Automated Resume Parsing and Ranking using Natural Language Processing

Thangaramya K
School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India.
thangaramya.k@vit.ac.in

Logeswari G
School of Computer Science and Engineering, Vellore Institute of Technology, Chennai Campus, India.
logeswari.g@vit.ac.in

Sudhakaran Gajendran
School of Electronics Engineering, Vellore Institute of Technology, Chennai Campus, India.
sudhakaran.g@vit.ac.in

Deepika Roselind J
School of Computer Science and Engineering, Vellore Institute of Technology, Chennai Campus, India.
deepikaroselind.j@vit.ac.in

Neha Ahirwar
School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India.
neha.ahirwar2020@vitstudent.ac.in

2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT) | 979-8-3503-7212-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/AIIoT58432.2024.10574696

***Abstract*- Advancements in networking and communication have revolutionized the recruitment process, leading to the development of modern resume parsing and ranking systems. With the proliferation of internet-based recruiting, numerous resumes are now stored in recruitment systems. However, traditional methods, such as manual processing and the utilization of unique resume templates, have limitations when dealing with unstructured documents like resumes. Resume parsing is a crucial technique that involves extracting essential details pertaining to applicants to be shortlisted for interview. This paper introduces the "Resume Parser and Ranker," an application designed to efficiently and automatically rank resumes, saving considerable time and manpower during the recruitment process. The application is powered by Natural Language Processing (NLP) approach, incorporates heuristic calculations to evaluate the final score of each candidate. The system utilizes Deep Learning (DL) for Named Entity Recognition (NER), achieving an impressive 93% accuracy in information extraction. Notably, this accuracy is close to the level achieved by humans, which typically does not exceed 96%. The high accuracy of the resume parser ensures reliable results, enabling companies to identify the most suitable candidates who can be called for further interview rounds.**

Keywords—Natural Language Processing, Deep Learning, Named Entity Recognition, Resume Parser, Machine Learning.

I. INTRODUCTION

In today's competitive job market, recruitment processes are becoming increasingly complex and time-consuming due to the vast number of job applications received by companies and organizations. Human resource professionals often find themselves overwhelmed with the manual task of screening and evaluating numerous resumes to identify potential candidates for a job position [1].

To alleviate this burden and streamline the recruitment process, automated resume parsing and ranking systems have emerged as invaluable tools. Automated resume parsing involves the use of technology, specifically NLP, to extract relevant and essential information from resumes. Traditional parsing techniques are effective for structured documents, but they fall short when dealing with unstructured data like resumes, which often vary in format, layout, and content [2][3]. NLP, a subfield of artificial intelligence, empowers machines to understand and process human language, making it an ideal technology to handle the challenges posed by unstructured resumes [4][5].

The process of automated resume parsing begins with the conversion of resume files, typically in formats like PDF or DOCX, into machine-readable text using Optical Character Recognition (OCR) or text extraction libraries. Once the resumes are converted into plain text, NLP algorithms and techniques are applied to comprehend and extract valuable information from various sections [6]. Such information extracted are structured into a standardized format, making it easier for further analysis and comparison. Resume ranking complements the parsing process by evaluating and scoring candidates based on predefined criteria or job-specific requirements. Each candidate's parsed data is assessed using heuristic calculations or machine learning models, taking into account factors such as years of experience, education level, relevant skills, and previous job roles [7]. By ranking candidates automatically, the system significantly reduces manual efforts, enabling recruiters to focus on more strategic tasks, such as conducting interviews and engaging with potential hires.

The integration of NLP in automated resume parsing and ranking systems offers several advantages [8]. Firstly, it greatly accelerates the recruitment process, allowing companies to swiftly identify suitable candidates from a large applicant pool. Secondly, the system enhances the overall accuracy and consistency of candidate evaluation, as it eliminates human biases and ensures objective analysis based on predefined criteria. Additionally, the technology can be easily scaled to handle high volumes of resumes, making it suitable for organizations of all sizes.

This paper introduces a comprehensive and efficient "Resume Parser and Ranker" application that utilizes the features of NLP for automating the screening of resumes of applicants and to rank them. The application leverages advanced NLP algorithms, such as Named Entity Recognition (NER) and syntactic parsing, to attain near-human accuracy in information extraction. This will lead to more precise evaluations of candidates. The outcome is a potent tool that transforms the recruitment process by optimizing time and resources and identifying the most suitable candidates for different job positions.

979-8-3503-7212-0/24/$31.00 ©2024 IEEE

In this proposed methodology, the application of NLP is employed to parse resumes tailored to specific companies' requirements. A unified job portal is offered to both employers and employees, facilitating job applications and creation. Resumes received through the portal undergo parsing and ranking based on company preferences. Furthermore, the objective is to extract data from PDF-format resume folders to streamline the recruitment process, ensuring fair practices and avoiding discrimination across different regions. The primary aim is to automate the process, addressing the challenges mentioned above. This automation solution efficiently identifies suitable resumes from extensive databases, aligning them with the job descriptions provided by recruiters. The code significantly reduces the time-consuming manual process of individually opening and reviewing each resume, achieving this task automatically through content parsing.

II. Related Work

The literature survey focuses on recruitment policies and procedures, aiming to establish a systematic and repeatable approach to candidate selection. For larger organizations, the resource-intensive and complex recruitment process often results in the risk of overlooking talented candidates. The survey explores various methodologies and technologies, particularly NLP, to optimize and automate the process.

Duygu Celik et al. [9] presented a hierarchical extraction method for parsing resumes in PDF format to extract information. The proposed approach involved fragmenting the resumes into various blocks and classifying them using a Conditional Random Field (CRF) model for information extraction from each block. The classification model utilized content-based features and layout-based features parsed from the resumes. While the proposed method demonstrated promising results, it had limitations in handling table-style resumes as it mainly focused on list-style resumes. Additionally, the study primarily concentrated on extracting specific information, such as personal details and educational background, from certain sections of the resumes. Ayishathahira et al. [10] introduced one new parsing technique using DL classification algorithms for analyzing the resumes.

To facilitate word embedding, the system incorporated a pre-trained glove model. However, the model encountered challenges due to the ambiguity inherent in the English language, which is known for its vast and diverse linguistic variations. Kelkar B et al. [11] presented a new method by combining DL with CRF model. Their model was able to handle the resume processing more efficiently. However, one notable drawback of CRF was its high computational complexity during the training stage of the algorithm. This complexity posed challenges when re-training the model with new training data samples. Moreover, CRF was not compatible with unknown words, meaning it could not handle words not present in the training data samples.

Vedant Bhatia et al. [12] utilized SVM-based methods for resume parsing, focusing on both LinkedIn and non-LinkedIn CV formats to extract data in a structured manner. The parsing process was then followed by sentence-pair classification for ranking, employing Bidirectional Encoder Representations from Transformers (BERT).

Additionally, the extracted text played a role in generating a ranking score for applicants using BERT. However, this research had a primary limitation as it could only handle the parsing of resumes in the LinkedIn resume format. When presented with a document of any other format, the parsing process resulted in a significant loss of information. Bhavya et al. [13] conducted a comparative study on various approaches using the SVM algorithm for entity recognition from multiple languages. The study identified ambiguity and abbreviations as the main disadvantages of the process. Handling words with multiple meanings and those that can be part of different sentences posed significant challenges.

Liu et al. [14] proposed a resume classification model using Convolutional Neural Network (CNN) algorithm for text classification. They used scoring based on words for job matching. However, the algorithm lacked spatial invariance to the input data and required a considerable amount of training data. Nisha et al. [15] introduced the smart applicant ranker, a new recommendation system designed to assist recruiters in the candidate selection process. The system utilized the K-Nearest Neighbors (KNN) algorithm to recommend suitable candidates based on the job requirements provided by the recruiters. However, the KNN algorithm exhibited limitations, such as slow performance with a large number of observations, computational inefficiency, and difficulties in selecting the appropriate value of k. Overall, the surveyed studies provide valuable insights into various approaches for resume parsing and candidate ranking, each with its unique strengths and limitations.

III. Proposed Work

The proposed system incorporates four modules. Initially, the dataset undergoes a pre-processing phase to eliminate stop words, punctuations, and non-ASCII characters. Subsequently, the pre-processed data is subjected to various analytical methods for analysis. The extracted data is then compared with the company-provided CSV file using phrase matching to categorize the candidates based on resume factor identification. Lastly, the data is visualized using heuristics methods. Fig. 1. presents the system architecture of the proposed system.

A. Pre-processing

Resumes exhibit diverse formats, encompassing variations in document type, layout, writing style, and overall structure. The absence of a standardized resume format contributes to these differences. To effectively apply text mining techniques, it is necessary to perform raw text extraction from resumes. In the proposed methodology, PDFbox is employed for extracting the relevant raw texts from resumes that are in PDF format. For documents with file extensions other than .pdf, a prior conversion to PDF is performed to enable raw text extraction. This is necessary because PDFbox is limited to extracting text solely from PDF format documents.

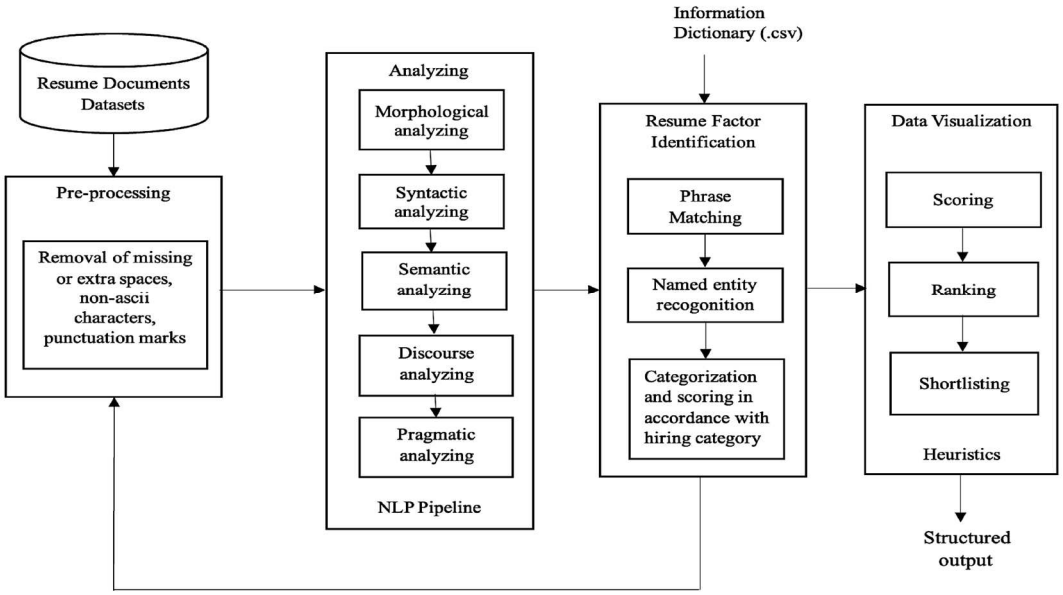

Fig. 1. System Architecture

The extracted data undergoes pre-processing to through whitespace and stop word removal.

Space Removal – Spaces are used in the resume to do better formatting. However, they must be removing before analysis. The spacing discrepancies introduce noise in the extracted text, which is subsequently addressed and cleaned during the analyzing phase.

Punctuations– The text extracted for analysis may have a large amount of punctuations these punctuations will misinterpret the analysis pattern. For instance, a skill such as "Adobe Illustrator" might be represented as "-Adobe Illustrator" due to punctuation, resulting in the model interpreting it as two different words. To mitigate such issues and avoid discrimination in similar text, the punctuation marks are removed during data pre-processing. This step ensures that the extracted text is cleaner and more suitable for accurate information extraction.

Non-ASCII characters – During the text extraction process, PDFbox may generate non-ASCII characters in the output text file, especially in response to bullet points, emoticons, or icons. To ensure data cleanliness and accuracy during the analyzing phase, all non-ASCII characters are removed.

Analyzing - The analyzing phase comprises five additional phases, which serve as key components of the NLP pipeline. These phases are as follows:

- Morphological Analyzer
- Parser (Syntactic Analysis)
- Semantic Analysis
- Discourse Analysis
- Pragmatic Analysis

Recent advances in DL for NLP have revolutionized information extraction, achieving state-of-the-art results without the need for hand-crafted features. In this system, a

These phases collectively contribute to the comprehensive analysis of the text during the NLP pipeline. The analyzing subsystem in the proposed methodology involves five key components of the NLP pipeline to process the raw text extracted from resumes and generate analysed data. The morphological analyzer examines word formation, analyzing root words, affixes, and parts of speech to understand the structure of a language. The parser constructs a tree assigning parent words to each word in a sentence, with the main verb as the root. The semantic analyzer determines the true meaning of sentences through NER and Coreference Resolution, identifying real-world concepts and assigning pronouns to corresponding nouns. The discourse analyzer comprehends the overall text, identifying involved entities, connections between sentences, and cause-consequence relationships. Lastly, the pragmatic analyzer considers context to extract meaningful information, minimizing ambiguity and providing crucial insights into the text's actual meaning.

B. Resume Factor Identification

Instead of manually searching for desired skills in candidates, the code utilizes phrase matching to automatically identify keywords, count their occurrences, and categorize them. Following the analyzing phase, the system obtains separate details of personal information, educational background, skills, work experience, and interests.

NER is then applied to extract named entities from these text blocks. NER plays a crucial role in information extraction, aiming to locate, extract, and classify named entities into predefined categories. Early NER techniques relied on rule and dictionary-based approaches, which required domain specialists to design and develop rule sets. However, these rules were not comprehensive, and the building process was time-consuming and less portable.

comma-separated file (.csv file) is utilized to specify the skills the company requires. This approach results in a structured output with each word tagged as a named entity, as depicted

in Fig 2. The most critical aspect is the acceleration it provides to the process. The algorithm generates a graph that aids in determining which candidates possess more keywords in each category, indicating their potential proficiency in those domains. Consequently, this approach significantly expedites the selection procedure. The heuristic scoring system involves points assigned to different skill categories, based on the Hiring Category. The top 10% of resumes with the highest scores are shortlisted for further consideration.

	Candidate Name	Subject	Keyword	Count
0	charles's resume	DE	sql	1
1	charles's resume	Stats	forecasting	1
0	gokul's resume	R	powerpoint	1
0	harini's resume	NaN	NaN	NaN
0	jumana's resume	NaN	NaN	NaN
..	...	...	...	...
1	preethi's resume	NLP	word	1
2	preethi's resume	R	powerpoint	1
3	preethi's resume	DL	accpac	1
0	sandhiya's resume	NLP	microsoft office	1
1	sandhiya's resume	DL	accpac	1

Fig. 2. Structured Output

C. Data Visualization

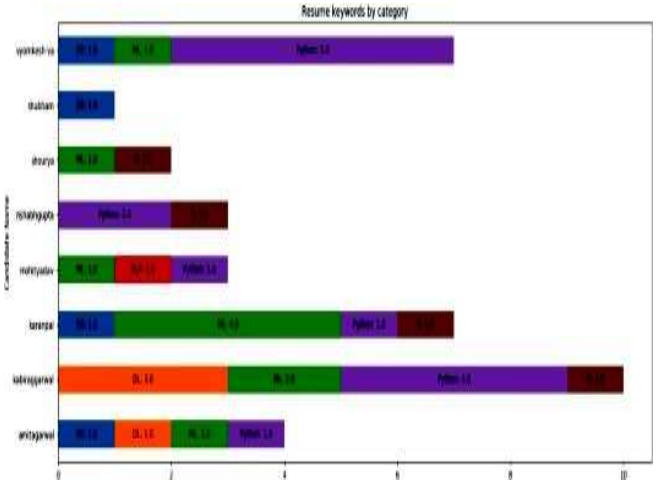

Fig. 3. Candidates Vs. Resume score

The heuristic calculation involves considering the number of skills (A1, A2, ..., An) a person possesses in various categories, where Ai represents the number of skills in the ith category. Additionally, X1, X2, ..., and XN are the multipliers or points assigned to the corresponding categories. The score is calculated as the sum of (Ai * Xi) for each category, where I ranges from 0 to N (Equation 1).

$$\text{Score} = \sum \text{Ai} * \text{Xi} \; \forall \; 0 \leq I \leq N \qquad (1)$$

Following the heuristic scoring and selection process, the resumes are sorted in descending order based on their scores. The top 10% of resumes are then chosen for further interview or test processes. To meet the visualization requirements, the proposed system includes visualization modules. The system utilizes the "Matplotlib" library in Python to create a bar graph depicting the candidates and their corresponding resume scores based on their skills. Fig 3 displays this graph, offering a visual representation of the candidates' scores and skill levels.

.

IV. Results and Discussion

A. Extracting the text

Resumes exhibit variations in their format, including differences in structure and format. To extract text and resumes from the folder, a function is defined, and each resume is loaded one by one. The PyPDF2.PdfFileReader is employed to read the pdf file, and the number of pages in the file is determined using .getNumPages(). The text is then extracted from the pdf using .extractText(), and the extracted content is stored in an array. For Natural Language Processing in Python, the spaCy library, which is a free and open-source tool, is utilized. Specifically, the largest English model of spaCy, en_core_web_lg, is employed for pre-processing English language text. The extracted data undergoes pre-processing by considering elimination of noise, punctuations and spaces.

B. Analyzing the Extracted Text

To begin, the matcher is initialized with a shared vocabulary that corresponds to the documents it will process. Using the matcher.add() function, we can provide an ID and a list of patterns to match within the documents. Here, matching is performed using the identity, starting place and the ending place. This process allows us to list specific subjects, keywords, and their respective counts along with the candidate's name. For matching the vocab of the documents, we utilize the company's requirements provided in the form of a comma-separated file (csv file). This file contains information about the skills and experience the company is seeking. We then extract the skills and experience of the candidates based on the data specifiedin the csv file provided by the company. Subsequently, a data frame is formed to organize the candidates' information. Fig 4 showcases the process of categorizing the skills of the candidates.

	Candidate Name	Subject	Keyword	Count
0	allen b	ML	svd	2
1	allen b	NLP	word	1
2	allen b	R	powerpoint	1
3	allen b	DL	accpac	1
4	allen b	DL	deep learning	1
5	allen b	DE	hive	2
6	allen b	Python	bokeh	2
7	allen b	DL	cuda	2
8	allen b	WebDev	webdev	1
9	allen b	DE	amazon redshift	1
10	allen b	NLP	bag of words	1
11	allen b	Stats	probability	1
12	allen b	DL	neural networks	1
13	allen b	Python	scipy	1
14	allen b	DE	sql	1
15	allen b	Experience	2 years	1

Fig. 4. Categorizing the skills of the candidate's structured output

C. Named Entity Recognition

The candidate's strengths in specific subjects can be recognized using the count function of Python. By applying the count function to the extracted text or data, we can determine the frequency of specific keywords or subjects mentioned in the resumes. This enables us to identify the candidates' proficiency and expertise in particular areas based on the occurrence of relevant keywords or skills in their resumes.

D. Scoring and Ranking

First, the Python library "pretty table" is imported to frame the data in a visually appealing tabular format. Subsequently, specific scores are assigned to individual skills and experience levels. With this information, the resumes of the candidates are scored based on their mentioned skills and experience. The scores are calculated by summing up the assigned scores for each skill and experience level present in the resumes. Once the scoring is complete, the resumes are arranged in the descending order of scores. This sorting process allows the candidates with higher scores to be ranked at the top, providing a clearer view of their qualifications and suitability for the desired positions.

Fig 5 presents the scorecard of the candidates. The scorecard provides an overview of the candidates' scores, indicating their proficiency in various skills and experience levels. The scores are calculated based on the assigned scores for each skill and experience mentioned in their resumes. The scorecard allows easy comparison and evaluation of candidates, aiding in the selection process for the desired positions.

E. Shortlisting

After scoring the resumes of the candidates based on their skills and experience, the next step is to shortlist the top 10% of the resumes. This shortlisting process involves selecting the best resumes based on the scores representing the most qualified and suitable candidates for the desired positions. These top 10% of resumes will be further considered for the next stages of the recruitment process, such as interviews or tests. Shortlisting the top candidates helps streamline the selection process and ensures that only the most promising applicants move forward in the hiring process.

Fig 6 showcases the list of selected candidates. These candidates are the top 10% of the applicants who have been shortlisted based on their high scores and proficiency in the required skills and experience. The selected candidates are considered to be the most qualified and suitable individuals for the desired positions, and they will proceed to the next stages of the recruitment process, such as interviews or further assessments.

Score Card of the Candidates :

Name	Score
aravind t	206
gowtham d	204
anand n	184
dharan t	181
shruthi s	178
santhosh t	168
eshwaram v	164
sreeniti a	162
geetha t	156
harish u	154
guru l	145
nirmal s	138
feroz m	135
yash p	134
lekha s	134
nelson t	131
rishitha j	124
allen b	121
sai nadhan m	120
sowmiya v	119
kiran v	119
priya l	117
fayeka t	116
madhumitha c	110
meena r	109
karthik m	105
rinku s	102

Fig. 5. Scorecard of the candidates

Selected Candidates :

Name	Score
aravind t	206
gowtham d	204
anand n	184
dharan t	181
shruthi s	178

Fig. 6. Selected Candidates

Precision and recall are essential evaluation metrics, particularly when the costs associated with false positives and false negatives are significant. Relying solely on accuracy as an evaluation metric can sometimes be misleading, which is why the F1 score is incorporated as an additional supporting metric. The system's performance yielded impressive results, with a precision of 89.32%, a recall of 83.4%, and an F-score of 86.25%. The AI-enabled parser efficiently processes even the most complex resumes, taking only 1-3 seconds. To achieve highly accurate information extraction, Deep Learning techniques were applied to NER, resulting in an impressive 93% accuracy rate. Remarkably, the resume parsers achieved "near-human accuracy," which is remarkable considering that human accuracy typically does not exceed 96%. These advancements in resume parsing

contribute significantly to streamlining the recruitment process and ensuring the identification of the most suitable candidates for various job positions.

V. Conclusion and Future Enhancement

This paper highlights the application of machine learning and NLP to enhance ourdaily lives, demonstrated through the example of Resume Screening. The proposed approach aims to streamline the recruitment process, making it more efficient and effective for both companies and candidates. By leveraging NLP techniques, the system ranks resumes based on their technical skills, ensuring that the most qualified applicants are identified. Moreover, the system aims to mitigate unfair and discriminatory practices in the recruitment process, promoting a more equitableselection of candidates. Future works for this research include expanding the capabilities to parse resumes from various platforms such as LinkedIn, GitHub, Naukri.com, and other applications. Additionally, the system could be enriched by incorporating a wider range of psychometric tests to further assess the candidates' suitability for specific positions. To enhance the system's performance, future efforts could involve enlarging the resume dataset and optimizing the proposed algorithms. Continual improvements and advancements in this system hold the potential to revolutionize the recruitment process and make it even more versatile and efficient in the future.

References

[1] Sanyal, S., Hazra, S., Adhikary, S., & Ghosh, N. (2017). Resume parser with natural language processing. International Journal of Engineering Science, 4484.

[2] Satheesh, K., Jahnavi, A., Iswarya, L., Ayesha, K., Bhanusekhar, G., & Hanisha, K. (2020). Resume ranking based on job description using SpaCy NER model. International Research Journal of Engineering and Technology, 7(05), pp. 74-77.

[3] Nimbekar, R., Patil, Y., Prabhu, R., & Mulla, S. (2019, December). Automated resume evaluation system using NLP. In 2019 International Conference on Advances in Computing, Communication and Control (ICAC3) (pp. 1-4). IEEE.

[4] Bhor, S., Gupta, V., Nair, V., Shinde, H., & Kulkarni, M. S. (2021). Resume parser using natural language processing techniques. International Journal of Research in Engineering and Science (IJRES), 9(6), (pp. 01-06).

[5] Sadiq, S. Z. A. M., Ayub, J. A., Narsayya, G. R., Ayyas, M. A., & Tahir, K. T. M. (2016). Intelligent hiring with resume parser and ranking using natural language processing and machine learning. International Journal of Innovative Research in Computer and Communication Engineering, 4(4), pp. 7437-7444.

[6] Daryani, C., Chhabra, G. S., Patel, H., Chhabra, I. K., & Patel, R. (2020). An automated resume screening system using natural language processing and similarity. ETHICS AND INFORMATION TECHNOLOGY [Internet]. VOLKSON PRESS, pp. 99-103.

[7] Pimpalkar, A., Lalwani, A., Chaudhari, R., Inshall, M., Dalwani, M., & Saluja, T. (2023, February). Job Applications Selection and Identification: Study of Resumes with Natural Language Processing and Machine Learning. In 2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) (pp. 1-5). IEEE.

[8] 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), pp. 1265-1274.

[9] Çelik, D., Karakas, A., Bal, G., Gültunca, C., Elçi, A., Buluz, B., & Alevli, M. C. (2013, July). Towards an information extraction system based on ontology to match resumes and jobs. In 2013 IEEE 37th annual computer software and applications conference workshops (pp. 333-338). IEEE.

[10] Ayishathahira, C. H., Sreejith, C., & Raseek, C. (2018, July). Combination of neural networks and conditional random fields for efficient resume parsing. In 2018 International CET Conference on Control, Communication, and Computing (IC4)(pp. 388-393). IEEE.

[11] Kelkar, B., Shedbale, R., Khade, D., Pol, P., & Damame, A. (2020). Resume analyzer using text processing. Journal of Engineering Sciences, 11(5), pp. 353-361.

[12] Bhatia, V., Rawat, P., Kumar, A., & Shah, R. R. (2019). End-to-end resume parsing and finding candidates for a job description using bert. arXiv preprint arXiv:1910.03089.

[13] Bhavya S. K., Kavya S. G., Sai S. S., Pranathi S. C., & Durga B. Y. (2022). CV Parsing Using NLP. International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), 2(1), (pp. 2581- 9429).

[14] Liu, J., Shen, Y., Zhang, Y., & Krishnamoorthy, S. (2021, May). Resume parsing based on multi-label classification using neural network models. In Proceedings of the 6th International Conference on Big Data and Computing (pp. 177-185).

[15] Nisha, B., Manobharathi, V., Jeyarajanandhini, B., & Sivakamasundari, G. (2023, December). HR Tech Analyst: Automated Resume Parsing and Ranking System through Natural Language Processing. In 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS) (pp. 1681-1686). IEEE.