applied sciences

MDPI

Article

A Novel Stream Mining Approach as Stream-Cluster Feature Tree Algorithm: A Case Study in Turkish Job Postings

Yunus Doğan [1,*], Feriştah Dalkılıç [1], Alp Kut [1], Kemal Can Kara [2] and Uygar Takazoğlu [2]

[1] Department of Computer Engineering, Dokuz Eylül University, Izmir 35390, Türkiye
[2] Kariyer.net R&D Centre, Istanbul 34760, Türkiye
* Correspondence: yunus@cs.deu.edu.tr

Abstract: Large numbers of job postings with complex content can be found on the Internet at present. Therefore, analysis through natural language processing and machine learning techniques plays an important role in the evaluation of job postings. In this study, we propose a novel data structure and a novel algorithm whose aims are effective storage and analysis in data warehouses of big and complex data such as job postings. State-of-the-art approaches in the literature, such as database queries, semantic networking, and clustering algorithms, were tested in this study to compare their results with those of the proposed approach using 100,000 Kariyer.net job postings in Turkish, which can be considered to have an agglutinative language with a grammatical structure differing from that of other languages. The algorithm proposed in this study also utilizes stream logic. Considering the growth potential of job postings, this study aimed to recommend new sub-qualifications to advertisers for new job postings through the analysis of similar postings stored in the system. Finally, complexity and accuracy analyses demonstrate that the proposed approach, using the Cluster Feature approach, can obtain state-of-the-art results on Turkish job posting texts.

Keywords: big data; stream mining; machine learning; clustering; text mining; job posting analysis

Citation: Doğan, Y.; Dalkılıç, F.; Kut, A.; Kara, K.C.; Takazoğlu, U. A Novel Stream Mining Approach as Stream-Cluster Feature Tree Algorithm: A Case Study in Turkish Job Postings. *Appl. Sci.* **2022**, *12*, 7893. https://doi.org/10.3390/app12157893

Academic Editor: Shi-Jinn Horng

Received: 29 June 2022
Accepted: 4 August 2022
Published: 6 August 2022

1. Introduction

It can be seen that job postings are continually diversifying, where even the qualifications sought for the same position are changing on a day-to-day basis. In this rapidly changing era, new sectors are taking their place in the business world, leading to an increasingly differentiated range of qualifications. In [1,2], statistical analyses of job postings during the COVID-19 pandemic have shown that advertisements for some professions decreased or disappeared, while others became more popular. For example, it has been reported that job opportunities in technology-related fields in Canada obviously increased during the COVID-19 pandemic [3]. These changeable stream data require effective approaches for analysis. In this study, a novel algorithm is proposed, which keeps regularly growing job posting data in the flow. We make use of the data of Kariyer.net, which is Turkey's largest employment platform with more than 94,000 company memberships, for analysis. Thanks to the regular structure in which job postings are kept, new job postings are ensured to be comparable with similar ones in the developed system. The key contributions brought by this structure include successfully recommending qualifications to the user for a new posting and creating the most appropriate job posting for the desired position. In recent literature, it can be seen that machine learning approaches have taken over in many different areas [4,5]. In the relevant literature, one can find the thesis study, comparing machine learning approaches, conducted by Thun in 2020 on this subject [6], which motivated this study. In the thesis, an approach making use of a semantic network to ensure that current job postings and the most suitable CVs are matched, was proposed. It was emphasized that job postings can be kept in three ways. The first involves keeping a standard database and pulling job postings with queries. This idea is adapted in our study,

and we tested whether qualifications not included in a specific advertisement can still be used to recommend positions from the most frequently encountered section in the database to qualified candidates. The second approach involves grouping the job postings according to their similarities through the use of clustering algorithms. The algorithm proposed in our study is based on this approach, and other clustering algorithms were tested against the proposed one. The third approach emphasized in the thesis is the semantic network structure [6].

As job postings are maintained in a stream structure and their number has been tending to increase, a data structure in which new postings can be systematically accommodated has become a necessity. For this reason, we considered that the developed approach should include stream mining logic [7]. For this purpose, as the BIRCH (balanced iterative reducing and clustering using hierarchies) algorithm is one of the most prominent approaches in the literature [8,9], which has been used in up-to-date studies [10,11], we also focused on it in our study and experiments. Thanks to the balanced CF-tree (Cluster Features) data structure obtained, the neighbor job postings of a new job posting can be discovered quickly, and based on the texts of these neighboring job postings, new qualifications may be rapidly obtained to make recommendations for new job postings [12]. However, in our tests, we observed that, due to the CF-tree being a balanced tree and the features of hierarchical clustering, the neighborhood logic was not a primary focus. For this reason, in this study a new stream mining clustering algorithm is developed, and we evaluate the BIRCH algorithm as a hybrid approach. Using this novel approach, we are able to obtain focused clustering patterns.

Unlike existing methods, the proposed algorithm uses stream logic. In this manner, a live system is created in which new data can be constantly adapted. Moreover, although a similar approach to the clustering algorithms is proposed in the existing methods, a new and efficient clustering pattern is implemented, as a tree structure is used. The BIRCH algorithm also uses a tree structure; however, when comparing the proposed method with BIRCH, we found that the matching of similar advertisements could be carried out faster and with higher accuracy when using the proposed approach. The contributions and innovations of this study can be listed as follows:

- Analyzing the job postings in Turkish as an agglutinative language by using a stream clustering algorithm for the first time in the literature;
- Keeping these job postings, which are big and stream data, in a novel and more effective data structure than the others in the literature. In addition, this novel approach can be used in any area effectively;
- Discovering the fit positions of new job postings quickly to adapt them to this data structure for the first time in the job postings industry;
- Recommending up-to-date and significant qualifications for new job postings as a decision support system for the first time in the job postings industry

To carry out these operations, first, we aimed to transform the unstructured job postings using natural language processing methods and to keep them in a data warehouse. Then, tests were conducted, comparing the three most effective data retention methods: traditional object relational database usage, stream clustering, and semantic networks. These approaches are presented in Section 3. All details of the study and the proposed algorithm are provided in Section 4. The obtained test and comparison results are also presented, along with graphics and tables, in Section 5.

2. Related Works

Studies on job postings have increased in recent years. Original studies focused on matters such as the discovery of job trends [13], determining the job groups that affect a country's economy the most [14], and predicting the course of the economy [15] have become very popular. For example, Buchmann et al. have determined job trends in Switzerland using job postings [13]. Arthur has discovered the effect on job groups in the U.K. throughout the COVID-19 pandemic [14], and Campos-Vazquez et al. have studied job

postings in Mexico to determine the economic changes during the COVID-19 pandemic [15]. Studies analyzing job posting texts have been carried out using data from various countries and in various fields. Furthermore, a new study field, involving the extraction of soft skills and hard skills separately from job postings, has recently been expanding [16,17]. For example, while Zhang et al. have created an original dataset containing soft and hard skills from job postings about the labor market [16], Lyu and Liu have determined soft and hard skills in the energy sector in their study [17].

The texts of job postings have been evaluated as big data, and for more effective results, it can also be seen that the methods for analysis of job postings in current studies have been diversified by utilizing approaches from the computer science field [18,19]. The multi-method study of Cegielski and Jones-Farmer has shown that hybrid usage of computer science methods can provide more accurate results when analyzing job postings considered big data [20]. In this aspect, data mining and text mining algorithms are the most fundamental approaches [21,22]. In [23], text mining analysis of job postings in the field of librarianship was conducted, and the basic competencies in this field were revealed. It can be found that clustering algorithms are used predominantly for text mining analysis [24,25], even if classification algorithms are used to create general prediction models [26] or specific models, such as those for detecting errors in job postings [27]. Studies have utilized clustering methods for job postings associated with various occupational groups, such as those in the accounting [28], health [29], art [30], engineering [31], and education [32] fields.

In addition to these studies, it can be seen that the effective nature of clustering algorithms for big data has been proven in many applications, such as that in [33]. Thus, the approach proposed in our study was also designed as a clustering algorithm. Moreover, as traditional algorithms, such as *k*-Means, have been shown to yield successful results for specific languages, such as Chinese [34] and Arabic [35], in the analysis of big data job postings, it may be predicted that a new approach based on *k*-Means clustering for the Turkish language will be successful, too. For big data, a satisfying *k*-Means pattern in Chinese has been obtained [36]. Furthermore, non-negative matrix factorization, principal component factor analysis, and DBSCAN algorithms have been tested together with *k*-Means, and a satisfactory *k*-Means pattern was obtained on an Arabic dataset [37]. In line with these current studies, a novel stream mining algorithm using *k*-Means++ is developed in this study. Finally, it can been found that artificial intelligence algorithms, such as dynamic feature weight selection [38], deep learning algorithms [39], and optimization algorithms [38], have been implemented in various areas and obtained successful results.

3. Preliminaries

3.1. Traditional Object Relational Database

Job postings are very difficult to analyze, as they consist of long texts and include free form writing. In addition, due to the continuous increase in the number of postings, the large data volume complicates the analysis. For these reasons, the most traditional method involves keeping the data in a database and analyzing them through queries with keywords obtained after NLP processes. The first method used in our study was a Microsoft SQL Server, due to its ease of management; this approach has been traditionally used in the analysis [39–42]. In our study, 100,000 job postings were stored in the database, with respect to the 3NF rule. Based on the two important aims of the study, the job postings were kept in a regular structure and job postings resembling a new job posting were queried with relevant keywords. In this way, a simple and streamlined approach was obtained. However, as it operates on all job postings (i.e., it does not conduct a preliminary relevant filtering step), it could not give satisfactory results in terms of both complexity and accuracy compared to other approaches. The comparison results are provided in Section 4.

3.2. Semantic Network

A semantic network presents a data structure as a graph, which semantically details the connections between expressions and concepts. The most important example is the Word-

Net project, a Semantic Network product that includes natural language processing [43]. Considering the Semantic Network graph structure, it can be found that the words in a language actually show the concepts that a language expresses. As deep semantic features and Semantic Networks are widely used approaches [44–46], they were tested in our application, and their effectiveness was measured. In this study, a hybrid approach consisting of clustering algorithms and a semantic network is used to obtain the advantages of both approaches. Each cluster is illustrated as a node in our semantic network, allowing for the usage of a Self Organizing Map (SOM), due to its neighborhood feature containing both network and clustering structures [47].

3.3. Traditional Birch Algorithm

In the following, we describe the traditional BIRCH algorithm step by step [48]. First, all job postings converted to an instance format are converted to a *CF* form using the formula $CF = (N, LS, SS)$. When these instances are converted to a *CF* format, a *CF*-Tree is run to aggregate several of the created *CF*s. At this stage, the algorithm requires the *B* (Branching number) parameter. Before evaluating any instance from the dataset, the initial *CF*-tree threshold must be determined. This threshold value is used for each new entry and is static during the clustering process. In a traditional BIRCH, the L value denotes the number of leaves. Additionally, the parameters m and b are included for use in further calculations, where *b* denotes the counter value for branches on non-leaf *CF*s, and *m* denotes the counter value for the branches on the leaf *CF*s. In the next step, for each instance given, the location of the record is compared with the location of each *CF* at the root node, either using the average *CF* or a linear number. The algorithm continues, running the entry to the *CF* root node closest to the entry. Finally, the node is included with the non-leaf child node of the *CF* nodes chosen in the previous step. The locations of instances are compared with the location of each non-leaf *CF*. The algorithm continues running the routine, with respect to the non-leaf *CF* node closest to the entry. Figure 1 shows the process of locating a new centroid in the *CF* Tree.

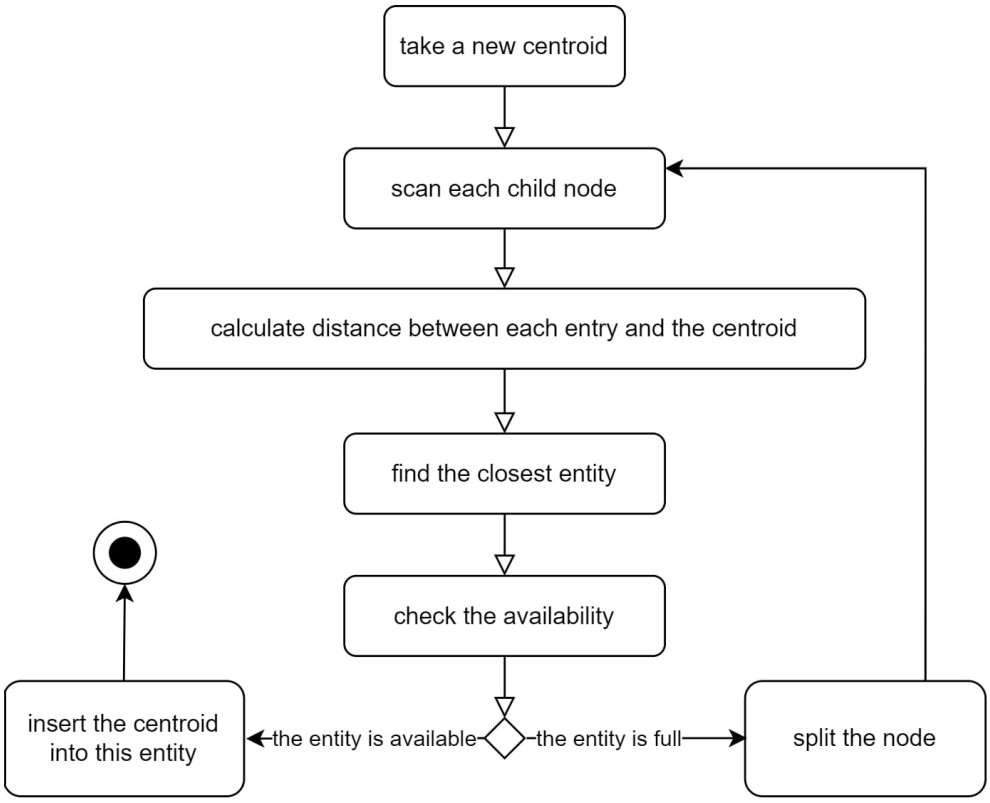

Figure 1. Locating a new centroid in a *CF* Tree.

The cluster size affects the accuracy of BIRCH. The selection of an appropriate threshold is necessary to increase this accuracy. In addition, the performance of BIRCH is affected by the connection methods used to create the nodes in the tree and can vary greatly with the distance measurements used to calculate the distance between data points and centroids [48].

The CF properties contain parameters that directly affect the flow of the algorithm. The descriptions of these properties are given as follows:

- CF properties: N, number of samples for a given data point (x); LS, linear sum of data points; SS, square sum of the data points. While LS consists of vectorial summation of the data points in an entity, SS is the vectorial summation of the squares of the data points in an entity. LS and SS are defined by Equations (1) and (2), respectively:

$$LS = \sum_{i=0}^{N} x_i \tag{1}$$

$$SS = \sum_{i=0}^{N} x_i^2 \tag{2}$$

- Centroid: Can be calculated from the data in a CF at run-time, defined as Equation (3):

$$x_0 = \sum_{i=0}^{N} x_i = \frac{LS}{N} \tag{3}$$

- Radius (R): As shown in Equation (4), this is the sum of the distances from the data points in any cluster to the centroid of the cluster, dividing the result by N, and taking the overall square root. This is calculated for each cluster. The radius of the current cluster is obtained as:

$$R = \sqrt{\frac{\sum_{i=0}^{N}(x_i - x_0)^2}{N}} = \sqrt{\frac{N * SS - 2 * LS^2 + N * LS}{N^2}} \tag{4}$$

- Diameter (D): As in Equation (5), this is taken as the sum of the distances from a data point in any cluster to every other data point of the current cluster, repeated for all clusters, dividing the result by N and its square root. For each cluster, it is calculated. Thus, the diameter of the current cluster is obtained as:

$$D = \sqrt{\frac{\sum_{i=0}^{N}\sum_{j=0}^{N}(x_i - x_j)^2}{N}} = \sqrt{\frac{2 * N * SS - 2 * LS^2}{N^2 - N}} \tag{5}$$

- Equation (6) indicates the change of CF when two clusters, $C1$ and $C2$, are merged. When a new cluster is joined into an entity, this formula is used, and the N, LS, and SS values are summed separately. Only all sub-parameters in CFs are summed separately:

$$CF = CF1 + CF2 = (N1 + N2, LS1 + LS2, SS1 + SS2) \tag{6}$$

4. Methodologies

4.1. Natural Language Processing Methods

Data preparation was carried out on 100,000 Turkish job postings belonging to 8200 random positions from Kariyer.Net. As Kariyer.Net is a web platform, they receive job postings from their users in HTML as free-text. Therefore, in addition to their regular structure (e.g., the position, province, and date), they keep job postings as unstructured data. These large data, which thus required pre-processing, were first converted into pure text using Natural Language Processing (NLP) processes. For this, a Windows Communication Foundation (WCF) project was implemented in the Microsoft Visual Studio framework. Apart from the pure text conversion function in this web service, another function was written in which basic NLP operations were performed for Turkish texts. In this function, the Zemberek library, which has been observed to have high accuracy in many studies, was used [49].

4.1.1. Parsing Job Postings in HTML Tags

As Kariyer.Net has a web-based system, job postings are stored in databases with HTML tags as features. In the WCF project, HTML commands were parsed. In addition, as they were written as free-text, it was necessary to determine the separators of sub-attributes in the job postings. The punctuation marks used for this purpose were determined manually, and sub-attributes were labeled when the use of these special braces was detected within the same function. As a result, the input of this function is the job posting written as free-text in HTML, while the output consists of the sub-attributes in the job posting.

4.1.2. Identifying Keywords in Sub-Attributes

Another function in WCF is the parsing function, which is used for each sub-attribute and determines the important keywords required for NLP operations and data warehousing. This function covers the operations that include all NLP steps, from extracting each word in the Turkish texts to finding the root of the word. For this purpose, the Zemberek library and its parsing, stop word extraction, and root finding methods were used. In this function, three different approaches were developed to create a keyword list. The first approach uses the roots of all words (except the stop words) to distinguish the texts. In the second approach, these word roots are grouped using the 2-Means++ clustering method. Thus, they are clustered as the most widely used and the least used; that is, those specific to the advertisement [50].

4.2. Stream-Cluster Feature Tree Algorithm

In the tests in the study, the problem of "separate clustering of similar data" was encountered due to the parameter dependence of the traditional BIRCH algorithm and the goal of creating a balanced tree. To eliminate this problem, a new stream mining algorithm is proposed in this study, retaining the *CF* features of the BIRCH algorithm.

This algorithm internally uses the *k*-Means++ algorithm to place the data in a binary tree with a top-down hierarchy, as in the DIANA (divisive analysis clustering algorithm). The number of clusters in the *k*-Means++ algorithm was assumed to be constant ($k = 2$). In this way, we aimed to obtain a binary balanced tree.

The algorithm was designed such that each child had 2 entity constants. Initially, all data was clustered with 2-Means++ to form the root, which creates the structure to be included in the entity to contain the CF capsule data.

The S-CF Tree data structure is illustrated in Figure 2 below.

To create the SC-F Tree data structure, job postings, as big data, should first be divided into two Entities using *k*-Means++. After the *k* cluster centers and *k* CF nodes obtained are saved in the Entities, other sub-branches are created recursively. In this way, the algorithm completes the SC-F tree. The algorithm can terminate in two ways:

- $N \leq p$, where N is the number of elements in the entity, and p is the threshold parameter. If $N \leq p$ is encountered in an entity, this entity is named as a leaf, and the algorithm does not continue for this branch;
- The optimal SSE is discovered by using *k*-Means++ for all instances. According to the number of clusters in the optimal pattern, the maximum level on the tree is determined; this number becomes the threshold of recursive calling for a branch.

The S-CF Tree algorithm flow is illustrated in Figure 3 below.

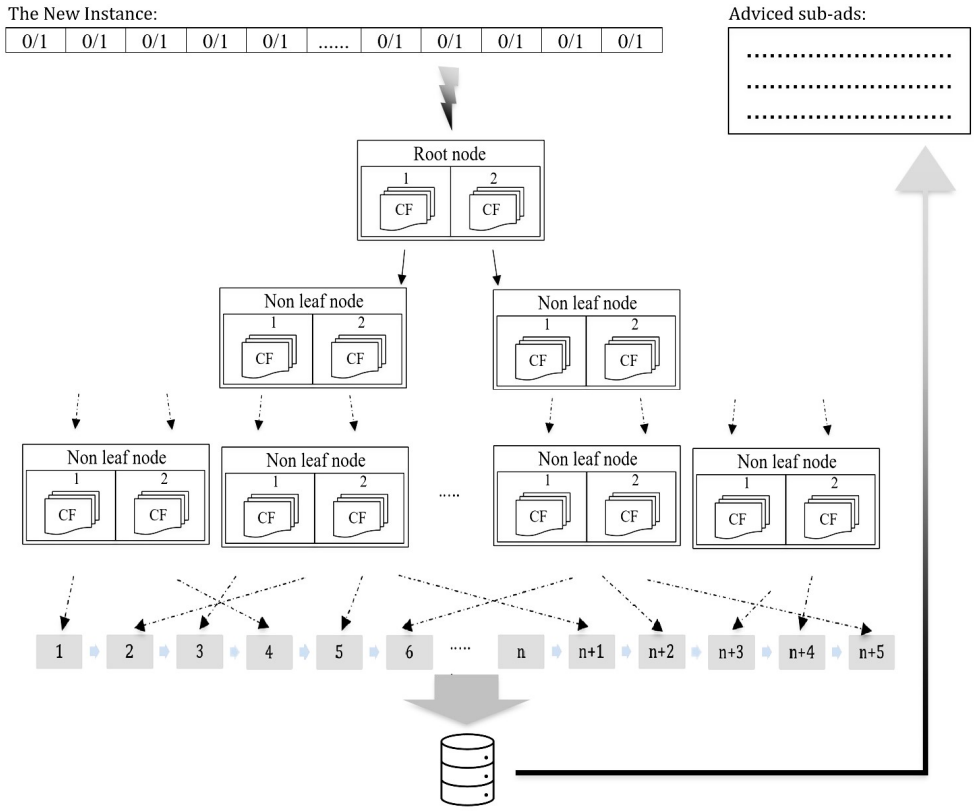

Figure 2. S-CF Tree Data Structure.

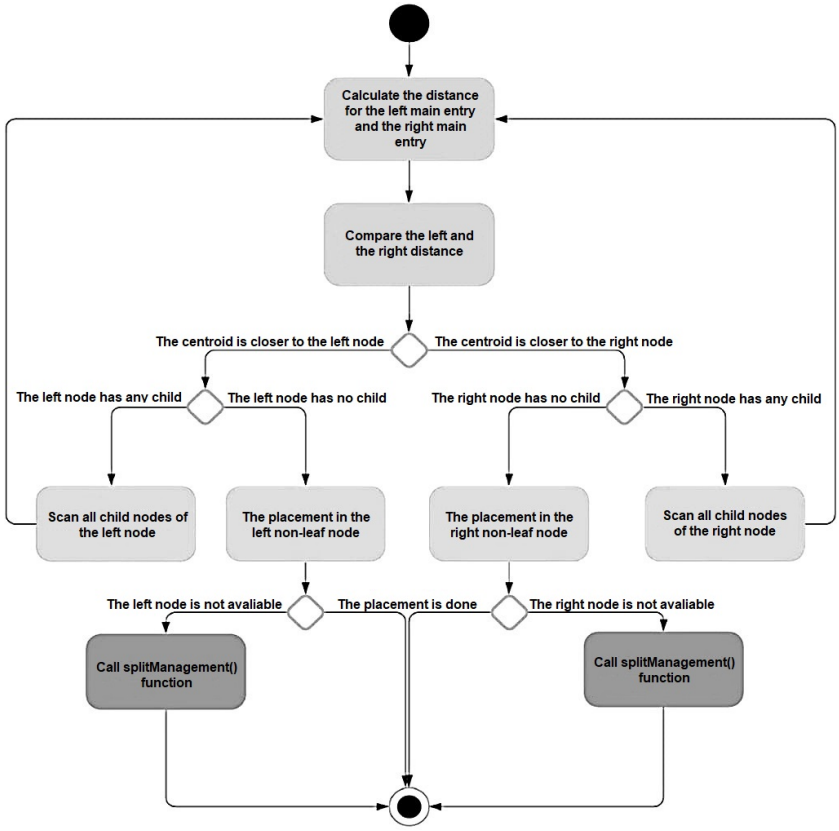

Figure 3. Locating the new centroid in an SC-F Tree for two clusters in each entity.

5. Experimental Studies and Tests

In our experimental studies, five datasets having various properties were tested. Furthermore, the job posting texts in these five datasets belonged to various job categories, from the most popular ones such as Accounting (Acc.), Computer Science and Engineering (CSE), and Marketing (Mar.), to the ordinary ones such as Machinery Industry (MI) and Cooking (Coo.). These popularities were determined according to the associated densities in the Kariyer.net database.

The tests were implemented with two different structures for each dataset. While the features in the first-version datasets consisted of the roots of Turkish words or terms about the current job category, the features in the second-version datasets consisted of the dominant roots of Turkish words or terms about the current job category, which were obtained by analyzing the Frequent Pattern (FP) Growth algorithm. The FP-Growth algorithm can be used for successful feature selection [51]; therefore, in this study, dimensionality reduction through the FP-growth algorithm was implemented for each dataset separately.

5.1. Properties of the Data sets

The tests were conducted in two stages: complexity and accuracy tests. In the first version of the analysis, these tests were implemented for:

- 3109 features, 4312 instances, and 73 positions in the Acc. dataset;
- 2529 features, 3039 instances, and 30 positions in the CSE dataset;
- 1692 features, 2124 instances, and 184 positions in the Mar. dataset;
- 634 features, 982 instances, and 87 positions in the MI dataset;
- 154 features, 120 instances, and 8 positions in the Coo. dataset.

In the second version of the analysis, these tests were implemented for:

- 2081 features, 4312 instances, and 73 positions in the Acc. dataset;
- 1781 features, 3039 instances, and 30 positions in the CSE dataset;
- 1127 features, 2124 instances, and 184 positions in the Mar. dataset;
- 475 features, 982 instances, and 87 positions in the MI dataset;
- 97 features, 120 instances, and 8 positions in the Coo. dataset.

Figure 4 shows the distributions of the numbers of cells in the datasets. The number of cells was calculated by multiplying the number of instances and features. There were two observations: the number of cells before and after conducting feature selection with FP-Growth. It was observed that changes were obtained separately for all datasets in proportion to the original cell numbers. It can be stated that the tests were implemented for datasets ranging from big data to standard-sized data.

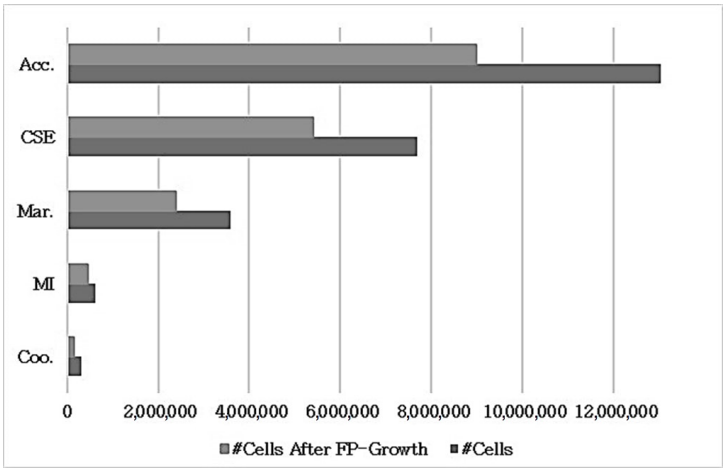

Figure 4. The number of cells in the datasets before and after use of the FP-Growth algorithm.

5.2. Results

While the complexity tests were measured in milliseconds for database queries and the semantic network, the response time was measured as the number of hops within the clusters for the machine Learning algorithms. This is because machine learning algorithms have a better complexity performance than other methods, and it is desirable to more accurately determine the differences between them. For accuracy tests, sum of square error (SSE) values—see Equation (7)—were calculated with respect to the elements in the cluster determined to be included in the new job posting by the clustering algorithms (including the semantic network, as SOM was used). The SSE was calculated between the instance cluster returning from the queries, using the accuracy values for the database query methods and the current new job posting.

$$SSE = \sum_{i=0}^{k} \sum_{j=0}^{n} (x_{ij} - c)^2, \tag{7}$$

where k is the number of clusters, n is the number of data points in the ith cluster, x_{ij} is a data point in the ith cluster, and c is the centroid of the ith cluster.

The complexities of the models were tested through two types of analyses, in terms of their running performance and the number of levels in the trees (i.e., the number of hops to reach the leaves). The number of levels in the trees is an important indicator, detailing how much effort is required to reach the data points in the leaves.

The comparison analyses were carried out using five different methods: database queries (DQ), SOM (used for semantic networking), k-Means++, BIRCH, and the proposed S-CF Tree. The S-CF Tree algorithm was diversified into cases of 2, 3, or 4 clusters in an entity, all of which were tested. In addition, the features of the server where the analyses were conducted were as follows: Intel(R) Xeon(R) CPU E5-2403 0 @ 1.80 GHz, 64 GB memory, 1 TB hard-disk capacity, and Windows Server 2012 R2 operating system. Implementations and tests were performed using the C# language in Microsoft Visual Studio 2019, the Python language in Google Colaboratory, and the R language in R Studio.

In Table 1, the complexity analyses performed for the five different job groups are detailed, preserving all features without the FP-Growth algorithm. The complexity was assessed in terms of milliseconds (ms) by measuring the time required to find job postings close to a new job posting. As expected, the highest time was observed with DQ for all datasets, as there was no previous clustering pattern in DQ. In the other methods, the performances were proportionally distributed according to the size of the dataset.

Table 1. Complexity tests (time) without feature selection.

Algorithms	Acc. (ms)	CSE (ms)	Mar. (ms)	MI (ms)	Coo. (ms)
DQ	32,004	19,010	12,741	7236	685
SOM	11,257	7889	6851	3980	420
KMeans++	9654	6890	6258	4019	478
BIRCH	1690	1409	1183	699	74
S-CF Tree(2)	1147	946	698	412	51
S-CF Tree(3)	1098	944	701	423	48
S-CF Tree(4)	1134	948	697	411	49

The SC-F Tree acts like a balanced search tree. When each node has two children (or, in other words, two clusters), it acts a binary search tree. When each node has three children, it acts as a ternary search tree. Therefore, it can be said that the height of the SC-F Tree becomes log_2^n for the binary structure or log_3^n for the ternary structure. However, while searching, it is also necessary to compare the number of clusters (k) for each node. Therefore, the time complexity of SC-F Tree Operations was O($log_k^n \times$ k).

As an important outcome, we observed that the versions of the proposed S-CF Tree algorithm presented the lowest values, and they had almost the same results, regardless of

whether the number of clusters in any entity was 2, 3, or 4. The explanation for this outcome is that when the number of clusters in any entity increases, the number of levels in the tree decreases. Thus, to reach the cluster where a new job posting is located, it is necessary to compare the number of clusters in each entity from the root. No matter how low the level is to reach the leaves, there will be a performance loss, which is relative to the number of clusters, for the comparison to find the closest cluster in each entity. For example, let the total number of clusters be 16. If we create the tree in a binary structure, the number of levels will be $log_2^{16} = 4$. This means that we must carry out $2 \times 4 = 8$ comparison operations. If we create the tree in a triple structure, the number of levels will be $log_3^{16} = 2.5237$. This means that we must carry out $3 \times 2.5237 = 7.5711$ comparison operations. If we create the tree with a structure of 4, the number of levels will be $log_4^{16} = 2$. This means the $4 \times 2 = 8$ comparison operations must be carried out. In other tests, it was expected that the number of clusters would not have affected the performance obtained by the S-CF Tree algorithm.

As detailed in Table 2, complexity analyses were performed for five different job groups, again preserving all features without the FP-Growth algorithm. The analyses were performed by measuring the number of hops (for tree structures) or comparisons (for cluster patterns) required to find job postings close to a new job posting. As there was no clustering model, DQ could not be evaluated according to the number of hops. In the other methods, it was again observed that the hops were proportionally distributed according to the size of the dataset. Moreover, due to the advantageous nature of the tree structure for the search operation, the BIRCH and S-CF Tree algorithms had the lowest hop counts.

Table 2. Complexity tests (#hops) without feature selection.

Algorithms	Acc.	CSE	Mar.	MI	Coo.
DQ	-	-	-	-	-
SOM	737	307	278	142	16
KMeans++	736	307	278	142	18
BIRCH	29	10	9	8	6
S-CF Tree(2)	20	16	16	16	8
S-CF Tree(3)	21	18	18	18	9
S-CF Tree(4)	20	16	16	16	8

In Figure 5, SOM patterns for all datasets are shown. The coding for this was performed using the R programming language. The outputs were obtained in hexagonal structure to supply the semantic network structure. The densities of instances in the clusters are indicated by different tones of gray. These patterns had the number of clusters set according to the optimal SSE values for all datasets. The patterns show that the counts of instances in each cluster were distributed nearly homogeneously, with only a few clusters presenting small agglomerations. This even distribution provides a smooth environment in which to search for job postings similar to a new job posting.

As shown in Table 3, complexity analyses were performed for five different job groups, again preserving all features without the FP-Growth algorithm. The analyses were performed by measuring the accuracy according to the SSE values. For the accuracy tests of DQs, each job posting was evaluated as a new job posting, and the returned job postings were assumed to be a cluster. The count of job postings illustrated as new job postings was determined as the count of clusters in SOM and *k*-Means++ patterns; furthermore, they were selected randomly. Cross-validation was used for the measurements; thus, each job posting could be evaluated. This approach was repeated until the count of clusters in the SOM and *k*-Means++ patterns had been obtained separately for each dataset. Finally, cluster patterns were obtained, as in the clustering algorithms, and these approaches could be compared with the clustering algorithms. The SSE values were calculated according to the centroids of all clusters. For the first outcome in the table, it was observed that the SSE values of DQs were the highest. The reason for this is that there was not any categorization operation for the queries associating irrelevant job postings to the new job postings. For

the second outcome, the SOM, *k*-Means++, and BIRCH algorithms presented almost the same patterns, with lower SSE values for each dataset. For the last outcome, all versions of the proposed S-CF Tree had the lowest SSE values for each dataset.

Table 3. Accuracy tests without feature selection.

Algorithms	Acc.	CSE	Mar.	MI	Coo.
DQ	10,425	8579	6021	3274	407
SOM	6952	5938	3124	1952	185
Kmeans++	7108	5459	3157	2075	179
BIRCH	7012	5798	3195	1973	183
S-CF Tree(2)	3341	2259	2875	1862	179
S-CF Tree(3)	3358	2129	2856	1903	176
S-CF Tree(4)	3398	2332	2882	1907	181

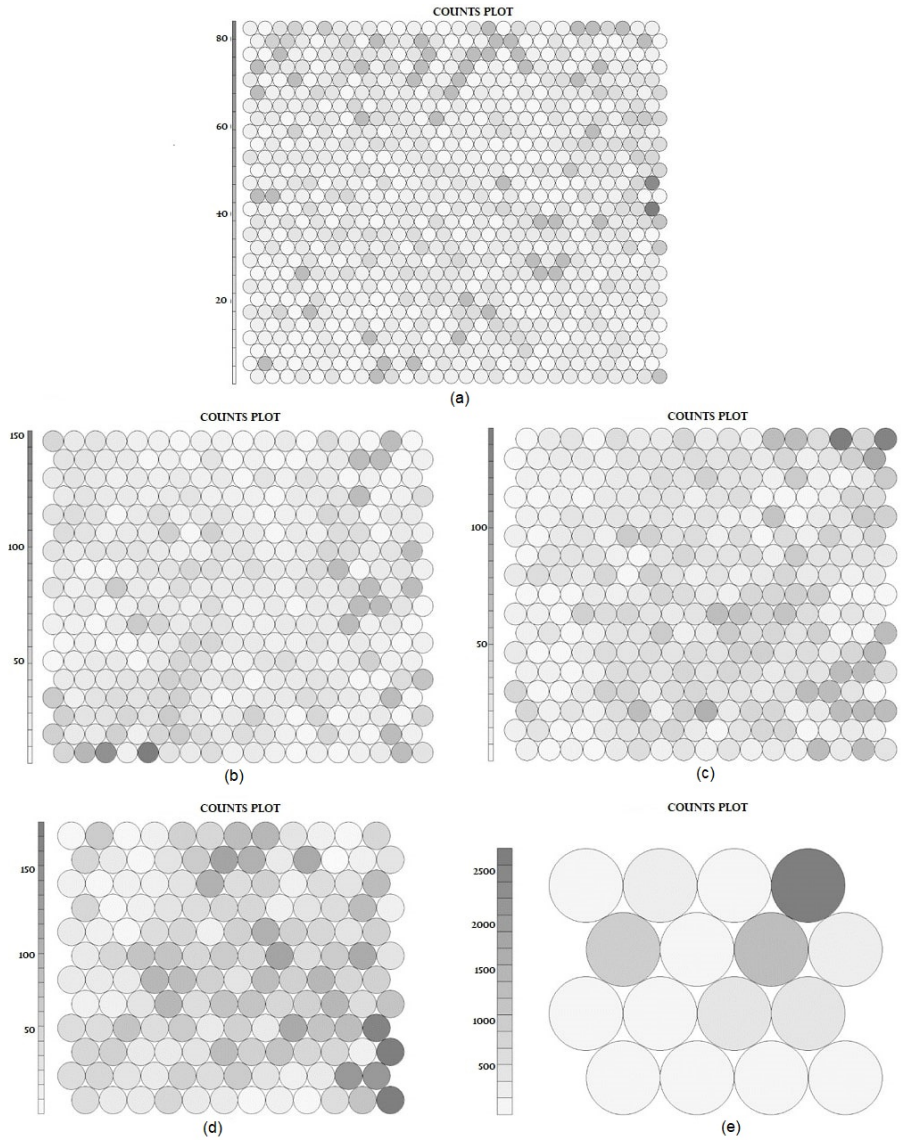

Figure 5. SOM patterns for: (**a**) the Acc. dataset; (**b**) the CSE dataset; (**c**) the Mar. dataset; (**d**) the MI dataset; (**e**) the Coo. dataset.

As the BIRCH algorithm is also a tree algorithm, it was expected to have values close to those of the S-CF Tree algorithm. However, the SSE values of the BIRCH algorithm were almost twice as high. This is because the BIRCH algorithm forces its tree to be created as

a balanced tree. Although the BIRCH algorithm is a very efficient algorithm, in terms of search pattern, it obtained patterns close to those of standard clustering algorithms in our study focused on capturing similar job postings.

In Tables 4 and 5, the complexity analyses for the five different job groups are detailed, where we selected all features using the FP-Growth algorithm.

Table 4. Complexity tests (time) with feature selection (FP-Growth).

Algorithms	Acc. (ms)	CSE (ms)	Mar. (ms)	MI (ms)	Coo. (ms)
DQ	25,122	16,197	9712	5982	496
SOM	7972	4654	4387	2765	287
Kmeans++	6712	3423	4162	2871	301
BIRCH	1187	1019	879	497	58
S-CF Tree(2)	873	578	492	259	57
S-CF Tree(3)	871	581	501	271	59
S-CF Tree(4)	877	583	498	268	58

Table 5. Complexity tests (#hops) with feature selection (FP-Growth).

Algorithms	Acc.	CSE	Mar.	MI	Coo.
DQ	-	-	-	-	-
SOM	737	307	278	142	16
Kmeans++	737	307	278	142	18
BIRCH	29	10	9	8	6
S-CF Tree(2)	20	16	16	16	8
S-CF Tree(3)	21	18	18	18	9
S-CF Tree(4)	20	16	16	16	8

The first outcome of this analysis is that due to the dimensionality reduction operations, noticeable increases in performance were observed for all datasets. As the second outcome, the best performances in terms of both time and number of hops were observed for algorithms using a tree data structure. However, in all versions of the proposed S-CF Tree algorithm, better performances were achieved in terms of time compared to the BIRCH algorithm. Finally, for all datasets, it was found that the variety of the number of clusters in any entity in the S-CF Tree did not have an effect on the hop counts and time.

In Table 6, the complexity analyses performed for five different job groups are detailed, again eliminating some features with the FP-Growth algorithm. The analyses were performed by measuring the accuracy according to the SSE values. The accuracy values of DQs were obtained similar to those in Table 3. The values in Table 6 show that the SSE values slightly increased as some significant features had been eliminated by FP-Growth. It can be observed that the SSE values of the S-CF Tree versions increased, which presented the highest accuracy values by between 6% and 7%.

Table 6. Accuracy tests without feature selection (FP-Growth).

Algorithms	Acc.	CSE	Mar.	MI	Coo.
DQ	12,982	10,079	7129	3609	598
SOM	7235	7001	3862	2198	190
Kmeans++	7249	6909	3871	2012	187
BIRCH	7148	5991	3810	2911	190
S-CF Tree(2)	3597	2388	3019	2617	187
S-CF Tree(3)	3590	2398	3029	2603	182
S-CF Tree(4)	3599	2487	3022	2612	182

6. Conclusions

In this study, a new stream clustering algorithm, S-CF Tree, which uses a tree data structure, was proposed. The proposed model was tested using Turkish data in different job categories from Kariyer.net, the company with the richest job posting content in Turkey. Our model was compared with the BIRCH algorithm, which has similar characteristics, and other approaches previously used in the literature (SOM, *k*-Means++, and DQ). The S-CF Tree model presented the most satisfactory results, both in terms of complexity and accuracy. Consequently, unlike the existing methods, the proposed algorithm uses stream logic. In this way, a live system in which new data can be constantly adapted was created. Moreover, although a similar approach to clustering algorithms was proposed in the existing methods, a new and efficient clustering pattern could be implemented since a tree structure was used. Additionally, it was found that the matching performance for similar job postings was faster and had higher accuracy than that of the existing methods. It is believed that the approach described in this study will encourage the use of new libraries and algorithms in the field of artificial intelligence, considering clustering and association analysis algorithms that have not been used before in the considered field. We believe that these approaches, which are expected to be very important in text mining, will provide fast and effective guides for producing solutions in future studies. In addition, to date, the technical skills (i.e., hard skills) that are effective for recruitment have been studied, and the inference of these qualities from advertisement texts has been studied. In the future, besides technical skills, we can foresee that new studies will be initiated on this subject with new algorithms to be focused on, including studies on soft skills, which will demonstrate the continuity and success of the recruitment process, which have not yet been implemented in Turkish job posting texts.

Author Contributions: Conceptualization, Y.D. and F.D.; methodology, Y.D.; software, Y.D.; validation, Y.D., F.D. and A.K.; formal analysis, Y.D. and K.C.K.; investigation, Y.D. and U.T.; data curation, K.C.K. and U.T.; writing—original draft preparation, Y.D.; writing—review and editing, Y.D., F.D., A.K., K.C.K. and U.T.; visualization, Y.D. and F.D.; supervision, Y.D.; project administration, Y.D. and A.K.; funding acquisition, K.C.K. and U.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Acc	Accounting
B	Branching number
BIRCH	Balanced Iterative Reducing and Clustering Using Hierarchies
CF-tree	Cluster Features
Coo	Cooking
CSE	Computer Science and Engineering
D	Diameter
DIANA	Divisive Analysis Clustering Algorithm
DQ	Database Queries
FP	Frequent Pattern
k	Number of clusters

LS	Linear sum of data points
Mar	Marketing
MI	Machinery Industry
ms	Milliseconds
N	number of samples for a given data point
NLP	Natural Language Processing
R	Radius
SOM	Self Organizing Map
SS	Square sum of the data points
SSE	Sum of Square Error
WCF	Windows Communication Foundation
x	a data point

References

1. Cerioli, M.; Leotta, M.; Ricca, F. COVID-19 hits the job market: An 88 million job ads analysis. In Proceedings of the 36th Annual ACM Symposium on Applied Computing, Gwangju, Korea, 22–26 March 2021; pp. 1721–1726.
2. Marinescu, I.E.; Skandalis, D.; Zhao, D. Job Search, Job Posting and Unemployment Insurance during the COVID-19 Crisis. 2020. Available online: https://ssrn.com/abstract=3664265 (accessed on 29 April 2022).
3. Bellatin, A.; Galassi, G. What COVID-19 May Leave Behind: Technology-Related Job Postings in Canada. 2022. Available online: https://www.iza.org/publications/dp/15209 (accessed on 29 April 2022).
4. Chen, H.; Miao, F.; Chen, Y.; Xiong, Y.; Chen, T. A hyperspectral image classification method using multifeature vectors and optimized KELM. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2021**, *14*, 2781–2795. [CrossRef]
5. Wu, D.; Wu, C. Research on the Time-Dependent Split Delivery Green Vehicle Routing Problem for Fresh Agricultural Products with Multiple Time Windows. *Agriculture* **2022**, *12*, 793. [CrossRef]
6. Thun, A. Matching Job Applicants to Free Text Job Ads Using Semantic Networks and Natural Language Inference. Master's Thesis, Kungliga Tekniska Högskolan School of Electrical Engineering and Computer Science, Stockholm, Sweden, 2020.
7. Verma, A.; Lamsal, K.; Verma, P. An investigation of skill requirements in artificial intelligence and machine learning job advertisements. *Ind. High. Educ.* **2021**, *36*, 63–73. [CrossRef]
8. Ismael, N.; Alzaalan, M.; Ashour, W. Improved multi threshold BIRCH clustering algorithm. *Int. J. Artif. Intell. Appl. Smart Devices* **2014**, *2*, 1–10.
9. Lorbeer, B.; Kosareva, A.; Deva, B.; Softić, D.; Ruppel, P.; Küpper, A. A-BIRCH: Automatic threshold estimation for the BIRCH clustering algorithm. In *Advances in Intelligent Systems and Computing*; Angelov, P., Manolopoulos, Y., Iliadis, L., Roy, A., Vellasco, M., Eds.; Springer International Publishing: Berlin, Germany, 2017; pp. 169–178.
10. Alzu'bi, A.; Barham, M. Automatic BIRCH thresholding with features transformation for hierarchical breast cancer clustering. *Int. J. Electr. Comput. Eng.* **2022**, *12*, 2088–8708. [CrossRef]
11. Gong, J.; Kou, X.; Zhang, H.; Peng, J.; Gong, S.; Wang, S. Automatic web page data extraction through MD5 trigeminal tree and improved BIRCH. In Proceedings of International Conference on Electronic Information Engineering, Big Data, and Computer Technology, Sanya, China, 20–22 January 2022; pp. 387–396.
12. Zhang, T.; Ramakrishnan, R.; Livny, M. BIRCH: An efficient data clustering method for very large databases. *ACM SIGMOD Rec.* **1996**, *25*, 103–114. [CrossRef]
13. Buchmann, M.; Buchs, H.; Busch, F.; Clematide, S.; Gnehm, A.S.; Müller, J. Swiss Job Market Monitor: A Rich Source of Demand-Side Micro Data of the Labour Market. *Eur. Sociol. Rev.* **2022**, jcac002. [CrossRef]
14. Arthur, R. Studying the UK job market during the COVID-19 crisis with online job ads. *PLoS ONE* **2021**, *16*, e0251431. [CrossRef]
15. Campos-Vazquez, R.M.; Esquivel, G.; Badillo, R.Y. How has labor demand been affected by the COVID-19 pandemic? Evidence from job ads in Mexico. *Lat. Am. Econ. Rev.* **2021**, *30*, 1–42. [CrossRef]
16. Zhang, M.; Jensen, K.N.; Sonniks, S.D.; Plank, B. SkillSpan: Hard and Soft Skill Extraction from English Job Postings. *arXiv* **2022**, arXiv:2204.12811.
17. Lyu, W.; Liu, J. Soft skills, hard skills: What matters most? Evidence from job postings. *Appl. Energy* **2021**, 300, 117307. [CrossRef]
18. 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. *Inf. Process. Manag.* **2018**, *54*, 807–817. [CrossRef]
19. Debortoli, S.; Müller, O.; Vom Brocke, J. Comparing business intelligence and big data skills: A text mining study using job advertisements. *Bus. Inf. Syst. Eng.* **2014**, *6*, 289–300. [CrossRef]
20. Cegielski, C.G.; Jones-Farmer, L.A. Knowledge, Skills, and Abilities for Entry-Level Business Analytics Positions: A Multi-Method Study. *Decis. Sci. J. Innov. Educ.* **2016**, *14*, 91–118. [CrossRef]
21. Pejic-Bach, M.; Bertoncel, T.; Meško, M.; Krstić, Z. Text mining of industry 4.0 job advertisements. *Int. J. Inf. Manag.* **2020**, *50*, 416–431. [CrossRef]
22. Wowczko, I. Skills and Vacancy Analysis with Data Mining Techniques. *Informatics* **2015**, *2*, 31–49. [CrossRef]

23. Yang, Q.; Zhang, X.; Du, X.; Bielefield, A.; Liu, Y.Q. Current market demand for core competencies of librarianship—A text mining study of American Library Association's advertisements from 2009 through 2014. *Appl. Sci.* **2016**, *6*, 48. [CrossRef]
24. Lund, B. A cluster and content analysis of data mining studies in Library and Information Science. *Qual. Quant. Methods Libr.* **2021**, *10*, 33–48.
25. Benabderrahmane, S.; Mellouli, N.; Lamolle, M. On the predictive analysis of behavioral massive job data using embedded clustering and deep recurrent neural networks. *Knowl.-Based Syst.* **2018**, *151*, 95–113. [CrossRef]
26. Nasser, I.; Alzaanin, A.H. Machine learning and job posting classification: A comparative study. *Int. J. Eng. Inf. Syst. (IJEAIS)* **2020**, *4*, 6–14.
27. Chern, A.; Liu, Q.; Chao, J.; Goindani, M.; Javed, F. Automatically detecting errors in employer industry classification using job postings. *Data Sci. Eng.* **2018**, *3*, 221–231. [CrossRef]
28. Uwizeyemungu, S.; Bertrand, J.; Poba-Nzaou, P. Patterns underlying required competencies for CPA professionals: A content and cluster analysis of job ads. *Account. Educ.* **2020**, *29*, 109–136. [CrossRef]
29. Goldfarb, A.; Taska, B.; Teodoridis, F. Artificial intelligence in health care? evidence from online job postings. *AEA Pap. Proc.* **2020**, *110*, 400–404. [CrossRef]
30. Wang, V.; Wang, D. The Impact of the Increasing Popularity of Digital Art on the Current Job Market for Artists. *Art Des. Rev.* **2021**, *9*, 242–253. [CrossRef]
31. 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. *Inf. Syst.* **2017**, *65*, 1–6. [CrossRef]
32. Pedulla, D.S.; Muñoz, J.; Wullert, K.E.; Dias, F.A. Field Experiments and Job Posting Sources: The Consequences of Job Database Selection for Estimates of Racial Discrimination. *Sociol. Race Ethn.* **2022**, *8*, 26–42. [CrossRef]
33. Ibrahim Hayatu, H.; Mohammed, A.; Barroon Isma'eel, A. Big Data Clustering Techniques: Recent Advances and Survey. *Mach. Learn. Data Min. Emerg. Trend Cyber Dyn.* **2021**, 57–79._3. [CrossRef]
34. Debao, D.; Yinxia, M.; Min, Z. Analysis of big data job requirements based on K-means text clustering in China. *PLoS ONE* **2021**, *16*, e0255419. [CrossRef]
35. Al Junaibi, R.; Omar, M.; Aung, Z.; Alibasic, A.; Westerman, G.; Woon, W.L. Evaluating Skills Dimensions: Case Study on Occupational Changes in the UAE. In Proceedings of 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, 3–7 November 2019; pp. 1–8.
36. An, Z.; Wang, X.; Li, B.; Xiang, Z.; Zhang, B. Robust visual tracking for UAVs with dynamic feature weight selection. *Appl. Intell.* **2022**. [CrossRef]
37. Hongru, C.; Haidong, S.; Xiang Z.; Qianwang D.; Xingkai Y.; Jianping X. Unsupervised domain-share CNN for machine fault transfer diagnosis from steady speeds to time-varying speeds. *J. Manuf. Syst.* **2022**, *62*, 186–198. [CrossRef]
38. Zhou, X.B.; Ma, H.J.; Gu J.G.; Chen, H.L.; Deng, W. Parameter adaptation-based ant colony optimization with dynamic hybrid mechanism. *Eng. Appl. Artif. Intel.* **2022**, *114*, 105139. [CrossRef]
39. Singh, A.; Rose, C.; Visweswariah, K.; Chenthamarakshan, V.; Kambhatla, N. PROSPECT: A system for screening candidates for recruitment. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, Barcelona, Spain, 16–18 October 2010; pp. 659–668.
40. Muthyala, R.; Wood, S.; Jin, Y.; Qin, Y.; Gao, H.; Rai, A. Data-driven job search engine using skills and company attribute filters. In Proceedings of 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 18–21 November 2017; pp. 199–206.
41. Omasa, A.; Inoue, U. Extracting Related Concepts from Wikipedia by Using a Graph Database System. In Proceedings of the 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Toyama, Japan, 8–10 July 2019; pp. 268–273.
42. Guo, S.; Alamudun, F.; Hammond, T. RésuMatcher: A personalized résumé-job matching system. *Expert Syst. Appl.* **2016**, *60*, 169–182. [CrossRef]
43. Fellbaum, C; Miller, G. A semantic network of English verbs. In *WordNet: An Electronic Lexical Database*; Fellbaum, C., Ed.; MIT Press: Cambridge, UK, 1998; pp. 153–178.
44. Jung, H.; Lee, B.G. Research trends in text mining: Semantic network and main path analysis of selected journals. *Expert Syst. Appl.* **2020**, *162*, 113851. [CrossRef]
45. Maree, M.; Kmail, A.B.; Belkhatir, M. Analysis and shortcomings of e-recruitment systems: Towards a semantics based approach addressing knowledge incompleteness and limited domain coverage. *J. Inf. Sci.* **2019**, *45*, 713–735. [CrossRef]
46. Suchanek, F.M.; Kasneci, G.; Weikum, G. "Yago: A Core of Semantic Knowledge". In Proceedings of 16th International Conference on the World Wide Web, Banff, AB, Canada, 8–12 May 2007; pp. 697–706.
47. Melin, P.; Monica, J.C.; Sanchez, D.; Castillo, O. Analysis of spatial spread relationships of coronavirus (COVID-19) pandemic in the world using self organizing maps. *Chaos Solitons Fractals* **2020**, *138*, 109917. [CrossRef]
48. Lorbeer, B.; Kosareva, A.; Deva, B.; Softić, D.; Ruppel, P.; Küpper, A. Variations on the clustering algorithm BIRCH. *Big Data Res.* **2018**, *11*, 44–53. [CrossRef]
49. Akın, A.A.; Akın, M.D. Zemberek, an open source NLP framework for Turkic languages. *Structure* **2007**, *10*, 1–5. [CrossRef]

50. Arthur, D.; Vassilvitskii, S. K-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, New Orleans, LA, USA, 7–9 January 2007; pp. 1027–1035.
51. Vishwakarma, S.K.; Sharma, A.K.; Verma, S.S.; Utmal, M. Text Classification Using FP-Growth Association Rule and Updating the Term Weight. In Proceedings of Innovations in Information and Communication Technologies (IICT-2020), Delhi, India, 7–8 November 2020; pp. 401–405.