Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. 2022. Т. 22, вып. 2. С. 250–265
Izvestiya of Saratov University. Mathematics. Mechanics. Informatics, 2022, vol. 22, iss. 2, pp. 250–265
https://mmi.sgu.ru https://doi.org/10.18500/1816-9791-2022-22-2-250-265

Article

Skill-based clustering algorithm for online job advertisements

A. A. Ternikov

HSE University — St. Petersburg, 3A Kantemirovskaya St., Saint Petersburg 194100, Russia
Andrei A. Ternikov, aternikov@hse.ru, https://orcid.org/0000-0003-2354-0109, AuthorID: 956559

Abstract. Clustering on the basis of categorical data is one of the challenging problems in data mining. The paper provides the clustering algorithm for job vacancies using information about the skills required. In the first step, the procedure of unstructured textual information standardization is proposed. The resulting procedures include stages of synonyms and general terms identification based on the combination of TF-IDF and n-grams approaches for translated and transliterated terms. Then, the algorithm is provided and validated on the data obtained from the cross-regional hiring platform. The algorithm provides validation of clusters' extraction, including hierarchical cluster analysis and Girvan – Newman coalition search. Output number of clusters is verified with internal validity scores and suggests disjoint sets of terms that describe particular job occupation groups in the IT sector. Based on obtained clusters well-matched and mismatched terms are identified using Silhouette scores. Given procedures allow to minimize human involvement in clustering itself and produce reasonable clusters for the following interpretation and analysis. In general, the approach for clusters identification based on categorical data is provided and tested on a sample of online job advertisements. It has a high potential in use for feature engineering tasks in machine learning research and applied labor market research in economics.

Keywords: online job advertisements, skill-sets in IT, occupational mismatch, clustering of vacancies, natural language processing

For citation: Ternikov A. A. Skill-based clustering algorithm for online job advertisements. *Izvestiya of Saratov University. Mathematics. Mechanics. Informatics*, 2022, vol. 22, iss. 2, pp. 250–265. https://doi.org/10.18500/1816-9791-2022-22-2-250-265

Научная статья
УДК 51-77

Алгоритм кластеризации на основе навыков для онлайн-объявлений о вакансиях

А. А. Терников

Санкт-Петербургский филиал Национального исследовательского университета «Высшая школа экономики», Россия, 194100, г. Санкт-Петербург, ул. Кантемировская, д. 3А
Терников Андрей Александрович, старший преподаватель департамента менеджмента, aternikov@hse.ru, https://orcid.org/0000-0003-2354-0109, AuthorID: 956559

Аннотация. Кластеризация на основе категориальных данных — одна из сложных задач интеллектуального анализа данных. В статье представлен алгоритм кластеризации вакансий с использованием информации о необходимых навыках. На первом этапе предлагается процедура стандартизации неструктурированной текстовой информации. Полученные процедуры включают этапы идентификации синонимов и общих терминов на основе сочетания подходов TF-IDF и n-граммов для переведенных и транслитерированных терминов. Затем предложенный алгоритм проверяется на данных, полученных с межрегиональной платформы online-рекрутмента. Алгоритм обеспечивает проверку количества извлеченных кластеров, включая иерархический кластерный анализ и коалиционный поиск Гирвана – Ньюмана. Результирующее количество кластеров проверяется при помощи внутренних оценок достоверности и предлагает непересекающиеся наборы терминов, которые описывают определенные группы профессий в секторе информационных технологий. На основе полученных кластеров хорошо совпадающие и несовпадающие термины идентифицируются с использованием индексов Силуэта (Silhouette Index). Указанные в статье процедуры позволяют минимизировать участие человека в процессе кластеризации и создавать интерпретируемые кластеры для последующего анализа. В целом, подход к идентификации кластеров на основе категориальных данных представлен и протестирован на выборке онлайн-объявлений о вакансиях. Он имеет большой потенциал использования для задач формирования факторов в исследованиях машинного обучения и для прикладных исследований рынка труда в экономике.

Ключевые слова: онлайн-объявления о вакансиях, набор навыков в ИТ, несоответствие профессий, кластеризация вакансий, обработка естественного языка

Для цитирования: *Ternikov A. A.* Skill-based clustering algorithm for online job advertisements [*Терников А. А.* Алгоритм кластеризации на основе навыков для онлайн-объявлений о вакансиях] // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. 2022. Т. 22, вып. 2. С. 250–265. https://doi.org/10.18500/1816-9791-2022-22-2-250-265

Introduction

Job seeking process involves employers, employees, state authorities, and the educational system. In order to provide powerful matching signals, different parties disclose skills that are required or possessed. From the side of the employer, skills contain extended information about relevant experience, knowledge, and character required. Especially in the IT-sphere, this information is highly valuable due to incessant changes arising in the era of digitalization. However, new job occupations and professional tasks with their own mix of competences are inevitably created. Thus, the particular set of skills determines not only job occupations themselves but also new job directions in a broad sense. The last-mentioned allows discovering issues of job occupations clustering on the basis of skills required. Accordingly, the appropriate clustering approach should satisfy at least the criteria such as a reasonable number of clusters for interpretation and a stable connection between cluster elements. This paper introduces the novel clustering approach for data structures based on linked sets of elements.

The paper's structure is the following. Section 1 provides an overview of related work about online job advertisements clustering issues and state-of-the-art approaches. Section 2 describes the data sample extraction and processing of unstructured information about skills. Section 3 contains the algorithm description. Section 4 provides results. The last two sections relate to the discussion and concluding remarks.

1. Related work

Several studies highlight issues of the importance of certain skills in the IT-sphere. Firstly, this branch of the labor market relates to the high volatility of technical and soft skills required [1–3]. Secondly, skills, especially technical, have an outstanding structure due to the presence of precise formulation of programming languages, technological stack, interface instruments, etc., so it is easier to classify them in attribution to several job positions [4–7]. Thirdly, the adoption of new technologies requires to changing combinations of skills of workers in order to perform newly created tasks [8–12]. The current paper is aimed to provide the clustering algorithm of vacancies based on information about skills.

The existing clustering approaches used in related works have their advantages and disadvantages. In general, algorithms are sensitive to the data structure and the initial parameters set for a number of clusters. We can highlight the following state-of-the-art clustering algorithms, which take associations between elements in set-based data structures. For example, some researchers use Structural Topic Modeling, in particular, Latent Dirichlet Allocation [13–16] and Latent Semantic Analysis [17, 18], that provide probabilistic matching between textual tokens but the number of topics is set beforehand. The other uses Hierarchical Clustering, which takes a dissimilarity matrix as input, but the results can be unstable because of the empirical choice of appropriate linkage function and the direction of clusters' aggregation [19–21]. In addition, several works introduce multiple-view clustering algorithms, which are based on several input matrices [22–24]. However, the initial parameter setting issues, such as choice of clusters number and cluster collaboration strength is chosen empirically.

Concerning the experimental setting with job advertisements data, many studies provide algorithms for information extraction from online vacancies. However, the way of their implementation differs from the stated research task. For example, if the main research objective relates to the process of matching the unstructured text fields from job advertisements with the official classifier for occupations and skills classification algorithms are implemented for the job titles and extended job descriptions [25–28]. The other researchers use a data-driven approach where obtained data is manually corrected by domain experts in order to provide the appropriate systematization [14, 29].

2. Data collection and processing

2.1. Online job advertisements sample

The data are collected from one of the largest hiring platforms in the CIS (Commonwealth of Independent States) region named HeadHunter (https://hh.ru). The typical structure of an online job advertisement (vacancy) includes the main following fields: vacancy ID, job name, specialization codes (from 1 to 6 professional area codes) publishing date, area (region), description (unstructured text), skills (the set of the size from 0 to 30 elements that consist of unstructured texts each up to 100 symbols). The main stages of the sample creation with the number of observations in parentheses are presented below.

1. IT vacancies extraction (3,066,707 obs.) with "description" fields, containing not less than 30 symbols and at least one "IT" specialization code according to HeadHunter classifier ("1.*": "IT, Internet, Telecom")[1]. Time frame: from May 2015 till September 2019. The subset of specialization codes for "IT" includes 37 IDs in total.

2. Removal of ambiguous occupations by the classifier (2,815,605 obs.). Seven groups are dropped: "Art Director", "Content", "Marketing", "Sales", "Producer", "Business Development", "Other" (8.2% reduction).

[1] HeadHunter API: Specializations, https://api.hh.ru/specializations?locale=EN

3. Identification of portion with skills specified[2] (501,512 obs.): 28.7% out of 1,744,839 vacancies created since 2015.

4. Checking of the sample representation by professional codes: no significant difference between the structure of specialization codes in the sample with skills specified and in vacancies with non-specified skills is found by paired t-test (582 pairs of IT specialization codes mixed with non-IT).

5. Checking of yearly difference for aggregation purposes by IT specialization codes: slight diversification in combinations with the first presented year (2015) in terms of the structure of specializations but in general no significant differences in median values according to sign-test (Table 1).

6. Removal of duplicated entries by both "description" and "skills" fields (351,623 obs.). So, obtained yearly distribution of vacancies: 17,551 in 2015; 53,110 in 2016; 74,560 in 2017; 116,262 in 2018; 90,140 in 2019.

7. Checking the sample represented by the structure of regions where vacancies are published. According to the t-test, the geographical structure of given data preserves in the sample (p-value for 4,457 regions comparison is close to 1). The top-10 regions (cities) in the cleaned sample represents 61.1% of the data (Table 2).

Table 1

Sign-test of paired yearly differences over the structure of IT specialization codes

Year #1	Year #2	p-value
2015	2016	0.01
2015	2017	0.03
2015	2018	**0.07**
2015	2019	**0.24**
2016	2017	**0.87**
2016	2018	**1.00**
2016	2019	**0.41**
2017	2018	**0.62**
2017	2019	**0.24**
2018	2019	**0.62**

Table 2

Distribution of vacancies by regions

Region (city)	Country	Share, %
Moscow	Russia	27.53
Saint-Petersburg	Russia	11.68
Minsk	Belarus	6.02
Novosibirsk	Russia	3.29
Kiev	Ukraine	2.44
Voronezh	Russia	2.34
Ekaterinburg	Russia	2.05
Nizhny Novgorod	Russia	2.01
Almaty	Kazakhstan	1.95
Kazan	Russia	1.77

8. Extraction of all skills from the sample (70,198 unique).

9. Removal of extra punctuation symbols (except #, + at the end) and white-spaces; lowercase applying; then, separation by punctuation symbols followed by white-spaces (60,111 unique).

10. Reduction of rare terms. The setting of the threshold was obtained with Zipf's curve (Fig. 1). The term with the rank $\log 8$ relates to 28 occurrences in the sample, thus, all terms with the lower frequency are removed. As a result of data processing, the sample of IT vacancies consists of 351,623 observations. In addition, the set of 3,034 unique frequent skills is prepared for further standardization.

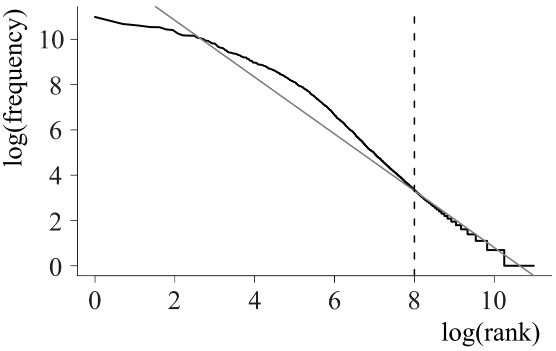

Fig. 1. Zipf's curve and threshold detection

[2] The first vacancy with "skills" field was published in June 2015.

2.2. Skills' standardization

The general logic of skills' (terms') standardization is followed by the steps of similar terms finding (synonyms) and generalized terms aggregation (the common term for the particular subset of skills). Moreover, the steps of matching abbreviations and multilingual terms processing are highlighted separately. In order to minimize the stage of manual processing the following steps are made:

1) removal of extra punctuation and digits;
2) splitting of terms into smaller ones with *regexp* “/| : | : ”;
3) tracking of terms with white spaces (words reordering and stemming);
4) finding potential abbreviations by the first letters extraction from terms with white spaces;
5) translation of terms with Cyrillic letters using Yandex.Translate API[3] (from Russian to English).

In the next step the following databases are obtained:

1) perfectly matched synonyms (281 observations);
2) potential synonyms after stemming (102 groups of 249 observations);
3) potential generalized terms (83 groups of 699 observations);
4) potential abbreviations (35 groups of 99 observations);
5) perfect synonyms after translation (81 observations);
6) potential synonyms after translation (38 groups of 128 observations);
7) potential generalized terms after translation (56 groups of 179 observations).

After manual processing of obtained databases, the terms that are not matched at least in synonyms or generalized terms (1,969 terms) are processed with TF-IDF uni-grams and bi-grams. Then, such processed terms are added to given databases. Finally, such terms are merged with the database of synonyms (2,297 terms) and generalized terms (1,394 terms generalized to 251).

Using the notation, which is given in the initial dataset such terms are matched with obtained synonyms and generalized terms by the number of occurrences of the initial term (or the written in Latin translated analog). Moreover, some narrow in their meaning terms are substituted with generalized terms by the co-occurrence in the initial dataset. So, at the endpoint of this stage 1,730 terms are obtained (including both synonyms and generalized terms only) that are matched with the initial sample data. After matching 343,669 vacancies are stayed with at least one matched term from the obtained dataset out of 351,623 (97.7%).

3. Vacancies' clustering algorithm

The data structure of vacancies (343,669 entries) obtained after the previous step is a 2-tuple $(id, skills)$, where id stands fo ID of the vacancy entry and $skills$ is a subset of standardized skills (several from 1,730 terms) corresponding to such ID. Then, this data structure is transformed into $skills$ representation (1,730 entries), so, each skill has a set of vacancy IDs. In order to perform the clustering procedure, the measure of similarity between $skills$ is needed. Following the data structure (categorical data), the Jaccard similarity index and confidence metrics (over pairs of $skills$) seem to be the most appropriate. In order to find Jaccard indexes, the MinHash procedure (introduced in [30]) with one hash function over 100 members is performed. Based on obtained indexes the relative confidence metrics were calculated.

[3] Yandex.Translate API, https://tech.yandex.com/translate/

The main logic of an algorithm, introduced in this paper, is based on the automatic finding of disjoint clusters of relatively equal size that are ready for human interpretation. Inputs are taken from categorical data and represent dissimilarity matrices. One of the relevant clustering algorithms, hierarchical agglomerative clustering (HAC), deals with symmetric dissimilarity matrices [31], so, three dissimilarity matrices were obtained. Denote them as $\mathbf{J}$ (filled with $1 - Jaccard(A, B)$), $\mathbf{C1}$ (filled with $1 - conf(A \to B)$), and $\mathbf{C2}$ (filled with $1 - conf(B \to A)$), where A, B are sets of IDs for corresponding $skills$.

The concise description of the algorithm and its supporting procedures (see Appendix), which is presented below, includes the following steps.

1. Three dissimilarity square matrices $(\mathbf{J}, \mathbf{C1}, \mathbf{C2})$ are taken as input. Indexes of such matrices represent terms (skills in particular), where the i-th row and column attribute to the same skill. Threshold t (in current data it takes a value of 100) describes the reasonable (for research objectives) number of elements in a particular cluster, where the HAC procedure is needed to be stopped.

2. Detection of aggregated items for all dissimilarity matrices. Iterated HAC procedure with 2-cluster separation is run till the number of elements in the bigger cluster is greater than t.

3. Filtering of obtained clusters by their average size and introducing the pairwise aggregation procedure over clusters using the Girvan – Newman algorithm [32, 33]. Then, rearrangement of duplicated items in obtained clusters for each input matrix. Finally, rearrangement of items in obtained clusters for three input matrices between each other with recalculation of dissimilarity measures.

4. Iterated reduction of the number of clusters by merging smaller-sized clusters with bigger-sized ones obtained with the cut-off.

5. Matching of non-matched terms obtained on the second step with given clusters. Then, the previous step is repeated.

6. The result of the algorithm is a disjoint set of clusters including all input terms.

4. Results

The algorithm proposed in the previous section is applied to the data sample obtained in Section 2. The final number of clusters is 13. Seeing the fact that the data itself is non-labeled and the algorithm takes three dissimilarity matrices as an input only internal validity could be assessed. The choice of the indexes is adapted for the input data in accordance with [34]. So, three measures are assessed: C index [35, 36], Γ [37] and $G(+)$ [38].

The results for all different input matrices and their transformations are presented in Table 3. The clusters, their sizes, and main representatives are given in Table 4.

Table 3

Internal validity

Parameters	C index	Γ	$G(+)$
$\mathbf{J}$	0.19	0.12	0.10
$\mathbf{C1}$	0.25	0.02	0.11
$\mathbf{C2}$	0.39	0.04	0.11
$\max_i \mathbf{H}_i$	0.20	0.12	0.10
$\min_i \mathbf{H}_i$	0.36	0.01	0.11
$\sum_i \mathbf{H}_i/3$	0.30	0.05	0.10

According to provided measures, clusters have a low proportion of disconcordant elements. However, due to the presence of terms that are highly related to several clusters, certainly matched terms could be separated from the other terms for further analysis. Silhouette scores [39] are calculated for such separation. So, 293 terms (16.9% from all) with negative Silhouette scores are supposed to be mismatched (negative values indicating the presence of the

```
**Algorithm** Skill-based Clustering
**Input:** $\mathbf{H} = (\mathbf{J}, \mathbf{C1}, \mathbf{C2})$ is a set of dissimilarity matrices, $t$ is a threshold
**Output:** $C$ stands for obtained clusters
Let $D = (d1, d2, d3)$; $U$ is an ordered set of terms (indexes of input matrices)
*Iterated HAC*
**for** $i$ in $\mathbf{H}$ **do**
    $j := t + 1,\ m := 1$
    **while** $j > t$ **do**
        $s \leftarrow HAC(\mathbf{H}_i)$ with 2 clusters: $s = (s_1, s_2)$
        $s_k = (n_k, u_k)$, $k = \{1, 2\}$, where $n_k$ is a number of items, $u_k$ is a set of items
        $D_i[m] \leftarrow u_g$, where $g = \arg\min(n_1, n_2)$
        $m := m + 1,\ j := \max(n_1, n_2)$
        $\mathbf{H}_i \leftarrow$ sub-matrix of $\mathbf{H}_i$ without rows and columns for $u_g$ terms
    **end while**
    $D_i[m] \leftarrow u_g$, where $g = \arg\max(n_1, n_2)$
**end for**
*Relatively big clusters extraction*
**for** $i$ in $D$ **do**
    $\overline{|d|}$ is a mean number of items among $d_l \in D_i$, where $l \in [1, |D_i|]$
    $D_i \leftarrow \{d_k \subset D_i : |d_k| < \overline{|d|},\ k \in [1, |D_i| - 1]\}$
**end for**
$\hat{D} \leftarrow \textsc{JClust}(D_2, D_3, 0)$
$U \leftarrow \textsc{CClust}(\textsc{JDupl}(\hat{D}_1), \textsc{JDupl}(\hat{D}_2), 0)$
$\hat{U} \leftarrow \textsc{JClust}(U, D_1, 0)$
$S \leftarrow \textsc{CClust}(\textsc{JDupl}(\hat{U}_1), \textsc{JDupl}(\hat{U}_2), 0)$
$A \leftarrow \textsc{JClust}(U, S, 1)$
$B \leftarrow \textsc{CClust}(\textsc{JDupl}(A_1), \textsc{JDupl}(A_2), 1)$
$B1 \leftarrow B_i(1) :\ B_i(2) = 1,\ i \in [1, |B|]$
$B2 \leftarrow B_i(1) :\ B_i(2) = 2,\ i \in [1, |B|]$
$C \leftarrow \textsc{SBClust}(B2, B1, \frac{1}{3})$
**while** $|C| \neq |\textsc{SBClust}(\{c_i(1) \in C^{[i]} : c_i(2) = 1\}, \{c_i(1) \in C^{[i]} : c_i(2) = 2\}, \frac{1}{3})|$ **do**
    $C \leftarrow \textsc{SBClust}(\{c_i(1) \in C^{[i]} : c_i(2) = 1\}, \{c_i(1) \in C^{[i]} : c_i(2) = 2\}, \frac{1}{3})$
**end while**
$F \leftarrow \{U \setminus \{c_i(1) \in C^{[i]}\}\}$ is a set of non-matched terms
$C \leftarrow \textsc{TMatch}(F, C)$
$v := \frac{1}{3}$
**while** $\{c_i(1) \in C^{[i]} : c_i(2) = 1\} \notin \varnothing$ **do**
    **while** $|C| \neq |\textsc{SBClust}(\{c_i(1) \in C^{[i]} : c_i(2) = 1\}, \{c_i(1) \in C^{[i]} : c_i(2) = 2\}, v)|$ **do**
        $C \leftarrow \textsc{SBClust}(\{c_i(1) \in C^{[i]} : c_i(2) = 1\}, \{c_i(1) \in C^{[i]} : c_i(2) = 2\}, v)$
    **end while**
    $v := \frac{v}{2}$
**end while**
$C \leftarrow \{c_i(1) \in C^{[i]} : c_i(2) = 2\}$
**return** $C$
```

term in different clusters), the rest of the terms with non-negative scores indicate the relation to one cluster (well-matched terms). The resulted word clouds are presented in Fig. 2.

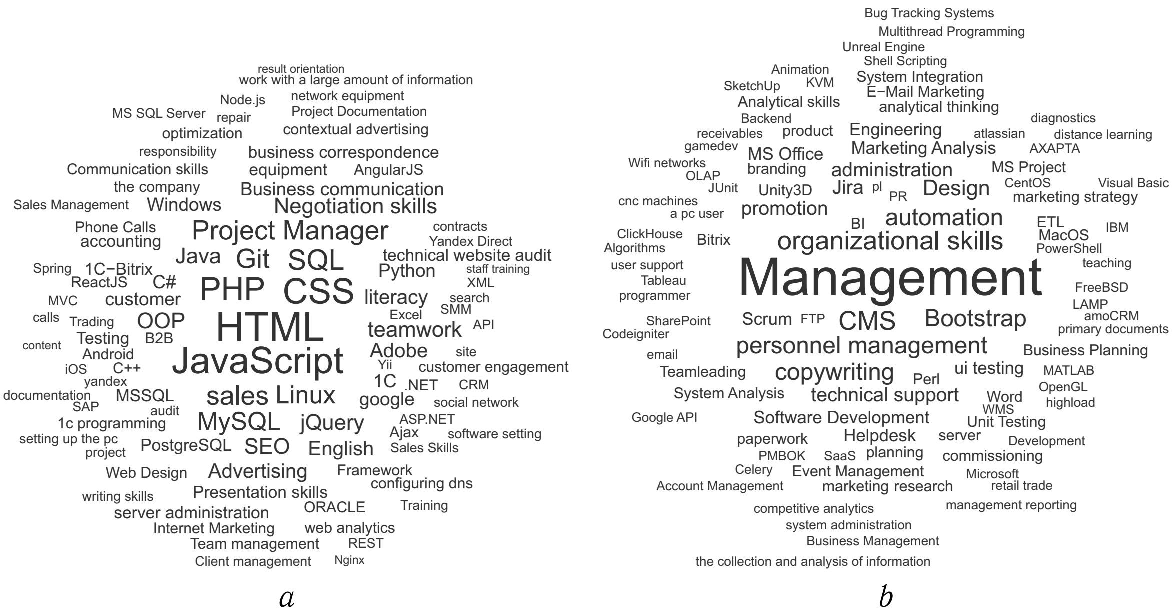

Fig. 2. Word clouds of terms and their frequencies separated by Silhouette scores: *a* stands for top-100 well-matched terms; *b* stands for top-100 mismatched terms

The results of clustering provide some insights for further research. For example, the algorithm detects some supporting areas such as marketing and management that are obtained separately from the main provided IT technologies. So, the researcher may decide the comparison of managerial skills and senior positions or just concentrate on the technical skills only. The other advantage relates to the quite clear separation between "hard" and "soft" skills. It allows to maintain and update already existing databases of such skills (see e.g. [4–6]).

Comparing the introduced approach with the traditional clustering algorithms (for 13 clusters), we mention the relative cluster size and internal validity scores. Our approach allows us to obtain relatively the same-sized clusters based on three dissimilarity matrices (the variation coefficient of the number of cluster elements equals 0.83). However, by providing traditional Hierarchical Clustering with several agglomeration methods[4] over different variants of the input matrix (provided in Table 4) we obtain an unbalanced clusters distribution. Intervals for variation coefficients of cluster numbers in attribution to different methods are as follows: $[1.33, 3.17]$ for "ward.D", $[1.95, 3.18]$ for "ward.D2", $[3.37, 3.57]$ for "UPGMA", $[3.39, 3.53]$ for "WPGMA", $[3.54, 3.58]$ for "WPGMC", $[3.55, 3.58]$ for "UPGMC".

Observing the relative distribution of elements among clusters, we can note that traditional clustering approaches are inferior to Skill-based Clustering, but due to the highly-skewed proportion of cluster members (on average one of the clusters

[4] Clustering was implemented with "hclust" function in "R", https://rdrr.io/cran/fastcluster/man/hclust.html

Table 4

Obtained clusters

Name	Size	Top-15 representatives
Marketing	110	SEO, Advertising, google, technical website audit, contextual advertising, Internet Marketing, optimization, web analytics, Yandex Direct, yandex, social network, site, SMM, search, project
Hardware	184	Linux, Windows, equipment, server administration, configuring dns, network equipment, setting up the pc, software setting, IP, TCP, pc repair, server configuration, Active Directory, Windows Server, maintenance
Big Data	57	Python, C++, BigData, Data Analysis, Machine Learning, SCALA, Hadoop, ElasticSearch, Mathematical Statistics, Data Mining, Spark, Mathematical Modeling, Data Science, Kafka, Cassandra
Software	467	HTML, JavaScript, CSS, PHP, SQL, Git, MySQL, Java, OOP, jQuery, C#, PostgreSQL, MSSQL, 1C-Bitrix, Framework
Administration	96	SAP, C, Unix, Unit Testing, Qt, STL, System Integration, Teamleading, Boost, ABAP, Unreal Engine, ARM, Embedded, teaching, sed
Security	52	audit, Information Security, Cisco, security, competitive analytics, antivirus protection network, means of cryptographic protection of information, technical means of information protection, implementation of information systems, domains, Juniper, audit information systems, SIEM, DLP, Check Point
Web Design	104	Adobe, Testing, organizational skills, copywriting, Bootstrap, Web Design, writing skills, content, Functional testing, UI, Graphic Design, UX, layout, video, writing
Engineering	70	Design, Project Documentation, repair, documentation, Engineering, control, AutoCAD, Visio, gost, automation of processes, process control system, circuitry, programming, circuit design, normative-technical documentation
Analytics	83	Excel, technical support, processing, database, powerpoint, ERP, Reporting, paperwork, Financial Analysis, VBA, financial statements, SAP ERP, work with current customer base, analytical studies, primary documents
Soft Skills	140	Communication skills, responsibility, result orientation, stress resistance, diligence, Customer Support, care, analytical thinking, Event Management, Bitrix, dedication, punctuality, Game Development, E-Mail Marketing, initiative
Management	76	Management, personnel management, administration, ui testing, Recruitment, dbms, BI, Delphi, Business Planning, Web Application Development, Xcode, ExtJS, Strategic Planning, mobile app, personnel evaluation
Testing	80	QA, Business Analysis, Selenium, modeling of processes, UML, BPMN, ITIL, Redmine, ITSM, Blockchain, banking software, IDEF, Test Automation, Quality Control, manual testing
ERP	211	sales, Project Manager, teamwork, Negotiation skills, English, literacy, Business communication, 1C, customer, Presentation skills, business correspondence, accounting, 1c programming, B2B, the company

contains more than 55% of observations) internal validity scores are better. Accordingly, for "ward.D" method the internal validity scores varies as follows: $[0.05, 0.18]$ for C index, $[0.55, 0.89]$ for Γ, $[0.02, 0.11]$ for $G(+)$. Interpreting the obtained results, we denote the higher proportion of compact clusters (with smaller distances between elements) compared with Skill-based Clustering in accordance to Γ. Such performance is caused by the relatively high number of small-sized clusters (typically from five to ten elements) obtained after traditional Hierarchical Clustering. Moreover, the C index also indicates that overall pairwise distances among elements from the same cluster are lower. Interestingly, the $G(+)$ index, which indicates the proportion of disconcordant pairs of elements is not substantially different from Skill-based Clustering procedures.

Thus, the proposed in the paper algorithm mainly deals with the task of relatively same-size clusters identification provides their number, and precisely allocates sub-clusters. Taking into account internal validity scores estimations, the point for the discussion is between interpretability of obtained clusters and sub-clusters detection.

5. Discussion

The algorithm provided in the current paper is performed to prevent unnecessary split of clusters into smaller ones and at the same time to avoid too much clusters aggregation. The research objectives are satisfied in the matter of detection of relatively same-sized clusters which could be described by a human. The novelty of the current study is the skill-based approach and the usage of the initial categorical data structure where only unweighted links between categories exist. However, it is hard to validate the result of the clustering (especially based on multi-lingual data) on the basis of specific classifiers of occupations and skills that are officially introduced and used for job position names validation in the other studies. Moreover, the database of some software developers' skills (e.g. the data from the annual Stack Overflow Developer survey[5]) cannot cover the whole IT sphere with hardware and support specialists. In addition, newly created sets of vacancies broaden the links between several skills from different professional occupations and tasks performed in the modern IT sector.

The overall algorithms' procedures are aimed to automate data pre-processing for further use. Followed by the steps of skills standardization the main proposed algorithm solves the task of particular skills reallocation among the reasonable (interpretative) number of clusters. Thereby, it is difficult to talk about the time and memory complexity of the algorithm because of the combination of different algorithmic techniques and their iterations from one side. On the other side, research objectives do not rely on the inner performance but the result expressed in the content of obtained clusters. Such clusters could be interpreted by the sub-field or purpose of usage in the IT sector. However, there is a tiny amount of entries that might be put into different groups because the algorithm itself is data-driven and it processes only the categorical data flows.

Conclusion

The provided algorithm extends the traditional Hierarchical Clustering by the proposition of several dissimilarity matrices as input, which allows for obtaining more

[5] Stack Overflow Annual Developer Survey, https://insights.stackoverflow.com/survey

stable results (clusters) compared to a traditional approach. Given procedures help to get relatively same-sized clusters for further interpretation by providing the cluster-size threshold. Moreover, the algorithm is scalable for applied data science tasks, which are based on associations inside the sets of elements, in particular for categories in natural language. For example, market basket analysis (clustering of customers' receipts), topic modeling (detection of common semantic patterns), recommendation systems (user experience pattern recognition), etc.

References

1. Bensberg F., Buscher G., Czarnecki C. Digital transformation and IT topics in the consulting industry: A labor market perspective. In: V. Nissen, ed. *Advances in Consulting Research. Contributions to Management Science*. Springer, Cham, 2019, pp. 341–357. https://dx.doi.org/10.1007/978-3-319-95999-3_16
2. Kappelman L., Jones M., Johnson V., McLean E., Boonme K. Skills for success at different stages of an IT professional's career. *Communications of the ACM*, 2016, vol. 59, iss. 8, pp. 64–70. https://dx.doi.org/10.1145/2888391
3. Litecky C., Arnett K., Prabhakar B. The paradox of soft skills versus technical skills in IS hiring. *Journal of Computer Information Systems*, 2004, vol. 45, iss. 1, pp. 69–76. https://doi.org/10.1080/08874417.2004.11645818
4. Börner K., Scrivner O., Gallant M., Ma S., Liu X., Chewning K., Wu L., Evans J. A. Skill discrepancies between research, education, and jobs reveal the critical need to supply soft skills for the data economy. *Proceedings of the National Academy of Sciences*, 2018, vol. 115, iss. 50, pp. 12630–12637. https://dx.doi.org/10.1073/pnas.1804247115
5. Deming D., Kahn L. Skill requirements across firms and labor markets: Evidence from job postings for professionals. *Journal of Labor Economics*, 2017, vol. 36, iss. 1, pp. 337–369. https://dx.doi.org/10.1086/694106
6. Sayfullina L., Malmi E., Kannala J. Learning representations for soft skill matching. In: *Analysis of Images, Social Networks and Texts. AIST 2018*. Lecture Notes in Computer Science (R0), vol. 11179. Springer, Cham, 2018, pp. 141–152. https://doi.org/10.1007/978-3-030-11027-7_15
7. Wowczko I. Skills and vacancy analysis with data mining techniques. *Informatics*, 2015, vol. 2, iss. 4, pp. 31–49. https://dx.doi.org/10.3390/informatics2040031
8. Bailey J., Mitchell R. Industry perceptions of the competencies needed by computer programmers: Technical, business, and soft skills. *Journal of Computer Information Systems*, 2006, vol. 47, iss. 2, pp. 28–33.
9. Brooks N., Greer T., Morris S. Information systems security job advertisement analysis: Skills review and implications for information systems curriculum. *Journal of Education for Business*, 2018, vol. 93, iss. 5, pp. 213–221. https://doi.org/10.1080/08832323.2018.1446893
10. Casado-Lumbreras C., Colomo-Palacios R., Soto-Acosta P. A vision on the evolution of perceptions of professional practice: The case of IT. *International Journal of Human Capital and Information Technology Professionals*, 2015, vol. 6, iss. 2, pp. 65–78. https://doi.org/10.4018/IJHCITP.2015040105
11. Föll P., Thiesse F. Aligning is curriculum with industry skill expectations: A text mining approach. *Proceedings of the 25th European Conference on Information Systems (ECIS)*, Guimarães, Portugal, June 5–10, 2017, pp. 2949–2959.
12. Stal J., Paliwoda-Pękosz G. Fostering development of soft skills in ICT curricula: A case

of a transition economy. *Information Technology for Development*, 2019, vol. 25, iss. 2, pp. 250–274. https://doi.org/10.1080/02681102.2018.1454879

13. Gurcan F., Cagiltay N. Big data software engineering: Analysis of knowledge domains and skill sets using LDA-based topic modeling. *IEEE Access*, 2019, vol. 7, pp. 82541–82552. https://doi.org/10.1109/ACCESS.2019.2924075
14. De Mauro A., Greco M., Grimaldi M., Ritala P. Human resources for Big Data professions: A systematic classification of job roles and required skill sets. *Information Processing & Management*, 2018, vol. 54, iss. 5, pp. 807–817. https://doi.org/10.1016/j.ipm.2017.05.004
15. Xu T., Zhu H., Zhu C., Li P., Xiong H. Measuring the popularity of job skills in recruitment market: A multi-criteria approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI, 2018, vol. 32, iss. 1, pp. 2572–2579.
16. Wu J., Shi H., Yang J. Are big data talents different from business intelligence expertise?: Evidence from text mining using job recruitment advertisements. *2017 International Conference on Service Systems and Service Management*, IEEE, 2017, pp. 1–6. https://doi.org/10.1109/ICSSSM.2017.7996289
17. Debortoli S., Müller O., vom Brocke J. Comparing business intelligence and big data skills. *Business & Information Systems Engineering*, 2014, vol. 6, iss. 5, pp. 289–300. https://doi.org/10.1007/s12599-014-0344-2
18. Radovilsky Z., Hegde V., Acharya A., Uma U. Skills requirements of business data analytics and data science jobs: A comparative analysis. *Journal of Supply Chain and Operations Management*, 2018, vol. 16, iss. 1, pp. 82–101.
19. Aken A., Litecky C., Ahmad A., Nelson J. Mining for computing jobs. *IEEE Software*, 2010, vol. 27, iss. 1, pp. 78–85. https://doi.org/10.1109/MS.2009.150
20. Pejic-Bach M., Bertoncel T., Meško M., Krstić Ž. Text mining of industry 4.0 job advertisements. *International Journal of Information Management*, 2020, vol. 50, pp. 416–431. https://dx.doi.org/10.1016/j.ijinfomgt.2019.07.014
21. Poonnawat W., Pacharawongsakda E., Henchareonlert N. Jobs analysis for business intelligence skills requirements in the ASEAN region: A text mining study. In: *Advances in Intelligent Informatics, Smart Technology and Natural Language Processing. iSAI-NLP 2017*. Advances in Intelligent Systems and Computing, vol. 807. Springer, Cham, 2019, pp. 187–195. https://dx.doi.org/10.1007/978-3-319-94703-7_17
22. De Carvalho F., Lechevallier Y., de Melo F. Partitioning hard clustering algorithms based on multiple dissimilarity matrices. *Pattern Recognition*, 2012, vol. 45, iss. 1, pp. 447–464. https://doi.org/10.1016/j.patcog.2011.05.016
23. Pedrycz W. Collaborative fuzzy clustering. *Pattern Recognition Letters*, 2002, vol. 23, iss. 14, pp. 1675–1686. https://doi.org/10.1016/S0167-8655(02)00130-7
24. Cleuziou G., Exbrayat M., Martin L., Sublemontier J.-H. CoFKM: A centralized method for multiple-view clustering. *ICDM 2009 IEEE 9th International Conference on Data Mining*. Miami, USA, IEEE, 2009, pp. 752–757. https://doi.org/10.1109/ICDM.2009.138
25. Amato F., Boselli R., Cesarini M., Mercorio F., Mezzanzanica M., Moscato V., Persia F., Picariello A. Challenge: Processing web texts for classifying job offers. *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, 2015, pp. 460–463. https://dx.doi.org/10.1109/ICOSC.2015.7050852
26. Boselli R., Cesarini M., Mercorio F., Mezzanzanica M. Classifying online job advertisements through machine learning. *Future Generation Computer Systems*, 2018, vol. 86, pp. 319–328. https://dx.doi.org/10.1016/j.future.2018.03.035
27. Colombo E., Mercorio F., Mezzanzanica M. AI meets labor market: Exploring the link

between automation and skills. *Information Economics and Policy*, 2019, vol. 47, pp. 27–37. https://dx.doi.org/10.1016/j.infoecopol.2019.05.003

28. Lovaglio P., Cesarini M., Mercorio F., Mezzanzanica M. Skills in demand for ICT and statistical occupations: Evidence from web-based job vacancies. *Statistical Analysis and Data Mining*, 2018, vol. 11, iss. 2, pp. 78–91. https://dx.doi.org/10.1002/sam.11372
29. Karakatsanis I., AlKhader W., MacCrory F., Alibasic A., Omar M. A., Aung Z., Woon W. L. Data mining approach to monitoring the requirements of the job market: A case study. *Information Systems*, 2017, vol. 65, pp. 1–6. https://doi.org/10.1016/j.is.2016.10.009
30. Broder A. On the resemblance and containment of documents. In: *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*. IEEE, 1997, pp. 21–29. https://doi.org/10.1109/SEQUEN.1997.666900
31. Murtagh F., Legendre P. Ward's hierarchical agglomerative clustering method: Which algorithms implement Ward's criterion? *Journal of Classification*, 2014, vol. 31, iss. 3, pp. 274–295. https://doi.org/10.1007/S00357-014-9161-Z
32. Girvan M., Newman M. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 2002, vol. 99, iss. 12, pp. 7821–7826. https://doi.org/10.1073/pnas.122653799
33. Newman M., Girvan M. Finding and evaluating community structure in networks. *Physical Review E*, 2004, vol. 69, iss. 2, Art. 026113. https://doi.org/10.1103/PhysRevE.69.026113
34. Milligan G. A Monte Carlo study of thirty internal criterion measures for cluster analysis. *Psychometrika*, 1981, vol. 46, iss. 2, pp. 187–199. https://doi.org/10.1007/BF02293899
35. Dalrymple-Alford E. Measurement of clustering in free recall. *Psychological Bulletin*, 1970, vol. 74, iss. 1, pp. 32–34. https://doi.org/10.1037/H0029393
36. Hubert L., Levin J. A general statistical framework for assessing categorical clustering in free recall. *Psychological Bulletin*, 1976, vol. 83, iss. 6, pp. 1072–1080. https://doi.org/10.1037/0033-2909.83.6.1072
37. Baker F., Hubert L. Measuring the power of hierarchical cluster analysis. *Journal of the American Statistical Association*, 1975, vol. 70, iss. 349, pp. 31–38. https://doi.org/10.1080/01621459.1975.10480256
38. Rohlf F. J. Methods of comparing classifications. *Annual Review of Ecology and Systematics*, 1974, vol. 5, iss. 1, pp. 101–113. https://dx.doi.org/10.1146/annurev.es.05.110174.000533
39. Rousseeuw P. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 1987, vol. 20, pp. 53–65. https://dx.doi.org/10.1016/0377-0427(87)90125-7

Поступила в редакцию / Received 07.08.2021
Принята к публикации / Accepted 08.02.2022
Опубликована / Published 31.05.2022

Appendix

Algorithm's procedures

JClust aggregates clusters on the basis of the Girvan – Newman algorithm and similarity indexes; returns enlarged clusters.

TMatch takes the set of terms and adds them into given clusters using similarity metrics; returns clusters, filled-in with new terms.

Procedure 1

JClust$(X, Y, switcher)$

```
Let $S = X \times Y$; $M$ is an edge weights vector for the Girvan – Newman algorithm
if $switcher = 0$ then
    for $s$ in $S$ do
        $M_s \leftarrow \frac{|S_s(x) \cap S_s(y)|}{|S_s(x) \cup S_s(y)|}$
    end for
    $M \leftarrow M_s : M_s > 1.5 \cdot IQR(M)$
else
    for $s$ in $S$ do
        $M_s \leftarrow \max\left(\frac{|S_s(x) \cap S_s(y)|}{|S_s(x)|}, \frac{|S_s(x) \cap S_s(y)|}{|S_s(y)|}\right)$
    end for
    $M \leftarrow M_s : M_s > \frac{2}{3}$
end if
Let $l$ number of communities obtained with the Girvan – Newman algorithm over $M$
$\mathcal{C}^{[l]} \leftarrow$ the set of community items, $\mathcal{N}$ stands for clusters that are not in communities
$\mathcal{M} \leftarrow (\mathcal{C}, \mathcal{N})$
return $\mathcal{M}$
```

Procedure 2

TMatch(F, C)

```
while $F \notin \varnothing$ do
    for $f$ in $F$ do
        $\widetilde{\mathbf{C2}} \leftarrow$ sub-matrix of $\mathbf{C2}$ with rows stands for the term $F_f$
        Let $k$ be the vector, where $|k| = |C|$; $m := 1$
        for $c$ in $C$ do
            $\widehat{\mathbf{C2}} \leftarrow$ sub-matrix of $\widetilde{\mathbf{C2}}$ with columns stands for $C_c$ terms
            if $\widehat{\mathbf{C2}}_{ij} \notin \varnothing \ \forall i \in [1, |F_f|], j \in [1, |C_c|]$ then
                $k[m] \leftarrow \sum_{i=1}^{|F_f|} \sum_{j=1}^{|C_c|} \widehat{\mathbf{C2}}_{ij}$
            else
                $k[m] \leftarrow 0$
            end if
            $m := m + 1$
        end for
        if $|\arg\max k[m]| = 1$ then
            $C_{\arg\max k[m]}(1) \leftarrow C_{\arg\max k[m]}(1) \cup F_f$
        end if
    end for
    $F \leftarrow \{U \setminus \{c_i(1) \in C^{[i]}\}\}$
end while
return $C$
```

JDupl rearranges duplicated items from the set of overlapping clusters and returns disjoint ones.

Procedure 3

JDupl(X)

```
$m := 1$
**while** $m > 0$ **do**
    Let $H = \{(x_i,\, l_i,\, i) |\, x_i \in X_i,\, l_i = |X_i|,\, i \in [1,\, |X|]\}$
    $H1 = \{(x_i,\, I(\arg\min L)) |\, L \subseteq H_i(2),\, I \subseteq H_i(3) :\, x_i \in H_i\}$
    $\mathcal{X} \leftarrow$ the set of new communities obtained in $H1$
    **if** $\nexists\, i, j :\, H_i(1) = H_j(1),\, i \neq j$ **then**
        $m := 0$
    **end if**
**end while**
**return** $\mathcal{X}$
```

SBClust takes small and large clusters (by the number of elements), merge small clusters to bigger ones with similarity threshold; returns enlarged clusters.

Procedure 4

SBClust($X, Y, threshold$)

```
$t := 1, s := 1$
**for** $x$ in $X$ **do**
    $\widetilde{\mathbf{C1}} \leftarrow$ sub-matrix of $\mathbf{C1}$ with rows stands for $X_x$ terms
    Let $k$ be the vector, where $|k| = |Y|$; $m := 1$
    **for** $y$ in $Y$ **do**
        $\widehat{\mathbf{C1}} \leftarrow$ sub-matrix of $\widetilde{\mathbf{C1}}$ with columns stands for $Y_y$ terms
        **if** $\widehat{\mathbf{C1}}_{ij} \notin \varnothing\ \forall i \in [1, |X_x|], j \in [1, |Y_y|]$ **then**
            $k[m] \leftarrow \max_{1 \leqslant i \leqslant |X_x|, 1 \leqslant j \leqslant |Y_y|} \widehat{\mathbf{C1}}_{ij}$
        **else**
            $k[m] \leftarrow 0$
        **end if**
        $m := m + 1$
    **end for**
    **if** $\max k > threshold$ **then**
        $R_s \leftarrow (t,\, \arg\max k[m])$
        $s := s + 1$
    **end if**
    $t := t + 1$
**end for**
**for** $r$ in $R$ **do**
    $Y_{r(2)} \leftarrow Y_{r(2)} \cup X_{r(1)}$
**end for**
$X \leftarrow X_i : i \notin \{R_j(1) \in R : j \in [1,\, |R|]\}$
$\mathcal{J} \leftarrow \{\{(x, 1) : x \in X\} \cup \{(y, 2) : y \in Y\}\}$ is the set of item sets
**return** $\mathcal{J}$
```

CClust takes item sets and returns rearranged clusters based on recalculated similarity metrics.

Procedure 5

```
CClust($X, Y, switcher$)
Let $S = X \times Y$
for $s$ in $S$ do
    $M_s \leftarrow \frac{|S_s(x) \cap S_s(y)|}{|S_s(y)|}$
end for
$G = \{(i, T_i) : T_i = \sum_{j=1}^{n} M_i,\ i \in [1,\ |X|],\ j \in [1,\ |\hat{X}|] \mid \hat{X} \leftarrow X_i \subseteq S\}$
$G = \{g \in G_i \mid i \in [k]\}$, where $[k]$ stands for indexes of $G_k(2)$ sorted increasingly
for $g$ in $G$ do
    $\hat{S} = \{(x_i, y_i) \mid X_{g(1)} \subseteq S,\ i = g(1)\}$
    for $\hat{s}$ in $\hat{S}$ do
        $M_{\hat{s}} \leftarrow \frac{|\hat{S}_{\hat{s}}(x) \cap \hat{S}_{\hat{s}}(y)|}{|\hat{S}_{\hat{s}}(y)|}$
    end for
    $\hat{S} = \{\hat{s} \in \hat{S}_i \mid i \in [\hat{k}]\}$, where $[\hat{k}]$ stands for indexes of $M_{\hat{s}}$ sorted increasingly
    for $\hat{s}$ in $\hat{S}$ do
        $A := \hat{s}(x),\ B := \hat{s}(y)$
        if $|\hat{s}(x)| < |\hat{s}(y)|$ then
            $A := \hat{s}(y),\ B := \hat{s}(x)$
        end if
        $Q := A \cap B$
        if $M_{\hat{s}} < \frac{1}{3}$ then
            $\hat{S}_{\hat{s}}(y) \leftarrow \hat{S}_{\hat{s}}(y) \setminus Q$
        else
            $\hat{S}_{\hat{s}}(x) \leftarrow \hat{S}_{\hat{s}}(x) \setminus Q$
        end if
    end for
    $S \leftarrow \hat{S}$: re-clustered items
end for
if $switcher = 0$ then
    $\mathcal{J} \leftarrow \{\{X \subseteq S\} \cup \{Y \subseteq S\}\}$ is the set of item sets by rearranged clusters
else
    $\mathcal{J} \leftarrow \{\{(x, 1) : x \in X \subseteq S\} \cup \{(y, 2) : y \in Y \subseteq S\}\}$
end if
return $\mathcal{J}$
```