

SmartHire -Hiring Market Place

Aiswarya B¹, Lakshmi R Nair², Anusree Ganga Nair³, Rohith S⁴, Sanjeev Shaji⁵, Sangeeth Subhash⁶

^{1, 3, 4, 5, 6}Computer Science and Engineering College of Engineering Cherthala

²Assistant Professor, Computer Science and Engineering College of Engineering Cherthala
rnairlekshmi90@gmail.com

Abstract— Resume parsing is a critical task for modern human resources departments and recruitment firms. Our application is designed to automatically extract relevant information from resumes and store it in a structured format. By leveraging natural language processing (NLP) our application is able to accurately extract information such as work experience, education, and skills. Our application can be easily integrated with existing applicant tracking systems (ATS) and provides a streamlined solution for managing large volumes of resumes. With our application, HR departments and recruitment firms can save time and resources by automating the resume parsing process, while improving the accuracy and consistency of candidate evaluation.

Keywords— Component, formatting, style, styling, insert.

I. INTRODUCTION

In today's competitive job market, effective recruitment strategies are crucial to identifying top talent and building successful teams. Our resume parsing application provides a comprehensive solution for automating the resume screening process and identifying the most qualified candidates. Using advanced NLP techniques, our application extracts key information from resumes such as work experience, education, and skills, and stores it in a structured format for easy analysis. In addition, our application features a Duplicate Profile Identification system, which helps to identify and remove duplicate resumes from the screening process. We also offer a Social Media Behaviour Analysis feature, which uses data from candidate's social media profiles to provide insight into their personality, communication style, and overall suitability for a given role. By leveraging these powerful features, our application provides a streamlined, efficient solution for recruitment firms and HR departments, allowing them to quickly and easily identify top talent and build winning teams.

II. PROBLEM STATEMENT AND OBJECTIVES

Problem Statement: Hiring managers face the challenge of efficiently evaluating and screening a large volume of resumes received for a job posting. The process of manually reviewing and assessing each resume is time-consuming and can lead to inconsistencies and oversight in candidate selection.

Objective: The objective is to develop a solution that assists hiring managers and other stakeholders involved in the hiring process to fast track the initial screening of resumes. The solution should leverage natural language processing (NLP) and automation techniques to extract relevant data from resumes, identify duplicate profiles, build social profiles from candidate information on platforms like LinkedIn and GitHub, match skills with job requirements, rank and score resumes, enable keyword search, allow customizable screening criteria, integrate with Applicant Tracking Systems (ATS), provide reporting and analytics capabilities, ensure user-friendly interface, prioritize security and data privacy, and continuously improve the screening process through machine learning and

artificial intelligence techniques. By achieving these objectives, the solution aims to streamline the resume screening process, enhance the efficiency and accuracy of candidate evaluation, and ultimately facilitate the identification of the most suitable candidates for further consideration in the hiring process.

III. LITERATURE SURVEY

A. *Resume Parser with Natural Language Processing by Satyaki Sanyal, Souvik Hazra, Soumyashree Adhikary, Neelanjan Ghosh*

The authors begin by defining resume parsing as the process of extracting relevant information from a resume in a structured format. They then discuss the importance of resume parsing in the recruitment process and the challenges associated with it, such as the variability in resume formats, language, and the use of synonyms and abbreviations.

The paper goes on to explore the different techniques used in resume parsing, such as rule-based, statistical, and machine learning approaches. The authors provide a detailed description of each technique, including their advantages and limitations.

The authors then focus on natural language processing (NLP) techniques used in resume parsing. They discuss the different NLP tasks such as part-of-speech tagging, named entity recognition, and dependency parsing, and how they can be applied to extract information from resumes. The authors also discuss the use of word embeddings and deep learning models for resume parsing.

The paper concludes by summarizing the key findings and outlining some of the future research directions in the field of resume parsing. The authors highlight the need for more research in developing multilingual resume parsers, handling unstructured data, and addressing privacy concerns.

B. *Resume Classification using various Machine Learning Algorithms - Riya Pal, Shahrukh Shaikh, Swaraj Satpute and Sumedha Bhagwa*

The paper "Resume Classification using various Machine Learning Algorithms" by Riya Pal, Shahrukh Shaikh, Swaraj Satpute, and Sumedha Bhagwa presents a literature survey

of the different machine learning algorithms used for resume classification.

The authors begin by discussing the importance of resume classification in the recruitment process. They highlight the benefits of automated resume classification, including increased efficiency, reduced bias, and improved candidate matching.

The paper then provides an overview of the different machine learning algorithms used in resume classification, including decision trees, support vector machines (SVMs), naive Bayes, logistic regression, and neural networks. For each algorithm, the authors describe the underlying principles, advantages, and limitations.

The authors then discuss the different features used in resume classification, such as education, work experience, skills, and personal information. They highlight the importance of feature selection in improving classification accuracy and reducing computation time.

The paper also discusses the challenges associated with resume classification, such as the variability in resume formats and language, and the need for large amounts of training data. The authors then present a comparative analysis of the different machine learning algorithms applied to resume classification, using datasets from various sources. They compare the accuracy, precision, recall, and F1 score of each algorithm and highlight their strengths and weaknesses.

The paper concludes by summarizing the key findings and outlining some of the future research directions in the field of resume classification. The authors suggest the need for more research in developing multilingual resume classifiers, handling unstructured data, and addressing privacy concerns. Overall, this literature survey provides a comprehensive overview of the different machine learning algorithms used for resume classification and highlights the potential for further research in this area.

C. Resume Parsing and Categorizing by AbeerZaroor

The paper "Resume Parsing and Categorizing" by Abeer Zaroor presents a literature survey of various techniques used for resume parsing and categorization.

The author begins by defining resume parsing as the process of extracting relevant information from a resume and converting it into a structured format. The paper then discusses the importance of resume parsing in the recruitment process and the challenges associated with it, such as the variability in resume formats and language.

The paper provides an overview of the different techniques used for resume parsing, including rule-based, statistical, and machine learning approaches. The author highlights the advantages and limitations of each approach and provides examples of tools and software that utilize these techniques for resume parsing.

The paper also discusses the importance of resume categorization, which involves classifying resumes into different categories based on factors such as skills, experience, and education. The author presents various techniques used for resume categorization, including clustering, topic modeling, and rule-based approaches.

The author then highlights the use of natural language processing (NLP) techniques in resume parsing and categorization. The paper provides an overview of different NLP tasks, such as named entity recognition, part-of-speech tagging, and dependency parsing, and how they can be used to extract relevant information from resumes.

The paper concludes by summarizing the key findings and outlining some of the future research directions in the field of resume parsing and categorization. The author suggests the need for more research in developing multilingual resume parsers, improving the accuracy of resume parsing and categorization, and addressing privacy concerns.

Overall, this literature survey provides a comprehensive overview of the different techniques used for resume parsing and categorization, highlighting the potential for further research in this area.

D. Resume Information Extraction with A Novel Text Block Segmentation Algorithm

The paper "Resume Information Extraction with A Novel Text Block Segmentation Algorithm" presents a literature survey of various techniques used for resume information extraction, with a focus on a novel text block segmentation algorithm.

The paper begins by discussing the importance of resume information extraction in the recruitment process and the challenges associated with it, such as the variability in resume formats and language. The authors highlight the need for an accurate and efficient resume information extraction system.

The paper provides an overview of the different techniques used for resume information extraction, including rule-based, statistical, and machine learning approaches. The authors discuss the advantages and limitations of each approach and provide examples of tools and software that utilize these techniques for resume information extraction.

The authors then introduce a novel text block segmentation algorithm, which involves the automatic segmentation of a resume into different text blocks based on their semantic meaning. The algorithm utilizes a combination of keyword-based and pattern-based techniques to identify different text blocks, such as personal information, education, work experience, and skills.

The paper provides a detailed description of the algorithm and its implementation, including the use of natural language processing (NLP) techniques such as part-of-speech tagging and named entity recognition. The authors also present experimental results that demonstrate the effectiveness of the algorithm in accurately extracting relevant information from resumes.

The paper concludes by summarizing the key findings and outlining some of the future research directions in the field of resume information extraction. The authors suggest the need for more research in developing multilingual resume information extraction systems, improving the accuracy of resume information extraction, and addressing privacy concerns.

Overall, this literature survey provides a comprehensive overview of the different techniques used for resume information extraction, with a focus on a novel text block

segmentation algorithm. The authors demonstrate the effectiveness of the algorithm in accurately extracting relevant information from resumes, highlighting its potential for further research and development in this area.

E. Automated Resume Evaluation System using NLP, Rohini Nimbekar; Yoqesh Patil; Rahul Prabhu; Shainila Mulla

The paper "Automated Resume Evaluation System using NLP" presents a literature survey of various techniques used for automated resume evaluation, with a focus on natural language processing (NLP) techniques.

The authors begin by discussing the importance of automated resume evaluation in the recruitment process and the challenges associated with it, such as the large volume of resumes received by organizations. The authors highlight the need for an accurate and efficient automated resume evaluation system.

The paper provides an overview of the different techniques used for automated resume evaluation, including rule-based, statistical, and machine learning approaches. The authors discuss the advantages and limitations of each approach and provide examples of tools and software that utilize these techniques for automated resume evaluation.

The authors then introduce the use of NLP techniques in automated resume evaluation, including techniques such as named entity recognition, part-of-speech tagging, and sentiment analysis. The authors provide a detailed description of how these techniques can be used to extract relevant information from resumes, such as skills, experience, and education.

The paper also discusses the use of machine learning algorithms in automated resume evaluation, such as decision trees and support vector machines. The authors provide a detailed description of how these algorithms can be used to classify resumes into different categories based on their content.

The authors then present an example of an automated resume evaluation system developed using NLP techniques. The system involves the automatic extraction of relevant information from resumes, such as skills and experience, and the classification of resumes into different categories based on their content.

The paper concludes by summarizing the key findings and outlining some of the future research directions in the field of automated resume evaluation using NLP techniques. The authors suggest the need for more research in developing multilingual automated resume evaluation systems, improving the accuracy of automated resume evaluation, and addressing privacy concerns.

Overall, this literature survey provides a comprehensive overview of the different techniques used for automated resume evaluation, with a focus on NLP techniques. The authors demonstrate the potential of these techniques in developing accurate and efficient automated resume evaluation systems, highlighting the need for further research and development in this area.

F. Resume Parser Using Natural Language Processing Techniques, Shubham Bhorl Vivek Gupta Vishak Nair

Harish Shinde Prof. Manasi S.Kulkarni

The paper "Resume Parser Using Natural Language Processing Techniques" presents a literature survey of various techniques used for resume parsing, with a focus on natural language processing (NLP) techniques.

The authors begin by discussing the importance of resume parsing in the recruitment process and the challenges associated with it, such as the variability in resume formats and language. The authors highlight the need for an accurate and efficient resume parsing system.

The paper provides an overview of the different techniques used for resume parsing, including rule-based, statistical, and machine learning approaches. The authors discuss the advantages and limitations of each approach and provide examples of tools and software that utilize these techniques for resume parsing.

The authors then introduce the use of NLP techniques in resume parsing, including techniques such as part-of-speech tagging, named entity recognition, and dependency parsing. The authors provide a detailed description of how these techniques can be used to extract relevant information from resumes, such as skills, experience, and education.

The paper also discusses the use of machine learning algorithms in resume parsing, such as support vector machines and random forests. The authors provide a detailed description of how these algorithms can be used to classify resumes into different categories based on their content.

The authors then present an example of a resume parsing system developed using NLP techniques. The system involves the automatic extraction of relevant information from resumes, such as skills and experience, and the classification of resumes into different categories based on their content. The paper concludes by summarizing the key findings and outlining some of the future research directions in the field of resume parsing using NLP techniques. The authors suggest the need for more research in developing multilingual resume parsing systems, improving the accuracy of resume parsing, and addressing privacy concerns.

Overall, this literature survey provides a comprehensive overview of the different techniques used for resume parsing, with a focus on NLP techniques. The authors demonstrate the potential of these techniques in developing accurate and efficient resume parsing systems, highlighting the need for further research and development in this area.

G. Social Media Analysis using Natural Language Processing Techniques-Jyotika Singh

The paper "Social Media Analysis using Natural Language Processing Techniques" presents a literature survey of various techniques used for social media analysis, with a focus on natural language processing (NLP) techniques.

The author begins by discussing the importance of social media analysis in understanding consumer behavior and sentiment, as well as the challenges associated with analyzing large volumes of social media data. The author highlights the need for accurate and efficient social media analysis systems.

The paper provides an overview of the different techniques used for social media analysis, including rule-based, statistical,

and machine learning approaches. The author discusses the advantages and limitations of each approach and provides examples of tools and software that utilize these techniques for social media analysis.

The author then introduces the use of NLP techniques in social media analysis, including techniques such as sentiment analysis, topic modeling, and entity recognition. The author provides a detailed description of how these techniques can be used to extract relevant information from social media data, such as sentiment and opinions, topics and themes, and entities such as people, places, and organizations.

The paper also discusses the use of machine learning algorithms in social media analysis, such as neural networks and support vector machines. The author provides a detailed description of how these algorithms can be used to classify social media data into different categories based on their content.

The author then presents an example of a social media analysis system developed using NLP techniques. The system involves the automatic extraction of relevant information from social media data, such as sentiment and topics, and the classification of social media data into different categories based on their content.

The paper concludes by summarizing the key findings and outlining some of the future research directions in the field of social media analysis using NLP techniques. The author suggests the need for more research in developing multilingual social media analysis systems, improving the accuracy of social media analysis, and addressing privacy concerns.

Overall, this literature survey provides a comprehensive overview of the different techniques used for social media analysis, with a focus on NLP techniques. The author demonstrates the potential of these techniques in developing accurate and efficient social media analysis systems, highlighting the need for further research and development in this area.

IV. EXISTING SYSTEM

After analyzing the literature surveys of several papers related to natural language processing techniques for various applications such as resume parsing, social media analysis, and automated resume evaluation, it can be concluded that NLP techniques are widely used and have shown significant potential in improving the accuracy and efficiency of these systems.

Existing systems based on NLP techniques have shown promising results in automating tasks such as resume parsing, information extraction, and sentiment analysis from social media data. These systems have the potential to save a significant amount of time and effort for recruiters, HR managers, and analysts in various industries.

However, there are still some challenges that need to be addressed, such as the need for multilingual systems, improving the accuracy of these systems, and addressing privacy concerns related to the use of personal data in these systems.

Overall, NLP-based systems have shown great potential in various applications, and with further research and

development, they can be improved to overcome the existing challenges and provide more accurate and efficient results.

V. PROPOSED SYSTEM

A. Natural Language Processing

Natural Language Processing (NLP) is a subfield of Artificial Intelligence (AI) that deals with the interaction between human language and computers. It involves the development of algorithms and computational models that can analyze and understand human language and generate language-like responses.

NLP techniques can be applied to various tasks such as text classification, sentiment analysis, information extraction, machine translation, and speech recognition. It involves a combination of linguistics, computer science, and statistical modeling techniques.

NLP algorithms work by analyzing the structure of language, including syntax, semantics, and context, and then using this analysis to extract meaning from text data. This analysis involves techniques such as tokenization, stemming, part-of-speech tagging, parsing, and named entity recognition. NLP has numerous applications in various industries such as healthcare, finance, customer service, and marketing. It can be used to automate tasks such as language translation, sentiment analysis, and chatbots.

Despite significant progress, NLP still faces some challenges, such as dealing with the ambiguity and variability of language, and the need for large amounts of training data to train the algorithms effectively.

Overall, NLP is an important field in AI that has the potential to revolutionize the way we interact with computers and automate various language-related tasks.

B. Flutter

Flutter is an open-source mobile app development framework developed by Google. It allows developers to build native apps for iOS and Android platforms using a single codebase. Flutter uses a programming language called Dart, which is similar to Java and JavaScript.

Flutter provides a rich set of pre-built widgets and libraries that can be used to create beautiful and highly performant mobile apps. It uses a reactive programming model, which means that the UI is automatically updated when the app state changes, making it easier to build complex user interfaces.

Flutter's hot-reload feature allows developers to see the changes they make in real-time, without having to recompile the entire app, which significantly reduces the development time.

Flutter also provides a rich set of tools and plugins that make it easy to integrate with other technologies such as Firebase, Google Maps, and third-party APIs.

Flutter has gained significant popularity among developers due to its ease of use, speed, and flexibility. It is suitable for developing various types of apps, including e-commerce apps, social media apps, and games.

Overall, Flutter is a powerful mobile app development framework that provides developers with the tools and flexibility they need to create high-quality mobile apps. Its rich

set of features, ease of use, and flexibility make it a popular choice among developers for building mobile apps.

C. FastAPI

FastAPI is an open-source web framework for building APIs (Application Programming Interfaces) using Python 3.7 and above. It is built on top of the Starlette ASGI (Asynchronous Server Gateway Interface) framework and uses the Pydantic library for data validation and serialization.

FastAPI is designed to be fast, easy to use, and highly performant. It uses asynchronous programming to handle a large number of requests concurrently, making it ideal for building high-performance APIs. FastAPI also provides automatic generation of OpenAPI and JSON Schema documentation, making it easy to create and maintain API documentation.

One of the key features of FastAPI is its automatic data validation and serialization, which allows developers to define the expected data models using Pydantic models. This ensures that the API always returns valid and consistent data, reducing the risk of errors in the client application.

FastAPI also supports dependency injection, allowing developers to easily inject dependencies into their API functions. This makes it easy to create modular and testable code, reducing the risk of bugs and making it easier to maintain the codebase.

Overall, FastAPI is a powerful and easy-to-use web framework for building APIs in Python. Its speed, performance, and automatic data validation make it a popular choice among developers for building high-performance APIs.

D. LinkedIn and GitHub Scraping

LinkedIn and GitHub are two popular social media platforms used by professionals and developers, respectively. Web scraping is a technique used to extract data from these platforms for various purposes such as recruitment, market research, and lead generation.

BeautifulSoup and Selenium are two popular tools used for web scraping. BeautifulSoup is a Python library used to extract data from HTML and XML documents, while Selenium is a web testing framework used to automate browser actions.

To scrape data from LinkedIn and GitHub, one can use BeautifulSoup to extract information from the HTML code of the webpage, or use Selenium to automate the browser actions and simulate user interactions.

For LinkedIn scraping, one can use BeautifulSoup to extract information such as job titles, company names, and location from the LinkedIn job search pages. Selenium can be used to simulate user interactions and extract information such as contact details, education, and work experience from LinkedIn profiles.

For GitHub scraping, BeautifulSoup can be used to extract information such as repository names, descriptions, and programming languages from the GitHub search pages. Selenium can be used to automate the browser actions and extract information such as contributor details and commit history from GitHub repositories.

However, it is important to note that web scraping may violate the terms of service of these platforms and may be illegal in some cases. Therefore, it is important to ensure that

the web scraping is done ethically and legally, and with the permission of the website owners.

VI. SYSTEM MODULES

A. NLP

1. Download required NLTK modules: stopwords, punkt, averaged-perceptron-tagger, maxent-ne-chunker, words.
2. Define a function named 'extract-name' that takes the resume text as input and returns the extracted name from it using a named entity recognition (NER) technique. The function uses spaCy's Matcher object to match a pattern of two consecutive proper nouns.
3. Define a list of education-related keywords and a set of English stopwords.
4. Define a function named 'extract-education' that takes the resume text as input and returns a list of tuples, where each tuple contains an education keyword and the corresponding education text. The function first splits the text into sentences, then for each sentence, it checks if any of the keywords appear in it. If a keyword is found, the function checks the following sentence to extract the year of the education. If a year is found, the function adds the education keyword and year to the output list.
5. Define a set of skills-related keywords.
6. Define a function named 'extract-qualification' that takes the resume text as input and returns a list of skills extracted from it. The function uses the same approach as the 'extract-education' function to extract the skills.
7. Define a function named 'extract-mobile-number' that takes the resume text as input and returns the extracted mobile number using regular expressions.
8. Define a function named 'extract-email-addresses' that takes the resume text as input and returns a list of email addresses extracted from it using regular expressions.
9. Define a function named 'extract-linkedin' that takes the resume text as input and returns the extracted LinkedIn profile URL using regular expressions.
10. Call the 'extract-name', 'extract-education', 'extract-qualification', 'extract-mobile-number', 'extract-email-addresses', and 'extract-linkedin' functions with the input resume text to extract the required information.
11. Return the extracted information as a dictionary with keys as 'Name', 'Qualification', 'Skills', 'Mobile', 'Email', and 'LinkedIn'.

B. GitHub Extractor

Sure! Here's an algorithmic representation of the provided code snippet:

1. Define the function 'gitfetch1()'.
 2. Set the "folder path" variable to the path of the folder containing the JSON files.
 3. Initialize an empty list named 'results' to store the fetched data.
 4. Iterate over each file in the folder using 'os.listdir(folder path)'.
 - For each 'filename':
 - a. Check if the file has a '.json' extension using 'filename.endswith(".json")'.

- b. If the file is a JSON file:
 - i. Read the contents of the file into a variable named 'jsonstring'.
 - ii. Parse the JSON string using 'json.loads(json string)' and assign the resulting object to 'git data'.
 - iii. Check if the 'git data' object has a key named "Github" using 'git data.get("Github")'.
 - iv. If the "Github" key exists:
 - Extract the GitHub URL from the first element of the list and assign it to 'github url'.
 - Check if the GitHub URL is in the format 'username (github.com)'.
 - If true:
 - Extract the username from the URL by splitting on "(" and taking the first part.
 - If false:
 - Extract the username from the URL by splitting on ")" and taking the last part.
 - Construct the URL for the Flask app by appending the extracted 'github username' to "http://localhost:5000/repos/".
 - Send a GET request to the constructed URL using 'requests.get(url)'.
 - If the response status code is 200:
 - Append the JSON response to the 'results' list using 'response.json()'.
 - If the response status code is not 200:
 - Append an error message dictionary to the 'results' list.
5. Return the 'results' list.

Note: This algorithm assumes that the necessary imports (e.g., 'os', 'json', 'requests') are already included and that the Flask app is running correctly.

VII. CONCLUSION

In conclusion, the proposed solution addresses the challenges faced by hiring managers in the initial screening of resumes. By incorporating functionalities such as resume data extraction, duplicate profile identification, social profile building, automated skill matching, resume ranking and scoring, keyword search, customizable screening criteria, integration with ATS, reporting and analytics, user-friendly interface, security and data privacy, and continuous improvement, the solution aims to streamline and expedite the resume screening process.

Implementing this solution can save valuable time and effort for hiring managers, allowing them to focus on evaluating the most qualified candidates. It reduces the risk of oversight or inconsistencies in the screening process and improves the

chances of identifying the best candidates for further consideration. Additionally, the solution provides a data-driven approach to hiring, enabling stakeholders to make informed decisions based on relevant metrics and analytics.

By leveraging NLP, automation, and machine learning techniques, the solution not only enhances efficiency but also adapts and improves over time, learning from user preferences and patterns. This ensures that the screening process becomes more accurate, effective, and aligned with the specific requirements of the job.

Overall, the proposed solution offers a comprehensive and technology-driven approach to fast-tracking the initial screening of resumes, empowering hiring managers and streamlining the hiring process for improved outcomes.

ACKNOWLEDGMENT

We would like to express our gratitude to all those who have contributed to the development of this resume parsing application. We are especially grateful to our team members who have worked tirelessly to ensure that this application meets the highest standards of accuracy and functionality. We also acknowledge the support and guidance provided by our advisors and mentors, whose expertise and insights have been invaluable in the development of this project.

We would also like to extend our appreciation to the individuals and organizations who generously provided their resumes for use in testing and refining this application. Their willingness to participate in our study has helped to improve the quality and effectiveness of this tool.

Finally, we would like to acknowledge the developers of the open-source libraries and tools used in the development of this application. Without their hard work and dedication, this project would not have been possible.

Thank you all for your contributions to this project.

REFERENCES

- [1] Resume Parser with Natural Language Processing by Satyaki Sanyal, Souvik Hazra, Soumyashree Adhikary, Nee- lanjan Ghosh
- [2] Resume Classification using various Machine Learning Algorithms - Riya Pal, Shahrukh Shaikh, Swaraj Satpute and Sumedha Bhagwa
- [3] Resume Parsing and Categorizing by AbeerZaroor
- [4] Resume Information Extraction with A Novel Text Block Segmentation Algorithm
- [5] Automated Resume Evaluation System using NLP, Rohini Nimbekar; Yoqesh Patil; Rahul Prabhu; Shainila Mulla
- [6] Resume Parser Using Natural Language Processing Techniques, Shubham Bhor1 Vivek Gupta Vishak Nair Harish Shinde Prof. Manasi S.Kulkarni
- [7] Social Media Analysis using Natural Language Processing Techniques- Jyotika Singh