Determining Column Numbers in Résumés with Clustering

Sultan N. Turhan, Şeref Recep Keskin, Yavuz Balı, Günce Keziban Orman, F. Serhan Daniş

Determining Column Numbers in Résumés with Clustering

Şeref Recep Keskin[1], Yavuz Balı[1], Günce Keziban Orman[2], F. Serhan Daniş[2], and Sultan N. Turhan[2]

[1] Kariyer.net Ümraniye, İstanbul, Turkey
{seref.keskin, yavuz.bali}@kariyer.net
[2] Department of Computer Engineering, Galatasaray University, İstanbul, Turkey
{korman, sdanis, sturhan}@gsu.edu.tr

Abstract. In the recruitment process, the workload of manual résumé reviews is quite time consuming for the recruiters. This review process can benefit from Artificial Intelligent-aided intelligent systems to extract the actual meaning within the résumés and structure their forms. However, writing résumés has no standards, and the personalized structure of each received résumé makes this task highly challenging. This work is dedicated to tackling a part of this issue on structuring résumés. More specifically, we firstly focus on finding the column number of any résumé since once the main parts of the résumé are separated, the subdivisions can easily be analysed. This study, thus, formalizes the problem of finding columns of a résumé as a clustering problem. The experiments are performed on a data set of custom Turkish résumés having up to two-columns, on which we apply two algorithms: K-means and Density-based spatial clustering of applications with noise. As a result of the experiments, we observe that an optimal cluster size relates strongly to the valid column number. Our method is not limited to résumés but can be applied to any unstructured textual data.

Keywords: Information Extraction · Résumé Parse · DBSCAN · K-means.

1 Introduction

The process of selectively structuring and combining implicitly or directly specified contents in textual data is called "Information Extraction" (IE) [7]. IE can be rule-based or model-based. One of the most recent famous IE problems is the information extraction from documents [2]. With the increasing data processing power, researchers are more confident in tackling IE tasks related to documents or texts. Résumés are regarded as valuable documents for document extraction with their varying structures and rich content. IE from résumés is the process of automatically generating or extracting specific phrases or meanings. Because manually assessing résumés is a time-consuming and labor-intensive task for recruiters, this process has a significant positive impact on the review process.

As Turkey's largest employment platform since 1999, Kariyer.net brings together job seekers and employers online with new generation technologies in job search and recruitment processes. On the platform offered by Kariyer.net, candidates are required to fill in various fields such as education information, past work experience, and personal information during registration. Besides, users can upload their free-style résumés to the system. These résumés are stored unprocessed in Kariyer.net databases. The unprocessed free-style résumés in PDF or other formats uploaded to the system by users will be called *unstructured*. After being exposed to IE procesdures, the data are stored with a particular hierarchy in the Kariyer.net database and will called *structured data*. It is essential to convert unstructured data into structured form since structured résumés allow IE processes. In this work, we are interested in this issue. More than 700,000 free-style unstructured résumés have been uploaded to the Kariyer.net database by users. Collecting information from each of these résumés, storing them in the database of the existing system with the human factor, and finally making them structured both cost time and are prone to errors. The main motivation of this work is to reduce this effort by proposing an automated system to replace any manual task of structuring. Additionally, this study aims to integrate the candidates into the Kariyer.net system by using the information obtained from these fields of the résumés in different formats [1]. The information extracted from a résumé is highly beneficial in terms of matching the candidate's qualifications with the right job by better analyzing them. We expect to increase the performance of further résumé-related operations, especially the job-candidate matching accuracy.

There is not a consensus about the résumé format and layout, that is to say, each résumé might have a different formatting style. This, of course, makes it difficult to develop an automatic structuring system working efficiently for any résumé format. The first difficulty of the process is that résumés in different file formats such as documents (DOC), portable file format (PDF), or any image format (PNG,JPEG, etc.) should be transferred to the computer environment as a text structure. Secondly, different layouts of the résumé files should be converted to a common format. For instance, because résumés are composed of different structures and the information is in different columns on a page, the extracted texts can mix with each other. In addition, the information in the extracted texts should be separated in a meaningful way. More clearly, the information should be separated and divided into the necessary information groups. In this study, we are specifically interested in this second difficulty. We concentrate on determining the number of columns in the résumés in order to provide a meaningful text extraction in the résumés containing different column numbers. By determining the number of columns in the résumés, it will be ensured that the extracted texts are separated into appropriate sections. Thus, the texts will be prevented from being mixed in the extraction stage.

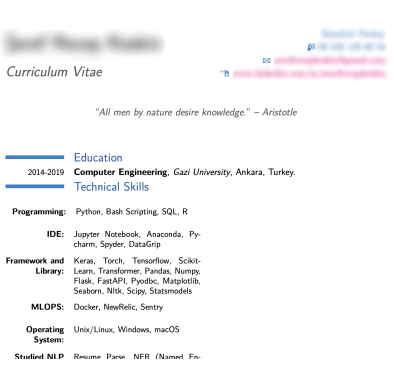
In the literature, the IE operations from the résumé texts are carried out by using regular expressions, natural language processing, machine learning methods, and named entity recognition [4, 6]. These works primarily seek to extract

the semantic meaning of documents or to make use of this type of information. However, there are only a few works that focus on the process of structuring itself. For instance, Tobing et al. examine the résumés in the Indonesian language [12]. In this study, different models of header segmentation were used for separating different segments such as personal information, work experience, etc. In a sense of the dedication of segmentation, this work can be similar to our aim. However, we are explicitly interested in determining the number of columns in this study. To the best of our knowledge, our work has originality due to the specific area of interest.

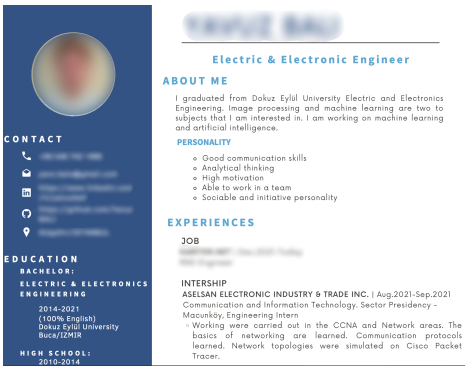
The rest of this paper is organized as follows. In Section 2, we introduce the data set that is used in this study in detail, and in Section 3, we describe the methods used in this study. In Section 4, we give a discussion about different approaches. Finally, in Section 5, we conclude our paper.

2 Data Set

In this work, we use the real résumés that are intended for job applications. Along with the standard templates available on the Internet, the applicants are observed to create their résumés in various forms. Free form résumés result with a set of different font faces, colors and types. This wide variety of résumé forms constitutes a challenge when transforming their unstructured form into a structured one. We handle the résumé data sets in both PDF and any image format, which are converted into free form texts by parsing the documents. Two different formed examples of résumés are given in Fig. 1. In Fig. 1a, a single-column résumé is shown while the sample résumé in Fig. 1b has two columns.



(a) A résumé with a single-column



(b) A résumé with two-columns

Fig. 1: Sample Résumés

This work uses use 1018 résumés of the format PDF. The résumés are extracted from Kariyer.net, which stores both the unstructured résumés uploaded

by the candidates and the résumés constructed by filling in some structured forms. The résumés uploaded directly to the system might have different formats and styles, with one-, two-, or three-columns, different headers, or writing details. We rigorously selected the experiment samples to reflect the true natural variety of the original database; that is, the data set consists of résumés with one- or two-columns. Moreover, the samples are manually labeled with their column sizes. There are almost 685 and 333 samples having one- and two-columns respectively.

Table 1: Descriptions of properties expressing texts

Parameter	**Explanation**
x_0	Left corners x coordinate
y_0	Top corners y coordinate
x_1	Right corners x coordinate
y_1	Bottom corners y coordinate
words	The output of text extraction

We aim to process and digitize the textual documents in various forms so that their outlines can be determined and structured. The unstructured but digitized intermediate text portions are obtained. They allow to catch the words containing the text and image content of the document page. It represents the hierarchical information structure of the document page, consisting of blocks, lines, spaces, and characters, each with its own sub-dictionary. We describe the features and explanations used in this study in Table 1. The geometric information of a text portion can be seen in Fig. 2.

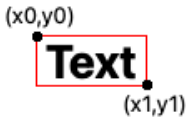


Fig. 2: Representation of coordinate parameters in the sample picture

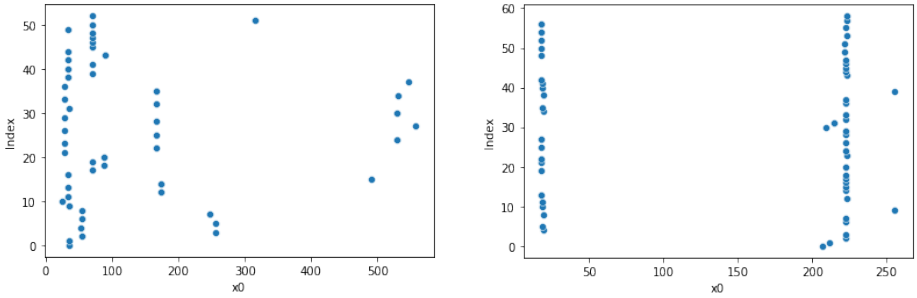
In the obtained dictionary, unwanted data may occur due to the whitespace characters in the résumés, so the data without text is cleared from the dictionary. Additionally, the page that each text belongs to is added. After all cleaning and preprocessing steps, the features of the résumés are represented with a data frame table. A representative sample of such data is shown in Table 2. The coordinate information and text information of the extracted texts are obtained in the table.

In Fig. 3, a scatter plot of the x_0 coordinates of the detected texts are shown in order for two sample résumés with one- and two-columns. We observe that the text are concentrated in one region for the résumés with a single-column (see

Table 2: Parsed Résumé Dataframe

	x_0	y_0	x_1	y_1	**words**	**page_number**
10	24.959	223.092	494.355	240.014	EDUCATION \n	1
11	33.720	240.522	116.180	303.062	University \n(Ba...	1
12	173.779	242.209	518.020	300.038	İzmir University...	1
13	33.720	309.882	108.500	358.381	University \n(Ba...	1
14	173.779	311.653	517.298	355.477	İzmir University...	1

Fig. 3a), and for the samples with two-columns, two distinct regions are easily separable (see Fig. 3b). Similar behavior for x_0 coordinates is observed in many résumés examined.



(a) A résumé with a single-column (b) A résumé with two-columns

Fig. 3: Scatter plot of x_0 coordinates of sample résumés

3 Methodology

To determine the number of columns, we focus on the coordinate information of the texts parsed from the résumés. From this point of view, the problem is handled as an analytical problem, which examines the coordinates of the parsed data. Accordingly, x_0 and y_0 represent the starting points for a text portion, while x_1 and y_1 represent the endpoints. For languages written from left to right, x_0 coordinate information is considered as a feature representing the beginning parts of the writing in the résumés. We assume that for the texts that belong to the same paragraph, only the y_0 information changes whereas the x_0 information remain in a certain tolerance margin. In the case of more than one-column, the x_0 coordinate is considered as a feature that indicates the starting positions of the text. In this case, it makes sense to use x_0 coordinates in languages written from left to right to determine the number of columns.

In this way, the problem of determining the number of columns in the résumés with different forms turns into a clustering problem of x_0 values extracted from the résumés. Closer x_0 values will be grouped to form rows starting in the same column based on clustering. Once the column number determination problem is considered as a clustering problem of x_0 values, we can employ several clustering solutions such as partitioning methods, hierarchical methods, density-based methods, etc. However, many clustering methods cannot determine the number of clusters automatically. The well-known K-means approach needs post-processing methods such as elbow or silhouette while the algorithm of the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [8], which has a proven performance on analyzing geographic data, can determine the number of clusters while clustering the data simultaneously. x_0 values show the coordinates on a text, similar to geographic coordinates. In the following subsections we first explain the details of clustering-related approaches, then we give the details of our problem setting and how we use the clustering techniques for determining the different columns.

3.1 K-means Algorithm

K-means clustering method is a method of partitioning a data group into clusters in the specified number of data sets [9]. It is one of the unsupervised machine learning techniques. Clustering operations aim to maximize the similarities between the data in a cluster and minimize the similarities between the clusters. It is a widespread method in the data mining world. The specified number of clusters is significant for the algorithm. The algorithm divides all the data into the specified number of clusters. Specifying too many or too few clusters can lead to meaningless data partitioning. The elbow or silhouette methods can determine the optimal number of clusters [11].

3.2 Elbow Method

Each number of clusters calculates the sum of the squares of the distances from the center of the cluster to which the data is included [13]. This calculation is also called Within-Cluster-Sum of Squared Errors (WSS). When the graph of the calculated values for each cluster number is drawn, a graph is formed as shown in Fig. 4. In the graph, the elbow point where the difference between the totals starts to decrease is indicated as the most appropriate number of clusters for K-means.

3.3 Silhouette Method

The silhouette method is a method that provides the most appropriate number of clusters and interpretation of consistency between data clusters. The method calculates the silhouette coefficients of each point, which measures how similar a point is to its cluster compared to other clusters. We evaluate the classification

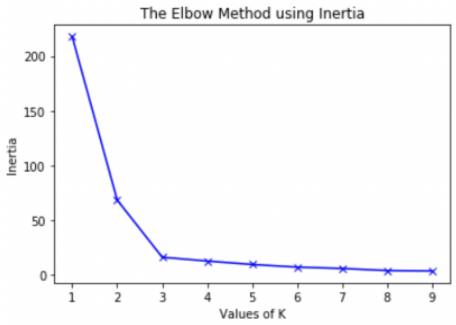


Fig. 4: Example of an elbow graph.

performance of each data point with silhouette coefficients. The formula for calculating the silhouette coefficient is given in Eq. 1. In this equation, $a(i)$ is the average distance function of a point from all other points in the same cluster [10]. $b(i)$ is the average distance function from all points in the other cluster closest to the cluster to which a point belongs. The distance calculation functions $a(i)$ and $b(i)$ can be used as Euclidean distance, Manhattan distance, Etc., any other distance metric.

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \tag{1}$$

The silhouette coefficient ranges from -1 to +1. A high positive value indicates that the data matches well with the cluster to which it belongs, and a low negative value indicates that the data is poorly matched with the cluster to which it belongs. Considering all the data, the fact that most of the silhouette coefficients of the data have a high value indicates that the clustering is appropriate. Most of the silhouette coefficients have low or negative values, indicating that the number of clusters is low or high.

3.4 DBSCAN Algorithm

DBSCAN algorithm is a clustering algorithm that depends on the neighborhood of data points in two or multidimensional space. Since the data is handled from a spatial point of view, it is mostly used in the analysis of spatial data. In the original work of Ester Et al. [5], it is accredited as "A density-based algorithm for discovering clusters in large spatial databases with noise". DBSCAN produces successful results in large-volume databases, even when clusters are separated in arbitrary ways.

Unlike K-means algorithm, it does not require the number of clusters to be specified beforehand. It is also an outlier-resistant algorithm. Given a set of points in space, the algorithm aggregates points that are highly close and marks data points below a certain threshold in low-density regions as outliers. It contains two different parameter inputs distance ε and minPoint. ε specifies how

close the points must be to be considered part of a set. Euclidean distance is commonly used to measure the distance between two points. However, different distance methods can also be used. If the distance between two data points is less than or equal to the ε value, it means that the point is considered a neighbor. minPoint is the minimum number of points to create a dense region. For a region to form a dense region, a data point must contain at least as many points as specified by the number of minPoints within the distance specified in the ε value. The minimum value for minPoint should be 3.

In DBSCAN, for clustering purposes, points are divided into three groups core points, reachable points, and outliers [3]. A data point is a core point if it contains as many data points as minPoints, including itself, within the epsilon distance. Points that are not core points within the area of a core point are called adjacent points and reachable points. Data that do not seed points and fall outside the areas of the seed points are called outliers. The seed points form a cluster together with the reachable points covered by the points. Each cluster contains at least one core point.

The data generated by the parsed résumés constitute an applicable data set for the DBSCAN Algorithm. Font, size, coordinate, etc., properties of the texts in the résumé are extracted in blocks. Coordinate information of text blocks creates point data in the 2D coordinate plane. The number of text blocks clustered on the page can be determined using the DBSCAN Algorithm. The number of clusters detected can give information about the number of columns in the résumé.

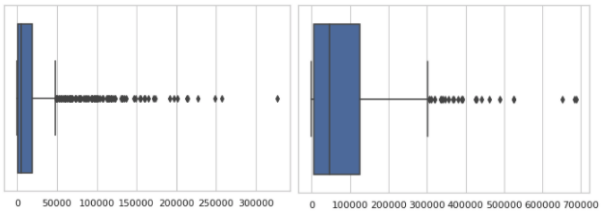
3.5 Column Detection via Clustering

We induced the problem of finding the different columns of a résumé to the problem of optimal clustering of x_0 coordinates of read text in résumés. In a real-world data set, we distinguish that some résumés have well-separated two-columns while some of them have vaguely separated ones. Thus, in our case, it is not certain that any clustering algorithm can easily detect the different segments, i.e. columns. We cannot find the best clustering of x_0 values in polynomial time because clustering is an NP-hard problem. That is why we need to choose one of the clustering approaches that gives the best results of all. Since clustering is an unsupervised problem by nature, and since the document column detection problem has never been studied from this perspective before, we do not know the most suitable algorithm yet. For this reason, we suggest using a supervised approach to find the clustering technique with the best performance for our case. That is why the numbers of the columns in our résumé set are labeled manually.

Among DBSCAN, K-means with elbow and K-means with silhouette, we choose the one that finds the correct column numbers for the labelled set. After preparing the data for the study and performing the calculations with the methods used in the experiment, we used multiple success metrics dedicated to measuring the performance of supervised modelling to monitor the results of this study. These metrics are accuracy, recall, precision, and F1-score. We used a confusion matrix to find these numbers. Here, our main purpose is to build an experimental setup for further similar studies.

4 Experiments and Results

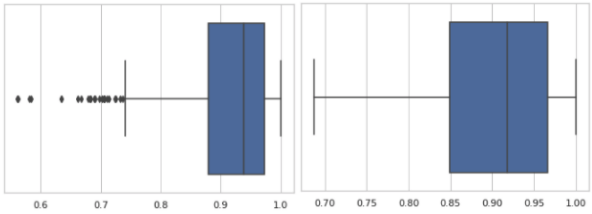
This section describes the parameters of the experiments and reports the results performed with different methodologies. We also elaborate on the evaluation of the results. The results of three different methods, including DBSCAN, elbow and silhouette, were evaluated. It were tested with a total of 1018 résumé file, consisting of 685 single-column and 333 double-column résumés. The positions of x_0 are clustered by the K-means algorithm choosing a certain number of clusters. In order to decide the number of k, the number of clusters, in K-means, we use the elbow and silhouette methods as described in Section 3.



(a) Single-column samples (b) Two-column samples

Fig. 5: Box plot of WSS scores of the elbow method

In Fig. 5a and Fig. 5b, the box-plots of WSS scores obtained by the elbow method for all résumés in the data set are shown for one- and two-column résumés respectively. According to these results, it is observed that there is no threshold value that can clearly distinguish one- and two-column résumés. On the other hand, for the best performance, a single-column résumé estimate can be given for résumés with a WSS below 50000 and a two-column résumé for résumés with a WSS above 50000.



(a) One-column samples (b) Two-column samples

Fig. 6: Box plot of the silhouette scores

We also report the scores obtained by the silhouette method according to the number of columns, as given in Fig. 6a and Fig. 6b. These results can be

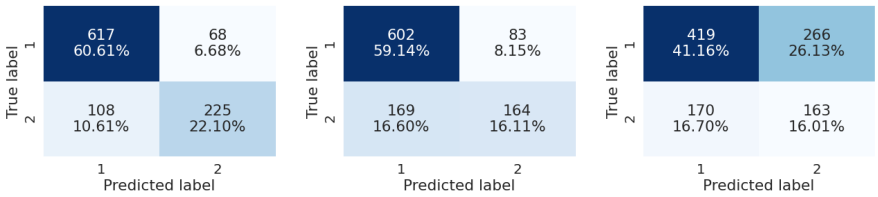
interpreted as a clear range of values cannot be observed for parsing one-column and two-column résumés. As in the elbow method, there is a range of values covering similar silhouette scores for both types of résumés. However, it can be determine that it includes résumés with one-column above a 0.95 silhouette score and six two-columns with a 0.95 silhouette score to ensure the highest accuracy.

With the estimation made by the DBSCAN algorithm on the x_0 coordinate values, we directly determine the column number of a résumé. The results obtained in the experiment performed on the test data are shown in Table 3. A high accuracy value was achieved with an accuracy rate of 83%. However, low accuracy was obtained for the double-column résumés. Accordingly, the F1-score 72% value was obtained.

Table 3: Column number determination performances

Method	**Test Accuracy**	**F1-Score**	**Recall**	**Precision**
DBSCAN	**83%**	**72%**	**68%**	**77%**
Elbow	75%	57%	49%	66%
Silhouette	57%	43%	49%	38%

Table 3 summarizes the column number estimation performances of three clustering strategies. The DBSCAN algorithm clearly outperforms the other K-means based strategies: silhouette and elbow methods. Moreover, Fig. 7 shows the confusion matrix results for each method. The success of the confusion matrix according to each label was examined. It is seen that DBSCAN achieved a success rate of 90.07% in single-column résumés and 67.56% in two-column résumés. It is seen that the Silhouette method reaches a success rate of 87.88% in single-column résumés and 49.24% in two-column résumés. On the other hand, the Elbow method has a success rate of 61.16% in single-column résumés and 48.94% in two-column résumés. Considering those results, all three methods outperform single-column resumes. Especially the DBSCAN method shows considerably high performance for single-column resumes. Nevertheless, all methods' performances seem to be one step behind when finding two-column.



(a) DBSCAN (b) The elbow method (c) The silhouette method

Fig. 7: Confusion matrices with respect to column numbers

The performance of the single-column résumés is usually higher. This study shows that the DBSCAN algorithm, with which the results are compared, performs much better than the elbow and silhouette methods. However, examining the confusion matrix in detail, we notice that the success of discovering two-columns résumés is low.

5 Conclusion

This study focuses on determining the number of columns for transforming the unstructured documents into structured ones. We employ the clustering methods to determine the number of columns. The performances of three different methods, K-means with the silhouette method, K-means with the elbow method and DBSCAN algorithm, on the data set are compared. When the discrete data of the elbow and silhouette methods on the data set are examined, a parsing threshold value could not be determined for the résumés with one- and two-columns. In this case, a threshold value that gives the best performance is determined empirically.

Although DBSCAN performance is acceptable, it is not sufficient to determine the number of columns in documents. When the résumés are examined, we observe that the information is transferred under the relevant headings. Accordingly, the headings contain information about the column number of a résumé. As an extension to this work, the heading information (semantic and positional information) can be employed to determine the headers and the number of columns at the same time. In this way, we presume that the success rates can be increased to reasonable rates for the résumé parse task. Although the study was carried out on résumés, the proposed methods are independent of résumés and can be used on different textual documents. In addition, since the logic on which the study is based is on the clustering of the coordinates where the texts are located, it is independent of the language. It can be used for any language. Also, the right-to-left or left-to-right spelling of the text does not affect the method.

There are several different clustering approaches (model-based, spectral, hierarchical, etc.) and different metrics for finding optimal clustering numbers (gap statistics, modularity, etc.) besides the ones which are used in this work. They can also be added to evaluate the performance of these approaches on this specific problem in further studies.

References

1. Çelik, D., Elçi, A.: An ontology-based information extraction approach for résumés. In: Proceedings of the 2012 International Conference on Pervasive Computing and the Networked World. p. 165–179. ICPCA/SWS'12, Springer-Verlag, Berlin, Heidelberg (2012)
2. Cowie, J., Wilks, Y.: Information extraction. In: Dale, R., Moisl, H., Somers, H. (eds.) Handbook of Natural Language Processing, pp. 241–260. Marcel Dekker, Inc., USA (2000)

3. Daranda, A., Dzemyda, G.: Novel machine learning approach for self-aware prediction based on the contextual reasoning. International journal of computers, communications and control **16**(4), 1–15 (2021)
4. Das, P., Pandey, M., Rautaray, S.S.: A cv parser model using entity extraction process and big data tools. International Journal of Information Technology and Computer Science (2018)
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. p. 226–231. KDD'96, AAAI Press (1996)
6. Gaur, B., Saluja, G.S., Sivakumar, H.B., Singh, S.: Semi-supervised deep learning based named entity recognition model to parse education section of resumes. Neural Comput. Appl. **33**, 5705–5718 (2021)
7. Grishman, R.: Information extraction. IEEE Intelligent Systems **30**(5), 8–15 (2015)
8. Li, J., Han, X., Jiang, J., Hu, Y., Liu, L.: An efficient clustering method for dbscan geographic spatio-temporal large data with improved parameter optimization. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences **42**, 581–584 (2020)
9. Oyelade, O.J., Oladipupo, O.O., Obagbuwa, I.C.: Application of k means clustering algorithm for prediction of students academic performance (2010)
10. Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics **20**, 53–65 (1987)
11. Shi, C., Wei, B., Wei, S., Wang, W., Liu, H., Liu, J.: A quantitative discriminant method of elbow point for the optimal number of clusters in clustering algorithm. EURASIP Journal on Wireless Communications and Networking **2021**(1), 1–16 (2021)
12. Tobing, B.C.L., Suhendra, I.R., Halim, C.: Catapa resume parser: End to end indonesian resume extraction. In: Proceedings of the 2019 3rd International Conference on Natural Language Processing and Information Retrieval. p. 68–74. NLPIR 2019, Association for Computing Machinery, New York, NY, USA (2019)
13. Yuan, C., Yang, H.: Research on k-value selection method of k-means clustering algorithm. J **2**(2), 226–235 (2019)