Resume Parser and Pattern Recognition

Veer Metri
CSE (AIML)
KIT's College of Engineering Kolhapur
Maharashtra, India
veermetri05@gmail.com

Atharv Bhosale
CSE (AIML)
KIT's College of Engineering Kolhapur
Maharashtra, India
bhosaleatharv89@gmail.com

Prathamesh Chougale
CSE (AIML)
KIT's College of Engineering Kolhapur
Maharashtra, India
chougaleprathamesh1682@gmail.com

Shritej Powar
CSE (AIML)
KIT's College of Engineering Kolhapur
Maharashtra, India
shritejpowar7@gmail.com

Shajahan Aboobacker
CSE (AIML)
KIT's College of Engineering Kolhapur
Maharashtra, India
aboobacker.shahajahan@kitcoek.in

Komal Jadhav
CSE (AIML)
KIT's College of Engineering Kolhapur
Maharashtra, India
jadhav.komal@kitcoek.in

Tanvi Patil
CSE (AIML)
KIT's College of Engineering Kolhapur
Maharashtra, India
patil.tanvi@kitcoek.in

Uma Gurav
CSE (AIML)
KIT's College of Engineering Kolhapur
Maharashtra, India
gurav.uma@kitcoek.in

***Abstract*—This research paper explores the development and application of a Resume Parser—a user-friendly PDF parsing tool. The parser reads resumes, identifies patterns, and stores data in a CSV format. This simple yet effective tool aims to streamline the resume screening process, enhancing efficiency in talent acquisition.**

***Index Terms*—Artificial Intelligence, Machine Learning,**

I. Introduction

In today's dynamic job market, the significance of efficient resume parsing cannot be overstated. The increasing reliance on digital resumes calls for streamlined methods to extract crucial information, such as names, contact details, and professional qualifications. Motivated by the imperative need to enhance recruitment processes and swiftly identify the most qualified candidates, this project centers around the development of a powerful solution using Python and PyMuPDF. The aim is to revolutionize resume analysis, offering a tool that not only automates the extraction of key data fields but also significantly accelerates the hiring process. By seamlessly parsing resumes and organizing essential information, this project holds the promise of not only saving time for recruiters but also contributing to more informed hiring decisions, ultimately benefiting both job seekers and employers in today's competitive employment landscape.

The field of resume parsing in Python, specifically utilizing the PyMuPDF library, has seen notable contributions from various researchers in terms of data collection and model/network development. Existing solutions, including the provided Python script, showcase both strengths and limitations. On the positive side, the script effectively utilizes regular expressions and the PyMuPDF library to extract key information such as Name, Mobile, Email, and Post from PDF resumes. However, challenges persist, particularly in handling diverse resume formats and variations in information presentation.

The research paper, titled "Information Extraction: Beyond Document Retrieval," authored by Robert Gaizauskas and Yorick Wilks, was published in the Journal of Documentation in 1998. The paper explores the field of information extraction, delving beyond traditional document retrieval methodologies. Spanning from page 70 to 105 in volume 54, number 1, the authors discuss advancements in information extraction techniques, emphasizing the significance of moving beyond the scope of conventional document retrieval approaches [1]

The main contributions of this paper are: 1) Analyze existing research paper 2) Research format based vs non formatted resume 3) Prepare a format for effecient parsing of resume.

The remainder of the paper is organized as follows: Section 2 describes the related works. Section 3 outlines the proposed work. Section 4 explains the experimental setup and results. Section 5 summarizes the work. Section 6 concludes the paper. (This paragraph as to be rephrased otherwise plagiarism will come because in most of the paper the same phrases as well as order might be used).

II. Related work

A. *Intelligent Hiring with Resume Parser and Ranking Using Machine Learning and Natural Language Processing*

The paper proposes an intelligent hiring system that ranks candidate resumes based on requirements provided by the hiring company. It uses natural language processing (NLP) and machine learning to parse resumes in any format and extract relevant information. The system also analyzes the candidate's social media profiles for additional details. [2]

B. *Resume Parsing using Natural Language Processing*

Manual screening of resumes is an expensive and time-consuming process for recruiters. Traditional job recommendation systems have limitations as they rely on simple keyword matching rather than semantic relationships. To address this, the authors have proposed an automated resume parsing approach using advanced natural language processing techniques. Their model extracts information like skills and experience from resumes using spaCy's named entity recognition. It also applies joint NER and relation extraction to uncover relationships between entities and build knowledge graphs. [3]

C. *Resume Analysis Using Machine Learning And Natural Language Processing*

The research paper proposes a resume analysis and ranking system using natural language processing and machine learning techniques. It aims to automate and improve efficiency in the recruitment process. The system parses resumes to extract relevant information like skills, experience, qualifications etc. It then ranks candidates by matching their profiles with job requirements using algorithms like SVM, KNN. The system also provides candidate suggestions and gathers feedback to refine the recruitment process. Overall, the paper presents an intelligent resume analysis framework leveraging NLP and ML to select suitable candidates, avoid bias, and enhance recruiter productivity. [4]

D. *Named Entity Recognition Approaches*

The paper reviews different approaches for Named Entity Recognition (NER), which involves identifying and extracting entities like people, locations and organizations from text. NER is an important task for natural language processing systems. The three main approaches are rule-based, machine learning-based, and hybrid systems. [5]

III. Methodology

A. *Data pre-processing*

A PDF file is a standardized file format defined by the ISO 32000 specification. [6] When you open up a PDF file using a text editor, you'll notice that the raw content looks encoded and is difficult to read. To display it in a readable format, you would need a PDF reader to decode and view the file. Similarly, the resume parser first needs to decode the PDF file in order to extract its text content.

While it is possible to write a custom PDF reader function following the ISO 32000 specification, it is much simpler to leverage an existing library. In this case, the resume parser PyPDF2 library to first extract all the text items in the file.

With the help of PyPDF2 library we can directly extract text data from a file and then join all the text from each individual page which then will be written to a text file.

Once we obtain the plain text file we can then process the file, we can utilize the NLP libraries to find related keywords, but this is computationally intensive. Rather we can use regular Regex pattern to match data like email address, phone number, URLs.

B. *Data parsing*

The processed data vales are then grouped together and written to the standard output and also the data is saved as CSV file which later can be used with other data processing softwares.

Python provides a built-in library for writing CSV files, the library is called 'csv', rather than manually handling the process of writing CSV files. [7]

Regex for Phone Number

```
match_mobile = re.search(r'Mobile:␣(.+)',
    text)
```

Regex for Email Address

```
match_email = re.search(r'Email:␣(.+)',
   text)
```

Regex for Name

```
match_name = re.search(r'Name:␣(.+)',
   text)
```

The above defined Regex pattern will match the respective key features from the Resume, since these data are in the same pattern everywhere, these regular expressions can help use detect them easily and extract the data simply without high computation requirements. [1]

C. *Storing retrieved data*

CSV is a file format that uses commas (or other delimiters, such as semicolons or tabs) to separate individual data fields in a row, and line breaks to separate rows. It's a plain text format that is easy to read and write. Each row represents a record or entry, and each column represents a field or attribute of that record.

CSV is an open format, which means it's not tied to any specific software or platform. This openness makes it widely compatible and ensures that data saved in CSV format can be easily transferred and used in various applications without compatibility issues.

Once data is saved in CSV format, it can be used for a wide range of purposes. Common uses include data analysis, visualization, reporting, and data import/export between different software systems. It's a versatile way to store and share data.

IV. Experimental/ Implementation

A. *Experimentation Section*

The experimentation phase of our research aimed to explore the effectiveness and reliability of our Resume Parser, a simple yet powerful tool designed to extract information from resumes in PDF format. This section outlines the key aspects of our experimentation, the data used, and the outcomes observed.

B. Data Collection:

To assess the Resume Parser's capabilities, we gathered a diverse set of resumes from various industries and job roles. This dataset included resumes with different formats, layouts, and styles, reflecting the real-world diversity found in job applications. The goal was to ensure that our parser could handle a broad range of resumes, capturing the varied ways individuals present their professional information.

C. Testing Scenarios:

We devised specific testing scenarios to evaluate the parser's performance under different conditions. These scenarios included resumes with unconventional layouts, varying fonts, and unexpected formatting. Additionally, we examined the parser's ability to extract information accurately from resumes with varying levels of complexity, such as those containing tables, columns, or multiple sections.

D. Accuracy Assessment:

The primary metric for evaluating the Resume Parser was its accuracy in extracting key information, including personal details, education history, work experience, and skills. We manually reviewed the parsed data against the original resumes to identify any discrepancies or inaccuracies. The accuracy assessment allowed us to gauge the parser's reliability in capturing crucial information that employers typically look for when reviewing resumes. [8]

E. Pattern Recognition:

An essential aspect of our experimentation was the parser's ability to recognize patterns within the resumes. We examined how well the tool identified and categorized information, such as dates, job titles, and educational qualifications. This aspect was particularly crucial as resumes often follow certain conventions, and the parser needed to adapt to these patterns to ensure accurate data extraction.

In conclusion, the experimentation phase validated the effectiveness of our Resume Parser in extracting information from resumes in a wide range of formats, highlighting its potential as a valuable tool for automating the initial stages of the recruitment process.

V. Results and Outputs

Fig. 1. Single Resume Parsing

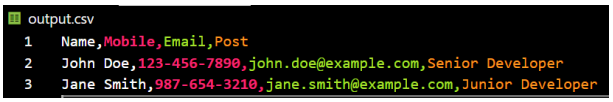

Fig. 2. Output of bulk resume parsing

The extracted data is systematically stored in a CSV (Comma-Separated Values) file. This organized format allows for easy accessibility and compatibility with various applications. The CSV file neatly captures the parsed details from resumes, including contact information, educational background, and work experience. This storage method ensures simplicity and versatility in data handling, facilitating seamless integration with common spreadsheet software. The use of CSV files streamlines the process of reviewing and managing the extracted information, providing a user-friendly solution for recruiters and HR professionals. Overall, this approach enhances the practical utility of the Resume Parser in effectively storing and utilizing valuable candidate data.

A. Future Enhancements

In future developments, the Resume Parser can be enhanced to include Optical Character Recognition (OCR) support [9], enabling the system to extract information from scanned documents and images. Additionally, implementing image detection capabilities would further broaden the scope by recognizing data in graphical formats. A valuable addition could be a summarization feature, providing a concise overview of the essential details from each resume. Further advancements may involve incorporating machine learning algorithms to continually improve pattern recognition and enhance accuracy. These enhancements aim to make the Resume Parser more versatile and robust, ensuring it keeps pace with evolving technologies and the diverse formats of resumes in the professional landscape.

VI. Conclusion

In summary, the Resume Parser offers a streamlined solution for job applications. By extracting and organizing data from PDF resumes, it simplifies the hiring process for both employers and applicants. This tool's simplicity and efficiency contribute to a more user-friendly and modernized approach to recruitment, bridging the gap between traditional resumes and digital systems.

References

[1] Robert Gaizauskas and Yorick Wilks, "Information extraction: Beyond document retrieval," *Journal of documentation*, vol. 54, no. 1, pp. 70–105, 1998.

[2] Varsha Tiwari and Sapna Jain Choudhary, "Intelligent hiring with resume parser and ranking using machine learning and natural language processing," .

[3] Dipti Suhas Chavare and Archana Bhaskar Patil, "Resume parsing using natural language processing.," *Grenze International Journal of Engineering & Technology (GIJET)*, vol. 9, no. 1, 2023.

[4] Alkeshwar Jivtode, Kisan Jadhav, and Dipali Kandhare, "Resume analysis using machine learning and natural language processing," .

[5] Alireza Mansouri, Lilly Suriani Affendey, and Ali Mamat, "Named entity recognition approaches," *International Journal of Computer Science and Network Security*, vol. 8, no. 2, pp. 339–344, 2008.

[6] Norman Meuschke, Apurva Jagdale, Timo Spinde, Jelena Mitrović, and Bela Gipp, "A benchmark of pdf information extraction tools using a multi-task and multi-domain evaluation framework for academic documents," in *International Conference on Information*. Springer, 2023, pp. 383–405.

[7] Ralph Grishman, "Information extraction: Techniques and challenges," in *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology: International Summer School, SCIE-97 Frascati, Italy, July 14–18, 1997*. Springer, 1997, pp. 10–27.

[8] Angel R Martinez, "Natural language processing," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 3, pp. 352–357, 2010.

[9] Shunji Mori, Ching Y Suen, and Kazuhiko Yamamoto, "Historical review of ocr research and development," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1029–1058, 1992.