) in a particular category. In other words, fractions of resumes consist of particular skills. Support is calculated using equation 1.

$$\text{Support}(Sp) = \sigma(PS)/n \tag{1}$$

Adamic Adar Score: For Job Recommendation Adamic Adar metric is used. Adamic Adar is a measure used to compute the closeness of nodes based on their shared neighbors. This score is popularly being used in social network for predicting future links. It is computed using the following formula:

$$A(x,y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{\log |N(u)|} \tag{2}$$

where N(u) is the set of nodes adjacent to u.

A value of 0 indicates that two nodes are not close, while higher values indicate nodes are closer.
We have verified results using adamic adar closeness score which gives a non-zero score value while applying link prediction which basically means that the recommended job description is close to the resume.

Common neighbors:

$$CN(x,y) = |N(x) \cap N(y)| \tag{3}$$

where N(x) is the set of nodes adjacent to node x, and N(y) is the set of nodes adjacent to node y.A value of 0 indicates that two nodes are not close, while higher values indicate nodes are closer.
The library contains a function to calculate closeness between two nodes.

1.4.3. Experimental Results:

Data has been analyzed at two stages. During the first stage, data has been visualized and addressed five questions. Second Stage, Heterogeneous bipartite graph has been constructed between Resume and Jobs for recommendation.

Q1: What are the hot and cold skills in available CV's.
Motivation: We examined popular and rare skills in Resume and Available Job Descriptions to find the level of proximity among them

Results and Inferences: We applied Association rule mining techniques, Apriori and FP Growth to find the frequent skills. We considered support to be 50%. Popular skills and rare skills from the Resume Dataset are shown in figure 2

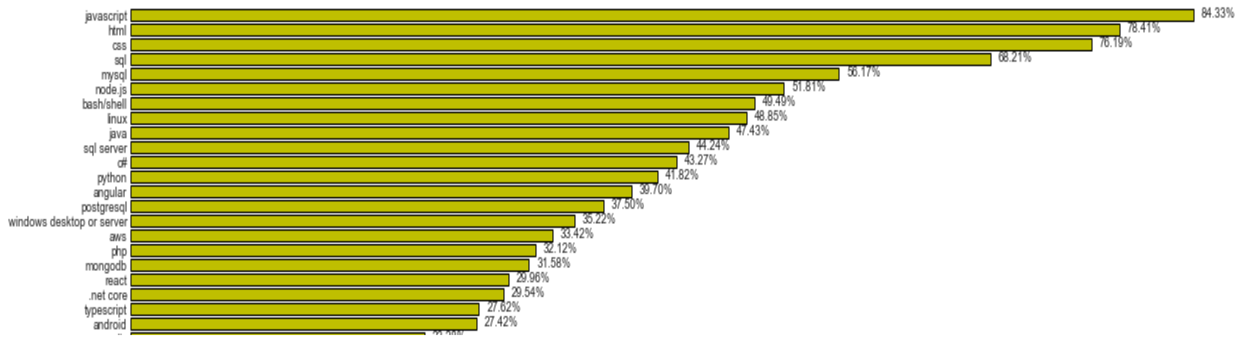


Figure 4: Bar Graph of percentage skills present in the resume dataset

popular_skills of threshold above 50%
['node.js', 'mysql', 'sql', 'css', 'html', 'javascript']

Rare_skills of threshold below 3%
['predix', 'hack', 'julia', 'ocaml', 'cobol', 'mainframe', 'gaming console', 'torch/pytorch', 'erlang', 'google home', 'apple watch or apple tv', 'ibm cloud or watson', 'apache hbase', 'clojure', 'f#', 'esp8266', 'delphi/object pascal', 'apache hive', 'haskell', 'google bigquery', 'rust', 'ibm db2', **'neo4j'**, 'windows phone', 'salesforce']

Similarly, we derived the popular domain and rare domain

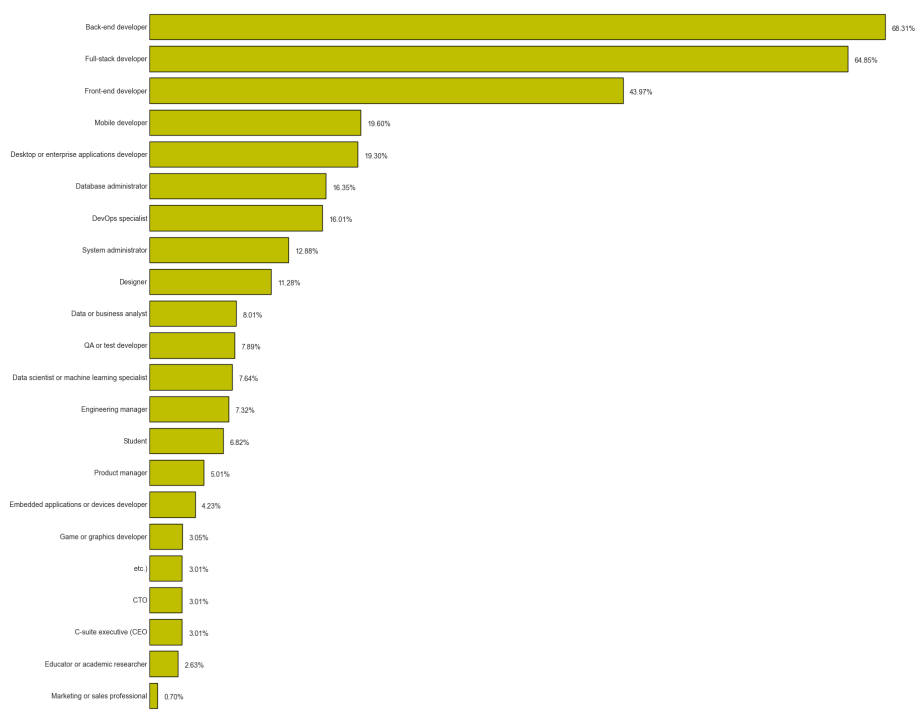


Figure 5: Bar Graph of percentage domain present in the resume dataset
Popular Domain includes ['Full-stack developer', 'Back-end developer']
Rare Domain includes ['Marketing or sales professional', 'Educator or academic researcher']

Analyzing the second dataset, we came across skills demanded by the **job description**. We classify those skills into 2 categories- popular skill and rare skill

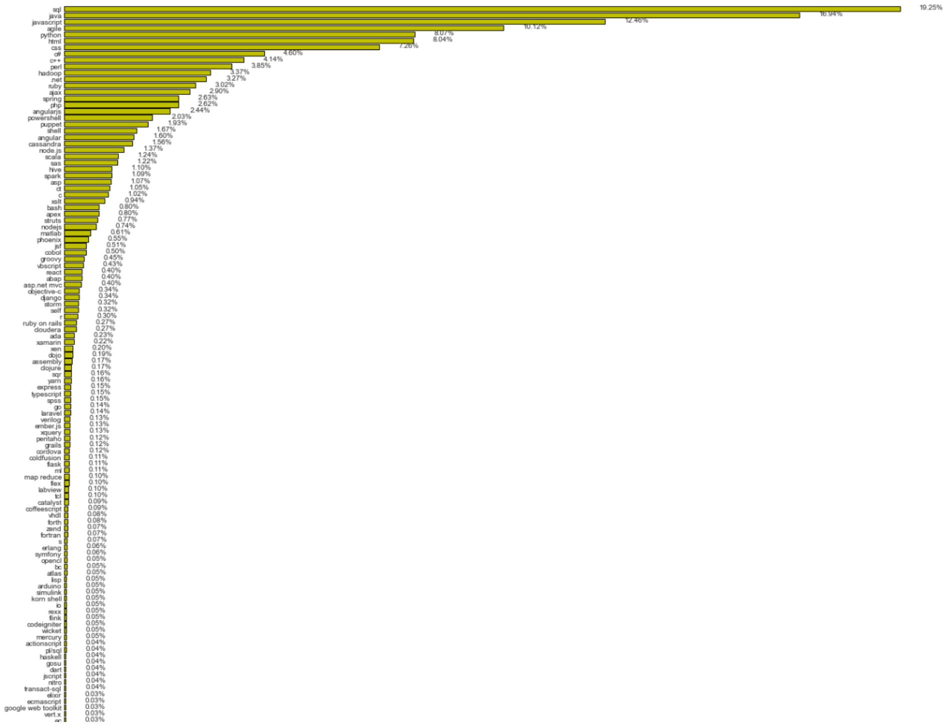


Figure 6: Bar Graph of percentage skills present in the job description dataset

Popular skill includes ['agile', 'javascript', 'java', 'sql']

Rare skill includes ['ml', 'flask', 'coldfusion', 'cordova', 'grails', 'pentaho', 'xquery', 'ember.js', 'verilog', 'laravel', 'go', 'spss', 'typescript', 'express', 'yarn','sqr', 'clojure', 'assembly', 'dojo', 'xen', 'xamarin', 'ada', 'cloudera', 'ruby on rails', 'r', 'self', 'storm']

Similarly, we derived the popular domain and rare domain

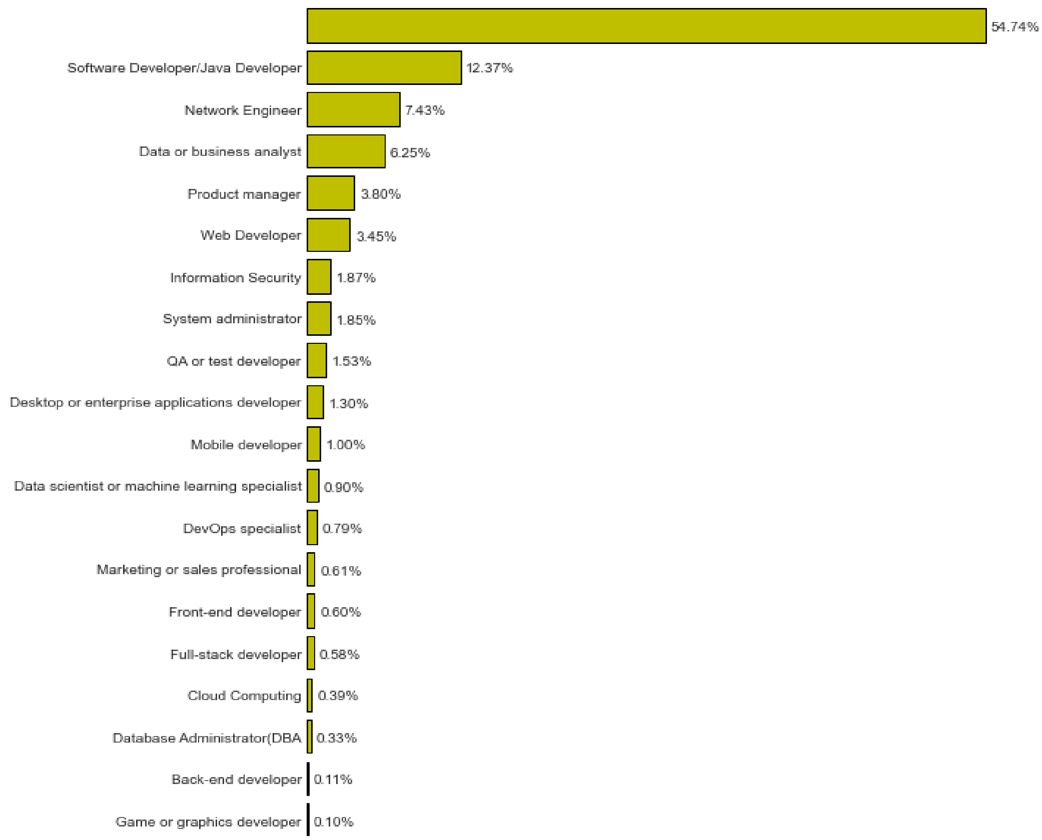


Figure 7: Bar Graph of percentage domain present in the job description dataset

Popular Domain includes['Data or business analyst','Network Engineer', 'Software Developer/Java Developer'']

Rare Domain includes ['Game or graphics developer', 'Back-end developer', 'Database Administrator(DBA)', 'Cloud Computing']

We found few insights from the data derived:-

- [“Clojure”] appears as a rare skill in both resume and job description. This suggests that Clojure is a skill that is relatively rare in both resumes and job descriptions, which could potentially make it a valuable skill to have if you are looking to stand out in the job market.
- Skills on the list of rare skills in job descriptions are generally more technical in nature, while the skills on the list of rare skills in resumes are more diverse and include a mix of technical and non-technical skills. This suggests that the skills that are considered rare in job descriptions may be more in demand by employers looking for specific technical expertise, while the skills that are considered rare in resumes may be more valuable for demonstrating a broader range of abilities and experiences
- According to resumes [“Backend”] is the most popular but in the job descriptions [“Backend”] is a rare domain. Meaning we have more backend developers then the opportunities available.
- [“Javascript”, “sql”] are popular skills for both resumes and job description dataset

Q2: What is the distribution of resumes gender wise corresponding to different ages?
Motivation: To find the gender wise impact on job search at different age groups.

Results & inference: The six groups are considered for analyzing the resume gender wise. Females resumes are skewed in comparison to males resumes. We found that female resumes are more in the age group 25-34 in comparison to rest age groups. Below the age 18 & post the age 55, females are not interested for any kind of jobs. Table 1 shows the gender statistics corresponding to different age groups.

Table 1: Gender statistics corresponding to different age groups.

Gender	Female	Male	other
Age			
18 - 24 years old	5.246818	92.890407	1.862776
25 - 34 years old	6.559225	91.975656	1.465119
35 - 44 years old	4.221721	94.424750	1.353529
45 - 54 years old	4.191617	95.059880	0.748503
55 - 64 years old	3.597122	94.964029	1.438849
65 years or older	0.000000	100.000000	0.000000
Under 18 years old	0.000000	100.000000	0.000000

Q3: What is the distribution of salaries corresponding to different age groups.
Motivation: To find salary variation at different age groups**.**

Results and Inferences: Salaries are given in dollars
Most common Salary group for different ages found are given in table 2.

Table 2: Salary statistics corresponding to different age groups.

Age Groups	**Salaries**
Under 18 years old	$500-$1500
18 - 24 years	$1501-5000
25 - 34 years	$1501-5000 & 5001-10000
35 - 44 years	$5001-10000
45 - 54 years	$5001-10000
55 - 64 years	$5001-10000 &10001-16000
65 years or older	$5001-10000

Few insights that can we drawn are:-
1. The most common salary group tends to increase as people get older. For example, the most common salary group for people under 18 years old is $500-$1500, while the most common salary group for people aged 45-54 is $5001-10000. This suggests that, on average, people tend to earn more authenticity, and that their salary may increase as they gain more experience and skills over time.

2. The salary groups for some age groups are more diverse than for others. For example, the salary group for people aged 25-34 includes both $1501-5000 and $5001-10000, while the salary group for people

aged 18-24 and 65 or older only includes $5001-10000. This suggests that the salary range for people in these age groups may be wider, potentially due to factors such as differences in experience, skills, or job roles.

Q4: What is the correlation between domain and skills?

Motivation: To find whether people are owing domain centric or diverse skills

Results & Inferences: Heat Map has been plotted to find the correlation between domain and skills.

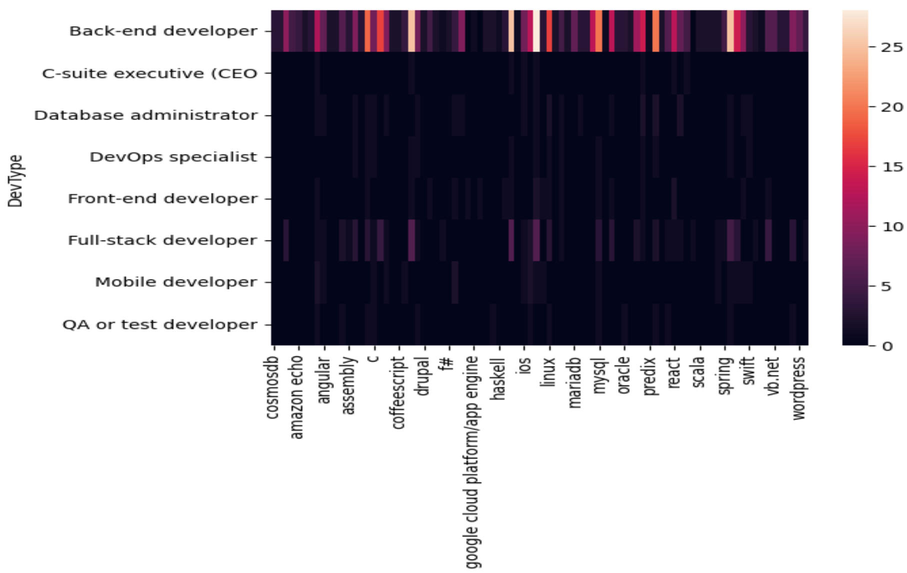


Figure 8: Heat Map to correlate Domain with Skills

We found few insights from the map:

- Most people are Back-end developer
- Full Stack Developer and Front-end developer occur frequently
- Coffeescript only occurs in Mobile Developer Domain
- Drupal only occurs in Full Stack.
- MariaDb only occurs in Database Administrator and Backend Developer
- CosmosDb and Amazon echo are skills of a Backend Developer
- Most important skill for Backend Developer is haskell

Q5. What are top skills used in various domains in the CVs?

Motivation: To check if the most popular skills in the overall dataset prefers any domain.

Results and inferences: We used the bar graph to visualize the top 3 skills used by the developer in each domain and noted that certain skills that are considered rare in one field may be very common in another.

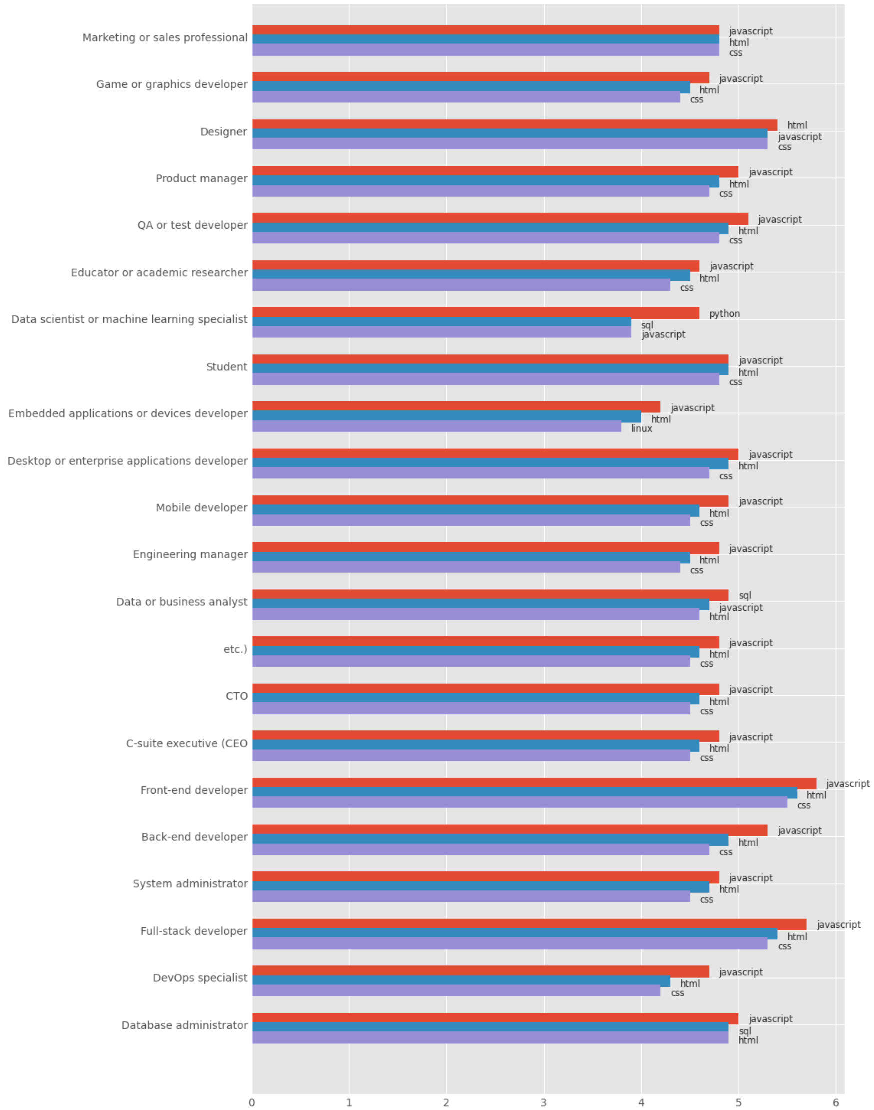


Figure 9: Bar Graph of Top 3 skills worked with particular domain

We found few insights from the graph:

- Mostly the top 3 skills of every domain are [‘javascript’, ‘html’, ‘css’]
- Domain types such as “Data Administrator” and “Data and business Analyst” have top skills such as [‘javascript’, ‘html’, “sql”]
- “Embedded applications or device developer” is the only domain which consist of [“**linux**”] as one of the top skills
- “Data scientist or machine learning specialist” domain has the most unique result where the top skills are [“**Python**”, “sql”, “javascript”].
- [“Python”, “linux”] are not the popular skills according to the result derived in the overall popular skill in the resume data but here in a few cases it is a popular skill.

In the second stage, we constructed the knowledge graph between resume and jobs available. And recommendation done on the basis of the number of connections formed between the resume node and the job description node. We form the connections with skill and domain nodes in between. To rank our results we calculated the score based on “commonSkillsCount” and “commonDomainCount”.

We visualized resume data and job description data using a bipartite graph. Both of these dataset individually creates their own graphs and are joined using common skills and domains only. We can answer certain questions using only resume data, job description data and both. As there is no direct connection between resume and job description node; we used our recommendation algorithm to match job seekers with suitable jobs based on their skills and domains. The queries can be used to find jobs that match specific skill sets, identify the skills and experience of potential candidates, and calculate compatibility between job descriptions and resumes. This information can be useful for job seekers and employers alike in finding and filling job opportunities.

Job Recommendation heterogeneous bipartite graph having Resume and Job Description

Q7: Find all the Job Description having similar skills as of resume.
Motivation: This query is useful for finding all the job descriptions that have similar skill requirements to a specific resume. This can be useful for job seekers who want to see which jobs might be a good fit for their skills and experience, and for employers who want to identify potential candidates with the skills they are looking for.
Results & Inferences: In this step we take all the potential JDs for a particular resume that may or may not be the best suitable recommendation. We show all the results even if there is only one connecting link between the jobId and the resumeId.

"jobDes1"	"resume1"
{"name":"8aec88cba08d53da65ab99cf20f6f9d9"}	{"name":3}
{"name":"a9ced403587b3bef3a0a20f46be23a54"}	{"name":3}
{"name":"359170ce45b8499420a871bc717bc388"}	{"name":3}
{"name":"8c87853f2cb2977e05df98497bcb7583"}	{"name":3}
{"name":"c43743b1fcf4ac35abe0832bf4cf1ef7"}	{"name":3}
{"name":"af7cee118bb4f7dbdcd7c44f8515dc42"}	{"name":3}
{"name":"981d868a1f6136fcc517a90bf5001ff8"}	{"name":3}
{"name":"c353b8b1bb3e98475262b566dc8e8790"}	{"name":3}
{"name":"6050a89bc1eda4de4c8f362569d9181b"}	{"name":3}
{"name":"6fbca09aae852c35214c982d102a95e5"}	{"name":3}
{"name":"637e50c7560f7edae8f02ba95066239b"}	{"name":3}

Figure 10: This is the list of jobs that have similar skills to a particular resume

Common neighbor score (link prediction):

This query is useful for calculating a score that represents the similarity between resumes and job description. This can be useful for job seekers who want to see how similar their skills are to the jobs in the market.

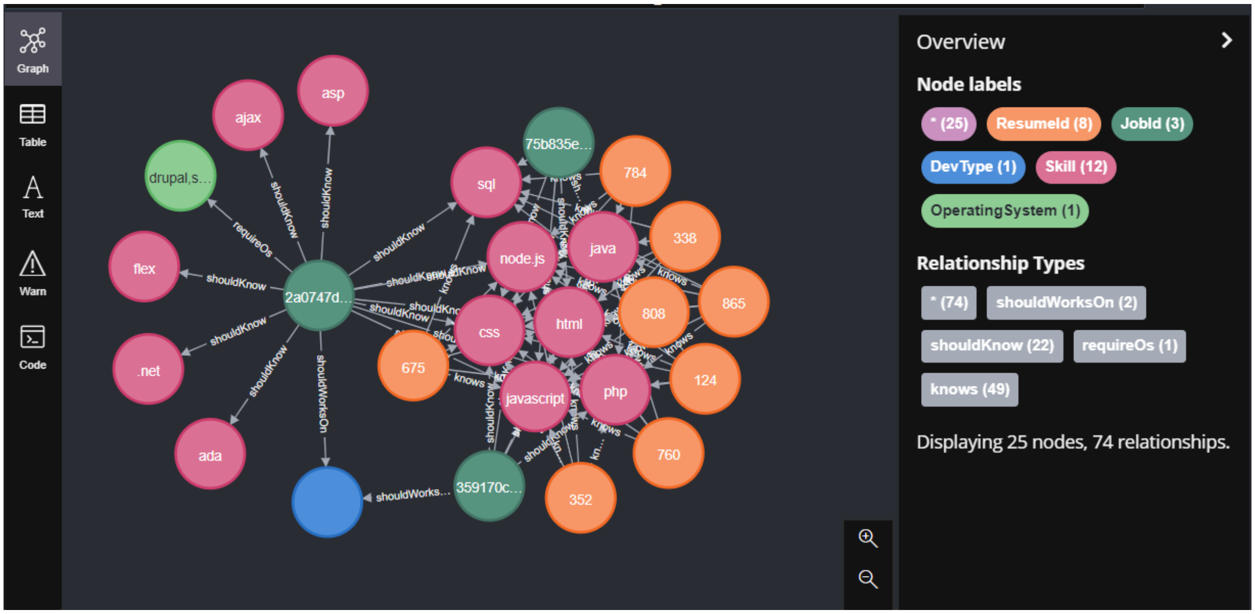
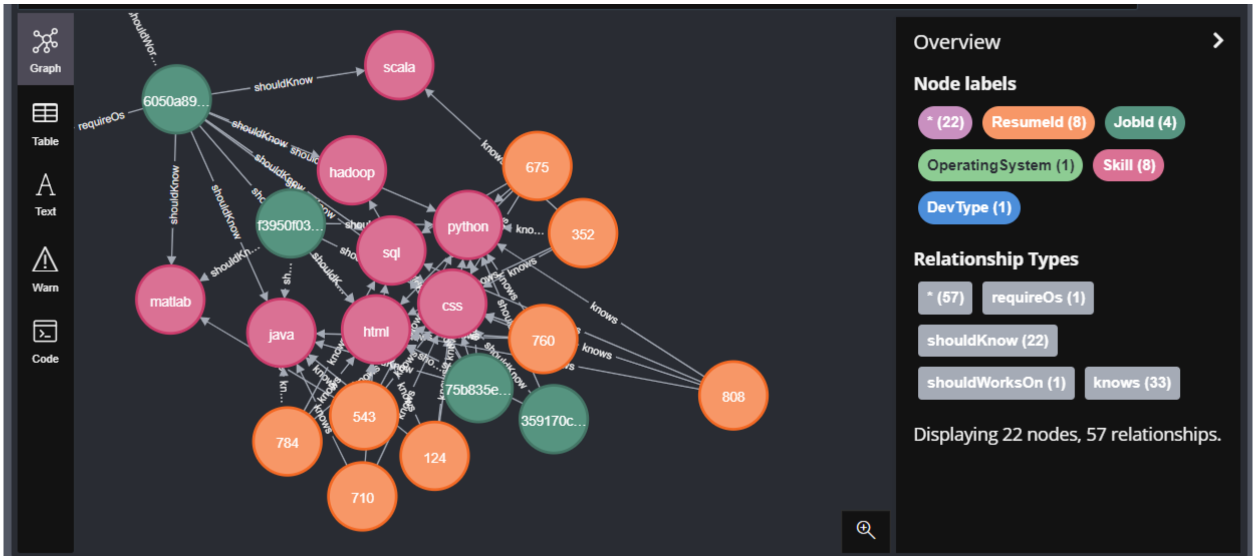


Figure 11&12: Common neighbor score between a resume and different Job

Score for JD and resume:

This query is useful for calculating a score that represents the compatibility between a specific job description and resume. This can be useful for job seekers who want to see how well their skills and experience match the requirements of a specific job, and for employers who want to identify the best candidates for a specific job based on their skills and experience.

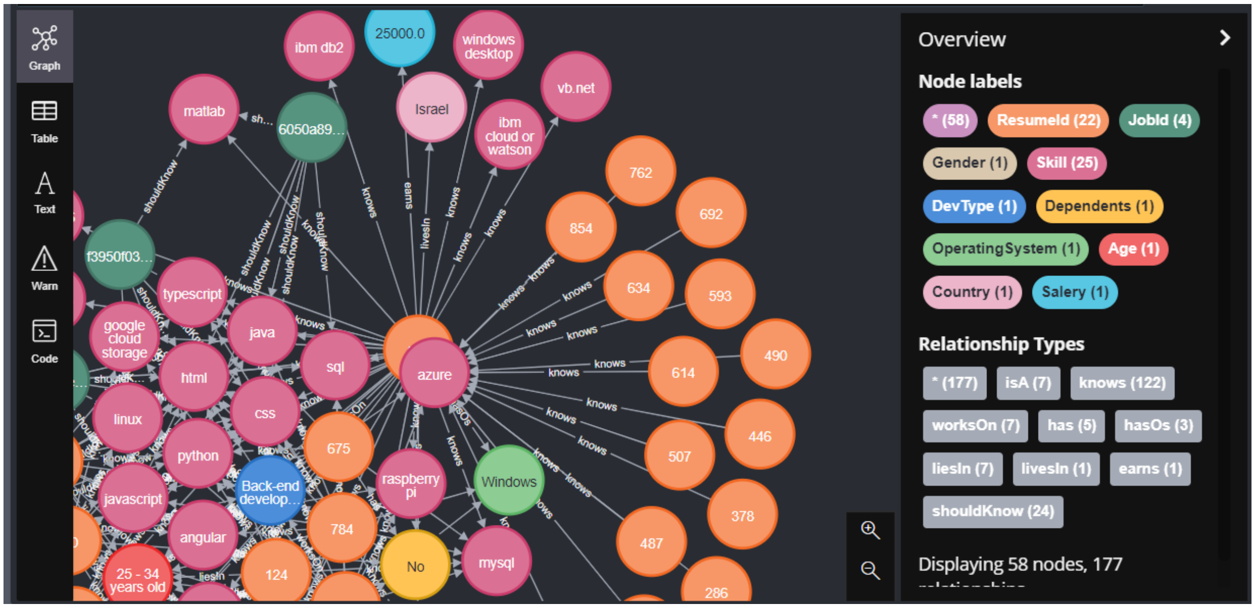


"adamic_score"	"common_neighbor_score"	"Resime"	"Job_Description"
2.3553605082877755	9.0	{"name":124}	{"name":"359170ce45b8499420a871bc717bc388"}
2.0941712517235516	8.0	{"name":760}	{"name":"359170ce45b8499420a871bc717bc388"}
1.9654892630364955	8.0	{"name":352}	{"name":"359170ce45b8499420a871bc717bc388"}
1.9529309039059308	8.0	{"name":675}	{"name":"359170ce45b8499420a871bc717bc388"}
1.741736095240056	6.0	{"name":543}	{"name":"f3950f031d1e12aefb4eb962e6f3702c"}
1.7398438573108084	6.0	{"name":543}	{"name":"6050a89bc1eda4de4c8f362569d9181b"}
1.7362518385703993	7.0	{"name":784}	{"name":"359170ce45b8499420a871bc717bc388"}
1.7015529594471577	7.0	{"name":808}	{"name":"75b835ecdf3ffc91f3a65969554d88b0"}
1.6205430152031444	6.0	{"name":710}	{"name":"359170ce45b8499420a871bc717bc388"}

Figure 13&14: Ordered list of Common neighbor score and Adamic Adar score between a resume and Job

Adamic Adar score:

This query is useful for calculating the Adamic Adar score for two nodes in the knowledge graph. This score represents the number of common neighbors that two nodes have, divided by the log of the degree of each of those common neighbors. This can be useful for identifying nodes that are closely related in the knowledge graph, and for recommending job seekers to jobs based on their common skills and domains.



score	u1	u2
2.3553605082877755	{ "identity": 202, "labels": ["ResumeId"], "properties": { "name": 124 } }	{ "identity": 791, "labels": ["JobId"], "properties": { "name": "359170ce45b8499420a871bc717bc388" } }
2.0941712517235516	{ "identity": 443, "labels": [	{ "identity": 791, "labels": [

Figure 15&16: Adamic Adar score between a resume and different Jobs

Verify result using Adamic Adar:

This query is useful for verifying the results of the recommendation algorithm by calculating the Adamic Adar score for each job description and resume pair. By comparing the Adamic Adar scores of different job descriptions and resumes, the algorithm can identify the most compatible matches and provide accurate and relevant recommendations to job seekers.

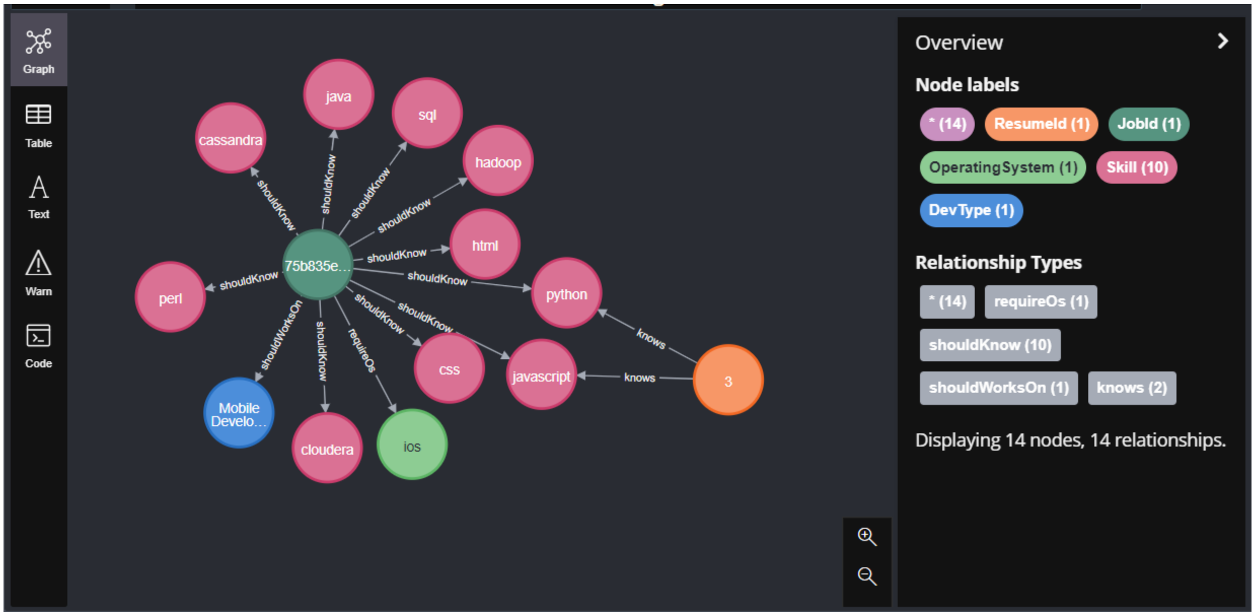


Figure 17&18: Resume and Job matched by our algorithm, has some closeness as score is positive

It is to be noted that in our work we cannot use the ROC curve as we do not have the ground truth regarding which job was taken by the resume. We used two separate databases and joined them using various parameters (skills, domain) , then recommended and ranked our matchings. We used Adamic Adar closeness score to give more reason to why our recommendation is accurate but we cannot verify it.

1.5 Conclusion

Recommender system technologies have gained considerable popularity in online domains due to its effectiveness in creating commercial and social value. As one of such domains, online recruiting services utilize recommender systems to serve millions of applicants with relevant and personalized job postings. To bring the full potential of advanced recommender systems into the job search domain, we proposed novel methods for a scalable and robust job RS. In this study, we studied and implemented a recommender system leveraging Knowledge Graph that recommends jobs based on resumes and vice versa

Our analysis of resumes and job descriptions using knowledge graph-based approaches revealed that popular skills in resumes include Node.js, MySql, and CSS, while popular skills in job descriptions include Javascript and Java. Our findings suggest that developers should focus on gaining proficiency in frameworks rather than just individual programming languages. We also found that developers tend to be more interested in Full Stack and Backend roles, while the job market currently needs more Network Engineers and Cloud Computing professionals. Our analysis of gender and age data showed that female resumes are most commonly in the 25-34 age group, and that people tend to earn more as they gain experience and skills over time. Most people in our dataset were Backend developers, and the most important skill for a Backend Developer was Haskell. We used common neighbor and Adamic Adar methods to calculate scores between job descriptions and resumes, and ranked the results according to these scores. Our Adamic Adar verification indicates that there is a strong relationship between recommended job descriptions and resumes.

Bibliography

[1] Chicaiza, J., & Valdiviezo-Diaz, P. (2021). A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions. Information, 12(6), 232.

[2] Sajisha, P. S., VS, A., & Ansal, K. A. Knowledge Graph-based Recommendation Systems: The State-of-the-art and Some Future Directions.

[3] Ehrlinger, L., & Wöß, W. (2016). Towards a definition of knowledge graphs. SEMANTiCS (Posters, Demos, SuCCESS), 48(1-4), 2.

[4] Li, N., Suri, N., Gao, Z., Xia, T., Börner, K., & Liu, X. (2017). Enter a job, get course recommendations. iConference 2017 Proceedings Vol. 2.

[5] Shalaby, W., AlAila, B., Korayem, M., Pournajaf, L., AlJadda, K., Quinn, S., & Zadrozny, W. (2017, December). Help me find a job: A graph-based approach for job recommendation at scale. In 2017 IEEE international conference on big data (big data) (pp. 1544-1553). IEEE.

[6] Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., & He, Q. (2020). A survey on knowledge graph-based recommender systems. IEEE Transactions on Knowledge and Data Engineering.

[7] Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G. D., Gutierrez, C., ... & Zimmermann, A. (2021). Knowledge graphs. ACM Computing Surveys (CSUR), 54(4), 1-37.

[8] Bellini, V., Biancofiore, G. M., Di Noia, T., Di Sciascio, E., Narducci, F., & Pomo, C. (2020, May). GUapp: a conversational agent for job recommendation for the Italian public administration. In 2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS) (pp. 1-7). IEEE.

[9] He, M., Zhu, Y., Lv, N., & He, R. (2022, January). A Feature Fusion-based Representation Learning Model for Job Recommendation. In 2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE) (pp. 791-794). IEEE.

[10] GRIDA, M., FAYED, L., & HASSAN, M. (2020). A STRUCTURED FRAMEWORK FOR BUILDING RECOMMENDER SYSTEM. Journal of Theoretical and Applied Information Technology, 98(07).

[11] Post-pandemic job search trends,
https://www.linkedin.com/news/story/post-pandemic-job-search-trends-4400521/
[12] Stack Overflow 2018 Developer Survey,
https://www.kaggle.com/datasets/stackoverflow/stack-overflow-2018-developer-survey
[13] Us jobs on Dice.com,
https://data.world/promptcloud/us-jobs-on-dice-com