Research Article GU J Sci, Part A, 12(1): 127-153 (2025) 10.54287/gujsa.1636051

Gazi University
Journal of Science
PART A: ENGINEERING AND INNOVATION
http://dergipark.org.tr/gujsa

Novel Approach for Detecting the Number of Columns of a Résumé

Yavuz BALİ[1] Günce Keziban ORMAN[1] Sultan Nezihe TURHAN[2*]

[1] *Galatasaray University, İstanbul, Türkiye*
[2] *Kariyer.net, Istanbul, Turkey, Ankara, Türkiye*

Keywords

Information Extraction
Résumé Parsing
Clustering
Text Processing
Text Coordinates

Abstract

In recruitment processes, manually reviewing résumés is a highly time-consuming job. In order to reduce the cost of these reviews, Information Extraction tasks have been introduced to extract the structure of the document and the personal information contained within. However, because there is no consensus on a standard structure of résumés, i.e., each résumé has its own distinctive layout, column numbers, or text properties, an accurate extraction process becomes highly challenging. This study addresses a part of this problem. We focus on the problem of estimating the number of columns in résumés, as we experience in the further processes that knowing the number of columns facilitates the separation of the main sections of the résumés, hence the analysis of the finer subsections. We employ the coordinates of the text blocks that build up a résumé. We hypothesize that the coordinates of the text blocks carry information on the number of columns. We define the problem in a clustering context. We proposed a novel clustering approaches dedicated to finding the number of columns in a résumé by the separation of the text block coordinates. The experiments are conducted on a dataset of the résumés of real applicants in two languages: Turkish and English. The results reveal that hybrid approaches that use the intermediate methods perform better than the individual methods. Furthermore, these findings could be extended to any unstructured textual data in any language and document format

Cite

Bali, Y., Orman, G. K., & Turhan, S. N. (2025). Novel Approach for Detecting the Number of Columns of a Résumé. *GU J Sci, Part A, 12*(1), 127-153. doi:10.54287/gujsa.1636051

Author ID (ORCID Number)

0000-0003-0621-3069 Yavuz BALİ
0000-0003-0402-8417 Günce Keziban ORMAN
0000-0001-9763-0882 Sultan Nezihe TURHAN

Article Process

Submission Date 10.02.2025
Revision Date 27.02.2025
Accepted Date 12.03.2025
Published Date 26.03.2025

1. INTRODUCTION

Recruitment processes in the early 2000s were based mainly on personal relations between employers, job seekers and intermediaries (Yakubovich & Lup, 2006; Holm, 2012). These personal relations are being replaced by the emerging technologies and techniques with the impact of the digital transformation in the business world, which lead a drastic reformation in the recruitment processes. The fundamental sub-processes of this transformation are transferring the documents into digital mediums, extracting textual data, and finally filtering valuable structured information automatically. This automated process aims to yield entities or relationships between entities and attributes as products (Sarawagi, 2008; Joan & Valli, 2019). All of these steps are usually performed manually, however, it is both time-consuming because of its repetitive document review and analysis, and costly because of the unstructured nature of the documents (Mao et al., 2003; Liu et

*Corresponding Author, e-mail: sturhan@gsu.edu.tr

al., 2019). These operational hurdles make the textual information extraction process highly important in order to have more automatized recruitment processes.

The evaluation of résumés is an important step in the recruitment processes, especially in the hiring stage (Çelik & Elçi 2012). Automating the information extraction from résumés, which means extracting the content properly, significantly facilitates the recruitment tasks (Zaroor, 2017), but this is a challenging duty, mainly due to the fact that the résumé files document structures are hidden behind their file format. Because one does not have document object structure, it is needed to parse the résumés for their content. But this is a challenging task too because there is no standard for preparing a résumé. Candidates produce unique résumés not only to present their professional background but also to leave a good impression on the employer through the résumé structure and content, assuming it as a way of expressing themselves. Thus, it is a natural fact that almost each résumé has its own unique structure. Due to their rich diversity, variety of structures, and unique forms, the analysis of résumés constitutes a valuable topic in the information extraction (IE) domain.

Previously, different methods such as regular expressions, NLP, machine learning, and named entity recognition have been applied to IE from résumés (Das et al., 2018; Gaur et al., 2021). Most of them aim to extract semantic content from résumés and to benefit from this information. These works have a motivation to extract each segment that includes personal information like skills, education, job experience, etc. They use supervised learning techniques and hence needs the labels of each segment of a résumé. Although these approaches propose sophisticated methodologies to extract each single part of a résumé, they suffer for the following reasons: First, they all need labeling for each section and subsection of résumés, which is a time- and human-resource-consuming task. Second, the architecture of the recent NLP and deep learning-based learning methods is hidden, which means the built model is not comprehensible easily. This makes the use of those models open to criticism. Moreover, the success of the proposed solutions is not high either.

A similar work overcomes the aforementioned challenges without using any semantic information (Tobing et al., 2019). Tobing et al. (2019), work on the Indonesian résumés to extract information such as personal information, work experience, etc. Rather than employing an NLP-based approach, they use the coordinates of the extracted texts. In their system, when a document is analyzed by an automated extraction tool, the content is restructured as a set of text blocks. These text blocks are defined by their coordinates and style-related attributes. Tobing et al., (2019) use also a dictionary of the possible headers to discover the headers as the essential feature in each segment of the résumé. They find different segments in a résumé according to the closeness of the text blocks to the headers. However, the non-standard shapes of the résumés still raise difficulties. Thus, their methodology suffers from not being able to understand non-standard résumé formats, which we believe constitute the majority of the real-world samples. The idea of using text block coordinates have been employed in (Keskin et al.2022) as well. Differently from the work of (Tobing et al., 2019), Keskin et al. (2022) emphasize that real-world résumés have various formats and the extraction of each segment is a secondary problem. Instead of finding each single segment, they focus on separating the résumé columns since

each single document column already includes more related sections and text. Thus, finding the columns of a résumé might result in more accurate segment separation.

In this work, we focus on detecting the columns of a résumé in order to extract the content texts correctly. The expertise and rich content of the Kariyer.net dataset (Kariyer.net, 2025) , which is a Turkish job recruitment system, benefit us. Kariyer.net stores the information provided by job seekers during their registration on the platform. This information includes the candidates' personal information, like educational or professional background, hobbies, skills, etc., stored in the form of structured data in databases. In addition, Kariyer.net allows candidates to upload their résumés to the system. These résumés are stored in databases in their original formats, mostly Portable Document Format (PDF), without being processed beforehand. It is crucial to extract the information from such documents and store it in a structured database for the well-being of the recruitment process. Currently, there are 700,000 freestyle résumés in the document storage of Kariyer.net, and this amount increases every day. Since the dataset is large and the samples are quite different from each other, supervised learning is difficult for the reasons explained above. But, separating each section into segments correctly is also crucial for further job recruitment assistance. That is why; we focus on the problem of column detection, which is a crucial step that should be handled before detecting each segment correctly.

Our two main contributions are: first, defining the problem of column detection in a given résumé as the problem of clustering the document text blocks; and second, proposing a new clustering algorithm for the explained problem. Moreover, a hybrid approach combining several different clustering methods is also proposed for the mentioned problem. The proposed approaches and two baselines, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Gaussian Mixture Model (GMM)-based clustering, are experimented on a novel data set provided by Kariyer.net. This dataset includes not only unprocessed résumés but also their number of columns are labelled by the domain experts. There are highly challenging cases whose number of columns cannot be decided easily even by the experts.

In the rest of the article, we first detail the related works in Section 2. Secondly, we explain the preliminaries and definitions in Section 3 and proposed methodologies and baselines in Section 4. Third, we explain in detail the experiments and their results in Section 5, where we also provide a discussion section that evaluates the performance of the approaches by our own interpretation. Finally, in Section 6, we conclude with the gained insights and the future perspectives on the work.

2. RELATED WORK

All organizations, from traditional companies focused on working with and optimizing conventional recruitment methods to employment-focused web portals based on intelligent recommendation systems, rely on ``résumés'' as the cornerstone of their business processes. Assessing résumés is, therefore, a critical and challenging step in the recruitment process, as it is not only essential but also difficult and costly, given the reliance of all organizations on résumés as the cornerstone of their business processes (Tejaswini et al., 2023).

The absence of standards in the résumés created by candidates is the key factor contributing to the difficult nature of résumé evaluation (Sinha et al., 2021). Depending on the industry it targets, each résumé is created in a particular language and structure with its own specific format. Furthermore, it is impossible to apply a consistent phrase for résumé evaluation due to the ongoing expansion of technology, work habits and business understanding. The complexities of the modern corporate environment make it impossible to even consider manually reviewing each résumé. As a result, every firm has created a variety of technologically advanced solutions to guarantee that résumés are swiftly and precisely assessed.

In order to get better results in "résumé assessing", all relevant candidate's information like skills, work experience, years of education, certifications, etc. should be extracted efficiently and accurately from the résumés (Roy et al., 2020). It is obvious that implementing an automated information extraction process from résumés significantly impacts recruitment tasks (Luo et al., 2018). There are a variety of methods used to extract information from résumés, according to the research in the literature. Information extraction is the process of mechanically obtaining structured data from structured, semi-structured or unstructured sources, such as entities, connections between entities, and characteristics (Adnan & Akbar, 2019). The process of extracting information from résumés varies depending on the structure of the résumé. In today's recruitment processes, the process of extracting information from relatively structured or semi- structured résumés prepared through e-recruitment portals, is done according to the DOM tree structure. For instance, Ji et al., (2010), created a tag tree technique in which they found and eliminated the common content between web sites using the same design while keeping the core text. Farkas et al. (2014), worked on a method of extracting information for career portal where the information of applicants is stored in a uniform data structure named HR-XML format. They used a résumé parser to automatically extract data from the résumé. These solutions, which rely on structured template files and/or DOM trees, are constrained by human labor. These techniques are challenging to scale out in large data due to the fact that it is impossible to determine how many groups of résumés have the same template.

Moreover, considering real-life applications, almost all of the résumés prepared by candidates are in an independent format. Undoubtedly, it is much harder to extract information from résumés in an unstructured format. On one of the pioneering studies on extracting information from unstructured résumés, Yu et al. (2020), describes a method for extracting structured information from unstructured résumés using a combination of rule-based and statistical approache. The first stage of the model uses regular expressions to identify and extract candidate information such as name, address, and phone number. The second stage uses a statistical model to classify each line of text in the résumé as belonging to a specific section (e.g., education, work experience). Finally, the third stage uses a combination of rule-based and statistical methods to extract specific information from each section. Therefore, they called their technique as a cascaded hybrid model, which combines rule-based and statistical approaches. However, the study is limited as the information extracted is only certain areas. Sonar & Bankar (2012) apply “chunking” to divide the résumé into different sectors based on information. Authors have divided the résumés into chunks and then parsed the information of candidates

using named entity recognition. This information was used potentially to implement the auto-fill feature of a form on a website, yet having potential to use better information extraction methods.

In the study realized by Zu et al. (2019) the text block segmentation is combined with the identification of résumé facts, and named entity recognition is carried out inside labelled text blocks using a variety of sequence labelling classifiers. Chen et al. (2018), have developed a two-step résumé information extraction approach. First, with the help of an open source tool, they detect and extracts text from résumés with different file types and they identify the raw text as different résumé blocks and then apply their proper grammatical rules to segment different blocks. Tobing et al. (2019), use also a two-stages pipeline for information extraction from the résumés. They first segment the résumés with indicative headers and they use heuristic rules to extract the necessary information from these segments. The implementation of these automated methods in real-world contexts is constrained by the fact that the aforementioned segmentation algorithms are ostensibly not resistant to various résumé layouts and styles. Qin et al. (2020), proposes a novel neural network-based approach to improve the accuracy and efficiency of person-job fit assessment in talent recruitment. They do not explicitly mention which technique they use to extract information from unstructured résumés. However, they describe their feature extraction method, which involves representing the text of the résumé as a bag-of-words and applying various transformations to generate dense feature vectors. Xu et al. (2020), use a technique called Recurrent Neural Networks (RNNs) to extract information from unstructured résumé documents. RNNs are a type of neural network that can model sequential data, making them well-suited for tasks such as natural language processing, speech recognition, and time series prediction. In the paper, the RNN model is used to learn the contextual information of different blocks in a résumé, such as the education section, work experience section, and skill section, to improve the accuracy of résumé parsing.

All these above-mentioned studies use segmentation and/or block identification techniques to extract information from unstructured résumés. Nevertheless, because the formats and designs used to create the résumés by the applicants vary greatly, they are only able to extract information from certain portions of the résumés. Their performance in real-world applications is drastically diminished as a result. In their paper, Mittal et al. (2020) use a combination of techniques to extract information from unstructured résumés. Specifically, they use the Term Frequency-Inverse Document Frequency (TF-IDF) technique to represent each résumé as a numerical vector, and they also employ Latent Dirichlet Allocation (LDA) to extract meaningful topics from the résumés. The TF-IDF technique is a well-known method in natural language processing (NLP) for measuring the relevance of a word to a document in a collection of documents. It is based on the idea that words that are rare in a collection of documents but common in a particular document are likely to be more important for understanding the content of that document

In their article, Yasmin et al. (2020), developed a framework of potential candidate selecting system by choosing a domain of document information extraction i.e. the CV/résumé documents. This development task involves the information extraction based on natural language processing i.e. tokenization, named entity recognizer (NER) and utilizes skyline query processing which works well in filtering the non-dominating objects from database and also makes a new addition to this domain. The approach described in the work of

Alamelu et al. (2021), is to use machine learning to train the dataset for the different jobs. In this instance, data extraction using NLP is done utilizing section-based segmentation. This is done to spare any company's recruiters from having to spend a lot of time and effort reading through and studying the résumés of numerous candidates. They will get the chance to view the résumés of the applicants as well as get the names of the best fit candidates for the required position, and they can then grade the résumés based on the results of this intelligent process.

As a result of recent developments in the field of NLP, several studies have been published recently that offer end-to-end frameworks to address the person-job match conundrum. The goal is to do semantic matching using sequential neural networks or modern transformer architectures to create thorough and efficient representations of the applicants and job advertisements in order to discover the best candidate for a position. Despite the systems' promising performance, the "information extraction" approach is still the most useful since deep learning-based systems have a substantial amount of opacity. An end-to-end system would raise concerns about the patterns it has learnt and the fairness of its judgements because current neural networks do not show their decision-making processes. Firstly, it should be emphasized that each résumé is tailored to the candidate's individual initiative, which results in distinct text placements when it is analyzed. Nonetheless, the data is organized into several columns on each page of the résumé. It is difficult to show the information in a relevant and sequential manner in the raw data retrieved from résumés with two or more columns since it has been noted that the text information collected at the conclusion of such a procedure is combined with other information. To divide the text elements in various places from one another, the CV's number of columns must be decided.

By concentrating on figuring out how many columns each résumé has in order to parse the raw data included in the résumés, this study aims to help with the extraction of information from unstructured résumés. Also, while considering the information extraction issue for résumés, we first consider Turkish résumés.

3. PRELIMINARIES AND DEFINITIONS

We aim to detect the number of columns of a résumé through the extracted positional information of the text. In the following section, we explain the preliminaries and the definitions that we use in our methodology.

3.1. Résumés and Their Document Characteristics

A résumé is a digital file including different pieces of text that characterize the personal and professional information. The résumés used in this work were obtained from Kariyer.net. Each résumé has different header styles, font sizes, and font faces that is characterized according to the preferences of the holder. The dataset consists of the résumés that have either one or two columns, since such résumés are the most frequently occurring types. Figure 1 shows two illustrations of various column types. Figure 1a represents a single-column sample, while Figure 1b represents a double-column one.

In a résumé, the text pieces of the same section, e.g., the information written under the *Experiences* section, are more related than the text pieces written under different sections. In fact, the relationships between different text pieces are all defined under a preformed document structure, but this structure is hidden in the PDF or Word-formatted résumés. As a result, one cannot solve the relationship between text pieces without having the source code of the document. That is why; we need to parse résumé documents without using their preformed document structure to find the features that explain the types, fonts, or places of the text pieces. But following a single parsing procedure is also impossible because there is no standard format for the résumés, i.e., each résumé is prepared with different templates or structures. Hence, we need to discover the relations between different text pieces for each single document. Parsing a résumé starts with reading its content. A document reader interprets a résumé as a two-dimensional Cartesian coordinate frame *C*. At each read process, it scans and interprets a piece of text which we call a *text block*. $(x_1 - x_0)$ is the width of *t*, $(y_1 - y_0)$ is its height. Coordinate information for a text block is exemplified in Figure 2.

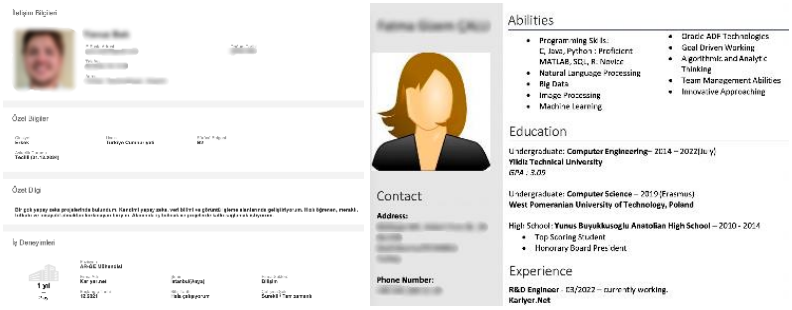


Figure 1. *Sample résumé* ***a)*** *A single-column résumé* ***b)*** *A double-column résumé*

Definition 1. A *text block*, t, is an object that represents a document area framed in a rectangular shape in *C*, whose top-left and bottom-right coordinate points will be represented by two coordinate pairs (x_0, y_0) and (x_1, y_1) respectively.

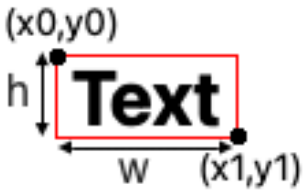


Figure 2. *Representation of coordinate parameters in the sample picture*

A document reader scans the content of the documents as several different text blocks, t_i. Because it is not aware of preformed document structure, at each scan, it reads text blocks at different widths and heights. Each obtained text block, t_i, has the properties which are listed in Table 1.

Once the entire document is read as a set of text blocks via the document reader, a data cleaning process is performed to remove the blocks whose contents are only spaces or tabs. The text's page information is also added. Table 2 shows a representation of a small sample after all these preprocessing steps. Each row in the table contains a text block read by a document reader at a time. Each column represents a feature of the relevant text. According to our empirical observations made in the résumés from the Kariyer.net set, we have discovered that the relation between text blocks in the same column is higher.

Table 1. *Descriptions of properties expressing texts.*

Parameter	Explanation
x_0	x coordinate of the top-left corner
y_0	y coordinate of the top-left corner
x_1	x coordinate of the bottom-right corner
y_1	y coordinate of the bottom-right corner
words	The content
font	The font types of the words
size	The font size of the words

More clearly, in a résumé, we come across either a single-column structure where each résumé section is sequential and the texts under each section are related, or a double-column structure, which is a split of the page. In a double-column structure, each column part is like a single-column structure. Thus, one needs to first find out if the studied résumé is single- or double-column before revealing the relation between different text blocks for determining the true content of a résumé. In the rest of this section, we concentrate on the problem of finding the column number of a résumé.

Table 2. *Parsed Résumé Text Blocks*

text block ID	x_0	y_0	x_1	y_1	words	page	font	size
10	2.49	22.3	49.43	24	EDUCAT...	1	Arial	16
11	3.37	24.05	11.61	30.3	Universi...	1	Arial	12
12	17.37	24.22	51.8	30	İzmir Üniver…	1	Arial	12
13	3.37	30.98	10.85	35.83	Universit...	1	Arial	12
14	17.37	31.16	51.72	35.54	İzmir Üniver…	1	Arial	12

3.2. Finding Column Number Problem

A document reader reads a résumé document file having an unknown predefined document structure as a set of text block objects, T, where each $t_i \in T$ has different properties. For the languages written left-to-right, x_{i0} and y_{i0} can be considered as the starting point of x and y axis of t_i respectively.

Definition 2. For the given T, the problem of finding its column numbers, c, is the problem of finding the size of the column set, C, of T.

Definition 3. The column set, C, is a partition of T in which $C_i \cap C_j = \emptyset$ for all $(C_i,C_j) \in C$ and $\cup C_i = T$.

Property 1. The column number of the T, which is noted as $c = |C|$, is limited to either 1 or 2 because of the nature of the résumé document files of experimental set.

If the column partition, C, of T was known, then finding c could be an easy problem of counting the elements of C. But C is unknown. Thus, we are interested in finding the column partition for the rest. In case that the

value of *c* is 1, then all tis are collected in a single column. If the value of *c* is 2, there can be as many different subsets *C* of *T* as the two-element subset. However, determining whether a résumé has 1 or 2 columns involves examining all two element subsets of *T* and a quality function which is used in decision making of the best column structure. This is a well-known NP-complete problem. In our work, we can reduce this problem thanks to document readers do never read the text blocks from different columns together. More clearly, we reduce this problem into an optimization problem that is trying to find the best column structure whose text blocks obey some criteria. We will first explain the finding columns problem definition, then reveal the criteria of the text blocks that can be found in the same column, and third, propose alternative solutions in the rest of the document.

Definition 4. The problem of finding the best column structure, Γ, of *T* is finding the most proper columns of *T* such that the t_i*s* in the same column have a set of positional relations, *P*.

In this work, we are interested in the positional relations of text blocks. Since the *word*, *page*, *font* and *size* of a *ti* is not related to its position but related to its content, we do not define any relation or criteria using these features. The y_{i0} and y_{i1} of a t_i represent its vertical axis properties. But column separation is related to the horizontal axis. Even if t_i and t_j belong to the same column, their y coordinates can be too different, or vice versa. A possible relation of y coordinates can be used to discriminate the different lines belonging to the same t_i however, it is out of scope of this work. That is why we are not interested in y_0 or y_1 for the rest of this article.

Among horizontal axis coordinates, x_1 describes where the read text block ends on the *x* axis. Different text blocks in different columns are not related to each other so that they can take too different x_1 values. Furthermore, even if different text blocks are in the same column, they can still have different x_1 values because they can be of different widths. Thus, the presence or absence of a relationship between x_1 values of text blocks is not significant to determine if they belong to the same column.

x_0 values of different text blocks in the same column, are expected to be related to each other. Although this relationship may vary according to the format of the résumé, it should more or less correspond to the starting point of a column on the horizontal axis. More specifically, the x_0 values in all text blocks from the same column are expected to have close values, that is, they should constitute a statistical mode of a distribution. An example of x_0 distribution of a single-columned résumé is represented in Figure 3a.

Definition 5. The problem of finding Γ is the problem of clustering the x_0 values of scanned text blocks.

Since the finding column number problem can be solved by clustering the x_0 values of text blocks, we propose different clustering techniques for this issue in the next part of this article.

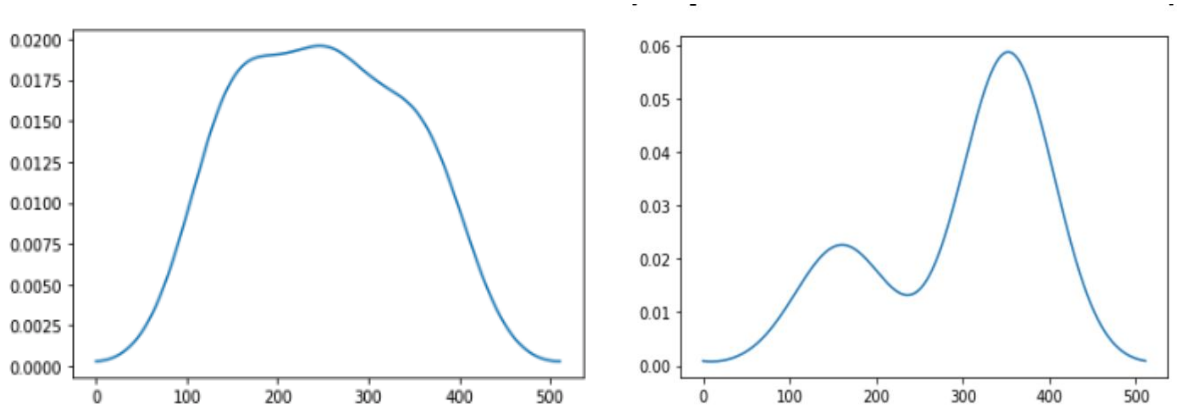


Figure 3. *Estimated densities of sample résumé* ***a)*** *A single – column résumé* ***b)*** *A double – column résumé*

4. CLUSTERING X_0 OF TEXT BLOCKS

Since all x_0s from the same column should have more or less close values, we assume that they should be part of a statistical distribution. As a baseline, we estimate the possible distributions of x_0. In the next part, we first explain this approach as our baseline, then explain Gaussian Mixture Model and DBSCAN, finally we present our proposed clustering algorithm for the mentioned problem.

4.1. Baseline Clustering

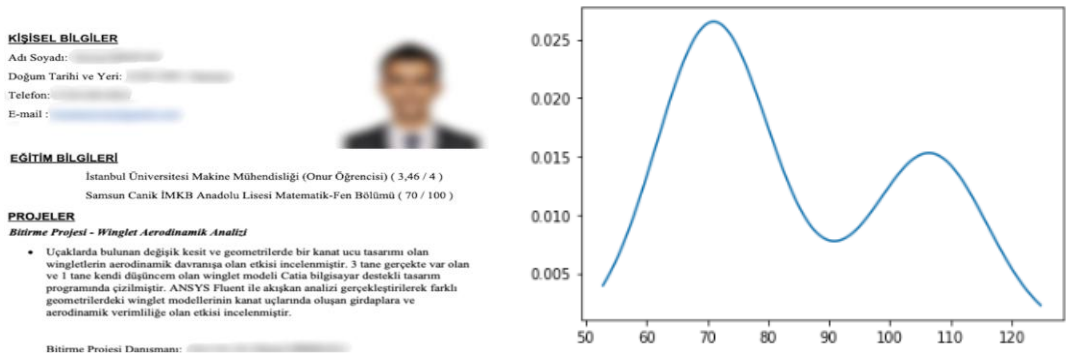
We make a Gaussian distribution assumption and estimation of the x_0 values of the text blocks of a résumé (Kokoska & Zwillinger, 2000). We expect to relate the number of columns to the number of peaks in the estimated distribution. If we obtain a distribution with a single mode, there is definitely a single column in the document. But, the main flaw of this method is that the distribution can have obvious two peaks when there happens to be a large space between the x_0 values of the same column due to the chosen layout of the résumé. Thus, two peaks do not imply two columns. We will call these types of résumés as irregular. An example to these irregular résumés is given in Figure 4. Due to the high indents of the original document, two-peak distributions are produced by x_0 values. An example to a genuinely double-column résumé is also represented in Figure 3b.

If a résumé is single-column but the estimated distribution has two peaks, the estimated x_0 values are far apart from the real x_0 values. To handle correctly the aforementioned irregularity of the résumés, we observe the difference between the estimated x_0 distribution and the real x_0s. However, in some extreme cases, the résumés are not directly classified as having one or two columns. We interpret these cases as *uncertain*, where the true number of columns can be either one or two.

4.2. Gaussian Mixture Model Clustering

As another statistical approach, we model the problem of clustering x_0 values as a Gaussian Mixture Model (GMM), which is a category of probabilistic models stating that all generated data points are said to be derived from a combination of finite Gaussian distributions with no prior parameters (Xuan et al., 2001). We are to estimate the parameters of the individual Gaussian distributions from a posterior distribution. The parameter

estimation is performed using the iterative expectation-maximization (EM) algorithm. We follow an empirical procedure to determine the number of clusters of the x_0 values. Since the clustering incorporates both information about in-cluster cohesion and out-of-cluster separation, we choose the cluster number that maximizes average silhouette width Rousseeuw (1987) which is a well-known quality metric for finding the best cluster separation of the sets. If a résumé is truly double-column, the resulting cluster structure is expected to have a high score. We set an empirical silhouette zone in order to categorize the results of the clustering performance as *single-column*, *double-column* and *uncertain*.



A single-column résumé Corresponding estimated density

Figure 4. *An example of an irregular résumé. Even though the résumé has only one column, the distribution appears to be bimodal*

4.3. DBSCAN Algorithm

We use the DBSCAN algorithm to simultaneously cluster the x_0 and determine the optimal number of clusters (Ester et al., 1996). The algorithm, unlike many other clustering methods, does not need a preset the number of clusters, which suits to our needs in detecting the number of columns in résumés. 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 requires two parameter inputs: ϵ, the minimum distance between data points in the same cluster, and *minPts*, the minimum number of data points required to form a cluster. ϵ specifies how close the points must be to be considered part of a set. The Euclidean distance is commonly used to measure the distance between two points. If the distance between two data points is less than or equal to the ϵ value, it means that the point is considered as a neighbor. *minPts* is the minimum number of points to create a dense region. A data point must contain at least as many points as specified by the number of *minPts* within the distance specified by the ϵ value for a region to form a dense region. The minimum value for *minPts* should be 3.

4.4. Overlapping Headers

We propose the Overlapping Headers method, which is an algorithm based on the overlapping property of the text blocks on the horizontal axis. In their work, Chen et al. (2013) employed the connected component analysis of the gaps in order to perform page segmentation. The authors propose a hybrid page segmentation procedure with hierarchical methods for removing the appropriate space rectangles and keeping the remaining ones. The following describes a similar strategy that we apply in the study.

4.4.1. Column Detection with Overlapping Text Blocks

The representation of different text blocks of a double-column résumé on the Cartesian coordinates is shown in Figure 5. The rectangular forms of the text blocks in the same column overlap at certain intervals on the horizontal axis, whereas they do not overlap and have large gaps if they belong to different columns.

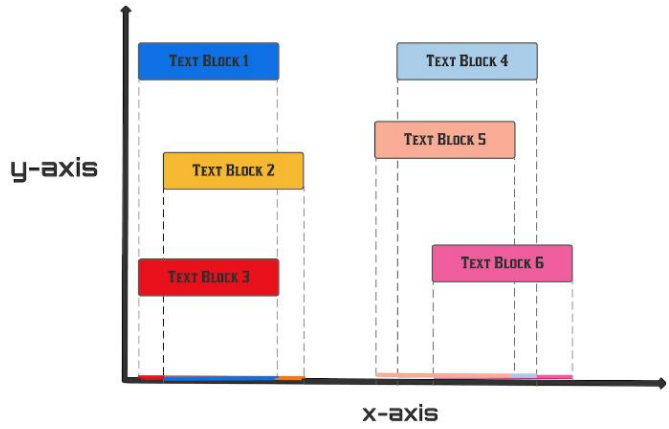


Figure 5. *Overlapping text blocks*

Definition 6. Let t_i and t_j be two text blocks that each of them is defined by two points $\{(x_0, y_0), (x_1, y_1)\}$ in the Cartesian space. t_i and t_j overlap if their binary overlap interval (BOI) value is zero, $BOI\ (t_i, t_j) = 0$, as given in (1). They are disjoint if $BOI\ (t_i, t_j) = 1$.

$$BOI\ (ti, tj)\ = \begin{cases} 0, & max(x_{i0}, x_{j0})\ -\ min(x_{i1}, x_{j1})\ \leq\ 0 \\ 1, & max(x_{i0}, x_{j0})\ -\ min(x_{i1}, x_{j1})\ >\ 0 \end{cases} \quad (1)$$

Definition 7. A column, C, is a set of text blocks in which $\forall\ (t_i, t_j) \in C^2$, $BOI\ (t_i, t_j) = 0$. In other words, every text block rectangle pair in a column *overlaps* with each other.

Property 2. $\exists t_i \in$ column Ck and $\exists t_j \in$ column C_l, if t_i and t_j *overlaps* then $C_k = C_l$.

Definition 8. A document is a set of n text columns. If $n > 1$, according to Property 2, $BOI\ (t_i, t_j) = 1$, $\forall t_i$ and t_j being member of different columns and vice versa.

A document is constituted by its columns. If the number of its columns is n, being larger than one, then it has n disjoint text block rectangular set, i.e. column. In each set, every text block pair overlap. Based on these definitions and their properties, we propose an iterative algorithm to find the number of columns (see Algorithm 1). The flow schema of the algorithm is presented in Figure 6.

The algorithm first retrieves data from the document in the form of text blocks (line #1). Then it builds the first column as a set of text blocks with the first processed text block (lines # 3-6). If there are new text blocks that overlap this already processed one, it adds them to the existing set (lines #7-14). However, if it has acquired new completely disjoint text blocks, it builds a new column for it (lines #7-12,15-18). For each new non-processed text block, the overlapping check continues until all text blocks are scanned. Finally, the overall columns are checked for disjunction (lines #19-23) to ensure Property 2 is correct.

Algorithm 1 Overlapping Headers Algorithm

```
Data ← extract_textblocks(PDF Sheet)
columns = ∅
for textblock ∈ Data do
    if columns == ∅ then
        C = createNewColumn(textblock)
        columns = insertColumn(C, columns)
    else
        for C ∈ columns do
            CheckOverlapInterval == 0
            for textblockofC ∈ C do
                CheckParameter = checkBOI(
                                 textblock, textblockofC)
            end for
            if CheckOverlapInterval == 1 then
                C = insertTextBlock(C, textblock)
            else
                C_new = createNewColumn(
                                 textblock)
                columns = insertColumn(
                                 C_new, columns)
            end if
        end for
    end if
end for
if isDisjoint(columns) then
    return size(columns)
else
    columns = mergeOverlappedColumns(columns)
    return size(columns)
end if
```

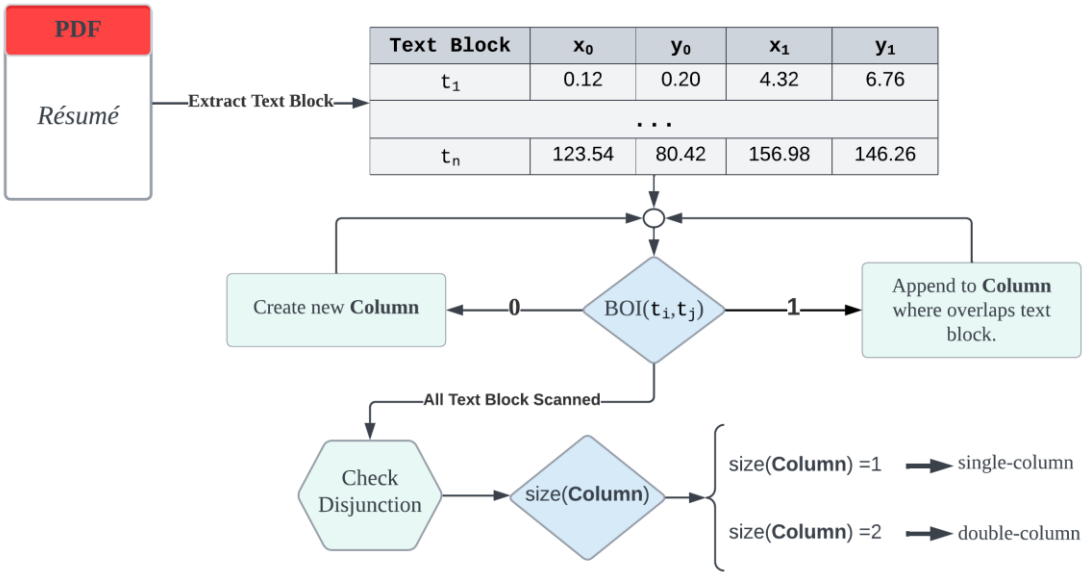
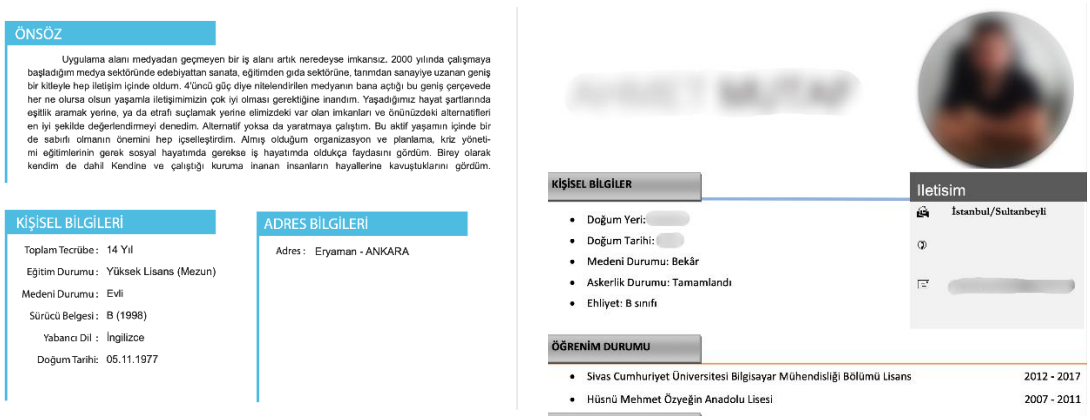


Figure 6. *Overlapping text blocks flowchart*

According to the properties and the iterative algorithm proposed, some of the résumé examples having two irregular column structures are also marked as single-column as shown in Figure 7.



(a) **(b)**

Figure 7. *Two résumé examples with irregular structures* ***a)*** *First single, then double-column résumé example* ***b)*** *First double, then single-column résumé example*

Algorithm 1 labels the résumé in Figure 7a as single-column from the beginning. Even if there are separate columns in the following, the rest is labelled as single-column as well, since the initial text blocks and the following blocks overlap with all of the unprocessed text blocks. For Figure 7b, the merge operation in the last part of the procedure follows Property 2, whereas there exist disjoint columns at the beginning. As a result, this résumé is also categorized as single-column. However, in these examples, the single-column sections are only short introductory paragraphs or short articles describing the career goal. The sections containing the main information of the résumé are distributed onto two columns in both examples. This is typical of real-world résumé sample. Considering all extracted text blocks may cause us to ignore such characteristics because of the reasons explained above. Nevertheless, instead of working with all text blocks, one can work with some proper text blocks such as headers, which may be more meaningful for column disjunctions. In the next section, we extend the proposed overlapping algorithm for overcoming such issues.

4.4.2. Header Detection

The currently offered methods provide a general framework for determining the number of columns of a résumé. However, we encounter irregular column structures in the real-world data as shown in Figure 7.

Another example is given in Figure 8. This example can be classified as double-column, because the text blocks that include the personal information are positioned highly disconnectedly from the text blocks of the labels of this information even though they are related on the line basis. As long as these text blocks are not positioned close to each other Algorithm 1 will classify such résumés as double-column. As we know that these documents are single-column résumés in reality, the separated but actually related text blocks of personal information and their labels should be associated. In the example of Figure 8, the main label sections all overlap with each other. As a result, they will all fit into a single set of columns.

In Figure 9, the scatter plot of the (x_0, y_0) coordinates of the text blocks belonging to two résumé examples is shown. The points are coded with different colors with respect to the types of the text blocks: personal

information and label. Considering all of the types of text blocks together, the separation of the coordinates is vague in the example of a single-column résumé while it is quite obvious in the double-column sample. However, we can distinguish clearly the coordinate separation in both cases by using only labels.

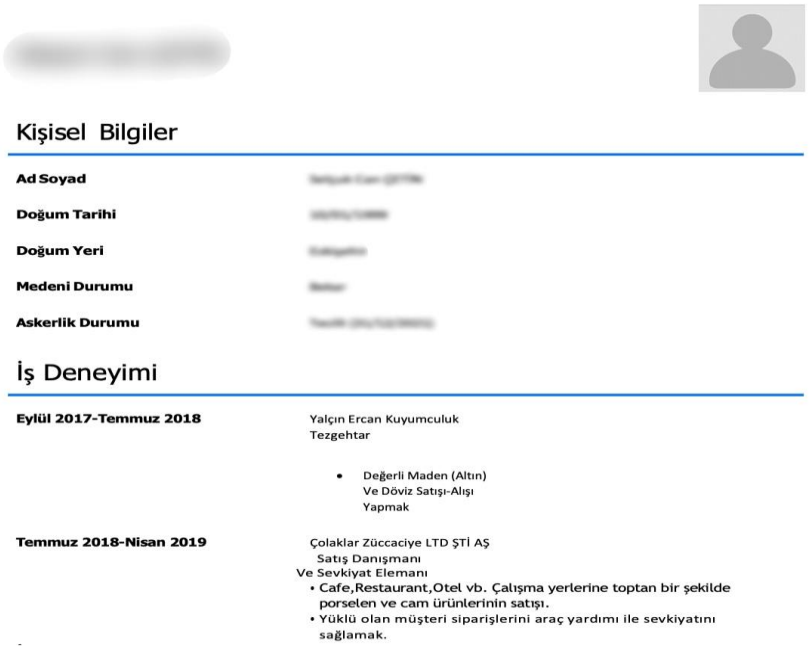
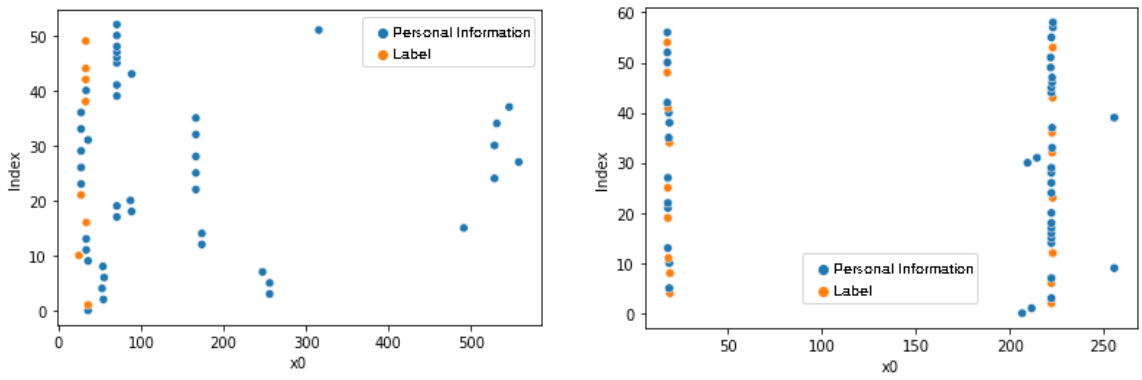


Figure 8. *A résumé example with one column but having a large distance between disjoint text blocks*



(a) **(b)**

Figure 9. *The scatter plot of the (x_0, y_0) coordinates of the text blocks* ***a)*** *A single column sample* ***b)*** *A double-column sample*

Not only for cases like the one shown in Figure 8 but also for any other résumé, checking the overlap of main labels is enough instead of checking all text block overlaps. In summary, determining the number of columns by extracting all text blocks may result in both unnecessary computational load and incorrect results, as shown in the previous examples.

We propose to consider the text block typology when selecting the important text blocks instead of all for overcoming the mentioned issue. We define the text blocks typology into three categories that are related hierarchically: *characteristics*, *headers* and *main headers*. In the hierarchy, we call a parent if the text block of interest is hierarchically superior and linked to a lesser text block. The lesser text block is called the child

of the text block of interest. This categorization allows us to use the important ones instead of considering all of them.

In the first category, there are text blocks that contain personalized explanations, which we call *characteristic*. Characteristics are at the lowest level in the text block hierarchy. They do not have any relationship with lower level text blocks, thus they have no children.

The second category is the *header*. We define a header as text blocks associated hierarchically with another header or characteristic. A header is not specific to a résumé, because they do not include personal information. Instead, the headers represent the meta-characteristics of personal information. For instance, "Education Level" is a header but "Master's degree" is a characteristic. Headers may be parents or children of other headers.

We call the third and the top category as *main header*. The main header text blocks don't have any parents. For example, a text block "Skills" is a main header whereas text blocks of "language", "certificates" or "hobbies" may be its children. But, in some résumés, "language" may become the main header as well.

A single-column résumé example is displayed in the Figure 10. Here, "Personal Information" is a main header but its child, "Language", is a header. Using the Algorithm 1 with header types text blocks, this résumé is classified as a double-column résumé because the headers are disjoint. The algorithm would classify it as single-column, if we were to use the main header.

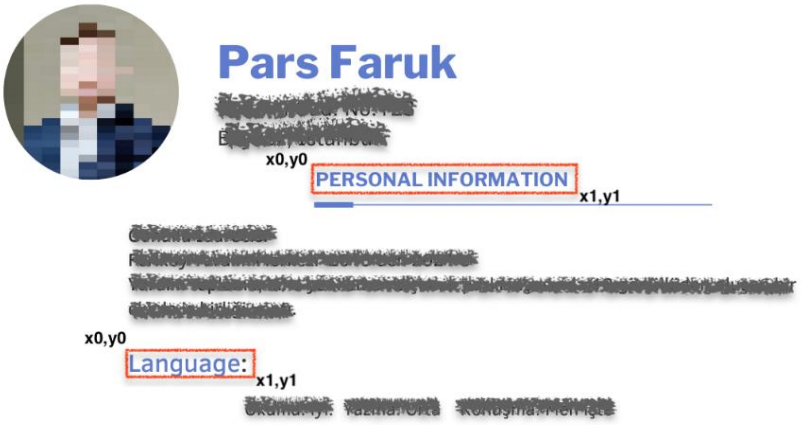


Figure 10. *A résumé example with headers and main headers*

As we mentioned previously, preformed document structure is unknown and can be built by using the properties of text blocks since usually, the text blocks belonging to the same section have the same font, size, or type, even though we encounter exceptions to this inference. Since we use Kariyer.net's data and there are professionals in this company who are especially specialized in reviewing résumés; we benefit from their expertise. Rather than developing a method for distinguishing the text block typology, we rely on the header and main header lists tailored by the experts of the company. We concentrate on the main headers list provided by Kariyer.net because their coordinates are the essential features for the detection of the number of columns.

We employ a text similarity check between each extracted text block and main header list elements by using sequence matcher approach (Rao et al., 2018). Once the main headers of studied résumé are detected, we apply the number of columns finder procedure given in Algorithm 1.

Even using the main headers, some résumés can be more challenging and their number of columns cannot be easily classified, even by the human experts. When we evaluate the example in Figure 7b, there are three main headers in one column and two main headers in the other column. In the résumés of regular formats, both columns have equal numbers of headers, while there can be more or less headers in the different columns in many cases. However, some real-world examples have even more complex structures with different numbers of main headers. In the method we propose to add new rules for such résumés. We mark their number of columns as uncertain in order not to make any wrong classifications.

4.5. Hybrid Column Detection

Each of the previously explained methods has different weaknesses and strengths. They cannot handle some complicated cases. Except DBSCAN, each one labels some types of résumés *uncertain*. The *uncertain* labels of different methods do not cover each other. In other words, one method can detect the number of columns while another one marks them as *uncertain*. We aimed to decrease the number of *uncertain* résumés. That is why we designed a *Hybrid Column Detection* by using Baseline, DBSCAN, and Overlapping Headers together. The sequential flow chart of the Hybrid Column Detection method is given in Figure 11. Coordinate values extracted from a résumé are sent to the Hybrid Column Detection Method. We give priority to the Overlapping Headers in the hybrid model. If the Overlapping Headers is *uncertain*, other methods are used.

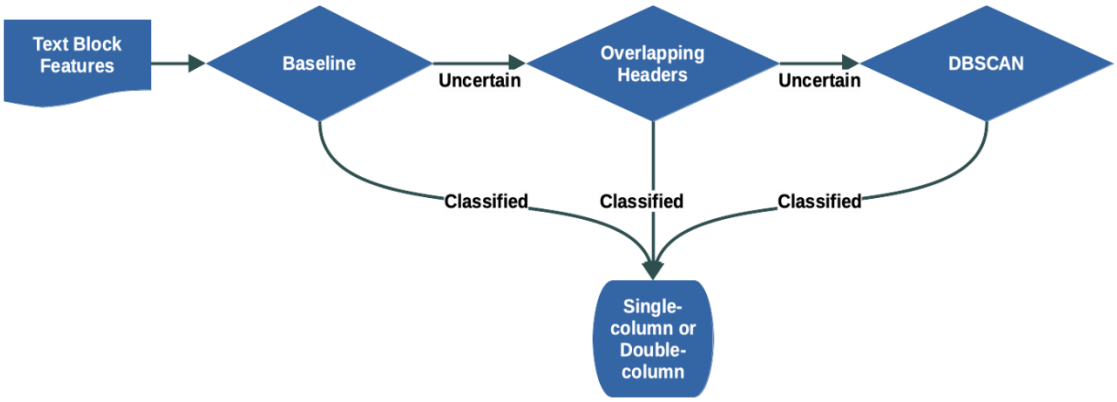


Figure 11. *Hybrid Column Detection Method Flowchart*

5. EXPERIMENTS AND RESULTS

The résumés used in the experiments are provided by Kariyer.net. They consist of a set of PDF documents having different fonts, colors and formats. All of them are authentic résumés created for real job applications in Turkey's marketplace. Some of them have been prepared in ready-made template formats, while some others are prepared by the applicants in free format. First, the text parsing process is put into practice. After removing any damaged files or those that could not be parsed, we finally have a total number of 1201 of documents in the dataset.

Prior to evaluating our algorithms, the résumés in the dataset are labeled with their true number of columns with the collaboration of human resources experts of Kariyer.net. This manually added ground truth information helps us to compare the efficacy of the methods objectively as in supervised learning even though all the approaches proposed in Section 3 are fundamentally unsupervised learning methods.

Each method has its own parameters that affect the performance: the ϵ and *minPts* in DBSCAN, or the silhouette value zone in GMM-based Clustering, which allows us to decide the number of clusters. Before evaluating the algorithms, we tune the parameters by using 15% of the master dataset (in 190 résumés). To prevent algorithmic biases, we reserve as many résumés as possible for testing. The parameter values for each method in these 190 résumés are chosen based on their best performances.

Usually, the number of columns may be clearly stated by the human expert without any question, but occasionally, their determination can be ambiguous even for the expert. As an example, a short part of a résumé at the beginning may be in single-column format where a short bio or personal information is placed, but the rest of the résumé that includes the other personal information like education, skills or experiences may be in double-column format. Some examples of such irregular résumés are displayed in Section 3, Figure 7a, 7b.

We perform two separate experiments on the test dataset. In the first batch of experiments, we only use résumés for which all experts agree on their numbers of columns. In other words, this set consists of the résumés whose columns can be distinguished relatively well by the human eye. Throughout our experiments, we refer to this set as the "reduced set". There is a total of 231 résumés, of which 195 documents are single-column and 36 are double-column. The methods are expected to estimate the true number of columns easily compared to the other résumés.

The second batch of experiments are performed on the set of résumés that are labeled according to the majority vote of the experts. We call this set as the "original set". In the original set, there are a total of 1011 résumés of which 678 and 333 are in single-column and double-column format respectively. The original set also includes the reduced set.

We report the results of the experiments on the reduced set of relatively easy-to-analyze résumés in Section 5.1. The results of the experiments on the original set are presented in Section 5.2. Finally, in Section 5.3, all outcomes are examined and discussed collectively.

5.1. Experiment with the Reduces Set

All of the aforementioned methods are initially evaluated on the reduced set, and the results of these experiments are reported in Figure 12 as confusion matrices. Additionally, Table 3 provides the F1-score, precision, recall and accuracy values attained to evaluate these methods. We also show the success on single- and double-column résumés separately on Table 4. The Hybrid Column Detection yields the same outcomes with the Overlapping Headers because the latter method performed on the reduced set does not produce any uncertain predictions.

First of al, DBSCAN and GMM-Based Clustering, give different results with an accuracy of 84% and 87% respectively. Concentrating on Figure 12a and 12b together, we distinguish that the DBSCAN performs poorly for double-column résumés. It could detect only 15 of 36 double-column résumés, which is less than 50% accuracy, while GMM-based Clustering catches 26 of 36, which is more than 70% accuracy. On the contrary of double-column résumés, DBSCAN gives better performance than the GMM-Based Clustering in single-column résumés. It seems DBSCAN has a bias towards detecting single-column. Considering F1-Score, precision, recall, and overall accuracy, GMM-based Clustering outperforms DBSCAN.

Table 3. *Results on the Reduced Set*

Method	Accuracy	F1-Score	Recall	Precision
Overlapping Headers	0.97	0.95	0.96	0.94
Baseline	0.87	0.77	0.81	0.75
DBSCAN	0.84	0.68	0.67	0.7
GMM-Based Clustering	0.87	0.77	0.81	0.75

Baseline predicts far less accurately than the Overlapping Headers in both single-column and double-column résumés (Table 4). Furthermore, Overlapping Headers has much less false negatives than the Baseline (Figure 12c and 12d). Baseline exhibits the same performance on the reduced set as GMM-based Clustering. We remind you that these two methods are based on statistical techniques. Both techniques use an assumption that x_0 follows a Gaussian distribution. Although their performances can be accepted as "good", we still underline that Overlapping Headers perform 10% better than them.

The reduced set includes résumés with relatively well-formed column structures, that is, the text blocks are not indented too far from the margins of the headers or the main headers, which offers in turn clearly detectable column margins. Thus, in fact, we expect that these two algorithms would perform as well as Overlapping Headers if the column structures could be identified by the x_0 coordinate Gaussian distribution properties. However, the experimental results reveal that this assumption cannot be sufficient for determining the number of columns.

The Overlapping Headers, which has an accuracy value of 97%, is the method with the highest accuracy among all. Almost any résumé can be predicted correctly by using this method. This result can be interpreted as the usage of the projection of main header coordinates to the horizontal axis being a good indicator for detecting the columns. In this relatively easy set, the three other methods also provide an accuracy rating of over 80%, which cannot be considered poor. But Overlapping Headers are highly successful by far. Compared to other methods, the DBSCAN includes more false negative predictions for double-column résumés. However, it behaves inversely on single-column résumés: the method produces a highly low rate of false negatives. As a result, it can be argued that single-column résumés generated using DBSCAN typically perform quite well. In particular, it is seen that it is much easier to analyze the more comprehensible résumés in the reduced set. All algorithms are effective at predicting the true labels of the résumés in the reduced set. Among them, Overlapping Headers is the most successful method.

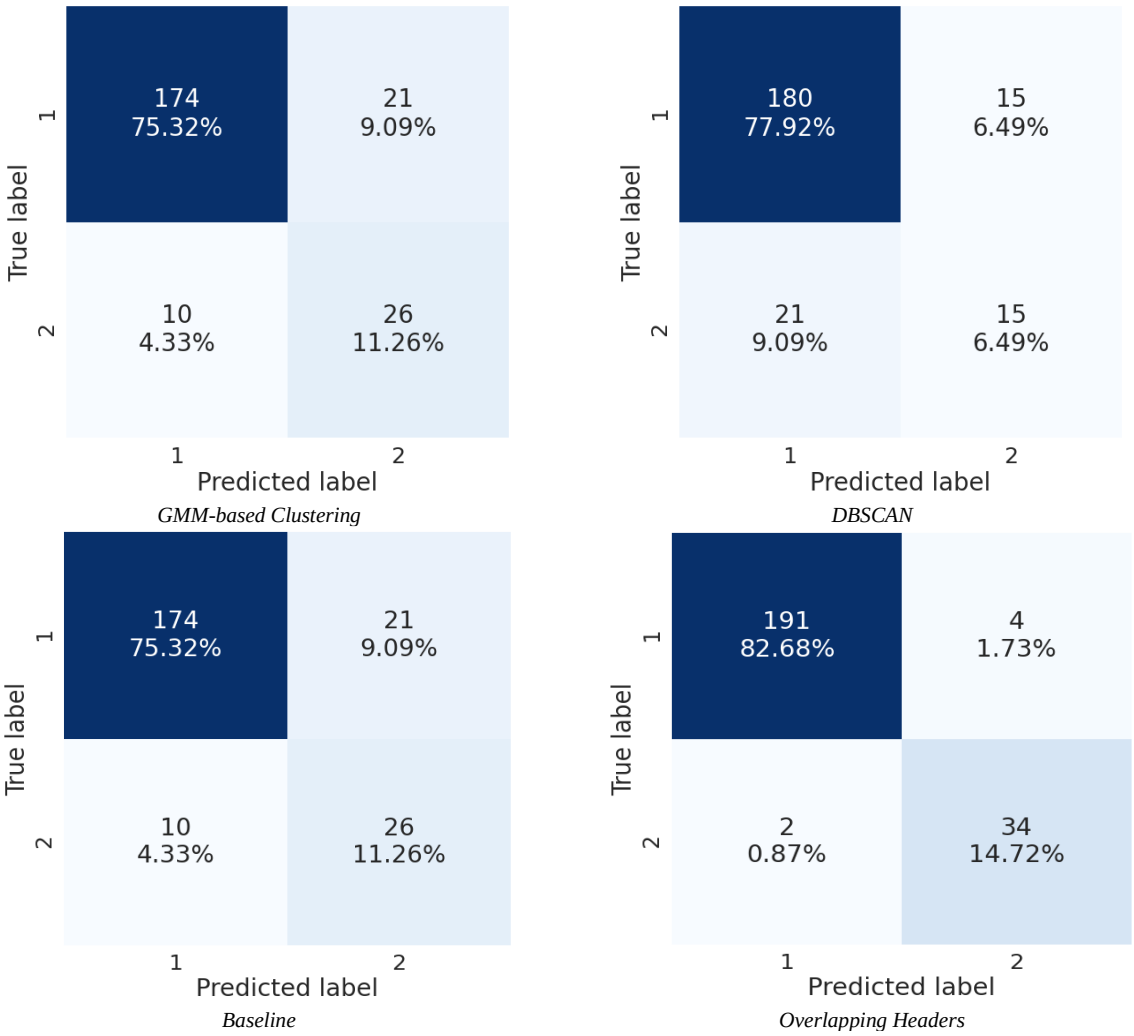


Figure 12. *Confusion matrices obtained after the experiments on the reduced set.*

Table 4. *Accuracies for Single- and Double-column résumés in the Reduced Set*

Method	Single-column	Double-column
Overlapping Headers	0.98	0.94
Baseline	0.89	0.72
DBSCAN	0.92	0.41
GMM-Based Clustering	0.89	0.72

5.2. Experiment with the Original Set

In the experiments carried out on relatively more understandable résumés in the previous section, the methods within the scope of this study perform promisingly. This section focuses on evaluating the performance of the set of all résumés, namely the original set. The corresponding confusion matrices are shown in Figure 13. Note that the methods except DBSCAN and the Hybrid Column Detection include the uncertain estimations within. These results report that DBSCAN yields an accuracy measure of 82%, dramatically higher than the GMM-based Clustering. When observing the separated performances on each column type regarding Table 5, we notice that not only on single- but also on double-column, GMM-based Clustering exhibits the worst performances. In both cases, in comparison with the other proposed methods, the GMM-based Clustering

produces too many *uncertain* estimations. In addition, it estimates 149 of 378 *non-uncertain* labels incorrectly. Its performance on both reduced and original sets reveals that it does not achieve much in deciding the number of columns when they are not obvious. When compared with GMM-based Clustering, Baseline is more successful, although they both use Gaussian distribution Assumptions. In particular, it has a moderate efficiency for single-column detection. But, as it was the case in the experiments with the reduced set, its performance is low for double-column detection. Considering the Baseline and Overlapping Headers, the Overlapping Headers outperforms the Baseline by correctly estimating 84% of the résumés, whereas the Baseline attains a percentage of 61% of the same set. Additionally, the Overlapping Headers yields less errors than the Baseline in its single- and double-column predictions. The résumés are labeled intermediately as *uncertain* by the Baseline and Overlapping Headers instead of being erroneously estimated. This results with less erroneous predictions in single- or double-column predictions.

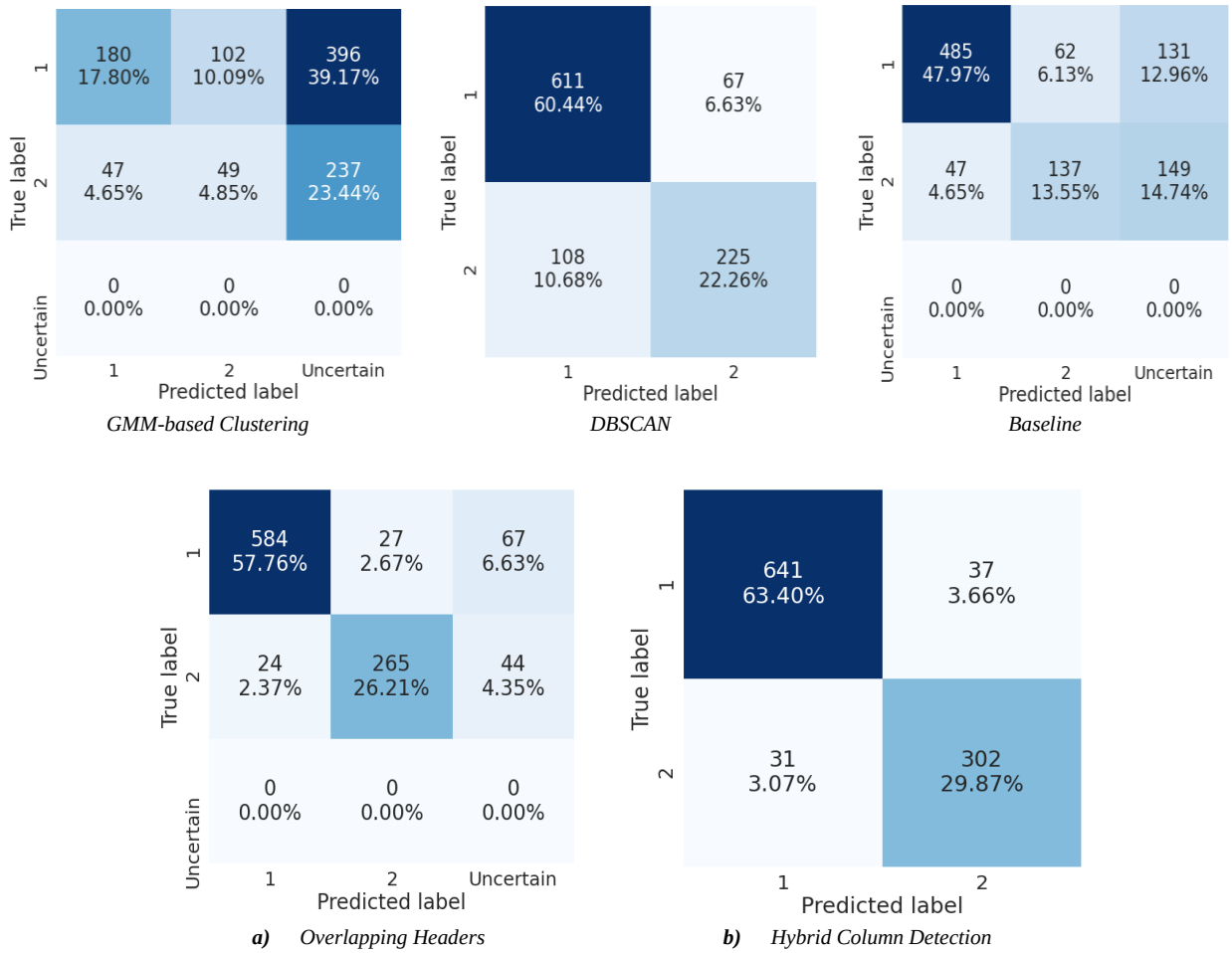


Figure 13. *Confusion matrices obtained after the experiments on the original set*

Among the two methods which does not use *uncertain* labels, the DBSCAN predicts 836 accurate predictions for single- and double-column, but it predicts the number of columns of 175 résumés incorrectly. Among all methods, DBSCAN produces the falsest negatives as it was the case in the reduced set. More importantly, it produces an incorrect assessment of the 108 double-column résumés. It estimates 32% of the résumé labels inaccurately. However, it estimates only 10% of single-column résumés incorrectly. Therefore, it can be said

that DBSCAN tends to accurately predict single-column résumés in the original collection, as it was the case in the reduced set. The Hybrid Column Detection, on the other hand, performs highly better than any other method, correctly predicting 641 single-column and 302 double-column résumés. It gives the highest accuracy on both single- and double-column. In Table 6, we report the F1-score, accuracy, precision and recall values of the experimental results on the original set. The Hybrid Column Detection outperforms the other number of columns prediction methods in terms of all of the aforementioned metrics. The GMM-Based Clustering, on the other hand, performs too poorly with an accuracy of 22%. Although the Baseline does not show a high performance with an accuracy of 61%, it estimates 622 of the résumés correctly (Figure 13c). It produces an incorrect single or double-column estimate for only 109 résumés.

Table 5. *Accuracies for Single- and Double-column résumés in the Original Set*

Method	Single-column	Double-column
Hybrid Column Detection	0.95	0.9
Overlapping Headers	0.86	0.79
Baseline	0.71	0.41
DBSCAN	0.9	0.67
GMM-Based Clustering	0.15	0.15

Table 6. *Results on the Original Set*

Method	Accuracy	F1-Score	Recall	Precision
Hybrid Column Detection	0.93	0.93	0.93	0.94
Overlapping Headers	0.84	0.89	0.84	0.93
Baseline	0.61	0.7	0.61	0.84
DBSCAN	0.82	0.82	0.82	0.82
GMM-Based Clustering	0.22	0.32	0.22	0.65

6. DISCUSSION

When we evaluate all the experimental results together, we see that the Overlapping Headers yields the best results in the experiments on the reduced set. Furthermore, on the original set, the Hybrid Column Detection, which employs the Overlapping Headers as an intermediate step to determine also the uncertain résumés, gives the most successful results. The poorest results are obtained with the two well-known clustering methods; DBSCAN and GMM-based Clustering. When we consider both the experimental performances and their own characteristics, we can reach the following conclusions for the studied methods;

1. Clustering the x_0 coordinates of the text blocks is a highly promising approach for detecting the number of columns of résumés.
2. Among the clustering approaches, clustering with DBSCAN algorithm tends to mark résumés as single-column.
3. The determination of the number of columns through statistical methods based on the Gaussian distribution fitting shows moderate performance if the columns are well separated. From these methods, simply stating the number of estimated Gaussian densities of the x_0 distribution performs better than more complex method of GMM-based Clustering.

4. The idea of hierarchical separation of text blocks and the use of more descriptive main headers seems effective for column number finding problem because Overlapping Header results with higher success rates in detecting the number of columns.
5. The Overlapping Headers approach performs better than both traditional clustering methods (DBSCAN or GMM-based Clustering) and the statistical fitting-based methods (Baseline or GMM-based Clustering).
6. We deduce that running other clustering methods on the x_0 coordinates of the headers or the main headers, instead of the x_0 coordinates of all text blocks can increase the performance in determining the number of columns. However, in this case, since the number of points that are to be clustered will be too small, it may be challenging to determine the parameters of the algorithms.
7. Every method has its own strengths and weaknesses. An incorrectly classified résumé by a certain method may be correctly classified by other methods. This leads us to combine the strengths of the methods successively.
8. The hybrid approach, built upon the strengths of the individual algorithms, makes use of the *uncertain* situations that are classified by the Overlapping Headers. This intermediate extension makes this approach the most successful approach in determining the number of columns. In the experiments that use the set of many challenging cases, the success rate of the Hybrid Column Detection is about 10% higher than that of Overlapping Headers.

Among the proposed approaches, we obtain the best results with the proposed hybrid approach. This one combines the strength of baseline, DBSCAN and proposed overlapping headers. Since it is based on several powerful methods, it captures the exception cases as well as the common cases. This method is used in real-world systems for detecting the headers of many different types of curriculum vitae. It is flexible for one and two columns and mixtures as well.

7. CONCLUSION

This study focused on determining the number of columns that a résumé file has, regardless of how the column structure is set up. This issue can be solved by methods such as natural language processing or text mining. However, configuring a file with these methods, whose document structure is unknown, requires labeling almost every section in that file. In many files, it is challenging to determine how the sections are divided, even with the human eye. As a result, such labeling tasks require a lot of time and effort but they might not be objective. Therefore, we refrained from using a supervised learning strategy, which necessitates meticulous document tagging. Instead, first, we suggested using the text blocks' coordinates that were taken from the résumés according to our empirical observations on the set of résumés taken from Kariyer.net; second, we defined the problem of finding the columns of a résumé as a clustering of their extracted text block coordinates; and third, we presented a new clustering method, Overlapping Headers, using these coordinates and compared its performance with the well-known clustering approaches.

In order to measure the performance of the proposed method, two separate experiments were conducted on the dataset provided by the Kariyer.net platform, with résumés whose number of columns were previously marked by experts. In the first experiment, column structures that are relatively easy to detect were used. In the second experiment, difficult résumés, in which it could not be easily decided how many columns there were, were in the majority. In both experiments, it was seen that the proposed method was more successful than the baselines. The Overlapping Headers in the easy set showed a high success rate of 97%. In the difficult set, the Hybrid Column Detection achieved 93% success. With the detailed evaluation of the methods, it was seen that traditional clustering approaches were one step behind the methods we suggested in determining the number of columns. Of these, DBSCAN tended to find a single column, while GMM-based Clustering performed quite unsuccessfully, with 22% accuracy, especially in experiments involving difficult cases. Thus, it turns out that novel clustering methods developed specifically for this problem give better results in determining the number of columns.

Although the study was performed on résumés, the suggested strategies are not résumé-specific and can be applied to any other types of text documents. Additionally, the proposed methodology is independent from the language of the résumé because it is based on texts' locations. It has the ability to work with any natural language. The approach is not affected by whether the text is written from left to right or right to left. This work can be extended along many different paths. First, all the methods can be tested on documents different from résumés. One expects that the methods should find the number of columns more accurately on other types of documents because résumés are one of the most irregular documents in terms of their structures. Second, an extension to the Overlapping Headers can be proposed because this method still cannot detect the number of columns for some of the difficult cases. In this work, we proposed using main headers rather than using all text blocks. A more definitive feature of the text blocks can be found by building their complete hierarchical structures. However, this can be an effort-demanding task since each résumé has almost its own free format. Third, in this work, we focus on finding only single and double columns. In some cases, a document might have three or more columns. Thus, Overlapping Headers can be generalized for more the number of columns. Finally, here we completely focus on unsupervised approaches. However, supervised learning based on the text block coordinates can also be successful. Since the résumé set that we use in our experiments is already labelled, a model can be trained.

AUTHOR CONTRIBUTIONS

Conceptualization, Y.B., G.K.O. and S.N.T; methodology, Y.B. and G.K.O.; fieldwork, Y.B. and S.N.T.; software, Y.B.; title, Y.B., G.K.O. and S.N.T.; validation, Y.B., G.K.O. and S.N.T.; laboratory work, Y.B., G.K.O. and S.N.T.; formal analysis, Y.B..; research, Y.B., G.K.O. and S.N.T.; sources, Y.B.; data curation, Y.B.; manuscript-original draft, Y.B., G.K.O. and S.N.T.; manuscript-review and editing, Y.B., G.K.O. and S.N.T.; visualization, Y.B..; supervision, G.K.O. and S.N.T.; project management, G.K.O. and S.N.T. . All authors have read and legally accepted the final version of the article published in the journal.

ACKNOWLEDGEMENT

This study was conducted within the scope of the R&D Project Cooperation signed between Galatasaray University and Kariyer.net in accordance with Article 58/K of the YÖK Law No. 2547.

CONFLICTS OF INTEREST

No conflict of interest was declared by the authors.

REFERENCES

Adnan, K., & Akbar, R. (2019). An analytical study of information extraction from unstructured and multidimensional big data. *Journal of Big Data*, 6(1), 1-38. https://doi.org/10.1186/s40537-019-0254-8

Alamelu, M., Kumar, D. S., Sanjana, R., Sree, J. S., Devi, A. S., & Kavitha, D. (2021, December). Resume validation and filtration using natural language processing. In *2021 10th International conference on internet of everything, microwave engineering, communication and networks (IEMECON)* (pp. 1-5). IEEE. https://doi.org/10.1109/IEMECON53809.2021.9689075

Chen, K., Yin, F., & Liu, C. L. (2013, August). Hybrid page segmentation with efficient whitespace rectangles extraction and grouping. In *2013 12th International Conference on Document Analysis and Recognition* (pp. 958-962). IEEE. https://doi.org/10.1109/ICDAR.2013.194

Chen, J., Zhang, C., & Niu, Z. (2018). A Two-Step Resume Information Extraction Algorithm. *Mathematical Problems in Engineering*, 2018(1), 5761287. https://doi.org/10.1155/2018/5761287

Çelik, D., & Elçi, A. (2012). An ontology-based information extraction approach for résumés. In *Proceedings of the 2012 International Conference on Pervasive Computing and the Networked World ICPCA/SWS'12* (p. 165–179). Berlin, Heidelberg: Springer-Verlag. https://doi.org/10.1007/978-3-642-37015-1_14

Das, P., Pandey, M., & Rautaray, S. S. (2018). A CV parser model using entity extraction process and big data tools. *International Journal of Information Technology and Computer Science*, 9(2), 21-31. https://doi.org/10.5815/ijitcs.2018.09.03

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd* (Vol. 96, No. 34, pp. 226-231).

Farkas, R., Dobó, A., Kurai, Z., Miklós, I., Nagy, Á., Vincze, V., & Zsibrita, J. (2014). Information extraction from Hungarian, English and German CVs for a career portal. In *Mining Intelligence and Knowledge Exploration: Second International Conference, MIKE 2014, Cork, Ireland, December* 10-12, 2014. Proceedings (pp. 333-341). Springer International Publishing. https://doi.org/10.1007/978-3-319-13817-6_32

Gaur, B., Saluja, G. S., Sivakumar, H. B., & Singh, S. (2021). Semi-supervised deep learning based named entity recognition model to parse education section of resumes. *Neural Computing and Applications*, 33, 5705-5718. https://doi.org/10.1007/s00521-020-05351-2

Holm, A. B. (2012). E-recruitment: Towards an ubiquitous recruitment process and candidate relationship management. *German Journal of Human Resource Management*, 26(3), 241-259 https://doi.org/10.1177/239700221202600303

Ji, X., Zeng, J., Zhang, S., & Wu, C. (2010). Tag tree template for Web information and schema extraction. *Expert Systems with applications*, 37(12), 8492-8498. https://doi.org/10.1016/j.eswa.2010.05.027

Joan, S. P. F., & Valli, S. (2019). A survey on text information extraction from born-digital and scene text images. *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, 89, 77–101. https://doi.org/10.1007/s40010-017-0478-y

Kariyer.net (2025) URL

Keskin, Ş. R., Balı, Y., Orman, G. K., Daniş, F. S., & Turhan, S. N. (2022, June). Determining Column Numbers in Résumés with Clustering. In *IFIP International Conference on Artificial Intelligence Applications and Innovations* (pp. 460-471). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-08337-2_38

Kokoska, S., & Zwillinger, D. (2000). CRC standard probability and statistics tables and formulae. *Crc Press.*

Liu, X., Gao, F., Zhang, Q., & Zhao, H. (2019). Graph convolution for multimodal information extraction from visually rich documents. *arXiv preprint arXiv:1903.11279.*

Luo, Y., Zhang, H., Wang, Y., Wen, Y., & Zhang, X. (2018, November). ResumeNet: A learning-based framework for automatic resume quality assessment. In *2018 IEEE International Conference on Data Mining (ICDM)* (pp. 307-316). IEEE. https://doi.org/10.1109/ICDM.2018.00046

Mao, S., Rosenfeld, A., & Kanungo, T. (2003). Document structure analysis algorithms: a literature survey. In *T. Kanungo, E. H. B. Smith, J. Hu, & P. B. Kantor (Eds.), Document Recognition and Retrieval (pp. 197 – 207). International Society for Optics and Photonics SPIE* volume 5010. https://doi.org/10.1117/12.476326

Mittal, V., Mehta, P., Relan, D., & Gabrani, G. (2020). Methodology for resume parsing and job domain prediction. *Journal of Statistics and Management Systems*, 23(7), 1265-1274. https://doi.org/10.1080/09720510.2020.1799583

Qin, C., Zhu, H., Xu, T., Zhu, C., Ma, C., Chen, E., & Xiong, H. (2020). An enhanced neural network approach to person-job fit in talent recruitment. *ACM Trans. Inf. Syst.*,38 https://doi.org/10.1145/3376927

Rao, G. A., Srinivas, G., Rao, K. V., & Reddy, P. P. (2018). A partial ratio and ratio based fuzzy-wuzzy procedure for characteristic mining of mathematical formulas from documents. *IJSC—ICTACT J Soft Comput*, 8(4), 1728-1732. https://doi.org/10.21917/ijsc.2018.0242

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*,20, 53–65 https://doi.org/10.1016/0377-0427(87)90125-7

Roy, P. K., Chowdhary, S. S., & Bhatia, R. (2020). A machine learning approach for automation of resume recommendation system. *Procedia Computer Science*, 167, 2318-2327. https://doi.org/10.1016/j.procs.2020.03.284

Sarawagi, S. (2008). Information extraction. Foundations and Trends® in Databases, 1, 261–377. https://doi.org/10.1561/1900000003

Sinha, A.K., Amir Khusru Akhtar, M., Kumar, A. (2021). Resume Screening Using Natural Language Processing and Machine Learning: A Systematic Review. In: *Swain, D., Pattnaik, P.K., Athawale, T. (eds) Machine Learning and Information Processing. Advances in Intelligent Systems and Computing,* vol 1311. Springer, Singapore. https://doi.org/10.1007/978-981-33-4859-2_21

Sonar, S., & Bankar, B. (2012). Resume parsing with named entity clustering algorithm. *Published paper, SVPM College of Engineering Baramati, Maharashtra.*

Tejaswini, K., Umadevi, V., Kadiwal, S. M., & Revanna, S. (2022). Design and development of machine learning based resume ranking system. *Global Transitions Proceedings*, 3(2), 371-375. https://doi.org/10.1016/j.gltp.2021.10.002

Tobing, B. C. L., Suhendra, I. R., & Halim, C. (2019, June). Catapa resume parser: end to end Indonesian resume extraction. In *Proceedings of the 2019 3rd international conference on natural language processing and information retrieval* (pp. 68-74). https://doi.org/10.1145/3342827.3342832

Xu, Q., Zhang, J., Zhu, Y., Li, B., Guan, D., & Wang, X. (2020). A blocklevel RNN model for resume block classification. In *2020 IEEE International Conference on Big Data (Big Data)* (pp. 5855–5857). https://doi.org/10.1109/BigData50022.2020.9377771

Xuan, G., Zhang, W., & Chai, P. (2001, October). EM algorithms of Gaussian mixture model and hidden Markov model. In *Proceedings 2001 international conference on image processing* (Cat. No. 01CH37205) (Vol. 1, pp. 145-148). IEEE. https://doi.org/10.1109/ICIP.2001.958974

Yakubovich, V., & Lup, D. (2006). Stages of the recruitment process and the referrer's performance effect. *Organization science*, 17(6), 710–723. https://doi.org/10.1287/orsc.1060.0214

Yasmin, F., Nur, M. I., & Arefin, M. S. (2020). Potential candidate selection using information extraction and skyline queries. In *Proceeding of the International Conference on Computer Networks, Big Data and IoT* (ICCBI-2019) (pp. 511-522). Springer International Publishing. https://doi.org/10.1007/978-3-030-43192-1_58

Yu, K., Guan, G., & Zhou, M. (2005, June). Resume information extraction with cascaded hybrid model. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)* (pp. 499-506).

Zaroor, A., Maree, M., & Sabha, M. (2017, May). A hybrid approach to conceptual classification and ranking of resumes and their corresponding job posts. In *International Conference on Intelligent Decision Technologies* (pp. 107-119). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-59421-7_10

Zu, S., Wang, X., & Darren, S. (2019). Resume information extraction with a novel text block segmentation algorithm. Linguistics, 8, 29–48. doi:10. 5121/ijnlc.2019.8503. In *Proceedings 2001 International Conference on Image Processing* (Cat.No.01CH37205) (pp. 145–148 vol.1). volume 1.