International Journal of Research in Interdisciplinary Studies
Volume 3, Issue 4, April 2025
https://www.ijris.com | ISSN (Online): 2584-1017

Automated Resume Parser Using Natural Language Processing

Shahabaz Shaikh[1*], Md Mustafa Mallebhari[2], Shakila Siddavatam[3]
[1,2]*Student, Department of Computer Science, Abeda Inamdar Senior College, Pune, India*
[3]*HoD, Department of Computer Science, Abeda Inamdar Senior College, Pune, India*

Abstract— Nowadays Recruiters invest a significant amount of time manually reviewing job seekers CV to identify key details such as skills, experience, and employment status etc. This process is quite time-consuming but also prone to human error. In this paper, we are going to propose an automated resume parsing system leveraging Natural Language Processing (NLP) to extract some specific details of the candidate's in efficiently. Our approach consists of Named Entity Recognition (NER), and deep learning models such as BERT to enhance the accuracy of information which is to be extracted. Unlike traditional rule-based parsers that rely on rigid keyword matching, our model demonstrates adaptability to different resume formats and text variations.

***Index Terms*— Resume Parsing, Natural Language Processing.**

1. Introduction

Recruitment of candidates is a very important process for organizations, and checking every resume individually to find the right candidate is time-consuming. Traditional methods rely on manual screening by humans, which is very likely to generate errors and inefficiencies. An automated resume parser using NLP can extract key information, from both structured and unstructured text format, and generate a summarized report for recruiters. This research paper focuses on developing a robust resume parser that can efficiently extract below data from the resume.

- Primary Skills
- Total Experience
- Employment Status
- Expected Salary
- Expected Designation

2. Literature Review

Several resume parsing solutions exist in real world, including rule-based and machine learning-based approaches. Traditional parsing methods fail to handle variations that are in job descriptions. Recent advancements in NLP, such as Named Entity Recognition (NER), transformer models (BERT, RoBERTa), and dependency parsing, have improved text extraction to its finest accuracy.

Resume Parsing Approaches Existing in real world

1. *Rule-Based Systems*: It is dependent on predefined patterns but lacks adaptability.
2. *Machine Learning Approaches*: It uses classifiers but require labelled data.
3. *Deep Learning & NLP-Based Systems*: Uses leverage transformers like BERT for contextual understanding.

3. Methodology

A. Data Collection

We use publicly available resume datasets to train and evaluate our model. The dataset includes our resumes in the PDF, DOCX, and TXT formats.

B. Pre-Processing

For text extraction we used libraries such as pdfplumber, PyMuPDF, and python-docx to extract text from different resumes.

In text cleaning we removed special characters, stop words, and redundant spaces.

For tokenization and pos tagging we used spaCy and NLTK to rephrase text into meaningful units.

C. Feature Extraction

We used predefined skill lists to extract skills from resumes.

Extracting job titles, start and end dates of previous jobs, and calculating years of experience candidates have.

Employment Status Detection: Classifying status of an employee using a rule-based and ML hybrid approach (e.g., “Currently working at ABC” -- Working, or “On notice period” -- Notice Period, or “Unemployed” -- Available).

D. NLP Models Used

- We used Named Entity Recognition (NER) to identify skills, job titles, and organizations.
- Dependency Parsing is used to Extracts hierarchical relationships in sentences.
- BERT-based Classification is used in our model.

4. Pseudo Algorithms

```
def extract_entities(text):
    doc = nlp(text)
    entities = {'NAME': [], 'SALARY': [], 'SKILLS': [], 'STATUS': [], 'DESIGNATION': []}
    for ent in doc.ents:
```

*Corresponding author: shahabaazshaikh.sj6@gmail.com

```
        if ent.label_ == 'PERSON' and not entities['NAME']:
            entities['NAME'] = ent.text
        elif ent.label_ == 'SALARY':
            entities['SALARY'] = ent.text
        elif ent.label_ == '': 'SKILLS'
            entities['SKILLS'] = ent.text
        elif ent.label_ == 'STATUS':
            entities['STATUS'] = ent.text
    return entities
    def extract_skills(text, skills_list):
        skills_found = []
        for skill in skills_list:
            if skill.lower() in text.lower():
                skills_found.append(skill)
        return skills_found
```

5. Implementation

Python Programming Language used in our model.

Libraries used in our model are spaCy, Transformers, NLTK, pdfplumber, Scikit-learn, Pandas

Pipeline:

1. First the text is extracted from the resume files.
2. Pre-processing (cleaning, tokenization)
3. Extraction of skills (NER-based and rule-based)
4. Calculation of the experience candidate has (duration parsing)
5. Employment status classification (BERT model)
6. Summary Generation

6. Results & Evaluation

Our model was tested on 150 resumes, achieving the following performance metrics:

The results show a significant improvement over traditional resume parsers, particularly in employment status classification which were done manually in previous days.

7. Discussion

1. While our model effectively extracts key details from resumes, some challenges still remain for our model to achieve.
2. Handling Variability is one of them, many resumes have inconsistent formats, making extraction process complex.
3. Ambiguity can occur in resumes and we need to train model accordingly.
4. Resumes can be written in many different languages and to support multiple languages requires additional training data.

8. Conclusion

The recruitment process mostly relies on automation and AI-driven solutions to ease candidate screening processes. This study presents an NLP-powered resume parser process developed to extract skills, experience, and employment status from both structured and unstructured resume text with high accuracy. Unlike traditional rule-based or keyword-matching parsers, our approach leverages Named Entity Recognition (NER), dependency parsing, and deep learning models such as BERT to improve the quality of information extraction.

9. Future Work

- Improving context-aware skill extraction.
- While our model effectively extracts skills, it lacks in some domains. Future work will focus on context-based skill identification.
- Multilingual and cross-cultural resume parsing.
- Many companies receive resumes in different languages. Expanding support for multilingual NLP will enhance adaptability for candidate selection.
- Real-time resume parsing API for HR systems.
- Analysing resumes in real time for quick HR judgements and for parsing API for HR systems.

References

[1] Devlin, J., Chang, M., Lee, K., & Tout nova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

[2] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and Their Compositionality.

[3] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural Architectures for Named Entity Recognition.