Optimizing Resume Parsing Processes By Leveraging Large Language Models

Vazzula Manish
Dept. of Information Technology
Vardhaman College of Engineering
Hyderabad, India
vazzulamanish@gmail.com

Yugandhar Manchala
Dept. of Information Technology
Vardhaman College of Engineering
Hyderabad, India
yugandhar1230@gmail.com

Y. Vijayalata
Dept. of Computer Science & Engineering
Vardhaman College of Engineering
Hyderabad, India
vijaya@ieee.org

Sudeep Banerjee Chopra
Dept. of Computer Science (AI & ML)
Vardhaman College of Engineering
Hyderabad, India
sudeepbanerjee@ieee.org

Kothi Yashwanth Reddy
Dept. of Computer Science (AI & ML)
Vardhaman College of Engineering
Hyderabad, India
yashwanthreddy@ieee.org

2024 IEEE Region 10 Symposium (TENSYMP) | 979-8-3503-6486-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/TENSYMP61132.2024.10752300

***Abstract*—In the present era of internet revolution, organizations have to go through numerous resumes in order to identify the most suitable candidates for their Job Description (JD), while ensuring that no acquisition of talent is overlooked through human mistakes. Thus, tools like Applicant Tracking Systems (ATS) have taken over the human process of resume screening, enabling assessment of thousands of resumes in a matter of seconds. Although these technologies are incredibly effective, they are not flawless. Consequently, highly qualified individuals may miss out on said opportunities if their resumes are not formatted correctly. Therefore, individuals must ensure that their resume is appropriately structured prior to submitting it to any organization. The study centers on the present literature and suggests an enhanced approach for parsing resumes by leveraging Large Language Models (LLMs).**

***Index Terms*—Large Language Model(LLM), Natural Language Processing (NLP), Deep Learning (DL), Automated Resume Parsing, Applicant Tracking Systems (ATS), Software as a Service (SaaS), Human Resources (HR).**

I. Introduction

In the modern landscape of online applications, the volume of job applications inundating companies has surged dramatically [1], driven in part by the ease of online job searching and application submission. As a result, Human Resources (HR) departments are faced with the daunting task of sifting through thousands of resumes to identify candidates who meet the criteria outlined in job descriptions. [1] [2]. Resumes are often structured documents that come in a variety of formats and don't follow a set guideline or convention while presenting their content. As a result, Human Resources (HR) must take their time reviewing resumes in order to identify the most qualified candidates. This demonstrates that the process of analyzing resumes is challenging, leading to the creation of many websites and job platforms to alleviate the strain of managing a wide range of disorganized resumes. This led to websites which provide forms for candidates to input their resume details into a structured web-based format, creating applicant metadata. However, this approach can be disadvantageous as it requires applicants to engage in repetitive tasks, resulting in incomplete fields in these formats. Additionally, it puts the possibility of greater human error into the equation. As of 2019, 99% of the Fortune 500 companies, rely on an Applicant Tracking System (ATS) to help them hire new employees [3]. Most of them are available as Software as a Service (SaaS) products and some its functions are as follows:

1) Storing job candidate information like resumes, cover letters, references, and other data.
2) Tracks job candidates and their application status throughout the hiring pipeline.
3) Automates time-consuming tasks such as manually screening applicants, reading resumes, scheduling interviews, and sending out notifications.

Resume parsing within an ATS entails scanning of documents for predetermined keywords specified by the employer. Our suggested method promotes the creation of a cutting-edge resume parsing system based on Large Language Models (LLMs) in order to complement the current existing paradigm, they achieve this by leveraging Natural Language Processing (NLP), these helps the LLMs not only recognize but also extract crucial information from resumes. Its strength is in understanding sequential data in resumes and identifying subtle connections between different contextual pieces of information.

The structure of the paper is as follows: Section II describes about works related to traditional resume parsing using NLP, DL, word matching. In Section III this paper elaborates on the proposed methodology in great detail. The experimental results of the model are analyzed in Section IV. The further expansion of the project is then explained through Section V, and Section VI provides a conclusive overview of the paper.

II. Literature Survey

D. Pant et. al. [1] in their paper proposes an Automated Resume Parsing and Ranking System to lower time, cost and effort with the help of natural language processing and resume label character positioning. Candidate information is mainly retrieved by regular expression checks such as word and phrase matching. In their testing on five random software engineering positions resume, correct extraction rate of 33.59% was obtained.
They state that in the case of an unstructured resume, extracting information is a much more tedious task. The model trained using spaCy produced good results by specifying the position of the character for a structured resume, but it produced bad results for an unstructured resume, hence they shifted from spaCy to keyword and phrase-matching to generate a summary of the resume, but it could not extract all the details from personal details and experience. Another drawback faced was that the proposed methodology could not fetch the skills and details that were not in the dataset but were important for the job description.

The study conducted by S. Bhoomika et. al. [2] focuses on using a 2Q learning technique coupled with Natural Language Processing to decide whether to reject or accept a resume. About 2400 types of resumes in both .PDF and string format were included in the Kaggle Dataset. After being preprocessed by a resume analyzer, the data was fed to the 2Q learning agent to reduce the overestimation in the resume.
The proposed work is more effective than the traditional resume parsing approaches that use string matching techniques, the 2Q learning algorithm updates its Q tables quickly to form new policies to get rewards, hence increasing its accuracy. The inability to make resume revision suggestions or offer employers that matched the skills listed on the resume were the negatives that surfaced.

The study conducted by B. Nisha et. al. [4] suggests a streamlined approach for recruitment by proposing an Automated Resume Parsing and Ranking System (ARRS) driven by natural language processing techniques. NLP algorithms assist in talent acquisition by scraping information on skills, work experience, and qualifications which are ranked based on predefined criteria. The recruiter receives the created list so they can concentrate on the most eligible candidates. The suggested solution might have an intuitive user interface, be tailored to the specific requirements of an employer, and be easily connected with the current Applicant Tracking System (ATS).

The methodology focused on by S. Bharadway et. al. [5] in their paper is on making job role recommendations to candidates based on the skills stated in their resumes. The model which was constructed with the help of Long Short Term Memory (LSTM) and NLP, operates on the grouping principle, which combines several related classes into a single, sizable group. Achieving an accuracy of approximately 90% was made possible by utilizing a natural language processing toolbox. Although the model was claimed to be able to extract data from GitHub and LinkedIn profiles, it was unable to interpret resumes in various formats.

R. C. Tripathi et. al. [6] state that during the hiring process, the study's suggested resume screening framework removes a number of resume-related issues. Regular language processing was used to extract a great deal of data from unstructured resumes, allowing for the creation of an overview of the complete resume that contained only the information a recruiter needed to see. Additionally, content-based recommendations were used in the suggested process to place the candidates. To match the extracted features with the intended ones, a vector space model was applied. A CF recommendation technique was put forth to maximize reaction time when dealing with heterogeneous data. Additionally, they combined semantic, lexical, and punctuation analysis to ascertain the attitude regarding lodging highlights. The suggested methodology is unable to provide candidates with employment recommendations and resume revisions and incorporate factors such as closeness, inclusion, and security.

R. Ransing et. al. [7] in their paper suggest a two-level stacked architecture automated resume screening program based on machine learning algorithms including K Nearest Neighbours (KNN), Extreme Gradient Boosting (XGBoost), and Linear Support Vector Classification (SVC). The framework in the same though process as this paper, shows the recruiter's rivals and gives applicants feedback on how to modify their resumes to make them more system-ready. The suggested approach, which made use of stack classifiers and supervised learning, produced an accuracy of 83%. The system fails to achieve semantically translating terms from job descriptions and resumes which would be possible with models that are based on Deep Learning techniques like LSTM, Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN) and others. It may have also lead to increased accuracy.

III. Proposed Methodology

The methodology proposed for this system is one that considers several problems faced within ATS including but not limited to:

- Formatting of different resumes
- Unstructured resume parsing
- Varied requirements for differing job descriptions
- Intuitive understanding of diverse but related technologies

A. Dataset Description

There exist several open source datasets available and this section aims to understand and describe the issues faced within the dataset of choice.

Resume Dataset by Snehaan Bhawal on Kaggle
"A collection of Resume Examples taken from livecareer.com for categorizing a given resume into any of the labels defined in the dataset."
The dataset is well compiled and diverse in its selection of fields, presenting the data in both the format of .pdf files as well as a pre-processed .csv file.
The dataset is highly voluminous in nature and its distribution (Refer to Table. I) may pose an issue as it does not consider the thousands of different formats possible.This poses a critical problem while building a robust resume parsing model. **Dimensions of the dataset:** (2111, 4)

TABLE I
TABLE OF DISTRIBUTION OF PROFESSIONS

Profession	***Count***
Business-Development	102
Information-Technology	102
Chef	100
Engineering	100
Advocate	100
Finance	100
Accountant	100
Aviation	99
Sales	99
Fitness	99
Consultant	98
Banking	98
Healthcare	98
Construction	95
Public-Relations	94
HR	93
Designer	91
Arts	88
Teacher	87
Apparel	82
Digital-Media	82
Agriculture	54
Automobile	31
BPO	19

B. Environmental Setup

A client environment has to be set up using an API key and a function call that defines the roles played by the LLM and user. This is to replicate the environment of user and LLM interaction as in the LLM playground through the API. The Groq Cloud LLM API runs on the following arguments:

model="llama3-70b-8192"
messages=[
{"role": "system", "content": prompt},
{"role": "user", "content": resumeText},
]
temperature=0
maxtokens=1024
topp=1
stream=True
stop=None

An Application Programming Interface (API) endpoint is used to connect to the Groq Cloud LLM's llama3-70b-8192 model over a network to pass information to and from the client side. This creates a client - server communication model through an API Port using an authorized API Key.
Eg: gsk_1388e4e4228076b9289531776f6ddf07120695c0

C. Architecture and Working

This system uses an LLM based model that helps the resume parser be context aware as to current and upcoming skills, this is in stark contrast to word or phrase matching as it is limited to the sheet of skills provided to it and does not perform any actions based on previous information such as resume sectioning. The architecture of the open-source LLM llama3-70b-8192 is provided in Fig.1

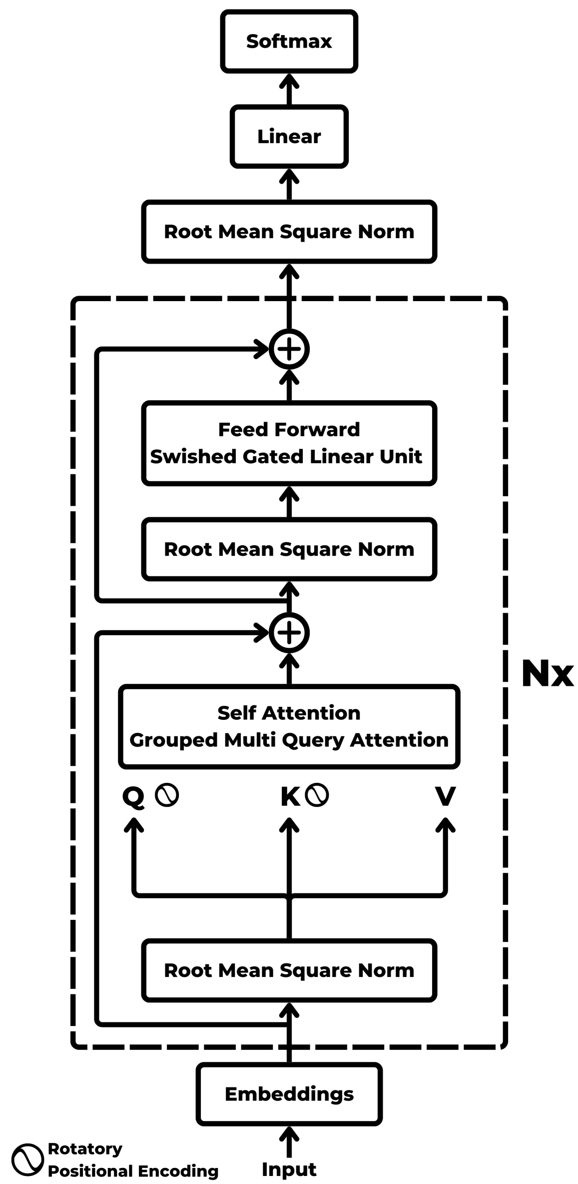


Fig. 1. Llama3-70b-8192 Architechture

The resume data parsing begins with the conversion of the .PDF format data into a type of data that our API can handle, mainly textual data. This also simultaneously helps lower the rate limit usage of the LLM API, which helps in further optimization of speed and response times.
The Groq Cloud LLM API was integrated to handle the resume text parsing tasks and the API call included prompts designed to extract specific sections listed below:

1) Full Name
2) Email Id
3) Github Id
4) Linkedin Id
5) Education
6) Experience
7) Skills (Technical, Soft, Other)
8) Projects (Title, Description, Technology)

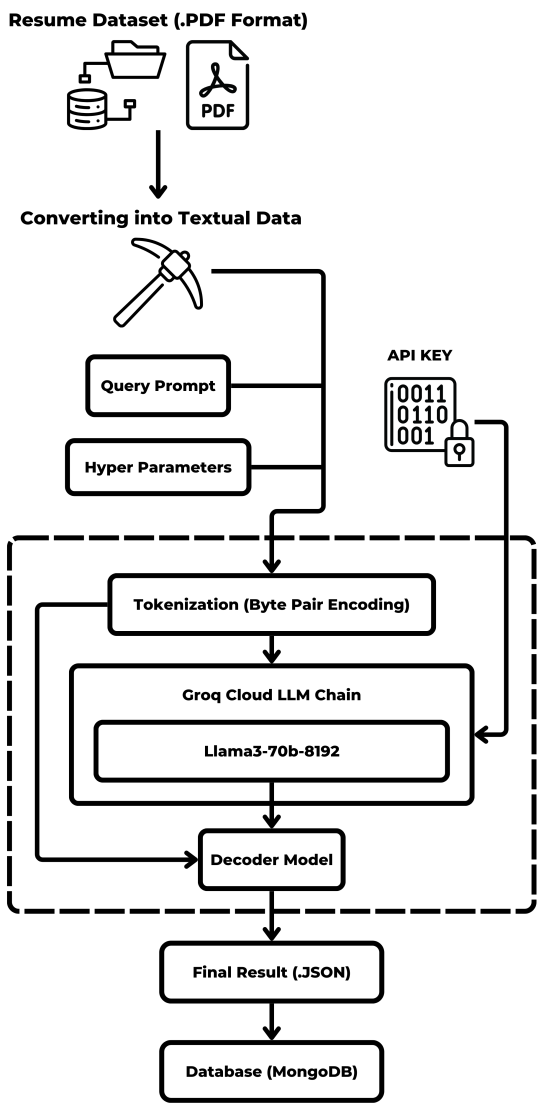
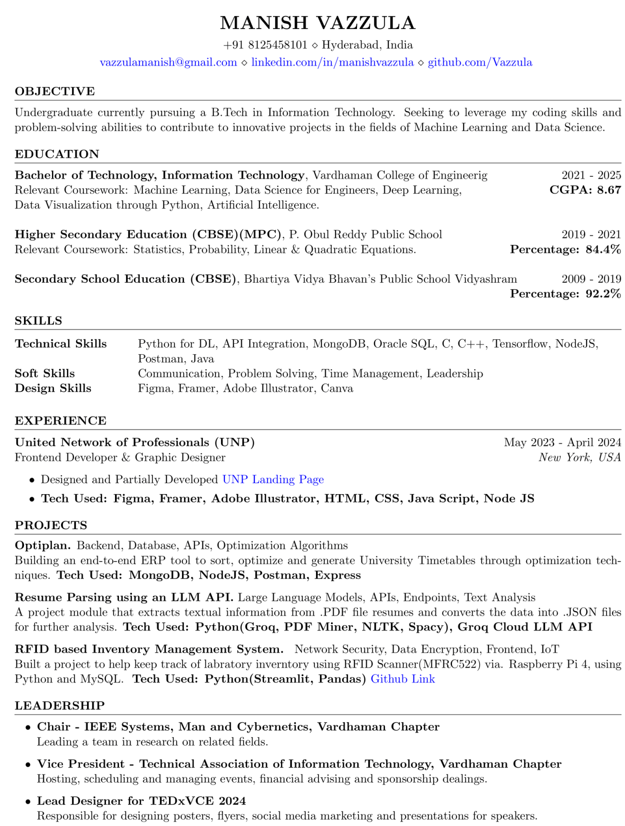


Fig. 2. Architecture for LLM based Resume Parsing System

The architecture described within Fig. 2. is used to help reduce ambiguity within the model and improve its performance.

IV. Results and Analysis

The objective of this analysis is to evaluate the context-aware resume parsing capabilities of the proposed model on a specified dataset and to test its generalization across various resume formats. The model demonstrates excellent performance with a rapid processing time of 2.0 seconds per resume. However, its efficiency is somewhat constrained by the limited diversity of resume formats within the dataset. To assess the model's ability to generalize, it was tested on a sample resume shown in Fig.3. The results are displayed in Fig.4, it shows that the model accurately extracted relevant information, which was then partitioned into distinct sections. This capability suggests that the model could be used to create structured datasets from unstructured resume data. While the model displayed high retrieval on the tested database, its reliance on standardized formatting indicates a potential risk of overfitting. This could lead to model's performance dropping with varied resume formats.



MANISH VAZZULA
+91 8125458101 ⋄ Hyderabad, India
vazzulamanish@gmail.com ⋄ linkedin.com/in/manishvazzula ⋄ github.com/Vazzula

OBJECTIVE
Undergraduate currently pursuing a B.Tech in Information Technology. Seeking to leverage my coding skills and problem-solving abilities to contribute to innovative projects in the fields of Machine Learning and Data Science.

EDUCATION
Bachelor of Technology, Information Technology, Vardhaman College of Engineerig 2021 - 2025
Relevant Coursework: Machine Learning, Data Science for Engineers, Deep Learning, Data Visualization through Python, Artificial Intelligence. **CGPA: 8.67**

Higher Secondary Education (CBSE)(MPC), P. Obul Reddy Public School 2019 - 2021
Relevant Coursework: Statistics, Probability, Linear & Quadratic Equations. **Percentage: 84.4%**

Secondary School Education (CBSE), Bhartiya Vidya Bhavan's Public School Vidyashram 2009 - 2019
Percentage: 92.2%

SKILLS
Technical Skills Python for DL, API Integration, MongoDB, Oracle SQL, C, C++, Tensorflow, NodeJS, Postman, Java
Soft Skills Communication, Problem Solving, Time Management, Leadership
Design Skills Figma, Framer, Adobe Illustrator, Canva

EXPERIENCE
United Network of Professionals (UNP) May 2023 - April 2024
Frontend Developer & Graphic Designer *New York, USA*
- Designed and Partially Developed UNP Landing Page
- **Tech Used: Figma, Framer, Adobe Illustrator, HTML, CSS, Java Script, Node JS**

PROJECTS
Optiplan. Backend, Database, APIs, Optimization Algorithms
Building an end-to-end ERP tool to sort, optimize and generate University Timetables through optimization techniques. **Tech Used: MongoDB, NodeJS, Postman, Express**

Resume Parsing using an LLM API. Large Language Models, APIs, Endpoints, Text Analysis
A project module that extracts textual information from .PDF file resumes and converts the data into .JSON files for further analysis. **Tech Used: Python(Groq, PDF Miner, NLTK, Spacy), Groq Cloud LLM API**

RFID based Inventory Management System. Network Security, Data Encryption, Frontend, IoT
Built a project to help keep track of labratory inverntory using RFID Scanner(MFRC522) via. Raspberry Pi 4, using Python and MySQL. **Tech Used: Python(Streamlit, Pandas)** Github Link

LEADERSHIP
- **Chair - IEEE Systems, Man and Cybernetics, Vardhaman Chapter**
  Leading a team in research on related fields.
- **Vice President - Technical Association of Information Technology, Vardhaman Chapter**
  Hosting, scheduling and managing events, financial advising and sponsorship dealings.
- **Lead Designer for TEDxVCE 2024**
  Responsible for designing posters, flyers, social media marketing and presentations for speakers.

Fig. 3. Example Resume


```
{'FULL_NAME': 'MANISH VAZZULA',
 'EMAIL_ID': 'vazzulamanish@gmail.com',
 'PHONE_NUMBER': '+91 8125458101',
 'ADDRESS': 'Hyderabad, India',
 'GITHUB_ID': 'github.com/Vazzula',
 'LINKEDIN_ID': 'linkedin.com/in/manishvazzula',
 'WEBSITE': None,
 'OBJECTIVE': None,
 'EDUCATION': [{'COURSE_NAME': 'Bachelor of Technology - Information Technology',
   'INSTITUTION_NAME': 'Vardhaman College of Engineering',
   'CGPA/PERCENTAGE': 8.67,
   'DATE': '2021 - 2025'},
  {'COURSE_NAME': 'Higher Secondary Education (CBSE)(MPC)',
   'INSTITUTION_NAME': 'P. Obul Reddy Public School',
   'CGPA/PERCENTAGE': 84.4,
   'DATE': '2019 - 2021'},
  {'COURSE_NAME': 'Secondary School Education (CBSE)',
   'INSTITUTION_NAME': 'Bhartiya Vidya Bhavan's Public School Vidyashram',
   'CGPA/PERCENTAGE': 92.2,
   'DATE': '2009 - 2019'}],
 'EXPERIENCE': [{'JOB_TITLE': 'Backend Developer & UI/UX Designer',
   'COMPANY_NAME': 'Enteam.ai',
   'DATE': 'May 2024 - July 2024',
   'RESPONSIBILITIES': 'Developing API's using Springboot to parse .PDF data for further analysis, Designed UI for marketing websites'},
  {'JOB_TITLE': 'Frontend Developer, UI & Graphic Designer',
...
   'DESCRIPTION': 'Built a project to help keep track of laboratory inventory using RFID Scanner(MFRC522) via. Raspberry Pi 4, using Python and MySQL.',
   'TECHNOLOGY': 'Python(Streamlit, Pandas)'}],
 'LEADERSHIP_ROLES': ['Chair - IEEE Systems, Man and Cybernetics, Vardhaman Chapter',
  'Vice President - Technical Association of Information Technology, Vardhaman Chapter',
  'Lead Designer - TEDxVCE 2024 and 2022']}
```


Fig. 4. Program Output Screenshot

V. Future Scope

The API endpoint is configured to return the information in a .JSON format which allows the construction of ”Compiled Database”, which underpins the deep learning evaluation phase through the use of Retrieval Augmentation. This database contains relational tables with thousands of job descriptions linked to their respective accepted and rejected resumes, allowing for the identification of trends and key phrases associated with successful applications. By mapping new job descriptions to those in the compiled dataset, the system can compare and contrast submitted resumes against historically accepted ones. The deep learning model utilizes this dataset to generate comprehensive reports with actionable recommendations for resume improvements, increasing the likelihood of passing ATS filters. Additionally, the system leverages LLMs to parse resumes to extract relevant information and provide intelligent suggestions, demonstrating the integration of advanced LLM capabilities to enhance resume screening processes.

VI. Conclusion

The methodology proposed in this research offers a novel approach to enhancing the effectiveness of resumes in the job application process, addressing critical issues such as formatting variability, differing job description requirements, and the intuitive understanding of technologies. By complementing current Applicant Tracking Systems (ATS) rather than competing with them, this system focuses on improving the quality of individual resumes to match specified job descriptions, thereby enhancing the chances of clearing resume screenings. Central to this approach is the LLM API on both resumes and job descriptions, ensuring that the resumes meet the minimum requirements specified by employers, such as GPA thresholds, mandatory skills, job location, experience, and language proficiency. The preprocessing avoids the step that involves meticulous cleaning of the resume text using an LLM API to remove unnecessary characters, thus preserving content integrity and extracting necessary information.

References

[1] D. Pant, D. Pokhrel and P. Poudyal, ”Automatic Software Engineering Position Resume Screening using Natural Language Processing, Word Matching, Character Positioning, and Regex,” 2022 5th International Conference on Advanced Systems and Emergent Technologies (IC_ASET), Hammamet, Tunisia, 2022, pp. 44-48, doi: 10.1109/IC_ASET53395.2022.9765916.

[2] S. Bhoomika, S. Likhitha, H. S. Chandana, S. A. Kavya and K. Bhargavi, ”2Q-Learning Scheme for Resume Screening,” 2023 4th International Conference for Emerging Technology (INCET), Belgaum, India, 2023, pp. 1-5, doi: 10.1109/INCET57972.2023.10169980.

[3] Sydney Myers, ”2023 Applicant Tracking System (ATS) Usage Report: Key Shifts and Strategies for Job Seekers.” (2023, October 2). https://www.jobscan.co/blog/fortune-500-use-applicant-tracking-systems

[4] B. Nisha, V. Manobharathi, B. Jeyarajanandhini and G. Sivakamasundari, ”HR Tech Analyst: Automated Resume Parsing and Ranking System through Natural Language Processing,” 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2023, pp. 1681-1686, doi: 10.1109/ICACRS58579.2023.10404426.

[5] S. Bharadwaj, R. Varun, P. S. Aditya, M. Nikhil and G. C. Babu, ”Resume Screening using NLP and LSTM,” 2022 International Conference on Inventive Computation Technologies (ICICT), Nepal, 2022, pp. 238-241, doi: 10.1109/ICICT54344.2022.9850889.

[6] R. C. Tripathi and Chandramma, ”An Approach of Intelligent Automated Resume Analysis & Recommendations,” *2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*, Mandya, India, 2022, pp. 1-5, doi: 10.1109/ICERECT56837.2022.10059603.

[7] R. Ransing, A. Mohan, N. B. Emberi and K. Mahavarkar, ”Screening and Ranking Resumes using Stacked Model,” *2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT)*, Mysuru, India, 2021, pp. 643-648, doi: 10.1109/ICEECCOT52851.2021.970797.

[8] A. Wahedna, A. Vakil, S. Shah, V. V. Kelkar and I. Shrivastava, ”Resume Screening–Testing For Data Stability,” *2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, Chennai, India, 2023, pp. 1-8, doi: 10.1109/ICSES60034.2023.10465529.