Natural Language Processing

Final Project

RESUME SCREENER

Chinnam Sohith

Kata Revanth

Alluri Sri Ram

GitHub: https://github.com/kata33/NLP_Final_Project/tree/main

Abstract

Each business has the most noteworthy obligation regarding picking the ideal individuals for the gig since the right arrangement of individuals can speed up business development dramatically. The technical team has a lot of big projects with big companies, so they don't have time to read resumes and pick the right one for their needs. The company always hires the third party to make the resume in accordance with the company's requirements to deal with this kind of issue. Hiring Service Organizations are the names given to these businesses. All that matters is the profile screen. Resume screening is the process of selecting the best candidates through coding competitions and other methods. For a very long time, software developers have struggled to create resume parsing tools that are accurate, effective, and capable of detecting all the information that recruiters require. We present a resume parser that makes use of natural language processing (NLP) to automate resume screening. There are a variety of resume parsers on the market, but their functionality makes them distinct. There will be many elements like schooling, abilities, experience, certification, and so forth. to evaluate the applicant. Our resume parser makes use of education and skills. Finally, we have developed a resume parser that extracts data from many unstructured pages with ease and precision.

1. Introduction

Numerous candidates apply for jobs on recruitment websites in today's large corporations and corporations. Every day, many resumes are screened by recruiters or departments of human resources. Because screening many resumes and selecting applicants for an interview takes a lot of time and can lead to errors due to human fatigue, this is not a job for humans. The format of an email, the content of a web page, and other data with defi, need, and others are different from the unstructured data found in resumes. The information on the resumes of applicants typically varies, as do the colors, fonts, order of presentation, and literary styles in which they are written. Additionally, resumes are available in a variety of file formats, such as ".txt," ".pdf," ".doc," ".docx," ".dot," ".rtf," and so on. Candidates typically use those file types. Thus, a computerized savvy framework considering regular language handling is expected to extricate all the data from unstructured resumes and different information sources. Converting all resumes into a similar structured format and selecting only the information that is relevant to screening, such as name, position, education, years of experience, work experience, certificates, email, phone number, and so on, is the method for parsing resumes. After that, the structured, parsed resume data will be saved in a database for later use.

You will learn how an end-to-end machine learning project is implemented to solve practical problems because this project involves resume parsing using machine learning and natural language processing (NLP). Our application will automatically parse resumes so that their keywords—experience, education, and skills—can be searched and filtered, saving you hours of scrolling through them. In this project, neural networks built with the Spacey library are used to build a model that can extract relevant fields like

location, name, and so on. from various resumes in various formats. Despite the abundance of text processing tools, the recognition and disambiguation of named entities are the primary focus of many. Demner-Fushman et al.'s MetaMap and Metapelite 2017) The two most widely used and supported tools for biomedical text processing are acronym recognition (NLP) tasks like the GENIA tagger and entity linking with negation detection (Tsoureki et al., new research innovations like word representations and neural networks are typically not used in phrase structure parsers like those presented in (McCluskey & Charmian, 2008) or 2005).

2. Objective

1) Using technology that is based on natural language processing to assist the human resource department in screening resumes before conducting interviews.

2) Parsing and matching the similarities between a candidate's resume and job description makes the hiring process easier and more efficient.

3) Help reduce human error and fatigue in screening resumes

3. Scope

Parse resume and matching resume to job description are the two functions of this system. The first function is to parse resumes. The user must upload a resume of the candidate file in PDF or DOC format. This project supports only PDF and DOC formats because they are the most popular for creating resumes nowadays. The system will read all text of the resume and the next part is to clean the resume text. From the cleaned text we extract only relevant data that is necessary for the selection of the resume: name (first Right now we implemented code to extract skills and education from the resume. The second function is to calculate the resume score for job descriptions to evaluate and filter the top candidates. The user can upload a job description file and see the displayed result as a percentage of similarity between the resume of the candidate and the job description. This system can reduce the HR department's time reading all text of a resume and reduce errors in the work.

The education, skills, and work experiences of candidates are essential types of information for recruiting by the human resource department. They also want this system to be able to rank or compare resumes to job descriptions provided by them to evaluate if there are any similarities. This will make it easy for them to work and make recruiting selections. As a result, where we must deal with a lot of data, converting a resume into formatted text or structured information to make it easier to review, analyze, extract relevant data, and understand is essential Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs and limit the use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

Table 1. Dataset details

DATA SET	NUM OF RESUMES	JOB CATEGORIES
TRAIN DATA SET	3446	48
TEST DATA SET	2490	24

4. Dataset

We have publicly available data from Kaggle.
We used 3 resume datasets Training Dataset: The training dataset is from Kaggle. Data is CSV format

https://www.kaggle.com/datasets/ Sneha Bhawan/resume-dataset?select=data

https://www.kaggle.com/code/ gauravduttakiit/resume-screening- using-machine-learning/data

Test Dataset: The test dataset is from a git repository

https://github.com/florex
/resume corpus (Jia Chie, 2021)
The dataset has a PDF format of resumes grouped by job category

The dataset of resumes has the following fields:

Location

Designation

Name

Years of Experience

College

Education degree

Experience

Professional Skills

Graduation Year

Companies worked at

Email address

Category	Resume
Testing	Computer Skills: â¯¢ Proficient in MS office (...
Testing	â Willingness to accept the challenges. â ...
Testing	PERSONAL SKILLS â¯¢ Quick learner, â¯¢ Eagerne...
Testing	COMPUTER SKILLS & SOFTWARE KNOWLEDGE MS-Power ...
Testing	Skill Set OS Windows XP/7/8/8.1/10 Database MY...

5.Techniques

NLP Tools and Techniques we used in this Resume Parser project is Spacey, NLTK, pandas for reading CSV data, slate3k for extracting text from pdf and python Here is an introduce to the exciting concepts we learned while building a python resume parser application system.

5.1Natural Language Toolkit

We used the nltk (Loper & Bird, 2002)library for NLP tasks such as stop word filtering and tokenization, parsing, and stemming. NLTK is a leading platform for building Python pro- grams to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

5.2Tokenization

It is the process of splitting textual data into different pieces called tokens. One can either break a sentence into tokens of words or characters; the choice depends on the problem one is interested in solving. It is usually the first step that is performed in any NLP project, and the same will be the case with this resume parser using an NLP project. Tokenization helps in further steps of an NLP pipeline which usually involves evaluating the weights of all the words depending on their significance in the corpus.

5.3 Lemmatization

The larger goal of this resume parsing python application is to decode the semantics of the text. For that, the form of the verb that is used does not have a significant impact. Therefore, lemmatization is

used to convert all the words into their root form, called ’lemma.’ For example, ’drive,’ ’driving, ’drove’ all have the same lemma ’drive.

5.4 Parts-of-Speech Tagging

If you consider the word” Apple,” it can have two meanings in a sentence. Depending on whether it has been used as a proper noun or a common noun, you will understand whether one is discussing the multinational tech company or the fruit. This CV parser python project will understand how POS Tagging is implemented in Python.

5.5 Spacey

Spacey(Neumann et al., 2019) The Python-based spacey library (Hannibal and Montani, 2017) 2 provides a variety of practical tools for text processing in multiple languages. Their models have emerged as the de facto standard for practical NLP due to their speed, robustness, and close to state-of-the-art performance. As the spaCy models are popular and the spaCy API is widely known to many potential users, we choose to build upon the spaCy library for creating a biomedical text processing pipeline. SpaCy is a library in Python that is widely used in many NLP-based projects by data scientists as it offers quick implementation of the techniques mentioned above. Additionally, one can use SpaCy to visualize different entities in text data through its built-in visualizer called display. Furthermore, SpaCy supports the implementation of rule-based matching, shallow parsing, de-pendency parsing, etc. This NLP resume parser project will guide you on using SpaCy for Named Entity Recognition (NER).

6. Methodology

This project implements Named Entity Recognition, a part of Natural Language Processing that analyzes large amounts of unstructured human languages. The initial step in extracting information and topic modeling is NER extraction. The system reads the whole paragraph and highlights the text’s key essential entity elements. Due to the resume text, being an unstructured text into predefined categories, you can utilize Stanford NER or Spacy for this project. Regular expressions have been used in this project, as well as regular expressions in scripts. A regular expression is a string of special characters that describes a search pattern by matching a character pattern to the string being searched. Regular expressions consist of literal symbols and special character combinations known as tokens, which indicate non-printable characters, symbols of a specific type, and the instructions for the regular expression engine. It is a formal language theory and theoretical computer science technique.

6.1 PDF and DOC to text conversion

This project uses slate3k library to convert PDF files to text format and python-docx library to convert Doc, Docx file to text format.

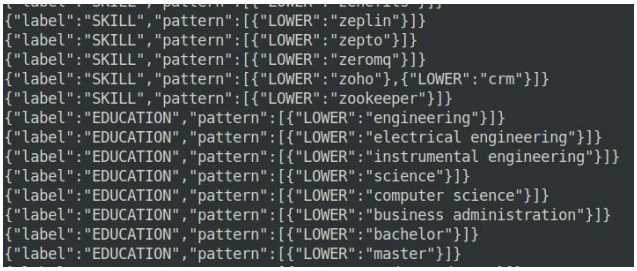
6.2 Cleaning text

```
resume_df2["Clean_Resume"][0]
```

```
'skill programming language python panda numpy scipy scikit learn
arning regression svm na bayes knn random forest decision tree bo
sentiment analysis natural language processing dimensionality red
abase visualization mysql sqlserver cassandra hbase elasticsearch
u others regular expression html cs angular 6 logstash kafka pyth
standing deep learning education detail data science assurance as
ung llp skill detail javascript exprience 24 month jquery exprien
ail company ernst young llp description fraud investigation dispu
ar technology assisted review assist accelerating review process
lped developing automated review platform tool scratch assisting
ng topic modelling automating review resulting reduced labor cost
solution research development classification model predictive ana
analyzing output precision monitoring entire tool tar assist pred
ey standard developed classifier model order identify red flag fr
earn tfidf word2vec doc2vec cosine similarity na bayes lda nmf to
matplot lib tableau dashboard reporting multiple data science ana
icle customer review data received customer feedback survey data
ve neutral time series analysis customer comment across 4 categor
quency word extracted positive negative word across survey catego
dashboard effective reporting visualization chatbot developed use
tion hour operation reservation option chat bot serf entire produ
latform also give recommendation response user question build cha
estion per user requirement asks relevant recommended question to
nltk spacy topic modelling sentiment analysis word embedding scik
governance organization make informed decision information store
```

First step is cleaning data by removing stop words that are a group of words that are regularly used in a language but contain relatively little valuable information, including punctuation on all text of resume. Stop words are the words like ’a’, ’the,’ ’am’, ’is’, etc., that hardly add any meaning to a sentence. These words are usually deleted to save on processing power and time. In their CV, an applicant may submit their work experience in long paragraphs with many stop words. For such cases, it becomes essential to know how to extract experience from a resume in python, which you will learn in this project.

6.2 Named Entity Recognition (NER)



Extracting key information from the resume such as skills, experience and education is essential. This project uses the Json file format in the train dataset. The Json file is loaded to spacy module as a custom entity ruler. Then, the custom entity uses Named Entity Recognition (NER)(Lample et al., 2016) for training model because this project is a finding and classifying text of the resume that is an unstructured text into predefined categories by tagging the dataset.

Table 2. Results for top 3 Data Science resumes

TOP	RESUME	MATCH SCORE
1	12011623.PDF	71.4
2	10624813.PDF	42.9
3	27152464.PDF	28.6

```
def get_skills(text):
    doc = nlp(text)
    myset = []
    subset = []
    for ent in doc.ents:
        if "SKILL" in ent.label_ :
            #print ("ent.label_ ", e
            subset.append(ent.text)
    myset.append(subset)
    return subset


def unique_skills(x):
    return list(set(x))
```

Extracting degree or educational background by using key- words such as Bachelor of, Master of, Doctor of, Degree, etc. After that, searching for all the characters around those keywords. Extracting skill. using keywords such as data science, machine learning, python, tableau, etc. Then, search for each token in the skills database.

	Category	Resume_str	Clean_Resume	skills	education
0	Data Science	Skills * Programming Languages: Python (pandas...	skill programming language python panda numpy ...	[bot, visualization, mysql, plotly, dimensiona...	[associate, associate, analytics, analytics]
1	Data Science	Education Details \r\nMay 2013 to May 2017 B.E...	education detail may 2013 may 2017 b e uit rgp...	[outlier, feature selection, github, ml, dimen...	[business]
2	Data Science	Areas of Interest Deep Learning, Control Syste...	area interest deep learning control system des...	[data analysis, debian, jupyter notebook, mysq...	[electrical, electrical, business]
3	Data Science	Skills â¢ R â¢ Python â¢ SAP HANA â¢ Table...	skill r python sap hana tableau sap hana sql s...	[data flow diagram, server, segment, visual st...	[analytics, analytics, analytics, management, ...
4	Data Science	Education Details \r\n MCA YMCAUST, Faridab...	education detail mca ymcaust faridabad haryana...	[data analysis, data structure, data science, ...	[management]

7.Result

The proposed system's results are shown in this part Table 2, which include extracting skills, education, like, degree, skills, experience. Another feature of this system is that it compares the Resumes and job description of the applicant. The similarity of the outcomes is expressed as a percentage. We tested resumes from test data in ENGINEERING with required skills as 'Data Science, Data Analysis, Database, SQL, Machine learning, Python, tableau' and got top 3 resumes. Fig. 6, show the entire system's results

```
Top 0 resume is dataset/data/ENGINEERING/12011623.pdf with match score : 71.400000

        Skillset of this resume is :
                ['testing', 'machine learning', 'shrinkage', 'schedule', 'material', 'knowledge base', 'data mini
ng', 'analytics', 'tableau', 'database', 'support', 'oracle', 'design', 'algorithms', 'visualization', 'linear reg
ression', 'collaboration', 'python', 'business', 'segmentation', 'engineering', 'database design', 'software', 'da
ta warehouse', 'data analysis']


Top 1 resume is dataset/data/ENGINEERING/19396040.pdf with match score : 42.900000

        Skillset of this resume is :
                ['java', 'python', 'gauges', 'engineering', 'documentation', 'image quality', 'software', 'materi
al', 'robot', 'data analysis', 'schedule', 'visual basic', 'database', 'industrial engineering']


Top 2 resume is dataset/data/ENGINEERING/50328713.pdf with match score : 28.600000

        Skillset of this resume is :
                ['python', 'linux', 'computation', 'machine learning', 'design', 'engineering', 'mechanical engin
eering', 'petroleum engineering', 'statistical model', 'libraries', 'tensorflow', 'medium', 'pandas', 'bash', 'sch
edule', 'regression analysis', 'simulation', 'support']
```

8. Limitation

Because of the data extraction limitations, it includes some data that cannot be processed, such as the year of graduation and date of birth, which makes it difficult to determine which class it is because the resume mentioned many dates or years. In addition, there is not enough dataset in this project, and the information extracted does not cover all the details of the resume, such as education. It can only retrieve a little amount of data that is closely connected to the word" education." As a result, data retrieval problems are possible. Resume parsing is also sensitive to ethical restrictions. Be- cause of this system, the result will be a text input only. As a result, this approach is only suitable for screening some positions. For example, a graphic designer position or other design positions that require a visual preview of the work, an image as evidence of work, and consideration of the resume's beauty and color may not be appropriate for this system. This system's bias appears to be causing firms to lose employees.

9.Conclusion

Because the online recruiting system has progressed, many resumes were submitted. Consequently, hiring new employees and reviewing many resumes is a challenge for the human resource department or employer. Therefore, this system has helped employers by using an automated intelligent system based on natural language processing. This system can convert various formats of resumes to text format and can extract some important information successfully. It is also possible to compare the applicant's resume and the job description to see the percentage of similarity as well. This system can assist the human resource department or employer in screening resumes before conducting interviews and finding the best candidate for the job position.

10. Further Development

This project intends to provide more datasets for training in the future because the existing datasets are insufficient for applications such as education, university, skill, etc. For future website development. This project will apply the model to the website and add a function to view the applicant's resume file or portfolio if the employer or human resource department is interested. To support the selection of resumes

in all positions. After the user confirms this candidate, the resume is saved in a database to be used as a future dataset, with the resumes being ranked based on the percentage of similarity between the applicant's resume and the job description.

References

- Demner-Fushman, D., Rogers, W. J., and Aronson, A. R. Meta map lite: an evaluation of a new java implementation of meta map. *Journal of the American Medical Informatics Association*, 24(4):841–844, 2017.
- Jiechieu, K. F. F. and Tsopze, N. Skills prediction based on multi-label resume classification using CNN with model predictions explanation. *Neural Computing and Applica- tions*, 33(10):5069–5087, 2021.
- Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.
- Langley, P. Crafting papers on machine learning. In Langley,
- P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stan- ford, CA, 2000. Morgan Kaufmann.
- Loper, E. and Bird, S. Nltk: The natural language toolkit.
- *arXiv preprint cs/0205028*, 2002.
- McClosky, D. and Charniak, E. Self-training for biomedical parsing. In *Proceedings of ACL-08: HLT, Short Papers*,
- pp. 101–104, 2008.
- Neumann, M., King, D., Beltagy, I., and Ammar, W. Scis- pacy: fast and robust models for biomedical natural language processing. *arXiv preprint arXiv:1902.07669*, 2019.
- Tsuruoka, Y., Tateishi, Y., Kim, J.-D., Ohta, T., McNaught, J., Ananiadou, S., and Tsujii, J. Developing a robust part-of-speech tagger for biomedical text. In *Panhellenic Conference on Informatics*, pp. 382–392. Springer, 2005.