European Proceedings of
Social and Behavioural Sciences
EpSBS

www.europeanproceedings.com
e-ISSN: 2357-1330

DOI: 10.15405/epsbs.2022.08.17

ICEST 2022
III International Conference on Economic and Social Trends for Sustainability of Modern Society

BUILDING AND ANALYSING A SKILLS GRAPH USING DATA FROM JOB PORTALS

Yuri A. Skobtsov (a), Denis M. Obolensky (b)*, Victoria I. Shevchenko (c),
Olga V. Chengar (d)
*Corresponding author

(a) Saint-Petersburg State University of Aerospace Instrumentation (SUAI), ul. Bolshaya Morskaya, 67, Saint Petersburg, Russia, ya_skobtsov@list.ru
(b) Sevastopol State University, ul. Universitetskaya, 33, Sevastopol, Russia, denismaster@outlook.com
(c) Sevastopol State University, ul. Universitetskaya, 33, Sevastopol, Russia, VIShevchenko@sevsu.ru
(d) Sevastopol State University, ul. Universitetskaya, 33, Sevastopol, Russia, OVChengar@sevsu.ru

Abstract

In this paper, the authors consider the process of building and analyzing a skills graph based on vacancies data from the job search portal. The connection between vacancies and skills is identified and formalized as labor market and employers' requirements to employee's qualification in the intellectual educational ecosystem model. The authors describe the collector program which can fetch vacancies data from job search sites with an example of site HeadHunter. Jupyter Notebook interactive notebooks and different Python-based libraries, such as Pandas and Numpy, are used for data processing. As a result, a skills graph is built and stored in the neo4j graph database platform. With the help of the Gephi application, the next properties of this graph are determined: degree distribution, the eigenvector centrality, closeness centrality, betweenness centrality, PageRank coefficient, and modularity. The graph is drawn using the ForceAtlas2 layout algorithm. The modularity coefficient for the vertices is used to find the main clusters of skills. The number of such skills clusters is almost the same as the count of specialties on the example job search portal. It results in the ability to determine specialty skillsets and interspecialty skills using vacancies data from job search sites.

Keywords: Job sites, machine learning, skills graph, skills, vacancies

https://doi.org/10.15405/epsbs.2022.08.17
Corresponding Author: D. M. Obolensky
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

1. Introduction

Every year, universities produce a large number of trained personnel for various professions (Abramov et al., 2021). After graduation, many people have a question about finding a suitable job for their level of skills. There is a difficulty in obtaining knowledge about the current requirements and essentials of the labor market.

A mandatory requirement for high-level educational programs is to orient students to obtain professional functions from the professional standards. Such standards usually refer to the education standard in this domain. Employment centers and online services specializing in job search can also become sources of information about professional skills, requirements, and working functions (Glebova et al., 2021).

Today many information resources can help find a suitable vacancy for any person following his professional skills. Among such information resources available on the Internet, online services are the most popular way to find a job (Glebova et al., 2021). In Russian Federation, the most popular job search sites are "Superjob.ru" (2021), "HeadHunter.ru (hh.ru)" (2021) and the state portal "TrudVsem" (2021), which also aggregates information from employment centers (Glebova et al., 2021).

2. Problem Statement

Analyzing the vacancies presented on these information portals, we can conclude that vacancies may have the following properties that are relevant to the applicant, in particular:

Skills requirements
Vacancy region
Minimum salary
Maximum salary
Publication date
Employer’s rating etc

The applicants' CVs also contain a description of professional skills, as well as, if specified, the desired salary, word region, and other parameters, according to data from job portal TrudVsem (2021).

A skill-based approach is one of the key research area in modern days. Intellectual educational ecosystem uses this approach to build vacancies recommendations for user based on user skills.

Skills are usually logically connected to each other and can be clustered based on their logical connections. Those clusters can be used to find related skills and new skills to learn. Those clusters may have intersections, since many skills are part of many specialities.

While it is possible to define skills and connections between skills in some speciality, in modern world, new skills are inventioned every day. And first place where new skills are becoming required are vacancies. Employers and companies wants to use the latest technologies to increase their revenue, so employee’s with new skills in their skillset are very relevant for them. Building and analysing skills clusters and graph in automated way can help employers and employees to find each other.

https://doi.org/10.15405/epsbs.2022.08.17
Corresponding Author: D. M. Obolensky
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

3. Research Questions

This article raises up next research questions:

i. Is it possible to automate gathering information about vacancies from job portals?
ii. How can be this information used to build a skills graph?
iii. Can we analyze a skills graph to find out clusters of connected skills using graph analysis methods? Are those clusters related to specializations?

4. Purpose of the Study

This study had four purposes:

i. to examine the possible gathering vacancies data from the job search portals,
ii. to explore ways to build a skills graph using gathered data,
iii. to analyse a skills graph and its main properties and metrics
iv. to analyse clusters of connected skills on the graph and examine their relationship with specialities.

5. Research Methods

The minimum unit for describing the requirements of the labor market, according to Obolenskii and Shevchenko (2020) is a vacancy. Let's specify the set of vacancies as V.

Each specialty sp can have a set of vacancies related to it. Let's mark this binary relation as a function VSP (1):

$$\forall sp \in SP\ VSP(sp): sp \rightarrow V'': V'' \subseteq V, \quad (1)$$

where SP – specialties set, V – vacancies set.

Each vacancy v can be assigned to some set of skills. Let's define this binary relation using multivalue function SV (2):

$$\forall v \in V\ SV(v): s \rightarrow S'': S'' \subseteq S, \quad (2)$$

where S – skills set, V – vacancies set.

It can be noticed that the set of skills $SV(v)$ for some vacancy v may include a variety of skills, including those that do not relate to the chosen specialty sp.

We can assume that skills in $SV(v)$ are logically connected. Then it is possible to form a fully connected graph of vacancy skills $K(v)$, where vertices will represent competencies in a given vacancy, and the edges will represent logical connections between those skills.

Then, for a given set of vacancies V, it is possible to form a set of skills graphs K (3):

$$K = \{K(v) \mid v \in V\} \quad (3)$$

where V – vacancies set.

https://doi.org/10.15405/epsbs.2022.08.17
Corresponding Author: D. M. Obolensky
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

The set of skills graphs K can be combined into a single competence graph K'. This graph can be built based on job vacancy data from job search portals. Let's use the social network model (Kolomeichenko et al., 2019) to analyze such a graph.

The social network model (Kolomeichenko at el., 2019) is a structure with a set of agents and a set of relationships defined on it (a set of connections between agents, for example, dating, friendship, cooperation, communication). Various methods can be used to analyze social network graphs (Aggarwal, 2011; Batura, 2012; Borgatti et al., 2013; Clauset et al., 2004; Palla et al., 2005), which make it possible to identify the structures of the graph of the interaction of subjects, for example, algorithms for identifying communities and clusters. To build such model and skills graph K' we need to be able to fetch vacancy data from job search portals.

5.1. Fetching Vacancy Data from Job Search Sites

To fetch job data from the job search portal, using HeadHunter as an example, we will use the API provided by this site. Figure 1 shows the main page of this job portal.

One can use HeadHunter's (2021) API to get information about job seekers and vacancies. It's also possible to leverage HeadHunter functionality to create applications.

The HeadHunter (2021) API has the following features:

All API works via HTTPS protocol
Authorization is performed via OAuth 2 protocol
All data is available only in JSON format
Base URL — https://api.hh.ru /
API supports paginated output
The maximum number of objects per page is 100.

The maximum number of returned objects from all the pages is 2000. To get more information from this resource, one should refine the query, for example, by specifying the region or other search parameters.

As a mechanism for collecting data, a C# collector program was implemented, which automatically receives information from this Internet resource, processes it, and saves it as a CSV file. The flowchart of the assembler program is shown in Figure 2, 3. The list of APIs used is shown in Table 1.

After the program's execution, it generates a file in a CSV format. This format is suitable for importing into other systems and is often used in data science. An example of the result of the work is shown in Figure 3. The structure of the CSV file is shown in Table 2.

https://doi.org/10.15405/epsbs.2022.08.17
Corresponding Author: D. M. Obolensky
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

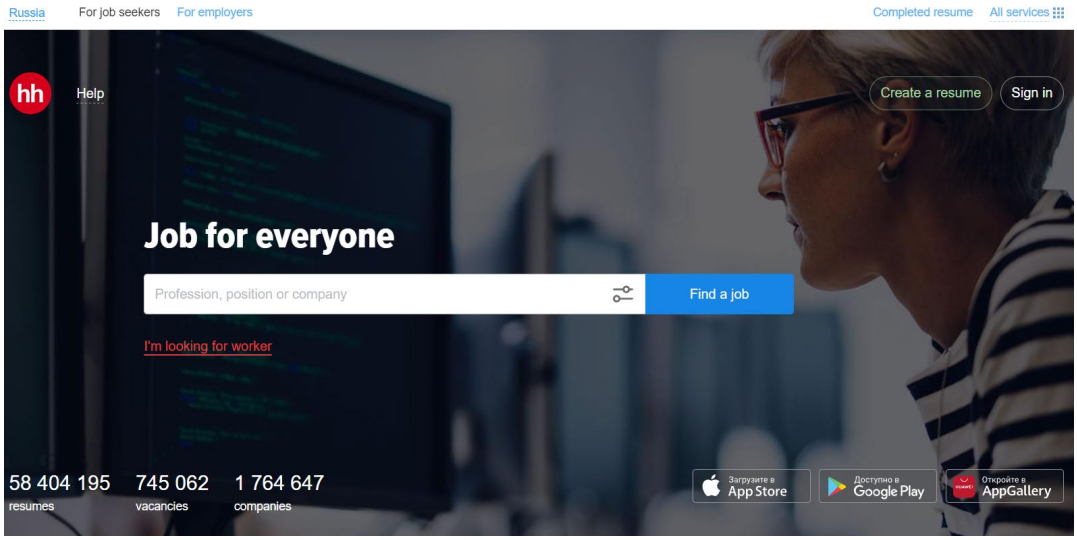

Figure 1. HeadHunter main page

Table 1. List of used HeadHunter APIs

URL	Description	Parameters
/areas	Get all regions	–
/vacancies?area&page&per_page	Get vacancies	area – region id or name page – number of page to fetch per_page – amount of items on the page
/vacancies/{id}	Get single vacancy	id – vacancy id

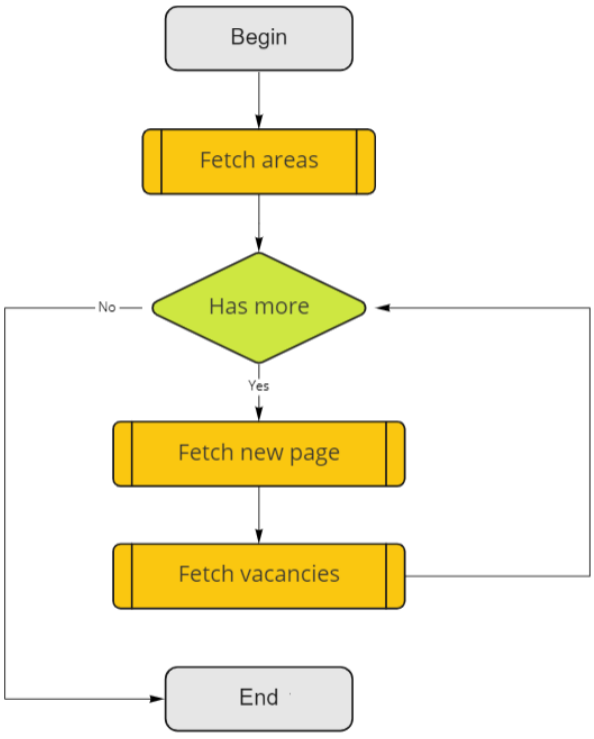

Figure 2. Diagram of fetching process

https://doi.org/10.15405/epsbs.2022.08.17
Corresponding Author: D. M. Obolensky
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

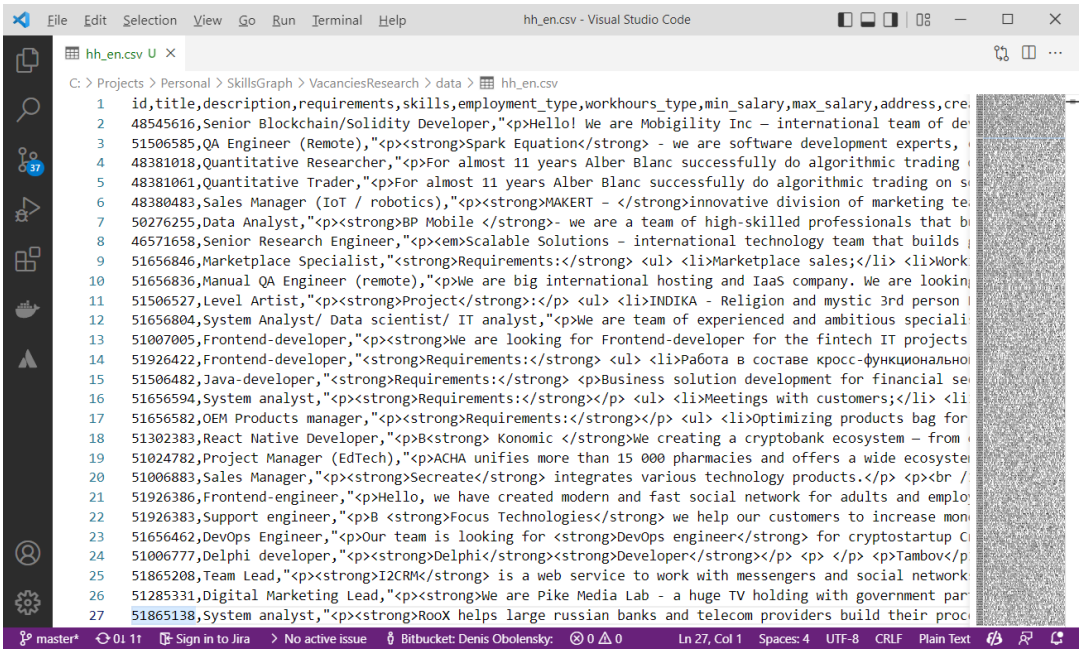

Figure 3. A CSV-file example

Table 2. CSV-file structure

Field	Type	Description
id	string	Unique identifier
title	string	Vacancy title
description	string	Vacancy description
requirements	string	Requirements for candidate
skills	list of strings	Required skills
employment_type	string	Employment type
workhours_type	string	Workhours type
min_salary	number (not required)	Minimal salary in rubles
max_salary	number (not required)	Maximum salary in rubles
address	string	Address
speciality	string	Speciality name
created_date	datetime	Creation date
published_date	datetime	Publication date
updated_date	datetime	Update date
is_deleted	boolean	Is vacancy closed

Using a separate CSV file with the structure above allows us to add support for new job search portals in the future without changing other intelligent educational ecosystem software modules.

As a result of the collector program's execution, it created a CSV file containing data on 130964 vacancies. Let's consider how to process these vacancies.

https://doi.org/10.15405/epsbs.2022.08.17
Corresponding Author: D. M. Obolensky
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

5.2. Data Processing

To build a skills graph based on vacancies data, the following steps should be performed:

1. Process input data and convert a row with a list of requirements into a set of skills
2. Define a list of skills and remove duplicates. Those skills will become vertices of the skills graph K'
3. Define connections between skills, create edges for graph K' and remove duplicates
4. Persist graph data in a graph database platform
5. Export graph into the file in a format suitable for further analysis, for example, GraphML

As graph storage, the *neo4j* graph database (Neo4J, 2021) can be used. It is an open-source graph database management system implemented in Java. As of 2021, it is considered the most common graph DBMS (Neo4J, 2021). An example of the interface of this DBMS is shown in Figure 4.

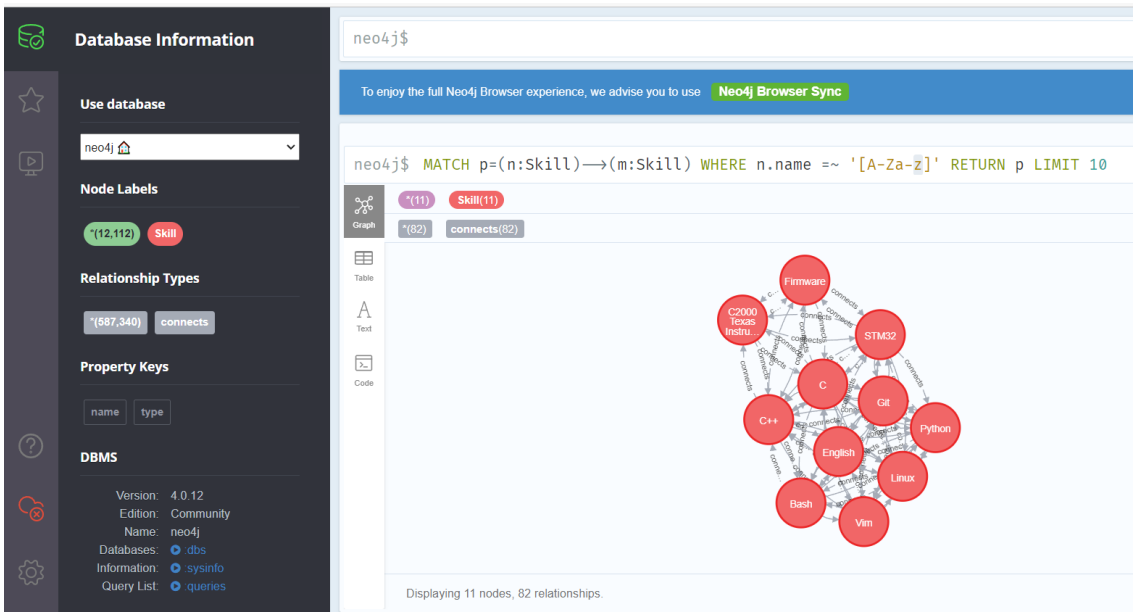

Figure 4. Neo4j user interface

Python was used to process vacancies data. A lot of popular data science tools and libraries, such as *Jupyter Notebook* interactive notebooks, *Numpy*, *Pandas*, and *Py2Neo*, were also used.

Loading data into an interactive notebook using Pandas dataframes is shown in Figure 5.

https://doi.org/10.15405/epsbs.2022.08.17
Corresponding Author: D. M. Obolensky
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

```
In [1]: import pandas as pd
        import numpy as np
        import requests

        from tqdm import tqdm
        tqdm.pandas()

In [2]: df = pd.read_csv("../data/hh.csv")

In [3]: df = df.sample(frac=1).reset_index(drop=True)

In [4]: df.head()
```

Out[4]:

	id	title	description	requirements	skills	employment_type	workhours_type	min_salary	max_salary	address	crea
0	44301859	Store Manager (Dmitrovsk)	<p>For everyone, who wants to try his...	NaN	NaN	Full-time	NaN	23000.0	33600.0	Orlovska oblast'	20
1	51643008	Salesman	<p>Road payments company...	NaN	Business Communications;Cash operations	Full-time	NaN	45000.0	54000.0	Moscowskaya oblast'	20
2	50603046	Barman	<p>«Altai Palace» is a shopping(...	NaN	NaN	Full-time	NaN	20000.0	NaN	Altai Republic	20
3	51134274	Foundry worker	<p>Huge facility on the North...	NaN	NaN	Full-time	NaN	77000.0	82000.0	Orenburgskaya oblast'	20
4	49951035	Auditor	<p>Company "Business Cities" is a well...	NaN	Financial control;Cost management;Purchases...	Full-time	NaN	35000.0	NaN	Republic Mordovia	20

Figure 5. Loading data into a dataframe

In Figure 6, one can see how string data transforms into a list of skills.

```
In [6]: df['skills'].isnull().any()
Out[6]: True

In [7]: df['skills'] = df['skills'].fillna('')

In [8]: df['title'] = df['title'].astype('str')

In [9]: df['id'] = df['id'].astype('str')

In [10]: df['skills_parsed'] = df['skills'].progress_apply(lambda x: x.split(';'))
         100%|██████████| 130964/130964 [00:00<00:00, 671557.12it/s]

In [11]: df['skills_parsed'] = df['skills_parsed'].progress_apply(lambda x: list(filter(lambda s: s.isspace() == False or s=='', x)))
         100%|██████████| 130964/130964 [00:00<00:00, 432160.53it/s]
```

Figure 6. Transforming string data into the list of strings

To determine a list of unique skills that will act as the vertices of the graph, for each vacancy, we transform a list of vacancy skills into a set, and then combine the resulting sets (Figure 7):

```
In [12]: all_skills = set()

In [13]: skills = [set(s) for s in df['skills_parsed']]

In [14]: for skill_list in tqdm(skills):
             all_skills = all_skills.union(skill_list))
         100%|██████████| 130964/130964 [00:20<00:00, 6495.24it/s]

In [15]: all_skills = set(filter(lambda s: s!= '', all_skills))

In [16]: len(all_skills)
Out[16]: 12112
```

Figure 7. Unique skills set calculation

https://doi.org/10.15405/epsbs.2022.08.17
Corresponding Author: D. M. Obolensky
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

After that, the result set contains 12112 unique skills. Let's get the set of edges of the desired graph K'. For each set of vacancy skills, we can get all permutations by 2, and then combine them into a single set of edges (Figure 8):

```
In [17]: from itertools import permutations

         df['skills_pairs'] = df['skills_parsed'].progress_apply(lambda skills: list(permutations(skills, 2)))

         100%|██████████| 130964/130964 [00:00<00:00, 172774.17it/s]

In [18]: df['skills_pairs']

Out[18]: 0         [(Team work, Civil law...
         1                                                []
         2         [(Team work, Literate speech), (Team ...
         3                                                []
         4         [(Team work, Merchendising...
                                    ...
         130959    [(Auditing, Internal control), ...
         130960    [(Communications, SAP ERP), (Communicatio...
         130961    [(Selling, Working with PC), (Selling,...
         130962    [(Working with PC, Team work), (Working...
         130963    [(Cash transactions, Transaction book management), ...
         Name: skills_pairs, Length: 130964, dtype: object

In [19]: skills_pairs = [set(s) for s in df['skills_pairs']]

In [20]: skills_connections = set()
         for s in tqdm(skills_pairs):
             skills_connections = skills_connections.union(s)

         100%|██████████| 130964/130964 [52:29<00:00, 41.58it/s]

In [21]: len(skills_connections)

Out[21]: 587687
```

Figure 8. Calculating permutations of skills and combining them into the set of edges

The use Python *Set* data structure guarantees that there are no duplicates in the resulting sets. With both sets defined it is possible to persist them as a graph in the neo4j graph DBMS. Let's use the Py2Neo library to interact with the neo4j DBMS, as well as the Numpy library to speed up the saving process. An example of saving vertex data is shown in Figure 9, and an example of saving edge data is shown in Figure 10.

```
In [23]: from py2neo.bulk import create_nodes, merge_nodes, merge_relationships
         from itertools import permutations

         np_skills = np.array(list(all_skills))

         for chunk in tqdm(np.array_split(np_skills, 100)):
             tx = graph.begin()
             skills_data = [{ "name": str(skillName), "type": "skill"} for skillName in chunk]
             create_nodes(tx, skills_data, labels={"Skill"})
             graph.commit(tx)

         100%|██████████| 100/100 [00:02<00:00, 47.27it/s]
```

Figure 9. Persisting vertices in neo4j DBMS

```
In [31]: from py2neo.bulk import create_nodes, merge_nodes, create_relationships
         from itertools import permutations

         np_skills_connections = np.array(list([(str(first), {}, str(second)) for (first, second) in skills_connections]))

         for chunk in tqdm(np.array_split(np_skills_connections, 1000)):
             tx = graph.begin()
             create_relationships(tx, chunk.tolist(), "connects", start_node_key=("Skill", "name"), end_node_key=("Skill", "name"))
             graph.commit(tx)

         100%|██████████| 1000/1000 [1:37:06<00:00,  5.83s/it]
```

Figure 10. Persisting edges in neo4j DBMS

https://doi.org/10.15405/epsbs.2022.08.17
Corresponding Author: D. M. Obolensky
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

Vacancies vertices and their connections can be persisted in the same way. A graph will be built automatically in neo4j as long as vertices and edges are given. An example of the structure of the resulting graph is shown in Figure 11.

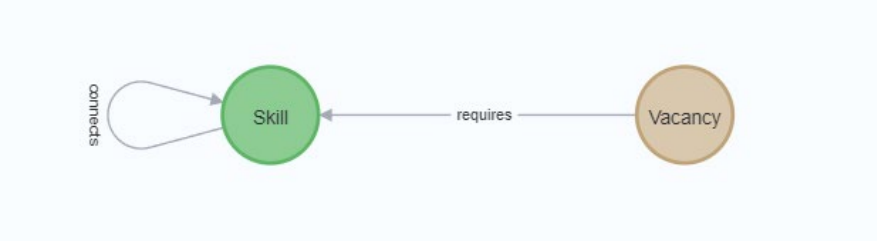

Figure 11. Graph database structure

The neo4j DBMS also can export data to various formats, such as CSV, JSON, GraphML. Figure 12 shows an example of a query to the neo4j DBMS in the Cypher language (Neo4J, 2021) which exports the entire graph to the specified file.

```
In [32]: graph.run("CALL apoc.export.graphml.all(\"skills.graphml\", {})")
```

Out[32]:

file	source	format	nodes	relationships	properties	time	rows	batchSize	batches	done	data
skills.graphml	database: nodes(12112), rels(587340)	graphml	12112	587340	24224	2255	599452	-1	0	true	null

Figure 12. Exporting the whole graph into GraphML file

GraphML format helps with the subsequent visualization and analysis of the graph in specialized software, for example, Gephi.

5.3. Building and Analysis of the Skills Graph

Let's use the Gephi (2021) application to visualize the created graph. Visualization of the graph is shown in Figure 13. The main properties of the graph are presented in Table 3.

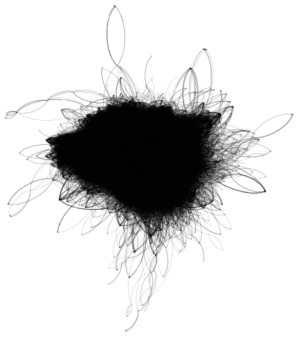

Figure 13. Graph built using imported data from GraphML file

https://doi.org/10.15405/epsbs.2022.08.17
Corresponding Author: D. M. Obolensky
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

Table 3. Skills graph properties

Property name	Value
Vertices	12112
Edges	293670
Type of graph	Undirected
Average degree	48.492
Dense	0.004
Diameter	7
Average path length	2.7136
Modularity	0.332
Clusters	566

6. Findings

Vertices degree distribution, a metric described by Kleinberg (1999) is shown on Figure 14.

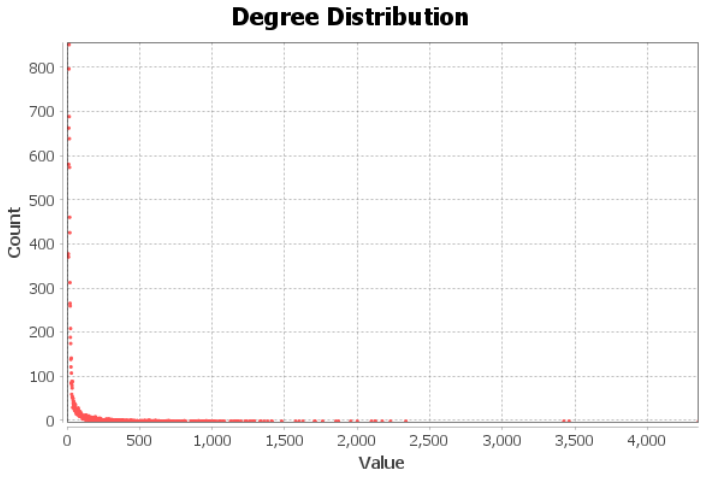

Figure 14. Degree distribution

Hubs distribution, another wideused graph metric according to Kleinberg (1999) and Tarjan (1972) is shown in Figure 15. Figure 16 shows PageRank coefficient distribution. PageRank metric was first described by Brin and Page (1998). Figures 17-20 show distributions for Betwennes Centrality, Closeness Centrality, Harmonic Closeness centrality and Eccentricity (Brandes, 2001).

https://doi.org/10.15405/epsbs.2022.08.17
Corresponding Author: D. M. Obolensky
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

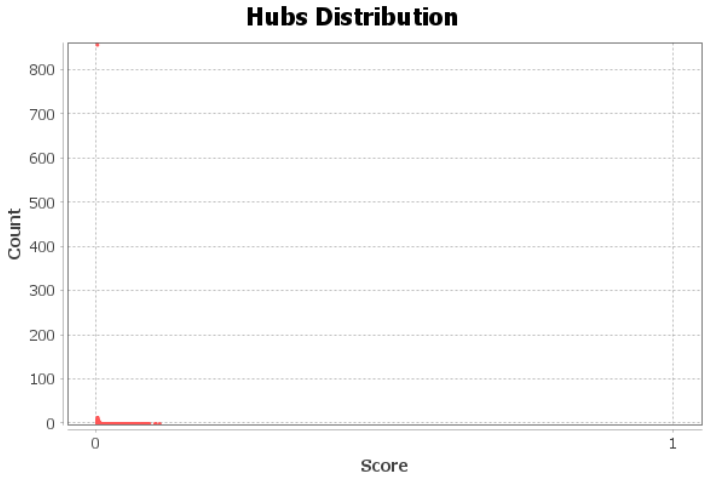

Figure 15. Hubs distribution

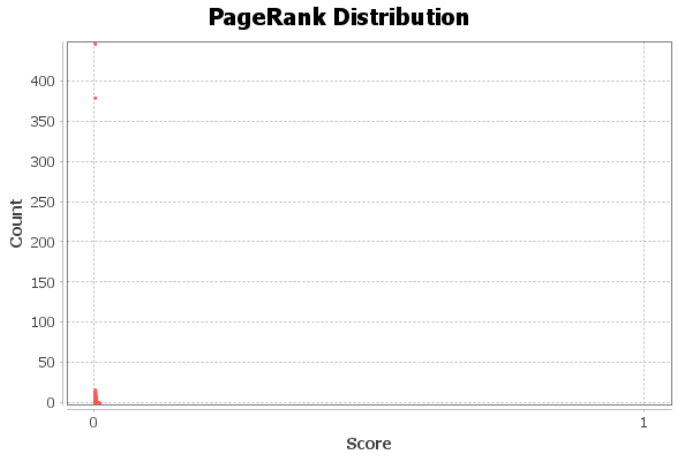

Figure 16. PageRank value distribution for vertices

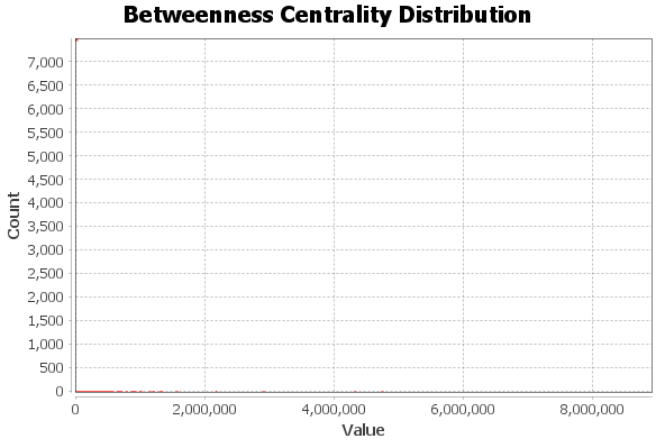

Figure 17. Betwenness Centrality distribution

https://doi.org/10.15405/epsbs.2022.08.17
Corresponding Author: D. M. Obolensky
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

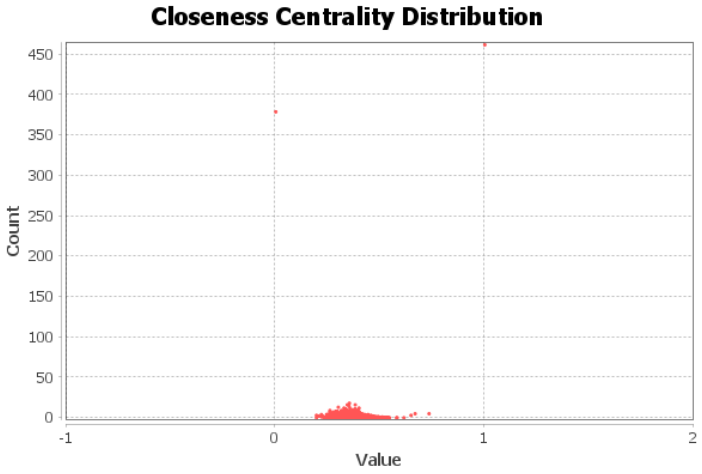

Figure 18. Closeness Centrality distribution

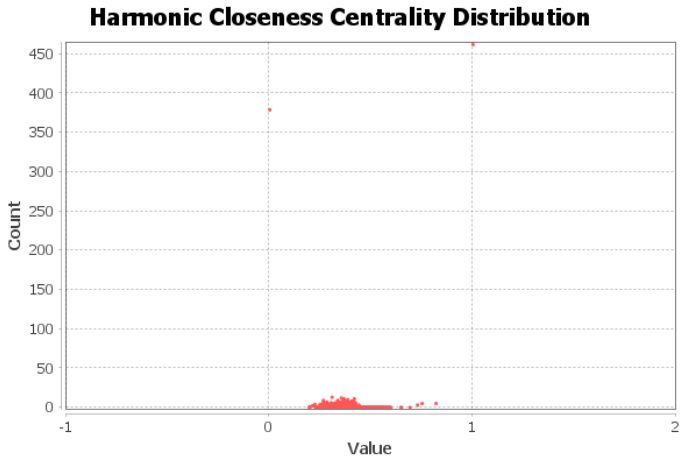

Figure 19. Harmonic Closeness Centrality distribution

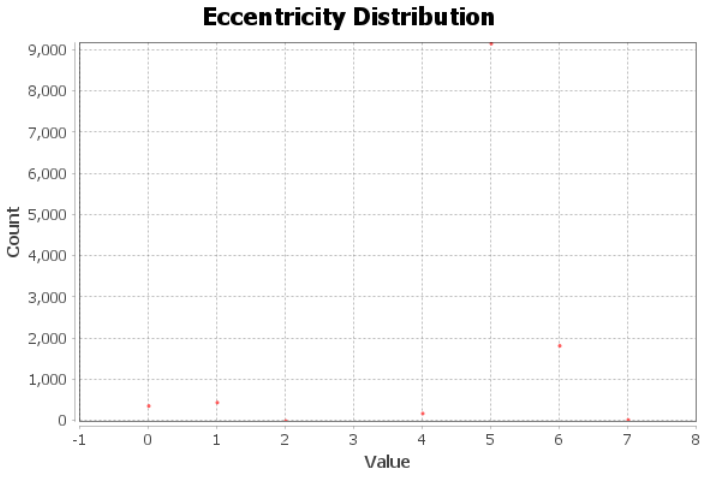

Figure 20. Eccentricity distribution

https://doi.org/10.15405/epsbs.2022.08.17
Corresponding Author: D. M. Obolensky
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

Modularity class distribution is show in Figure 21. Modularity class can be helpful to determine clusters in the graph.

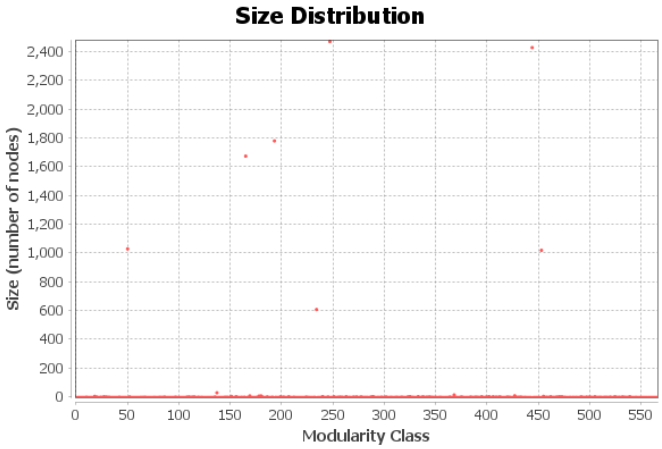

Figure 21. Modularity distribution

With modularity classes available, it's possible to determine clusters (Abramov et al., 2021; Blondel et el., 2008; Lambiotte et al., 2009) in the skills graph using modularity value. Each cluster combines skills that are often used together. The number of clusters (566) almost coincides with the number of specialties on the HeadHunter job search portal (570 excluding the group of specialties "Career Start, students").

Each cluster was assigned a specific color. Each vertex was assigned a size depending on its PageRank value. The higher the PageRank of the vertex is, the larger size the vertex will have in the visualization. Visualization of the skills graph using *ForceAtlas2* layout is shown in Figure 22.

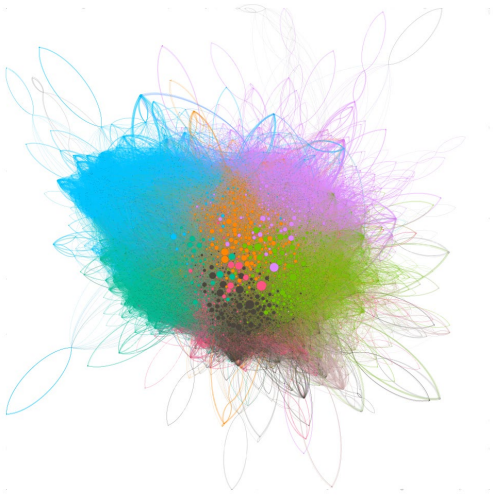

Figure 22. Skills graph with colored clusters, visualized in the Gephi

7. Conclusion

As a result, the connection between vacancies, skills, and specialities was established and formalized using a skill-based approach. Job sites APIs can be used to fetch data about vacancies, skills in automated way. This data can be later processed to build a skills graph.

The application of graph theory has made it possible to combine all the skills required in vacancies into a single graph. The analysis of this graph showed that the number of clusters obtained as a result of the calculation of the modularity class almost coincides with the number of specialties on the job search portal, which makes it possible in the future to determine skills for new specialties, as well as interdisciplinary skills, by analyzing the skills graph for given vacancies.

However, after analysis of graph main metrics, such as degree distribution, centrality and modularity we found, that many skills have semantical duplicates, for example, “Teamwork ability” and “Working in the team” are semantical duplicates for “Teamwork”. While it will go to the same cluster, those aliases, synonyms and semantical duplicates should be revisited in removed in the pre-processing phase.

Further work will continue research in this area. First, we need to analyse skills, find out how to treat aliases in automated way to improve resulting skills graph. Secondly, we will determine the similarity measures between vacancies based on the skill-based approach, which will formalize the principles of work of the recommendation system as the core of the intellectual educational ecosystem.

Also, important tasks that will be solved in the future are finding a way to perform dimensionality reduction of the graph using a minimal spanning tree, as well as the building an acyclic directed skills graph $DACG$.

References

Abramov, A., Filatov, K., Peregrimov, A., & Boganyuk, Y. (2021). Razrabotka servisa dlya opredeleniya aktual'nyh grupp navykov specialista na osnove tekstov vakansii [Development of a service for determining the actual skill groups of a specialist based on vacancy texts]. *Mathematical and information modeling Materials of the All-Russian Conference of Young Scientists. Tyumen, Tyumen: Tyumen State University*, 54-64. https://www.elibrary.ru/item.asp?id=47170628

Aggarwal, C. (2011). An Introduction to Social Network Data Analytics. *Social Network Data Analytics*. *Springer.* https://doi.org/10.1007/978-1-4419-8462-3_1

Batura, T. V. (2012). Metody analiza komp'yuternyh social'nyh setej [The method of analysis of computer social networks.]. *Vestn. NGU, Ser. Informatsionnye tekhnologii*, *10*(4), 13-28. https://lib.nsu.ru/xmlui/handle/nsu/250

Blondel, V., Guillaume, J., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment, 2008*, 1008. https://doi.org/10.1088/1742-5468/2008/10/P10008

Borgatti, P., Everett, G., & Johnson, C. (2013). *Analyzing Social Networks*. Los Angeles: SAGE Publications Ltd, 384. https://www.researchgate.net/publication/281294181_Analyzing_Social_Networks

Brandes, U. (2001). A Faster Algorithm for Betweenness Centrality. *Journal of Mathematical Sociology*, *25*(2), 163-177. https://doi.org/10.1080/0022250X.2001.9990249

Brin, S, & Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems, 30,* 107-117. https://doi.org/10.1016/S0169-7552(98)00110-X

Clauset, A., Newman, M. E., & Moore, C. (2004). Finding community structure in very large networks. *PHYSICAL REVIEW E, 70*(6), 066111. http://dx.doi.org/10.1103/PhysRevE.70.066111

Gephi. (2021). *The Open Graph Viz Platform*. https://gephi.org/

Glebova, E. V., Ivanchenko, P. P., & Anokhin, A. S. (2021). Identifikatsiya trebovanii k professional'nym navykam vypusknikov napravleniya 27.03.01. «Standartizatsiya i metrologiya» na osnove analiza on-lain servisov, spetsializiruyushchikhsya na poiske vakansii. [Identification trebovaniy k professionalnym navykam vypusknikov napravleniya 27.03.01 "standartizatsiya i metrology" na osnove analyza on-line servisov, spetsializiruyushchixsya na poiske vakansiy]. Innovative development of the fishing industry in the context of ensuring food security of the russian federationmaterials of the iv national scientific and technical conference. vladivostok: far eastern state technical fisheries university, 184-188. https://www.elibrary.ru/item.asp?id=44749839

HeadHunter. (2021). *Rabota v Moskve, poisk personala i publikatsiya vakansii [Work in Moscow, search for personnel and publication of vacancies]*. https://hh.ru/

Kleinberg, J. (1999). Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, *46*(5), 604-632. https://doi.org/10.1145/324133.324140

Kolomeichenko, M., Polyakov, I., Chepovskii, A. A., & Chepovskii, A. M. (2019). Detection of Communities in a Graph of Interactive Objects. *Journal of Mathematical Sciences*, *237*(3), 426-431. https://doi.org/10.1007/s10958-019-04168-2

Lambiotte, R., Delvenne, J., & Barahona, M. (2009). Laplacian Dynamics and Multiscale Modular Structure in Networks. *IEEE Transactions on Network Science and Engineering*, *1*(2), 76-90. https://doi.org/10.48550/arXiv.0812.1770

Neo4j. (2021). *Graph Data Platform. Graph Database Management System*. https://neo4j.com/

Obolenskii, D. M., & Shevchenko, V. I. (2020). Kontseptual'naya model' intellektual'noi obrazovatel'noi ekosistemy [Conceptual model intellectual imaging ecosystem] *Ekonomika. Informatika*, Technologies, *47*(2), 390-401. https://doi.org/10.18413/2687-0932-2020-47-2-390-401

Palla, G., Derenyi, I., Farkas, I., & Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, *435*, 814-818. https://doi.org/10.1038/nature03607

SuperJob. (2021). *Rabota v Moskve, svezhie vakansii v Moskve, poisk raboty i rezyume na SuperJob* [Work in Moscow, fresh vacancies in Moscow, job search and resume on SuperEb]. https://superjob.ru/

Tarjan, R. (1972). Depth-First Search and Linear Graph Algorithms. *SIAM Journal on Computing*, *1*(2), 146-160. https://doi.org/10.1137/0201010

TrudVsem. (2021). *Rabota Rossii Obshcherossiiskaya baza vakansii i rezyume [Jobs of Russia All-Russian database of vacancies and resumes]*. https://trudvsem.ru/