IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, November 2022
Impact Factor: 6.252

Resume Parser using Natural Language Processing

Mr. B. Venkata Satish Babu[1], R. Bharath[2], Sk. Parvez[3], S. Sreya[4], M. Yaswini[5]
Assistant Professor, Department of Information Technology[1]
B. Tech Students, Department of Information Technology[2,3,4,5]
Prasad V. Potluri Siddhartha Institute of Technology, Vijayawada, Andhra Pradesh, India

Abstract: *Manual extraction of information from the resume is very difficult and time taking process. The project mainly focuses on extracting the required information from the resumes using Natural Language Processing Techniques. The large set of resumes can be parsed using this kind of system. This project helps us know about the needed and important aspects while shortlisting the candidates and also helps the companies and high-level firms to select the quality employees from the extracted information. This project also helps to improve the shortlisting the process.*

Keywords: Resume Parser, Spacy, NLTK, Natural Language Processing, JSON, Pymongo, Information Extraction, Text Cleaning, BERT algorithm, Regular Expressions.

I. INTRODUCTION

Extracting the useful information from resume has become a difficult task for the recruiters and shortlisting the candidates based on the requirements company is also important. One cannot spend too much time on filling the data in forms manually. To overcome all these problems an extraction system has to be used to understand Human language. To make the task easier for recruiters using resume parser, information can be extracted from the submitted resume of candidates. The resume submitted can be in either .DOCX format or in PDF format, the system parses the whole resume and gives the results in JSON format that can be processed for further analysis.

II. PROPOSED SYSTEM

Processing the large amount of data of having huge set of resumes can be made easier using Natural Language Processing Techniques. With the use of NLTK, Spacy and some other python libraries like RegEx, the required information can be extracted from the resumes [8].The extracted Data can be used for further analysis which can be taken care of by the recruiting company or organization. The extracted information can be the candidate's name, email address, phone number, skillset, education and other required columns which are mentioned in the unstructured form of resumes.

III. TOOLS REQUIRED

3.1 Google Colaboratory

It is an open-source toolfor developing python and machine learning projects [1].It is cloud-based platform where you can access the entire code written and the results from the executed code from anywhere in the world. The libraries and the dependencies that are going to be installed in colab are by default global.

IV. LIBRARIES USED

The following are some of the most important libraries and modules:

1. **Pandas:** Pandas is a Python library which is an open source. It's a tool for analysing data [2].
2. **OS:** In Python, the OS module has methods for creating and deleting folders, retrieving their contents, altering and identifying the current directory, and so on.
3. **Spacy:** SpaCy is an open-source package library for advanced natural language processing implemented in Python. SpaCy has pre-trained pipelines with tokenization support [9].
4. **Matcher:** The Matcher assists in the finding of words and phrases by utilizing rules that describe their token properties. After applying the matcher on a Doc, we can see the matched tokens in context.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, November 2022
Impact Factor: 6.252

5. **Nltk.corpus:** The modules in this package provide functions for reading corpus files in a variety of formats. A corpus is simply a set of texts that are used as input [8].
6. **RegEx:** RegEx, also known as a Regular Expression, is a string of characters that defines a search pattern. This module's functions allow to see if a given string matches a given regular expression [3].
7. **NLTK:** NLTK is a Python toolbox for working with natural language processing. It gives us access to a number of text processing libraries as well as a large number of test datasets [8].
8. **Docx2txt:** Pure Python-based library for extracting text from docx files [10].
9. **Pdfminer:** PDFMiner is a tool that extracts text from PDF files. It focuses completely on gathering and processing text data, unlike other PDF-related tools [11].
10. **Pymongo:** To establish connection with the mongoDB database i.e., MongoDB Atlas, a cloud database we used Pymongo. The results are stored in Key Value pairs and they are used in apis in JSON format [15].

V. THE MODEL

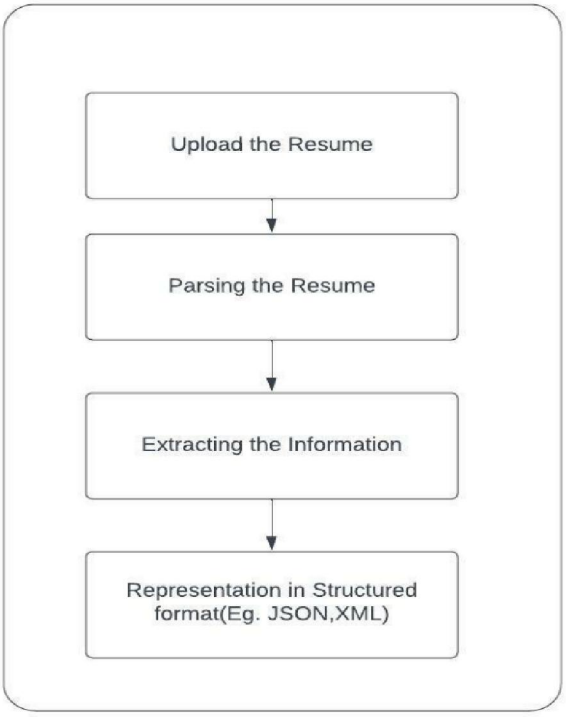


Fig: 5.1: The model of Resume Parser.

The resume should be uploaded before parsing it in the format of .pdf or .docx. After the resume got loaded completely then it can be parsed. The parsing process of resume is done on analyzing the syntax of the text in the resume which follows the grammatical form. The parsing process uses the Natural Language Processing (NLP).

VI. RELATED WORK

6.1 Text Extraction

```
#Extracting Text from the Resume

def doctotext(m):
    temp = docx2txt.process(m)
    resume_text = [line.replace('\t', ' ') for line in temp.split('\n') if line]
    text = ' '.join(resume_text)
    print(text)
    return (text)
```

Figure 6.1: Extracting text from resume.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, November 2022
Impact Factor: 6.252

Here, doctotext() is used to extract information from the resume by ignoring the images in the resume.

```
def pdftotext(m):
    # pdf file object
    # you can find find the pdf file with complete code in below
    pdfFileObj = open(m, 'rb')

    # pdf reader object
    pdfFileReader = PdfFileReader(pdfFileObj)

    # number of pages in pdf
    num_pages = pdfFileReader.numPages

    currentPageNumber = 0
    text = ''

    # Loop in all the pdf pages.
    while(currentPageNumber < num_pages ):

        # Get the specified pdf page object.
        pdfPage = pdfFileReader.getPage(currentPageNumber)

        # Get pdf page text.
        text = text + pdfPage.extractText()

        # Process next page.
        currentPageNumber += 1
    print(text)
    return (text)
```

Figure 6.2: Extraction of text from pdf.

From the library called pdfminer, we used PdffileReader which converts pdf file in to pdf file object. It parses every page in the resume and extracts the text using extractText () method [11].

6.2 Tokenization

Tokens are usually referred to as terms or words, but sometimes fabricating a type/token distinction is essential [5]. A specimen of an array of characters in a document that is assembled as a helpful acceptable unit for processing is called a token. Whereas, the group of tokens which consists of same character sequence is called type. And a type that is added to the dictionary of IR system is called term. We can completely differentiate a set of index terms from tokens. As an example, we can say, they can be acceptable identifiers in taxonomy, but mostly in modern IR systems, they have a strong relation with tokens in the document. Nevertheless, as a substitute for being totally the tokens appearing in the document, they are mostly derived from them by various processes of normalization.

6.3 Chunking

```
for i in noun_chunks_list:
  if i.lower() in skills:
    final.append(i)
```

Figure 6.3: Chunking.

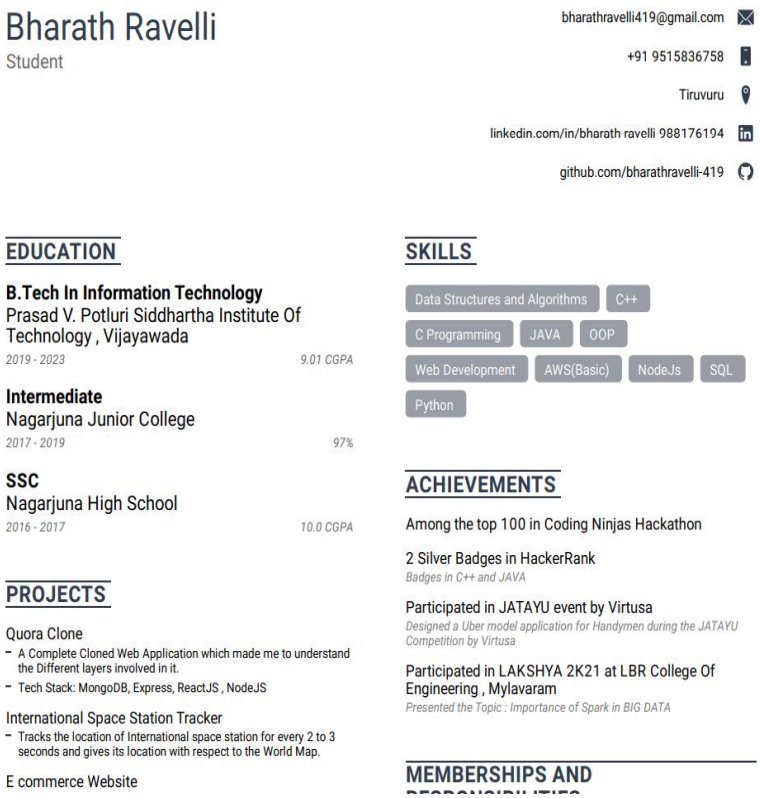
IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, November 2022
Impact Factor: 6.252

Also known as shallow parsing, chunking is actually the recognition of parts of speech and short phrases [6]. We can determine if the words are nouns, verbs, adjectives, etc by Parts of Speech tagging, but from this, we cannot get any clue about the sentence of phrase structure in the sentence. At times, some more information than parts of speech of words are useful, but the full parse tree that we would get from parsing is not needed. Named-entity recognition is a citation when chunking might be preferable. In NER, the goal is finding named entities, which mostly has a tendency to be noun phrases, so in the following sentence we would want to know if 'The angry bear' is there or not: "The angry bear chased the frightened little squirrel" But one wouldn't necessarily care if the angry bear is the subject of the sentence or not [7]. Chunking is also commonly used in tasks like example -based machine natural language understanding, speech generation, and others as a pre-processing step.

6.4 Lemmatization

In linguistics, lemmatization is a procedure of organising the altered form of a word, such that their analysis can take place as a single term, identified by the word's dictionary form (lemma) [12]. In computational linguistics, the procedure of concluding the lemma of a word depending upon its predetermined meaning. It depends on rightly identifying the predetermined part of speech and what a word in a sentence means, as well as in a bigger situation surrounding the sentences, which can include neighbouring sentences, and even an entire document, which contradicts stemming. So, an algorithm of lemmatization is an open platform for research.

VII. EXPERIMENTAL RESULTS



Bharath Ravelli
Student

bharathravelli419@gmail.com
+91 9515836758
Tiruvuru
linkedin.com/in/bharath ravelli 988176194
github.com/bharathravelli-419

EDUCATION

B.Tech In Information Technology
Prasad V. Potluri Siddhartha Institute Of Technology , Vijayawada
2019 - 2023 9.01 CGPA

Intermediate
Nagarjuna Junior College
2017 - 2019 97%

SSC
Nagarjuna High School
2016 - 2017 10.0 CGPA

PROJECTS

Quora Clone
- A Complete Cloned Web Application which made me to understand the Different layers involved in it.
- Tech Stack: MongoDB, Express, ReactJS , NodeJS

International Space Station Tracker
- Tracks the location of International space station for every 2 to 3 seconds and gives its location with respect to the World Map.

E commerce Website

SKILLS

Data Structures and Algorithms | C++ | C Programming | JAVA | OOP | Web Development | AWS(Basic) | NodeJs | SQL | Python

ACHIEVEMENTS

Among the top 100 in Coding Ninjas Hackathon

2 Silver Badges in HackerRank
Badges in C++ and JAVA

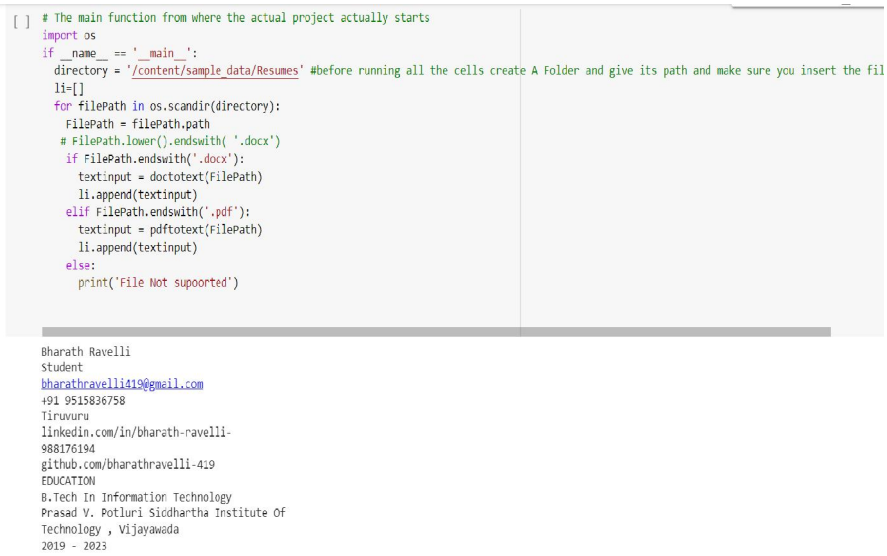
Participated in JATAYU event by Virtusa
Designed a Uber model application for Handymen during the JATAYU Competition by Virtusa

Participated in LAKSHYA 2K21 at LBR College Of Engineering , Mylavaram
Presented the Topic : Importance of Spark in BIG DATA

MEMBERSHIPS AND

Figure 7.1: Input Resume

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, November 2022
Impact Factor: 6.252



```
[ ] # The main function from where the actual project actually starts
    import os
    if __name__ == '__main__':
      directory = '/content/sample_data/Resumes' #before running all the cells create A Folder and give its path and make sure you insert the fil
      li=[]
      for filePath in os.scandir(directory):
        FilePath = filePath.path
       # FilePath.lower().endswith( '.docx')
        if FilePath.endswith('.docx'):
          textinput = doctotext(FilePath)
          li.append(textinput)
        elif FilePath.endswith('.pdf'):
          textinput = pdftotext(FilePath)
          li.append(textinput)
        else:
          print('File Not supoorted')
```

```
Bharath Ravelli
Student
bharathravelli419@gmail.com
+91 9515836758
Tiruvuru
linkedin.com/in/bharath-ravelli-
988176194
github.com/bharathravelli-419
EDUCATION
B.Tech In Information Technology
Prasad V. Potluri Siddhartha Institute Of
Technology , Vijayawada
2019 - 2023
```

Figure 7.2: (Resume) pdf to text.

When the resume is uploaded in the form of pdf and by using pdfminer, the information in pdf using deep neural networks converts to text format as shown above [11].

```
[ ]  def extract_name(resume_text):
         nlp_text = nlp(resume_text)

         # First name and Last name are always Proper Nouns
         pattern = [{'POS': 'PROPN'},{'POS': 'PROPN'}]

         matcher.add('NAME', [pattern])

         matches = matcher(nlp_text)

         for match_id, start, end in matches:
             span = nlp_text[start:end]
             return span.text
     print('Name: ',extract_name(li[0]))
     #na = extract_name(textinput)
```

```
Name:  Bharath Ravelli
```

Figure 7.3: Name extraction.

The above output shows the name extraction from the inputted resume (.pdf) using matcher.

```
# Extracting skills
def extract_skills(resume_text):
    final =[]
    resume_list = []
    skill_list = []
    doc = nlp(resume_text)
    data = pd.read_csv("https://raw.githubusercontent.com/OmkarPathak/ResumeParser/master/resume_parser/resume_parser/skills.csv")
    skills =[]
    noun_chunks_list=[]
    for row in data:
      skills.append(row.lower())
    for noun_chunks in doc.noun_chunks:
      test_str = ''.join(letter for letter in str(noun_chunks.text) if letter.isalnum())
      noun_chunks_list.append(test_str.lower())
    #print(noun_chunks_list)
    #print(skills)
    for i in noun_chunks_list:
      if i.lower() in skills:
        final.append(i)
    return list(set( final))
extract_skills(li[0])
```

```
['c', 'spark', 'engineering', 'java', 'spacy']
```

Figure 7.4: Skills Extraction.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, November 2022
Impact Factor: 6.252

Using pandas [2], a .csv file consisting of all the skillset for a job is read. Using chunking, the noun chunks matching with the chunks of data extracted from the resume are taken out. Finally, they are appended to a list.

```
def extract_email_addresses(string):
    r = re.compile(r'[\w\.-]+@[\w\.-]+')
    return r.findall(string)
print('Mail id: ',extract_email_addresses(li[0]))
#ea = extract_email_addresses(textinput)
```

```
Mail id:  ['bharathravelli419@gmail.com']
```

Figure 7.5: E-mail extraction.

Using the RegEx library,the e mail id is extracted from the resume text [3].

```
list(records.find())
```

```
[{'_id': ObjectId('6375f10deb975a09fda782e4'),
  'name': 'Bharath Ravelli',
  'phone_number': '919515836758',
  'skills': ['c', 'spark', 'engineering', 'java', 'spacy'],
  'email_id': ['bharathravelli419@gmail.com'],
  'Qualification': ['BTech']}]
```

Figure 7.6: Storing the information into the Database

The results are stored in a NOSQL database (MongoDB).It is being stored in a cloud database MongoDB atlas [14] .

VIII. SCOPE OF FUTURE USE

This Project can be developed and made more efficient with usage of some enriched libraries like pyresparser [4]. By Choosing some other algorithms along with BERT [13]. We can also make use of Deep Learning techniques to extract text and useful information from images also. The trained models like pyresparser usage can be of better choice to enhance the performance of Resume Parser and to have more accuracy.

IX. CONCLUSION

Hence, with this project the recruitment process can be made more efficient and faster. There are many companies and organizations which need a huge recruitment every year. Receiving large number of resumes and employing the right candidates for making better quality of products can be made easier with the Resume Parser.

REFERENCES

[1]. https://colab.research.google.com/
[2]. https://pandas.pydata.org/
[3]. https://docs.python.org/3/library/re.html
[4]. https://snyk.io/advisor/python/pyresparser
[5]. https://www.tutorialspoint.com/python_text_processing/python_tokenization.htm#:~:text=In%20Python%20tokeization%20basically%20refers,in%20programs%20as%20shown%20below.
[6]. https://pythonprogramming.net/chunking-nltk-tutorial/
[7]. https://blog.devgenius.io/named-entity-recognition-ner-nlp-python-6504d5843f98
[8]. https://www.analyticsvidhya.com/blog/2021/07/getting-started-with-natural-language-processing-using-python/
[9]. https://realpython.com/natural-language-processing-spacy-python/
[10]. https://pypi.org/project/docx2txt/
[11]. https://pypi.org/project/pdfminer/
[12]. https://www.geeksforgeeks.org/python-lemmatization-with-nltk/

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, November 2022
Impact Factor: 6.252

[13]. https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/

[14]. https://www.mongodb.com/cloud/atlas/lp/try4?utm_source=google&utm_campaign=search_gs_pl_evergreen_atlas_core_prosp-brand_gic-null_apac-in_ps-all_desktop_eng_lead&utm_term=mongodb%20atlas&utm_medium=cpc_paid_search&utm_ad=e&utm_ad_campaign_id=12212624347&adgroup=115749713263&gclid=CjwKCAiApvebBhAvEiwAe7mHSGzd1_olXBGp0RDVaCpIyqSSSVdXksVtJtBDP71t1_cWvBuafZDa8BoCUHYQAvD_BwE

[15]. https://pymongo.readthedocs.io/en/stable/