e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

SMART RESUME PARSER AND ANALYZER USING AI

Harsh Pratap*1, Prakhar Singh*2, Kunal Gautam*3, Rohit Sehwag*4,
Asst. Prof. Archana Mongia*5

*1,2,3,4Students, Department Of Information Technology, HMR Institute Of Technology And Management, GGSIPU, Delhi , India.

*5Assistant Professor, Department Of Information Technology, HMR Institute Of Technology & Management, GGSIPU, Delhi, India.

DOI : https://www.doi.org/10.56726/IRJMETS63933

ABSTRACT

Evaluating resumes poses a considerable obstacle in the hiring process due to the diverse formats and content found in applicants' documents. This study introduces an innovative system, Resume Parser Analyzer, aimed at enhancing the candidate assessment process by extracting crucial information from resumes in PDF and DOC formats. The system employs Natural Language Processing (NLP) and machine learning techniques to recognize and classify vital elements such as personal details, competencies, academic background, and professional history. Furthermore, it offers suggestions to applicants for resume improvement and screens applications according to criteria set by recruiters. This research outlines the creation, structure, and potential uses of the Resume Parser Analyzer, offering a valuable resource for improving recruitment effectiveness.

Keywords: Resume Analyzer, Python, Streamlit, Apache, NLP, Job Recommendation, Resume Parser.

I. INTRODUCTION

In the current highly competitive employment landscape, hiring managers are often inundated with numerous resumes for a single position, each featuring distinct formatting and content. Traditional manual resume screening approaches are time-consuming, labor-intensive, and susceptible to human error, potentially impeding the efficient selection of suitable candidates. However, with the progress in Natural Language Processing (NLP) and machine learning technologies, automated systems have emerged as practical solutions for resume parsing, information extraction, and matching applicants to job requirements.

The Resume Parser Analyzer offers an automated solution that addresses these challenges by not only parsing resumes but also providing individualized feedback to applicants. This system employs a blend of NLP techniques and machine learning algorithms to recognize crucial sections within resumes, including the applicant's name, contact details, skills, educational history, and professional experience. By combining rule-based methods with machine learning, the tool ensures high precision in information extraction, even when dealing with diverse resume layouts. Furthermore, it filters resumes according to specific recruiter criteria, facilitating more efficient candidate selection. This paper outlines the Resume Parser Analyzer's structure, methodology, and its influence on recruitment processes.

II. LITERATURE REVIEW

The field of automated resume parsing has been thoroughly explored due to its significance in streamlining recruitment processes. Various techniques and technologies are commonly utilized, including NLP, machine learning models, and rule-based systems, each with its own strengths and weaknesses.

NLP techniques have played a crucial role in the evolution of resume parsers. Tools such as named entity recognition (NER) and regular expressions are often employed to extract key elements from resumes, including names, contact information, and skills. Some systems implement rule-based NLP methods to analyze unstructured text and identify specific sections like education and work history. However, these approaches often encounter difficulties when processing resumes that deviate from conventional formats.

Machine learning has been pivotal in enhancing resume parsing, particularly through the application of supervised learning techniques for classifying and extracting pertinent information. Contemporary methods utilize algorithms such as support vector machines (SVM) and clustering techniques to categorize resume content according to predetermined criteria. For instance, models trained on extensive resume datasets can

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

accurately predict sections, facilitating effective extraction of candidate details.Despite these improvements, challenges persist, especially when parsing non-standard or intricate resumes, necessitating flexible and context-sensitive parsing models.

A recent advancement in resume parsing involves layout-aware methodologies, which consider the spatial organization of text elements within resumes. This approach incorporates tools like Optical Character Recognition (OCR) for PDF documents, as well as layout parsers, to handle complex document structures, such as tables and multi-column layouts. Layout-aware techniques have enhanced parsing accuracy by maintaining section boundaries and preserving the visual structure of the document during the extraction process. While powerful, these methods can be computationally demanding and require additional preprocessing steps.

Although current resume parsers have shown success, limitations remain. Many systems lack the ability to adapt to diverse resume layouts and often rely on restricted datasets for training, resulting in reduced performance on unconventional formats. Furthermore, few systems offer feedback for resume enhancement, a feature increasingly valued by candidates. The Resume Parser Analyzer sets itself apart by incorporating a feedback mechanism, along with a filtering feature that enables recruiters to establish custom selection criteria, addressing the needs of both recruiters and candidates.

Related Work

This section examines previous research in the field. The author of [1] introduces an innovative NLP-based approach for parsing resumes to identify key entity features. The method employs a dictionary to efficiently tally feature occurrences and determine the most frequent string's canonical structure. Entity features are then compared to the dictionary to update a mapping table and pinpoint entities within the resume. This named entity recognition (NER) model surpasses existing techniques in precision, recall, and F1 score. It can be utilized to rank resumes based on similarity and create shortlists for further evaluation. In [2], the author explores the necessity of automated systems for processing resumes and converting them into structured formats for job matching. The diverse formats of resumes pose challenges for parsing. The proposed system utilizes Natural Language Processing (NLP) for resume parsing, Optical Character Recognition (OCR) for plain text conversion, and a ranking algorithm for matching applicants to job requirements. Organizations can specify skill set criteria, and applicants' resumes are ranked according to extracted entities and required keywords. The findings are displayed using pie charts and bar graphs. The authors of [3] tackle the issue of recommending suitable jobs to individuals seeking new employment. They approach this as a supervised machine learning task, utilizing past job transitions and data related to employees and organizations to forecast a candidate's next career move. The study involves training a machine learning model using an extensive dataset of job transitions extracted from publicly available employee profiles across the World Wide Web. Experimental outcomes indicate that job transitions can be predicted accurately, outperforming a baseline approach that consistently predicts the most frequent institution in the dataset. In [4], the authors address the difficulties faced by companies and recruitment agencies due to the high volume of online resume submissions. They propose a solution that uses natural language processing (NLP) to parse and extract crucial information from resumes, transforming them into a structured format. Users can create accounts, upload resumes, and review the parsed data, which is stored in a NoSQL database. The aim is to enhance the recruitment process, making it more efficient and fair by matching job applicants with appropriate positions based on their parsed resumes. The study in [5] examines the difficulties encountered during recruitment due to the high volume of applicants for a single position. The manual review of individual resumes is time-intensive and burdensome for recruiters, particularly because of the varied sections and formats in each resume. To address this issue, the paper proposes an NLP-based system with two primary components: Job Seekers and Recruiters. Job seekers upload their resumes, which the system then parses for field extraction. The system analyzes the resumes, assigns rankings, and provides suggestions for improvements, such as needed skills or course recommendations. The rankings and extracted data are stored in a database for recruiter access. Research in [6] discusses the implementation of an Automated Job Recommendation System Based on Candidate Profiles. This study focuses on developing a recommendation system for online job hunting, aiming to reduce the time-consuming process job seekers face when searching for suitable positions on the internet. The paper compares user-based and item-based collaborative filtering algorithms to determine the most effective approach. It also incorporates various factors, including students'

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

resumes and job listing details, into the recommendation algorithm. The work presented in [7] employs a Bottom-Up Approach to Job Recommendation System. The authors describe their efforts in designing a job recommendation system for the career-based social networking website XING. They utilized a bottom-up approach, beginning with a thorough understanding and analysis of the available data before gradually constructing the recommendation system. The study explores traditional recommendation system methods such as collaborative filtering and discusses their performance. Research in [8] utilized the K-Means Clustering Method in a Job Recommendation System. The study addresses the challenge faced by recent graduates in finding job vacancies that match their criteria due to limited work experience. The researchers developed a web-based application that serves as an intermediary between companies and applicants, using K-Means Clustering to recommend specific job vacancies for undergraduates. The system calculates the match between the applicant's main salary, location, and other skills with a company's requirements. The researchers conducted black-box testing and questionnaire testing to evaluate user satisfaction and system quality. Results showed that the system performed well, with a user satisfaction percentage of 87.6%. The study in [9] employed NLP-Based Resume Parser Analysis to address the challenges faced by human resources departments due to the increasing volume of diverse resume submissions. It introduces a solution in the form of an NLP-based resume parser to streamline the hiring process. The proposed Employee Recommendation System emphasizes the need to extract key information from unstructured resumes for effective ranking and selection. This paper utilizes a combination of machine learning and natural language processing for extracting relevant features from resumes.

III. METHODOLOGY

The Resume Parser Analyzer employs NLP and machine learning to systematically evaluate resumes and extract pertinent data. The system consists of three main components: Data Extraction, Information Parsing and Analysis, and Filtering and Feedback Mechanisms. Each component is tailored to tackle common issues in resume parsing, including inconsistent layouts, various file types, and the need for accurate information extraction.

Data Extraction: The system processes resumes in PDF and DOC formats, transforming them into structured text for further analysis. PDFMiner is utilized for PDFs to extract text while preserving layout, while python-docx handles Microsoft Word documents. This initial step ensures uniform processing of different resume formats, streamlining the extraction process.

Information Parsing and Analysis: After text extraction, the spaCy library, a robust NLP toolkit, is used for entity recognition and information parsing. Key elements include: Named Entity Recognition (NER): spaCy's pre-trained NER model identifies and classifies entities such as names, locations, dates, and skills, facilitating precise extraction of candidate information. Pattern Matching and Regular Expressions: Custom algorithms and tailored regular expressions are employed to detect resume sections and identify specific information like email addresses, phone numbers, and skills. A keyword-based approach, supported by a technical term dictionary, is used for skill extraction. Content-Based Filtering: The system analyzes extracted skills and experience, comparing them to job-specific requirements set by recruiters, and provides feedback on potential skill gaps. In figure (1), The complete processing of the parsing is depicted.

Filtering and Feedback Mechanisms: In Figure (2), This module enhances user experience and aids recruiters in candidate shortlisting based on predetermined criteria. Filtering Mechanism: Recruiters can set filters for required skills, experience, or education. The system uses TF-IDF for term analysis and cosine similarity to assess alignment between resumes and job descriptions, ranking candidates accordingly. This process ensures only relevant resumes are shortlisted, optimizing screening efficiency. Feedback Module: A distinctive feature of the Resume Parser Analyzer is its ability to offer constructive feedback. It evaluates resume completeness by checking for essential sections like Objectives, Skills, and Projects. The system suggests improvements for missing or insufficient sections, helping candidates enhance their resumes before applying and aligning with industry standards.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

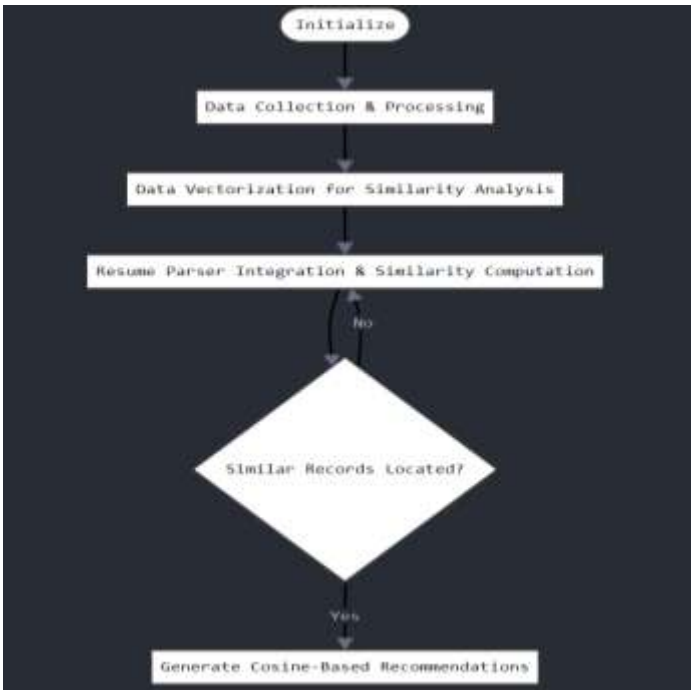

Figure 1: Resume document processing

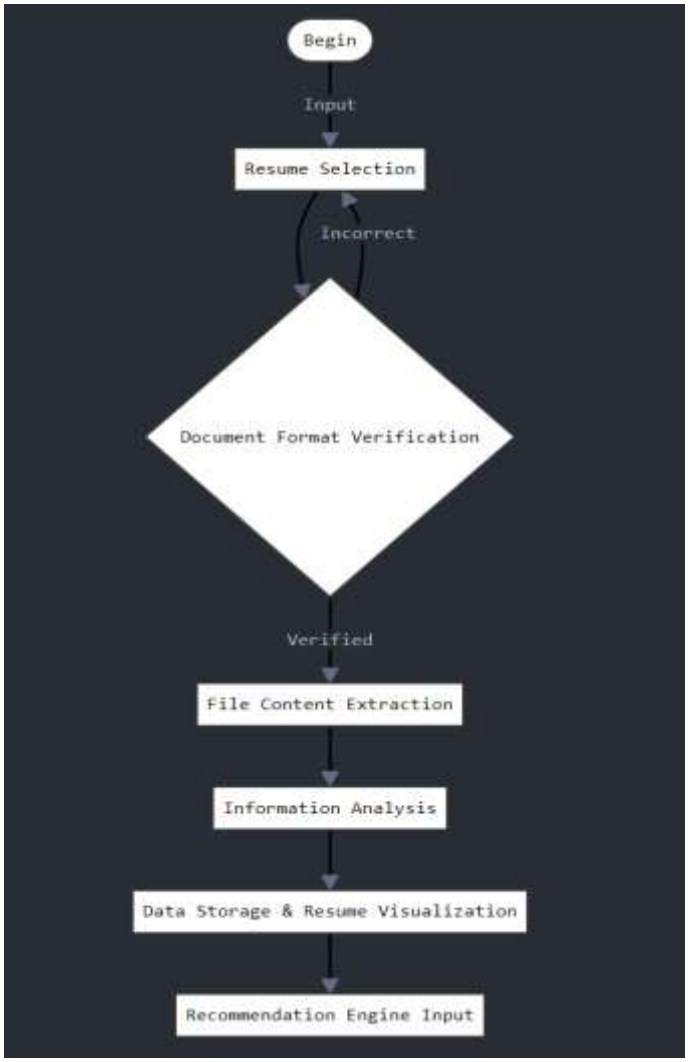

Figure 2: Recommendation system workflow

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

In Figure (3), The system implemented features a sophisticated database structure paired with a smart recommendation system. At its core is a well-designed SQL schema called "SRA," which includes a comprehensive "user_data" table. This table is carefully crafted to store detailed user information, such as personal data, analytical results, and suggested courses. The schema utilizes VARCHAR fields of varying lengths, like resume_score(8) for evaluation metrics and larger fields such as Recommended_courses(600) to store extensive recommendation data. To ensure data integrity, the implementation uses NOT NULL constraints and an auto-incrementing primary key for unique identification of records.

```
# Create the DB
db_sql = """CREATE DATABASE IF NOT EXISTS SRA;"""
cursor.execute(db_sql)

# Create table
DB_table_name = 'user_data'
table_sql = "CREATE TABLE IF NOT EXISTS " + DB_table_name + """
                (ID INT NOT NULL AUTO_INCREMENT,
                 Name varchar(100) NOT NULL,
                 Email_ID VARCHAR(50) NOT NULL,
                 resume_score VARCHAR(8) NOT NULL,
                 Timestamp VARCHAR(50) NOT NULL,
                 Page_no VARCHAR(5) NOT NULL,
                 Predicted_Field VARCHAR(25) NOT NULL,
                 User_level VARCHAR(30) NOT NULL,
                 Actual_skills VARCHAR(300) NOT NULL,
                 Recommended_skills VARCHAR(300) NOT NULL,
                 Recommended_courses VARCHAR(600) NOT NULL,
                 PRIMARY KEY (ID));
                """
cursor.execute(table_sql)
if choice == 'Normal User':
    pdf_file = st.file_uploader("Choose your Resume", type=["pdf"])
```

Figure 3: Database SQL snippet

The recommendation system employs an advanced method for analyzing career paths and suggesting skills. It organizes technology stacks into specific domains, creating specialized keyword collections for various career trajectories. In the realm of data science, it includes cutting-edge frameworks and technologies such as TensorFlow, Keras, PyTorch, and highlights machine learning capabilities. The web development stack thoroughly covers both frontend and backend technologies, including React, Django, Node.js, Laravel, and numerous other frameworks, enabling a comprehensive analysis of web development skills. The system's recommendation capabilities extend to mobile Application development, maintaining distinct keyword repositories for Android and iOS development. The Android stack incorporates modern development frameworks like Flutter and Kotlin, while the iOS stack includes Swift, Cocoa, and crucial development tools. Furthermore, the UI/UX domain encompasses contemporary design tools and methodologies, such as Figma, Adobe XD, and prototyping tools, ensuring thorough coverage of design-focused career paths. In Figure (4), The recommendation logic adopts a methodical approach, evaluating the user's current skills against these predefined technology stacks. It produces customized recommendations for skill improvement and proposes relevant courses based on the identified career path. The system utilizes a streamlit interface to display these recommendations in an interactive and user-friendly format, incorporating markdown formatting for improved readability. The implementation includes success messages and formatted output to offer clear guidance to users regarding potential career directions and skill enhancement opportunities. This integrated strategy, merging structured data storage with intelligent recommendation algorithms, establishes a comprehensive career guidance system capable of effectively analyzing resumes and offering personalized development paths across multiple technology domains. The system's modular architecture allows for simple expansion and updates to accommodate emerging technologies and evolving industry requirements.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

```
##  recommendation
ds_keyword = ['tensorflow','keras','pytorch','machine learning','deep Learning','flask','streamlit']
web_keyword = ['react', 'django', 'node jS', 'react js', 'php', 'laravel', 'magento', 'wordpress',
               'javascript', 'angular js', 'c#', 'flask']
android_keyword = ['android','android development','flutter','kotlin','xml','kivy']
ios_keyword = ['ios','ios development','swift','cocoa','cocoa touch','xcode']
uiux_keyword = ['ux','adobe xd','figma','zeplin','balsamiq','ui','prototyping','wireframes','storyframes','ado

recommended_skills = []
reco_field = ''
rec_course = ''
## Courses recommendation
for i in resume_data['skills']:
    ## Data science recommendation
    if i.lower() in ds_keyword:
        print(i.lower())
        reco_field = 'Data Science'
        st.success("** Our analysis says you are looking for Data Science Jobs.**")
        recommended_skills = ['Data Visualization','Predictive Analysis','Statistical Modeling','Data Mining',
        recommended_keywords = st_tags(label='### Recommended skills for you.',
        text='Recommended skills generated from System',value=recommended_skills,key = '2')
        st.markdown('''<h4 style='text-align: left; color: #1ed760;'>Adding this skills to resume will boost
        rec_course = course_recommender(ds_course)
        break
```

Figure 4: Recommendation system snippet

IV. RESULTS

The resume parser's performance is assessed using various metrics, including precision, speed of processing, and user contentment with the feedback system. This evaluation aims to gauge the parser's capability in extracting and sifting pertinent data from resumes, as well as its ability to fulfill the requirements of both recruiters and applicants.

1. **Precision of Data Extraction-** The accuracy of information retrieval is a crucial measure, reflecting the parser's proficiency in correctly identifying and classifying resume content. Evaluations were performed using a diverse set of resume formats to measure the parser's effectiveness in extracting fields such as personal details, contact information, competencies, educational background, and professional history. Named Entity Recognition (NER) Efficiency : The NER module demonstrated high accuracy in detecting names, contact information, and educational institutions, achieving an average precision rate of about 88%. This impressive score indicates the parser's reliability in identifying structured data, such as contact details, which typically follow consistent patterns across various resumes. Data Extraction by Section : For specific sections like "Skills," "Experience," and "Education," the parser exhibited an accuracy of roughly 82% when tested on a varied dataset. While satisfactory, this level of precision suggests room for enhancement, particularly for resumes with unconventional layouts. The layout-aware feature, which utilizes positional data and keywords, proved advantageous but occasionally struggled with sections that combined multiple headings (e.g., "Skills & Experience").
2. **Speed and Processing Efficiency-** The parser's efficiency was evaluated by examining its processing duration per resume. On average, the parser required 2 to 5 seconds to process each resume, varying based on file format and length. PDF files generally demanded slightly more processing time than DOC files due to additional steps needed for layout preservation and text extraction. This performance meets the criteria for scalable applications that require rapid processing of large resume volumes. Comparison with Manual Review : Traditional manual resume screening typically consumes several minutes per resume, especially for thorough evaluations. In contrast, the automated parser substantially reduces this time, enabling recruiters to manage large applicant pools more efficiently. Scalability : Tests revealed that the parser performs well

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

with larger batches, as its modular architecture allows for parallel resume processing. This scalability ensures the parser can handle peak recruitment periods without compromising performance.

3. **User Experience and Feedback-** This parser stands out due to its capacity to offer feedback to applicants, which was assessed through simulated user interactions. The quality of feedback was evaluated based on its relevance, clarity, and usefulness, resulting in an average user satisfaction score of 85%. Feedback Relevance: Applicants generally found the feedback pertinent, as it emphasized missing competencies, proposed structural improvements, and identified potential formatting issues Applicant Response: Users responded positively to this feature, indicating that it aided their understanding of what recruiters seek in resumes. This feedback mechanism not only enhances the applicant experience but also indirectly assists recruiters by minimizing the need for repetitive guidance.
4. **Constraints** Although the parser performed well overall, certain limitations were identified, particularly in handling unconventional or visually intricate resume formats. Moreover, challenges persist in extracting data from highly customized layouts and in filtering resumes based on nuanced recruiter preferences. Handling Non-Standard Layouts: Resumes with unique designs, such as those with extensive graphic elements or unconventional section headings, posed challenges. The current rule-based layout-aware approach struggles to recognize non-standard formats, as it relies on pattern matching and keyword identification. Reliance on Training Data: The accuracy of the parser's Named Entity Recognition (NER) and filtering components could be further enhanced with a larger, more diverse training dataset. Currently, the system performs well on common resume structures but may require additional training to handle varied formats and industry-specific terminologies. Filtering Constraints: The rule-based filtering system, while effective for straightforward criteria, may not fully capture complex recruiter requirements, such as nuanced skill proficiency levels or specific project experience. This could potentially be addressed by incorporating a more sophisticated machine learning-based filtering model in future iterations.
5. **Potential Enhancements**- Future work could improve the resume parser's accuracy and adaptability through the following upgrades: Advanced NER with Deep Learning: Utilizing transformer-based models, such as BERT or GPT, could enhance the accuracy of entity recognition across diverse resume formats. These models excel at understanding context, making them suitable for recognizing nuanced information in resumes. Improved Layout Recognition with OCR Integration: Incorporating Optical Character Recognition (OCR) capabilities would enable the parser to better handle resumes with intricate designs or graphical elements. OCR can help identify section boundaries and text flow in documents with complex layouts, improving overall extraction accuracy. Semantic Search for Filtering: This approach would assist recruiters in finding candidates with specific skill sets, even if the resume wording differs from the job description.

V. CONCLUSION

The Resume Parser Analyzer presents a comprehensive solution for automating the process of resume parsing and screening, effectively tackling prevalent issues encountered by both applicants and hiring managers. By employing natural language processing, machine learning, and content-based filtering methodologies, this tool efficiently extracts key information from various resume formats, delivers constructive input, and screens applications based on recruiter-specified criteria. The incorporation of feedback mechanisms sets this system apart from conventional resume parsers, providing substantial benefits to candidates seeking to improve their resumes. Future enhancements to the Resume Parser Analyzer could involve implementing sophisticated deep learning models to achieve even greater precision in data extraction, particularly for non-standard or intricate resume structures. Moreover, expanding the feedback component to offer more tailored recommendations for specific industries could further boost the tool's applicability and user-friendliness.

VI. REFERENCE

[1] Dr. Naveenkumar Jayakumar, Akshay Ramchandra Patil, Dr. Shashank Joshi, Dr. Prasanth Narayanan, Dr. Saurab Saoji. “An Approach to Parse Resumes and Recommend Jobs Using Modified K-mers and Firefly Algorithm (FFA) in Resume Parsing,” SJIS, vol. 35, 2023.

[2] Shubham Bhor, Vivek Gupta, Vishak Nair, Harish Shinde, Prof. Manasi S. Kulkarni, “Resume Parser Using Natural Language Processing Techniques,” International Journal of Research in Engineering and Science (IJRES), vol. 9, pp. 01-06, 2021.

[3] I. Paparrizos, B. B. Cambazoglu, A. Gionis, "Machine-Learned Job Recommendation," in Proceedings of the Fifth ACM Conference on Recommender Systems, pp. 325-328, October 2011.

[4] Abdul Wahab S., Dr. M. N. Nachappa, "Resume Parser with Natural Language Processing," International Research Journal of Engineering and Technology (IRJET), vol. 9, March 2022.

[5] Nimish Patil, Shubham Yadav, Vikas Biradar, "Resume Parser and Analyzer Using NLP," International Research Journal of Modernization in Engineering Technology and Science, 2023.

[6] Vinay Desai, Dheeraj Bahl, Shreekumar Vibhandik, Isra Fatma, "Implementation of an Automated Job Recommendation System Based on Candidate Profiles," International Research Journal of Engineering and Technology (IRJET), vol. 4, May 2017.

[7] Sonu K. Mishra, Manoj Reddy, "A Bottom-Up Approach to Job Recommendation System," International ACM, September 2016.

[8] Betty Dewi Puspasari, Lany Lukita Damayanti, Andy Pramono, Aang Kisnu Darmawan, "Implementation of K-Means Clustering Method in Job Recommendation Systems," in 7th International Conference on Electrical, Electronics, and Information Engineering (ICEEIE), October 2021.

[9] Mhaske Harshada, A. N. Kshirsagar, Nevase Sonali, Pimparkar Ankita, "NLP-Based Resume Parser Analysis,".