International Journal of Enhanced Research in Management & Computer Applications
ISSN: 2319-7471, Vol. 13, Issue 4, April-2024, Impact Factor: 8.285
Presented at "ICRETETM-2024", Organized by GSMCOE, Pune, on 22nd - 23rd April 2024

Ai Resume Analyzer Using Natural Language Processing and Data Mining

Jayshri Mankar*[1], Abhishek Chauhan*[2], Aniket Gophane*[3], Aditya Karle*[4],Taslimarif Makandar*[5], Akshay Funde*[6]

Department of Computer Engineering, G. S. Moze College of Engineering,Balewadi, Pune-411045, India

ABSTRACT

Imbalance data conversion into structured data is the very tedious task in data mining techniques, various techniques have been already introduced to extract the data from large textand extract the features using various feature extraction techniques, some machine learning algorithms have been already introduced by various researchers for classification and display the results on heterogeneous data. This work suggests a method of eliminating or resuming important information in a curriculum vitae from the semi-structured textformat, and rating it according to the preference and requirements of the client. The whole process was divided intothree basic sections to achieve the desired goal. The first section consists of segmenting the whole Summary according to the content of each part, the second section consists of extracting data in a standardized form from unstructured data and the final section consists of analyzing structured data using NLP and Machine learning algorithms. The Stanford NLP ruleextraction algorithm ha used to extract the various rules from raw data and select some important feature for classification as well as optimization. Experimental analysis shows the effectiveness of proposed system with classification accuracy.

Keywords: Resume parsing, Data Mining, Machine learning,NLP

INTRODUCTION

he need of thinking and designing an appropriate resume. Classification is specially essential in solutions to data processing and machine-learning. Nowadays, many outlets have generated the numerous types of data in row format, as well as its hard to process from existing environments and algorithms. Text classification requires assigning the text to one or more predefined groups using some kind of classification algorithm performed by the content of the document. A Generic classification corpus has been developedand a single assessment system has been introduced to identifyEnglish text based on machine learning, which has now madesignificant progress. Most of the evidence in the real world is contained in relational bases. Data clustering is an essential machine learning process in which a sub-set of candidate labels is allocated to an entity, the main issue with multi-label clustering is the redundant online clustering method and the offline data set for dealing with this issue. We plan to use unstructured data classification to structured conversion systems and maximize the accuracy of the final sub-cluster. Demonstrate two implementations of our method using logistic regressions and improved gradient trees, along with asimple procedure for Expectation Maximization preparation. We also get an efficient prediction approach dependent on dynamics programming.
Resume Analyzer is software developed to simplify the task ofcreating a resume in structured format. The system is flexible to be used and reduces t

LITERATURE REVIEW

According to [1] a recruiting case study as a basis for a statistical evaluation of several methods for calculating similarity scores. To this end, we suggest using a computer- aided resume evaluator on a group of resumes, then has professionals evaluate the same set of resumes, and finally look for a connection between the two sets of results. Finding the right computer-aided resume evaluator for digital human resources requires a consideration of the various similarity score calculation methodologies now available for processingresumes.

According to [2] an approach to resume writing that is both straightforward and straightforward to apply. We propose a program that, given some basic information about the applicant, may generate a professional-looking résumé. Users may sign up for an account and start working on their resume by entering their login details and receiving a one-time password (OTP).

According to [3] Recruiters have a hard time finding the greatest fit for a job position since the resumes candidates submit vary in format (e.g., font, color, font size, etc.). To combat these issues, recruiters may turn to natural language processing (NLP) to glean the specifics about potential candidates they need to move their candidacy ahead. In this paper, we suggest using the Stanford CoreNLP system's named entity recognition capabilities to glean data useful in the hiring process.

According to [4] Data-driven HR has been shown to significantly enhance the quality and speed of the whole recruitment process by using Natural Language Processing tools. First, a resume parser has been built utilizing natural language processing to evaluate the most important aspects of the hiring process. The computational framework of the parser was then used to create a potent instrument for resume matching based on job requirements, with the candidate's ability to produce a pie chart serving as an input.

According to [5] It might be challenging to find qualified candidates for an open position, particularly if there are numerous applications to choose from. It may be difficult for the team to find the most qualified candidate at the most opportune time if they have to go through each resume manually. An automated method of screening and rating applications might significantly reduce the time spent on the screening process. The KNN algorithm is used to select and rank Curriculum Vitaes (CV) based on job descriptions in large numbers, and the cosine similarity is used to find the CVs that are most relevant to the provided job description in our work.

According to [6], They have propose a substance extraction approach for getting content from news pages that joins a division like methodology and a thickness based methodology. A tool Block Extractor is used to identifies contents in three steps. First, it looks for all Block-Level Elements and Inline Elements blocks, which are designed to roughly segment pages into blocks. Second, it computes the densities of each BLE and IE block and its element to eliminate noises. Third, it removes all redundant BLE and IE blocks that have emerged in other pages from the same site. Compared with three other density-based approaches, our approach shows significant advantages in both precision and recall. BLE and IE blocks to gather related noises or contents. Next, we used this density- based approach and redundancy removal to obtain the final content. Based on our approach, a tool called Block Extractor was developed.

In this paper [7], The issue of naturally removing web information records that contain user generated content (UGC).To solve this problem MiBAT and MDR algorithms are used 1)MiBAT (Mining data records Based on Anchor Trees). They have represented two space imperative guided likeness measures, for example PM and PS. They have propose an information record mining calculation utilizing either PM or PS. Our instinct is exceptionally basic: each record comprises of one or a few sub trees, just one of which contains the rotate. We call such sub-trees that contain turns as grapple trees, since they give stay point data about where information records are found. 2)MDR (Mining Data Records in Web pages) MDR identifies a list of records by conducting a similarity test against a pre-defined threshold for two sub- trees in the DOM tree of a web page. Such a method is referred to as the similarity-based approach, because the underlying assumption is that data records belonging to the same list usually have similar DOM tree structures MDR and its Limitations:-A group of similar objects, which forms a data region, is usually presented in a contiguous region and format- ted using similar HTML tags. Every record in a data region is formed by the same number of adjacent child sub-trees under the same parent node. Novel mining algorithm called MiBAT which makes use of domain constraints to acquire anchor point information. Our methodology accomplishes an exactness of 98.9% and a review of 97.3% concerning post record extraction. On page level, it lawlessly handles 91.7% of pages without removing any o_-base posts or missing any brilliant posts.

This paper [8] depicts a framework for robotized continue data extraction to help fast resume search and the board. The framework is equipped for extricating a few significant educational fields from a free arrangement resume utilizing a lot of common language handling (NLP) strategies. We depict a working framework, for programmed continue the board. The framework is equipped for extricating six significant fields of data as characterized by HR-XML In this the main layer is made out of a few general data squares, for example, individual data, instruction and so forth. The second layer of structure is inside the principal layer and contains explicit data comparing to the layer 1. For instance, the layer 1 individual data square comprises of layer 2 data like name, address and email. While this probably won't be valid for every one of the resumes, the structure is by all accounts held in the greater part of resumes. Furthermore, the area of the data (like name, age

and so forth) in resumes differs fundamentally from resume to continue. Our framework can chip away at both layered structure and unstructured resumes. Data extraction module ismade out of a few sub modules, every one of which plays out the undertaking of removing explicit data. The primary sub modules are (a) Qualification module, (b) Skill module (c) Experience module and (d) individual data extraction. While the capability extraction sub-module separates the graduating college name, degree and the class acquired. The aptitudes extraction module extricates the abilities of the applicant. Experience extraction module is competent or removing the all-out understanding, in any event, when this data isn't expressly referenced in the resume of the candidate. The extraction procedure utilizes a lot of language preparing systems which are part heuristics and part example coordinating. Test results completed on countless resumes demonstrate that the proposed framework can deal with an enormous assortment of resumes in various record positions with an exactness of 91% and a review of 88%.

In this paper [9] we present a near investigation of 5 estimates utilizing distinctive vector loads done over an enormous arrangement of French list of qualifications. The point is to know how these measures act and whether they approve the idea that chose list of qualifications have more in a similar manner as themselves than with the dismissed list of qualifications. We utilize NLP systems and ANOVAs to do the relative examination. The outcomes demonstrate that the determination of measures and vector loads must not be viewed as insignificant in e-Recruitment projects; especially in those where the resumes' resemblance is estimated. Something else, the outcomes may not be dependable or with the normal execution. Four sorts of archives are dissected in this work: pdf, Microsoft Word, Open Document Text and Rich Format Text.

In this paper [10] the general target of this examination was toconcentrate such information as experience, highlights, and business and training data from resumes put away in HR archives. In this article, we propose a philosophy driven data extraction framework that is intended to work on a few million free-design printed resumes to change over them to an organized and semantically improved rendition for use in semantic information mining of information basic in HR forms. The engineering and working instrument of the framework, similitude of the idea and coordinating strategies, and a deduction system are presented, and a contextual investigation is displayed.

RESEARCH METHODOLOGY

In our proposed study, a machine learning strategy based on a recommendation algorithm is used to predict student achievement using both synthetic datasets and real-time student data. Consider figure for a thorough explanation of the machine learning classifier-based proposed design. The proposed system architecture with the machine learning classifier is shown in Figure 1. We first gathered information from a variety of sources, including different web applications, actual student information, company information and a few synthetic data sets from diverse sources**.**

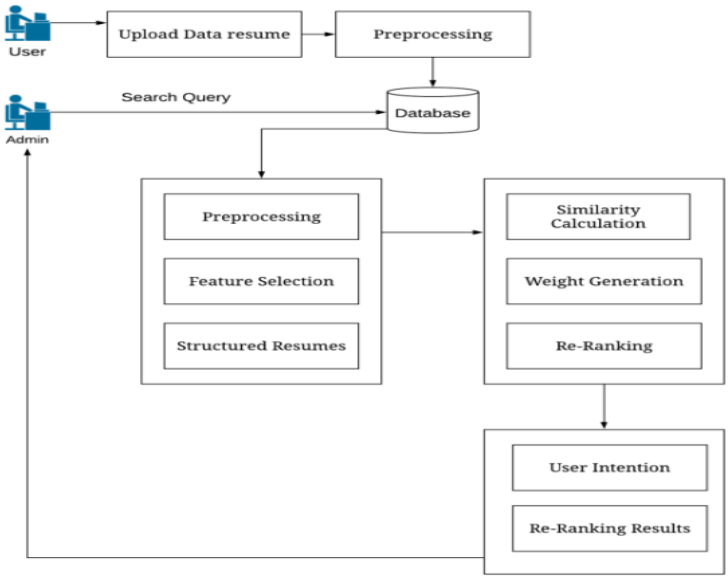

Fig 1: System Architecture of Proposed SystemList of Modules and Functionality

The proposed system follows below procedure for entire execution to convert unstructured data to structured conversion in entire process.

- Initially some raw resume data has given input to system, it should be unstructured format. (it should be doc, pdf file)
- Read data from entire document and apply stop wordremoval as well as porter stemming algorithm to get the lemmas features.
- Natural Language Processing (NLP) is another technique has used to extract the features from text using dependency parser.
- To identify the name of specific entity has used NameEntity Recognizer of Stanford NLP.
- Once semi-structured format has done, selected feature has place in structured format and using any respective machine learning algorithm.
- Once clarification has done we calculate the confusion matrix for entire test data and predict the precision, recall, accuracy etc. respectively.

ALGORITHM DETAILS:

Similarity (Machine learning) Algorithm:

Input: Training set with established standards or guidelines Performing a test using data that has been normalised from the Train_Data. Test_Data indicated the threshold Th_values.

Output: Output set using the parameters {Predicted_class, weight_Score}

Step 1: The following code can be utilised to read each test result from the Test_Data array and verify training criteria. Subsequently, the data is standardised and modified to align with the requirements of the algorithms.

$$\text{test\_Feature(data)} = \sum_{m=1}^{n} (.\ \text{Attribute\_Set[A[m] ..... A[n]} \leftarrow Test\_Data)$$

Step 2: Select the characteristics from the extracted attributes set of the test. Using the provided code, generate a feature map by utilising the data as features.

Test_Feature_List [t…..…n] = $\Sigma(t)_{nx=1} \leftarrow$ test_Feature(x)
Test_Feature_List [x] contains the features that were chosen for further inspection.

Step 3: Subsequently, it is necessary to thoroughly examine the entire training dataset to design the rule employed to categorise all test data.

$$\text{train\_Feature\_List(data)} = \sum_{m=1}^{n} (.\ \text{Attribute\_Set[A[m] ..... A[n]} \leftarrow Train\_Data)$$

Step 4: Generate the training dataset by utilising the provided function on the input dataset.

Train_Feature_list [t……..…n] = $\Sigma(t)_{nx=1} \leftarrow$ train_Feature(x)

The enumerated regulations The Train_Feature_list[t] generate the feature vector that is utilised to build the hidden layer. This utilizes the train data to evaluate all test cases.

Step 5: The similarity weight is determined once the feature map has been generated.

$$Gen\_weight = CalcSimilarity\ (Test\_Feature\_List \ || \sum_{i=1}^{n} Train\_Feature\_List[i])$$

Step 6: Assess the disparity between your current weight and your desired weight.

if(Gen_weight> =qTh)

Step 7: Out_List.add **(**trainF.class,weight**)**

Step 8: Advance to step 1 and proceed once the Test is finished. The data variable has a null value.

Step 9: ReturnOut_List.

1 : Stop word Removal Approach
Input: Stop words list L[], String Data D for remove the stop words.
Output: Verified data D with removal all stop words.
Step 1: Initialize the data string S[].
Step 2: initialize a=0,k=0
Step 3: for each(read a to L)

If(a.equals(L[i]))
Then Remove S[k]
End for
Step 4: add S to D.
Step 5: End Procedure

2 Stemming Algorithm.
Input : Word w
Output : w with removing past participles as well.
Step 1: Initialize w
Step 2: Intialize all steps of Porter stemmer
Step 3: for each (Char ch from w)
If(ch.count==w.length()) && (ch.equals(e))
Remove ch from(w)
Step 4: if(ch.endswith(ed))
Remove 'ed' from(w)
Step 5: k=w.length()
If(k (char) to k-3 .equals(tion))
Replace w with te.
Step 6: end procedure.

RESULTS

The partial implementation of proposed system has been completed for the training module. As per our first module wehave used standard pdf data set of 2 student data files for dataset. The below figure 2 shows the ranking show of wholedata

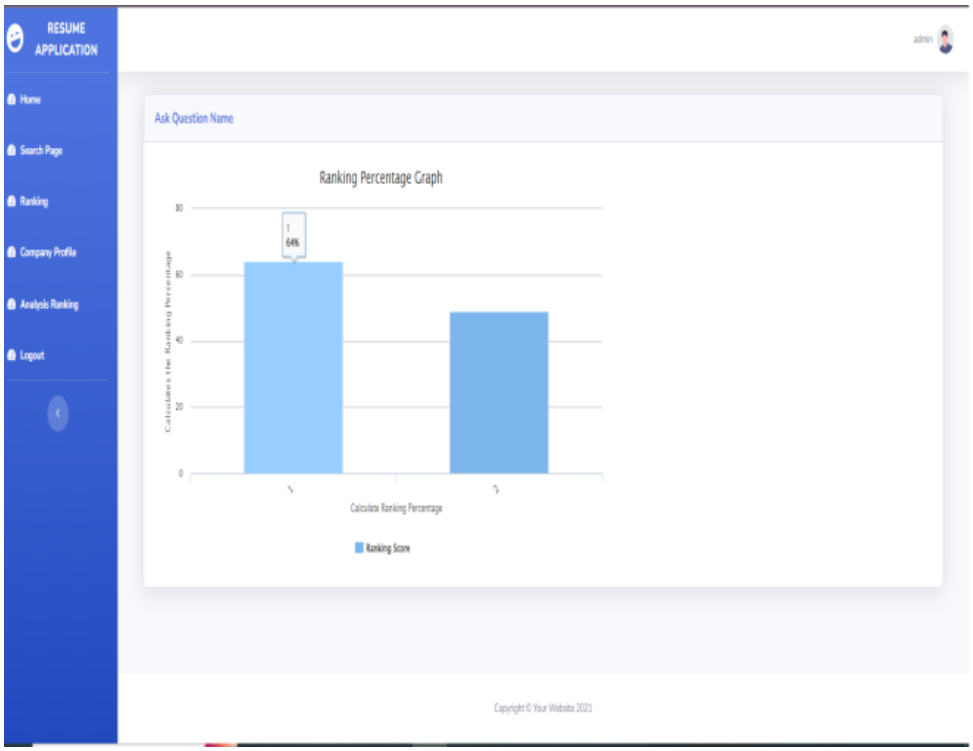

Fig 2: Ranking Graph

As per our module we have used standard Input pdf data set . The below figure 4 shows the output file of whole data

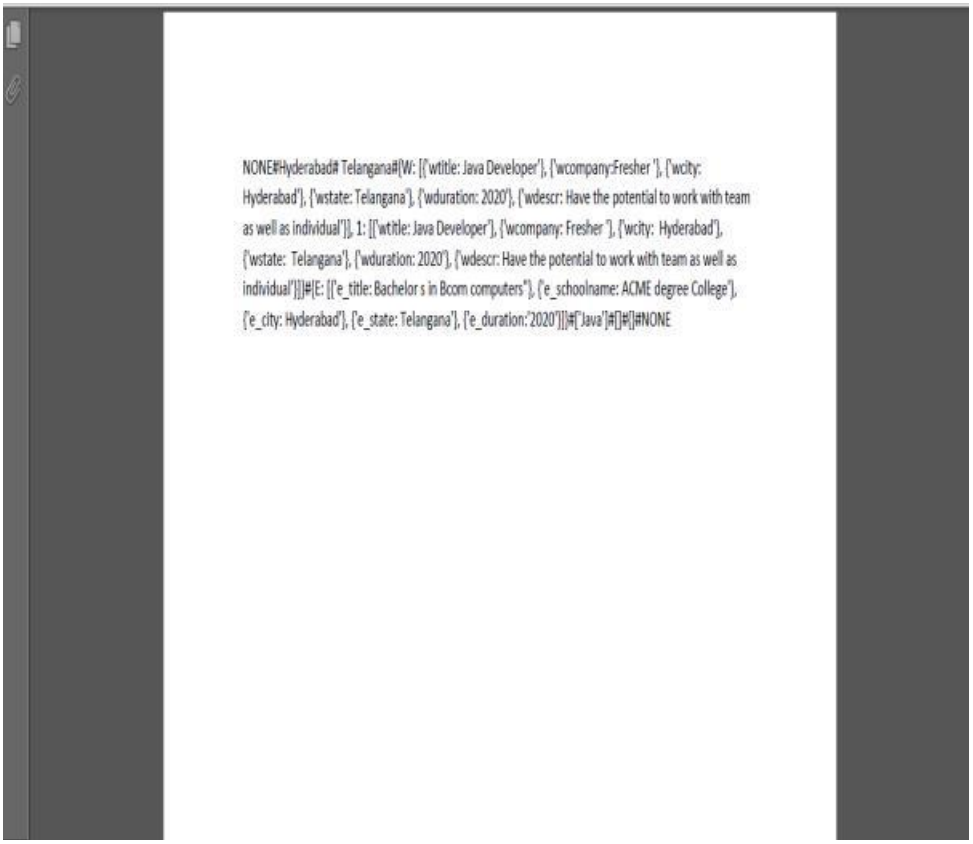

Fig 3: System Input File Format

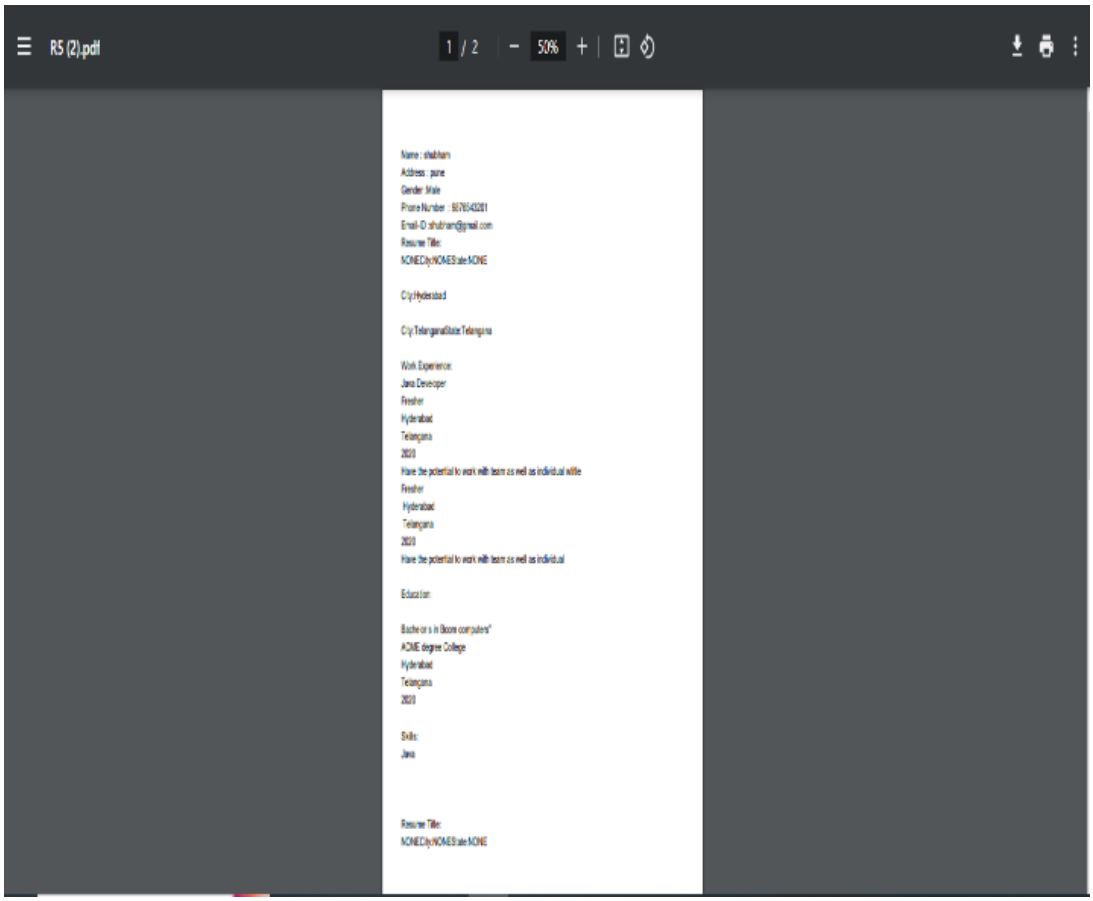

Fig 4: System Output File Format

CONCLUSION

Based on the proposed experimental analysis this system willprovide better and efficient solution to current hiring process. This will provide potential candidate to the organization and the candidate will successfully be placed in an organization which appreciates users skill set and ability and speed up the whole hiring process. To work with various kinds of large unstructured data will be future work for such systems.

REFERENCES

[1]. Özçevik, Yusuf, Fatih Yücalar, and Murat Demircioğlu. "Determining a Proper Text Similarity Approach for Resume Parsing Process in a Digitized HR Software." Celal Bayar University Journal of Science 18.4 (2022): 371-378.
[2]. Tyagi, Rinki, et al. "Resume Builder Application." International Journal for Research in Applied Science and Engineering Technology (IJRASET) Volume 8 (2020).
[3]. Mittal, Vrinda, et al. "Methodology for resume parsing and job domain prediction." Journal of Statistics and ManagementSystems 23.7 (2020): 1265-1274.
[4]. Deepak, Gerard, Varun Teja, and A. Santhanavijayan. "A novel firefly driven scheme for resume parsing and matching based on entity linking paradigm." Journal of Discrete Mathematical Sciences and Cryptography 23.1 (2020): 157- 165.
[5]. Tejaswini, K., et al. "Design and development of machine learning based resume ranking system." Global Transitions Proceedings 3.2 (2022): 371-375.
[6]. Shuang Lin, Jie Chen, Zhendong Niu, \Combining a Segmentation-Like Approach and a Density-Based Approach in Content Extraction \ , August 2017.
[7]. Xinying Song, Jing Liu, Yunbo Cao, Chin-Yew Lin, and Hsiao-Wuen Hon , "Automatic Extraction of Web Data Records Containing User-Generated Content", Jan. 2015.
[8]. Sunil Kumar Kopparapu , \Automatic Extraction of UsableInformation from Unstructured Resumes to Aid Search",

Oct.2016.

[9]. Paul, P. Efficient Graph-Based Document Similarity. In: The Semantic Web. Latest Advances and New Domains, Springer International Publishing, 2016, pp 334–349.

[10]. Wang, J, Dong, Y. 2020. Measurement of Text Similarity: A Survey. Information, vol 11(9).

[11]. Farouk, M. 2019. Measuring Sentences Similarity: A Survey. Indian Journal of Science and Technology, 12(25), 1–11.

[12]. Yujian, L, Bo, L. 2007. A Normalized Levenshtein Distance Metric. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 29 (6), pp 1091-1095.

[13]. Dreßler, K, Ngomo, A, N. 2017. On the Efficient Execution of Bounded Jaro-Winkler Distances. Semantic Web, vol 8, pp 185-196.

[14]. Del, M, Angeles, M, García-Ugalde, F, Valencia, R, Nava, A. Analysis of String Comparison Methods During DeDuplication Process. International Conference on Advances in Databases, Knowledge, and Data Applications, Rome, Italy, 2015, pp 57-62.

[15]. Bakkelund, D. 2009. An LCS-based string metric. Olso, Norway: University of Oslo. [19]. Kondrak, G. N-Gram Similarity and Distance. In String Processing and Information Retrieval, Springer Berlin Heidelberg, 2005, pp 115–126.

[16]. Luis Adri_an Cabrera-Diego1,2, Barth_el_emy Durette2, Matthieu Lafon2, Juan-Manuel Torres-Moreno1,3 and Marc El-B_eze1, "How Can We Measure the Similarity Between R_esum_es of Selected Candidates for a Job? Luis Adri_an Cabrera-Diego1,2, Barth_el_emy Durette2, Matthieu Lafon2, Juan-Manuel Torres- Moreno1,3 and Marc El-B_eze1" , Jan.2014.

[17]. Duygu C_EL_IK,"Towards a semantic-based information extraction system for matching resumes to job openings", June2016.