Automation of Demand Planning for IT Specialists Based on Ontological Modelling

Denis V. Yarullin, Rustam A. Faizrakhmanov, and Polina Y. Fominykh

Abstract The chapter addresses the issue of correspondence between the skills required by employers and the professional competencies of specialists in the IT field. The discrepancy between the said sets has been highlighted. An approach based on extracting skills from the natural language vacancies texts published on the job aggregators sites is proposed. The method allows for analyzing the required professional competences from the employers' point of view to eliminate the identified differences. Possible ways of structuring the selected skills including ontological modeling and cluster analysis are described. An ontological model has been created to proceed with the hierarchical structuring of the professional competencies set. Skill groups have been formed based on domain knowledge, and cluster analysis has been applied to form workload sets. The method of dynamic cluster formation and skill attribution to a particular group within the domain is described. The applied aspects of the approaches are examined using data of Russian regions and federal states of Germany. The differences between a set of workloads and a skill set are determined. The strengths and weaknesses of the highlighted approaches are described. The automation method of demand planning for IT specialists based on an integrated model combining the described approaches above is suggested. Prospects for its further application are outlined.

Keywords IT-specialist · Skill · Structuring · Ontology · Cluster analysis · Workload sets · Dynamic modeling

D. V. Yarullin (✉) · R. A. Faizrakhmanov · P. Y. Fominykh
Perm National Research Polytechnic University, 29 Komsomolsky prospect, Perm 614990, Russia
e-mail: d.v.yarullin@gmail.com

R. A. Faizrakhmanov
e-mail: Fayzrakhmanov@gmail.com

P. Y. Fominykh
e-mail: phominykh1997@gmail.com

A. G. Kravets et al. (eds.), *Society 5.0: Cyberspace for Advanced Human-Centered Society*, Studies in Systems, Decision and Control 333,
https://doi.org/10.1007/978-3-030-63563-3_4

1 Introduction

The study of the competencies required for a specialist in a particular field could be essential for both an employer and a potential employee. Currently, the skills employers expect from specialists often do not correspond to the skills specialists acquire during their training [1].

By 2013, 15% of Russian companies reported a lack of employees' qualifications. Moreover, the qualification gap in the Russian labor market is not a temporary issue, but a constant and worrying trend [2]. Such skills discrepancies are also observed in European countries. Data analysis revealed a gap between the skills of young professionals and the employers' requirements in different activity sectors. It is important to note that such skills discrepancies are more prevalent in intensively developing fields [3].

In the present study, the issue of skills discrepancy is examined on the example of software engineering specialists and their professional competencies. This field is rapidly developing in the labor market, and the demand for specialists is constantly growing. *HH.ru* job aggregator data analysis shows a vacancy rate 5.5% increase for IT specialists during the period from 2016 to 2018 [4]. At the same time, it is necessary to note that the necessity in IT specialists arises in other areas where intensive processes of automation, digitalization, analytics take place. The demand for IT specialists is observed in all the developed countries, and the staff shortage issue is rather acute [5, 6]. Therefore, closing the gap between the skills required by employers and real specialists' competences is especially important in this field.

We suggest that in order to successfully identify discrepancies between the employers' requirements and the actual skills of candidates, an approach allowing analysis and formalization of the said requirements is necessary. The skills systematization will provide an opportunity to take into account the most valuable competencies during the programs planning for future specialist training.

2 Approaches to Solving the Problem

The proposed approach is based on the method of extracting the employers' requirements from the vacancies texts published on the job aggregator sites. The vacancies descriptions published on the *HH.ru* aggregator were selected as the primary data source [7]. The data were retrieved via API using the "Programmer" query for each Russian region individually. After the vacancies' text retrieval, the skills were collected using Natural Language Processing (NLP) techniques. The method includes the following steps:

1. Splitting the text into sentences;
2. Splitting the sentences into words;
3. Words normalization;
4. Stop-words deletion and filtering;

5. Converting words into vectors [8, 9].

This algorithm was applied to vacancies text. The skills were identified during the filtering step. After the skills were extracted, it was necessary to create a model that would allow structuring the data. Two approaches were considered as possible methods: ontological model and cluster analysis. The ontological model does not require words to be converted into vectors, so the skills set was formed based on the ontology before the words to vectors' conversion step.

The ontological model is a hierarchical structure of the domain concepts. The formation of ontology includes a description of the studied issue using concepts, their attributes, and specific objects. This could be compared to the object-oriented programming paradigm, where concepts are presented as classes, properties are class attributes, and objects are class instances. The advantage of this structure of knowledge organization is the machine processing capability as well as the flexibility and scalability. At the same time, ontology is presented as a holistic model of knowledge [10, 11]. Retrieved skills do not initially have any system organization, so there are no logical links between certain skills. It should also be noted that often vacancies do not explicitly specify a full hierarchy of required skills. For example, a vacancy may mention the Flask framework proficiency, but it does not highlight that Flask is a Python programming language framework. Thus, the vacancy implicitly specifies the following requirements: proficiency in the Python programming language, proficiency in the Flask framework. The integrity of ontology allows us to restore missing dependencies and build a complete list of necessary skills.

A skill was chosen as a key concept forming the structure of the ontology, defined as a fragment of the domain knowledge, which allows performing specific tasks within the domain [12]. Using the automatic construction model of the hypertext denotation graph, the following groups of skills were defined [13]:

1. Programming language;
2. Development environment;
3. Library;
4. Framework;
5. Programming technologies;
6. Operating system;
7. Software;
8. Information transfer protocols/Server;
9. Non-specialized skills.

Figures 1, 2 and 3 show the ontology excerpts based on the vacancies localized in Moscow.

The presented graph has a rather complicated structure and a multilevel dependency. Moreover, an instance of one class can be an instance of another class. It is also worth mentioning that the list of classes in the method is constant.

For the second method of skills structuration, it is necessary to convert words to their vector representations as this method functions in vector space. The "Bag of Words" approach was used in the study. The given algorithm allows us to define

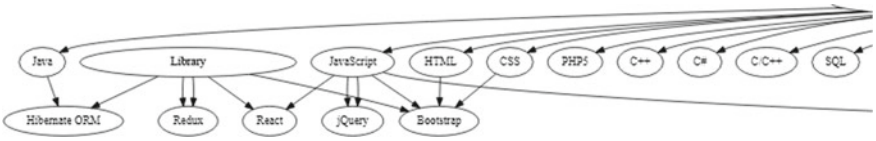
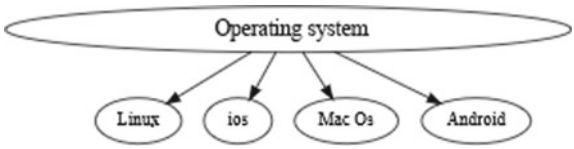


Fig. 1 Fragment of the "Programming languages and libraries" graph for Moscow

Fig. 2 Fragment of the "Operating system" graph for Moscow



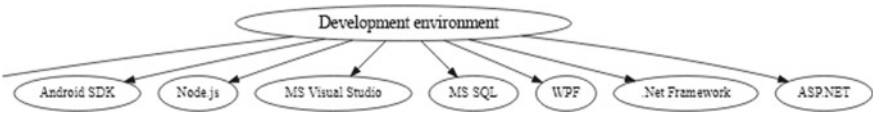


Fig. 3 Fragment of the "Development environment" graph for Moscow

the word usage frequency in the overall text scope. For the attribute calculation, the metric TF-IDF was used, where TF is term frequency, and IDF is inverse document frequency. This approach does not take into account the word order in the text, which can lead to data loss. This drawback is eliminated to some extent by using the N-gram algorithm, which makes it possible to consider not only words but also phrases. A combination of these methods reduces the number of errors in semantic understanding of words with the same spelling but different meanings [14, 15]. The algorithms were implemented using Python programming language, and NLTK library for natural language symbolic and statistical processing [16].

For further processing, cluster analysis was performed to create workload sets. Affinity propagation was chosen as the clustering algorithm. The algorithm automatically determines the structure and number of clusters by passing messages between vector representations of words. When passing the information about the points' location relative to each other, matrices are formed that define the "leader" of the cluster and the points that fall into the cluster with the said leader. Recalculation of the matrices occurs until the system is settled [17, 18]. Cosine similarity had been chosen as a metric determining the elements affinity [19]. The similarity level between vectors A and B is determined by scalar product and vectors normalization using the Formula (1).

$$\text{similarity} = \frac{A \times B}{AB} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \tag{1}$$

Cluster analysis grouped the skills defining workload sets. As a group designation, the cluster central element was chosen (cluster centers are marked in red). In Fig. 4 there is an example of clustering for Moscow.

In the region, 79 skills were identified and grouped into 11 clusters by the algorithm. It is noteworthy that the clusters overlap as a relatively low Silhouette coefficient (0.363) shows. Therefore, it is advisable to conduct an additional study using fuzzy algorithms such as fuzzy c-means.

The algorithm has identified the following groups by their central skills:

1. PHP	7. Android
2. C + +	8. JavaScript
3. PostgreSQL	9. ORACLE
4. iOS	10. JSON API
5. 1C programming	11. Spring Framework

(continued)

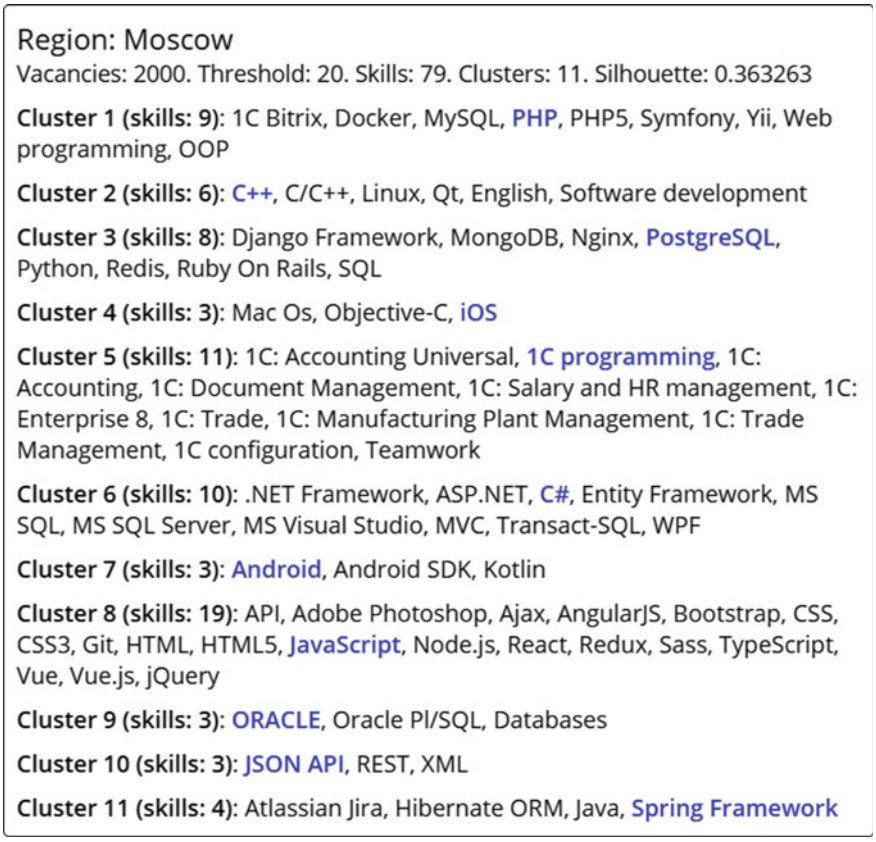


Fig. 4 Cluster analysis result for Moscow

(continued)

6. C#	

This approach involves a simple hierarchy with two levels, and an instance can belong to only one class at a time. It is worth pointing out that the centers and the number of clusters vary depending on the region.

Comparing both approaches to the skill structuring for Moscow, we can see that the ontological model provides a multilevel hierarchy of classes with a strict organization system. Another advantage of structuration via an ontology is the possibility of assigning one element to multiple classes while clustering is intended to create non-crossing sets and therefore does not provide such an opportunity. Nevertheless, developing an ontological model for each region requires significant resources.

Cluster analysis does not allow to identify the vertical hierarchy of the skills within the domain and to highlight the deep implicit dependencies between the skills, but the identification of the key element of the workload sets yields sufficiently good results regarding horizontal skill integration. For instance, the group in Moscow clusters determined by the "1C Programming" skill includes all the skills used for the 1C programming workload. The advantage of this approach is the automation of the group selection process, which reduces the data structuring labor intensity. The approaches differ in the principles that define a skill belonging to a class: in cluster analysis, the group includes a skill that is more similar to all the elements of a given cluster; in ontology, the class is an abstract concept to which the selected skills assigned.

Furthermore, it has to be emphasized that the ontological model is structurally more complex and the structure itself is stricter. Grouping in ontology occurs by category, with a specific category not being a set of skills that are required for a particular field of software engineering. On the contrary, the cluster approach groups skills according to a specific field of IT specialist expertise, the structure of such a model is more dynamic and flexible. A cluster may contain a set of skills that a specialist requires for a particular position.

The heterogeneous nature of the IT field leads us to emphasize the necessity of specifying a region during clusters and ontologies creation. To prove this, let us give an example of clustering results for the Sverdlovsk Oblast region (Fig. 5).

There were 73 skills identified in the Sverdlovsk Oblast region, and the algorithm formed 13 clusters with the following centers:

1. 1C programming	8. iOS
2. JavaScript	9. Spring Framework
3. Qt	10. Java SE
4. PHP	11. Android
5..NET Framework	12. Django Framework
6. SAP	13. MongoDB
7. Project management	

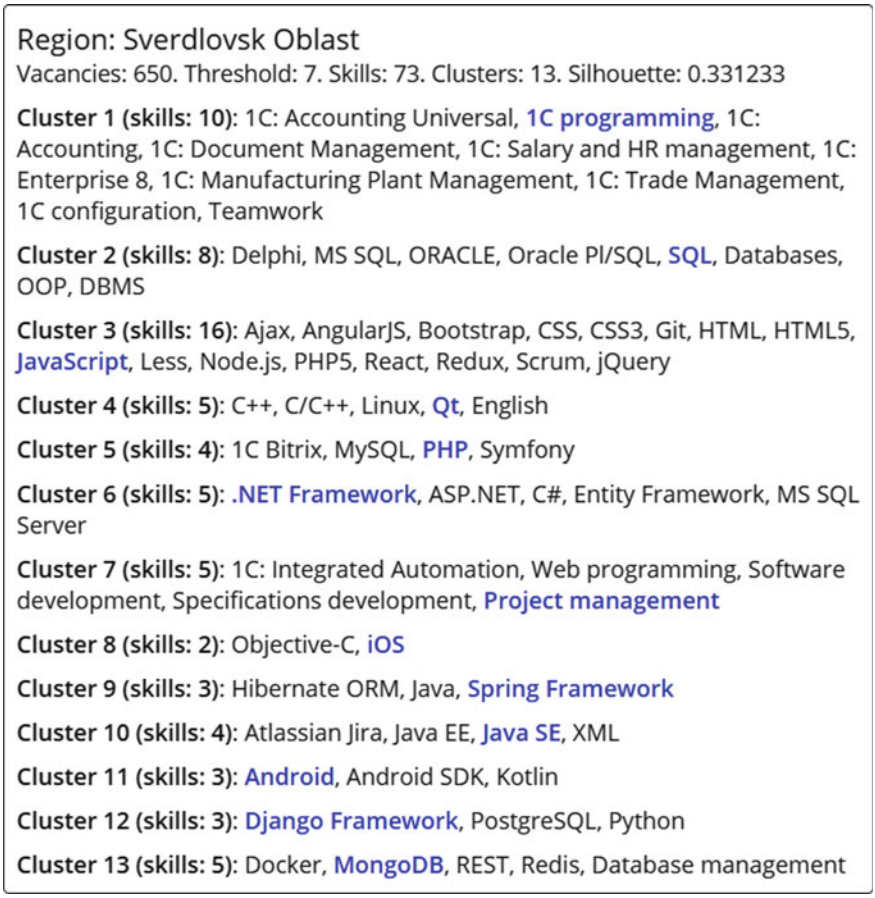


Fig. 5 Cluster analysis for the Sverdlovsk oblast region

Several clusters and their centers in the Sverdlovsk Oblast overlap with those in Moscow. There are even fully matching clusters, for instance, cluster 7 in Moscow and cluster 11 in the Sverdlovsk Oblast have both the same central elements and the same list of skills.

But there are also differences, for example, clusters with the "PHP" central element have different sizes: the set of skills of the Sverdlovsk Oblast region is a subset of Moscow skills. Also, there is no "JSON API" skill in the Sverdlovsk Oblast region, and in Moscow, this skill is the center of cluster 10.

According to this, one can assume that the workloads set of the Sverdlovsk Oblast are a subset of the workloads set of Moscow. However, a more detailed analysis of the cluster structure reveals non-overlapping skills and clusters, for instance, in the Sverdlovsk Oblast there is the "Development of technical tasks" skill, which is not

present in Moscow clusters. Thus, it is confirmed that it is necessary to analyze a set of workloads for each region individually to provide the most complete representation of the local companies demand IT specialists possessing certain professional competences.

To evaluate the quality of the identification of workloads sets for different countries, vacancies in the German federal states were also analyzed. The primary data source was the "Monster.de" job aggregator [20]. The query "Programmierer" was used for data retrieval for each federal state individually. In Fig. 6 the workloads set identified for the federal state of Saxony is presented.

In Saxony, 98 skills were divided into 18 clusters. Noteworthy that in Moscow, 2000 vacancies were analyzed and 79 skills were identified, while in Saxony 98

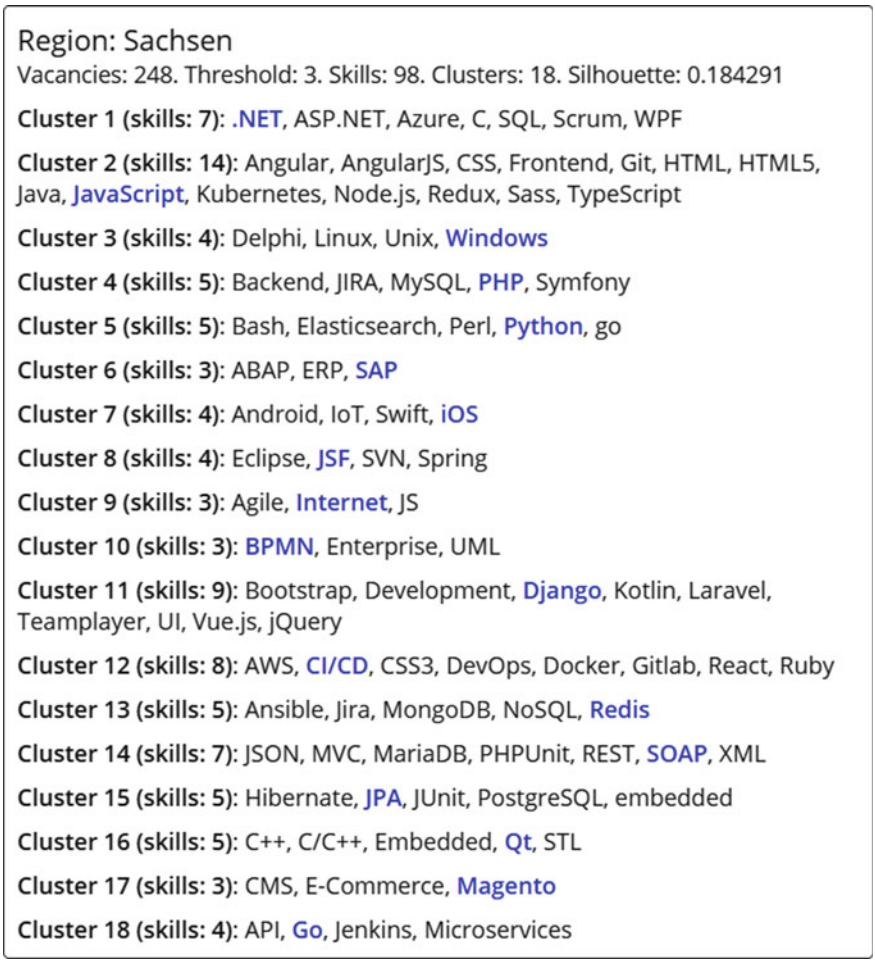


Fig. 6 Cluster analysis for the federal state of Saxony

unique skills were extracted out of 248 vacancies. Hence, there are more vacancies similar to each other in Moscow. It should also be noted that there is more fragmentation than in Moscow. The average cluster size for Moscow is 7.18, and for the federal state of Saxony, it is 5.44. The Silhouette coefficient is even lower than in Moscow (0.184) which leads us to the conclusion that there is indeed a need for further studies using fuzzy methods.

The algorithm has identified the following groups by their central skills:

1. NET	10. BPMN
2. JavaScript	11. Django
3. Windows	12. CI/CD
4. PHP	13. Redis
5. Python	14. SOAP
6. SAP	15. JPA
7. IOS	16. Qt
8. JSF	17. Magento
9. Internet	18. Go

The group formation principle is the same. Let us also remark that Moscow and Saxony both have "PHP", "iOS" and "JavaScript" cluster centers, with "JavaScript" clusters being relatively similar. However, the differences between the clusters as a whole are significant, which once again demonstrates that a skills structure should be created taking into consideration country and region specifics.

3 Conclusion

Both of the outlined approaches quite effectively structure the required skills. For strict formalization and identification of top-level skills, the ontological model is more appropriate. At the same time, the cluster analysis allows to identify a group of skills needed in a particular programming area and is more suitable for creating a list of a particular specialist professional competencies. A combination of these methods will allow, on the one hand, to form a workload-based list of the required skills, and on the other hand, to form a complete list of skills based on the professional competencies' hierarchy within the domain.

Therefore, we can suggest a possible approach to an integrated model combining the advantages of both ontology and cluster analysis. We assume that it can be an automated interpretation of the cluster based on its center and the skills closest to it and its further integration into a broader hierarchy of the domain ontology.

It is easy enough to interpret cluster 7 for Moscow (Fig. 4) as a group of skills required for an Android OS application developer. Describing the cluster as "an Android developer skills", we at the same time integrate it into the ontological model: the "Android" skill is linked to the "Operating systems" class, "Kotlin" is part of

the "Programming languages" class, and "Android SDK" belongs to development environments. Consequently, the requirements for general competencies related to operating systems, programming languages, as well as familiarity with development environments are implicitly imposed on an Android developer. Based on the given interpretation, it is possible to provide an individualized training program aimed at improving the professional level in these particular areas.

The clusters formed for the German federal states can be interpreted similarly. For example, for Saxony, cluster 16 (Fig. 6) can be interpreted as "a C/C++ developer skills". This cluster includes the programming languages themselves ("C" and "C++"), a standard template library for the C++ programming language ("STL"), a cross-platform framework for software development in C++ ("Qt"), a specialized microprocessor-based monitoring and control system that is compatible with Qt ("Embedded"). Thus, this cluster defines C/C++ developer skills as proficiency in programming languages and libraries, the ability to use the frameworks, and the ability to work with embedded systems.

The integrated model implies regional localization of workloads sets. In this case, in order to simplify the model, it is possible not to fragment the ontology by region. It is sufficient to create a unified ontology for all the skills. The skills required for a particular IT area in the region can be drawn from clusters, with the missing skills identified using the different hierarchy levels of the general ontological model.

The proposed approach may become the basis for the decision support system both in the field of human resources management and specialists training.

References

1. Popova, T.N.: (2011) Structural imbalance of the employment system in the region. Modern Econ. Probl. Trends Persp. **5**, 1–6 (2011)
2. Bondarenko N.V.: The nature of the current and expected shortage of workers' professional skills and qualities on the Russian labor market. Public opinion bulletin. Data. Anal. Dis. **3–4**(116), 34–46 (2013)
3. Cedefop: Insights into skill shortages and skill mismatch: learning from Cedefop's European skills and jobs survey, p. 106, 107. Publications Office, Cedefop Reference Series, Luxembourg (2018)
4. IT: Job Market Overview and Top 15 Professions; https://perm.hh.ru/article/24562
5. Zemnukhova, L.V.: (2013) IT workers on the labor market. Soc. Sci. Technol. **4**(2), 77–90 (2013)
6. Eurostat Statistic Explained: ICT Specialists in Employment. Eurostat Statistic Explained; https://ec.europa.eu/eurostat/statistics-explained/index.php/ICT_specialists_in_employment#Number_of_ICT_specialists (2019)
7. HeadHunter API; https://github.com/hhru/api.
8. Jurafsky D., Martin J.H.: Speech and Language Processing 3rd ed. draft, 613 p (2019)
9. Manning, C.D., Raghavan, Schütze, H.: Introduction to Information Retrieval; https://nlp.stanford.edu/IR-book/html/htmledition/irbook.html (2018)
10. Gruber, T.R.: (1993) A translation approach to portable ontology specifications. Knowl. Acquis. **5**(2), 199–220 (1993)
11. Uschold, M., Gruninger, M.: (1996) Ontologies: principles, methods and applications. Knowl. Eng. Rev. **11**(2), 93–136 (1996)

12. Faizrakhmanov R.A., Yarullin D.V.: Web-data driven ontological approach to modelling IT specialists recruitment needs. In: Proceedings of 2019 20th IEEE International Conference on Soft Computing and Measurements (SCM), pp. 252–255 (2019). https://doi.org/10.1109/SCM.2019.8903715
13. Kurushin D.S., Leonov E.R., Soboleva O.V.: A possible approach to automatic construction of the hypertext denotation graph. Information Structure of the Text, pp. 113–118. RAS.INION, Moscow (2018)
14. Kim S., Gil J.: Research paper classification systems based on TF-IDF and LDA schemes. Hum.-Cent. Comput. Info. Sci. **9**(30) (2019)
15. Zhang, Y., Jin, R., Zhou, Z.: (2010) Understanding bag-of-words model: a statistical framework. Int. J. Mach. Learn. Cybern. **1**, 43–52 (2010)
16. Bird, S., Loper, E., Klein, E.: Natural Language Processing With Python. O'Reilly Media Inc., 502 p (2009)
17. Thavikulwat, P.: (2008) Affinity propagation: a clustering algorithm for computer-assisted business simulations and experiential exercises. Develop. Bus. Simul. Experient. Lear. **35**, 220–224 (2008)
18. Frey, B.J., Dueck, D.: (2007) Clustering by passing messages between data points. Science **315**, 972–976 (2007)
19. Han J., Kamber M., Pei J.: Data Mining: Concepts and Techniques 3rd ed. 703 p. Elsevier (2012)
20. Monster Job Search API; https://partner.monster.com/job-search