Resume Parsing using NLP and Python

Define requirement.

To start with resume parsing, need to define libraries given below few important libraries are.

- Regex (re)
- Pdfminer
- Spacy
- Fitz
- Bert (summarizer)
- Textdistance and difflib
- Pandas and numpy

Define the requirements for user.

For user given below parameters are important and critical parameter for resume parsing and resume screening.

- Skills
- Education
- University
- Name
- Phone Number
- Email Address.

For given code of resume parsing, here three parameters defined that is skills, education, and university. This need to be pre-defined as user should know its requirements what skill set looking for, what sort of education is important and any specific university like Tier-1 or Tier-2 etc.

For assignment here I covered these three parameters from user perspective.

Language Used

Python and IDE is Jupyter notebook.

Parsing Data and input/output stored in format of xlsx and csv.

Data Wrangling

1. Starting with user input, will read all the user requirement. Before that will take all input into different sheet of workbook of excel file named as (data scientist skill), It have three sheets named as Skills, Education and University.
2. Now starting with .pdf file reading from the folder where all resume is stored. Defining the functions for different user defined parameter and few default parameters such as contact details, email address, skills, education, name , university .
3. Used looping to extract all .pdf files from the defined folder. And extract all data from each file into text format and after extracting data, will pass to each function of desired extraction mechanism such as university extraction, name extraction etc.
4. To extract text from .pdf here I used "pdfminer" library. To extract matching pattern from the resume as per user requirement I used regex pattern matching method. To match pattern of email, contact details skills and education etc.
5. Since university name have some name conversion problems like short name full name, for that I used close *match method* it basically looks for closest match pattern having higher cut off percentage, suppose standard for matching collage is defined that if 75% name is matching show the result of collage from resume.
6. Similarly, for name extraction method we defined grammar noun of name proper noun and all other pattern to match first name middle and last name. Here we used "Spacy python library". Loaded with 'en_core_web_sm' , basically this method will use English dictionary to extract all name pre-defined and this is trained model which help in finding new name out of given text. *It best work on English name*. To make it more reliable we defined pattern matching along with spacy library.

Function calls

1. After defining all function and applying all logic inside the function, it's time to call the function one by one and save the result for all the functions in form of list.
2. Creating an empty data frame and defining the columns for each parsed function and passing the list of each parsed value into that specific column.

BERT model

1. After passing all value to data frame, a proper list of passed data is ready. After that will use "BERT model" method for resume summary. Basically, this method is used to extract whole resume for each candidate and make a short summary or we can say gist of the resume in few lines instead of reading whole resume.
2. This BERT model have function called summary() , which will take all text and we can set parameter of maximum character we want to create summary and we can get done with summary within those number of words.
3. Will put summary of each resume into defined data frame.

Resume Screening

- Basically, Resume screening is first step before resume parsing, so basically in this method, will check how well resume matching with the requirements of user. Suppose user have 10 requirement and from resume its 6 are matching, then it means resume is 60% fulfilling the requirement of user. For resume screening will pass all the requirement into a text format and will you "text difference" library to check how well resume matching with the resume to match with requirements.
- On basis of resume matched percentage user can take decision what cut-off should be good to go with for resume selection.

NLP Based Summary generator like BERT model.

- Alike BERT model there is method in NLP with spacy to generate summary of whole text into short form. This is just another approach I am mentioning in the resume parsing. After generating short text from spacy will append it into data frame.

Save

- Once all the details are available in data frame and all requirements are full fill, will save the data frame into workbook, it can be in csv or excel format. Here I am exporting in xlsx (excel) format.

Possible Optimizations

1. While extracting text, it can be more viable and more effective solution if its text data get clean and more readable.
    a. To make text more readable, here it can use Lemmatization with parameter "**WordNet Lemmatization**".
    b. Also, for more understanding of part of speech in paragraph, **TF-IDF** (Term frequency and Inverse document frequency) method can be used to make text of resume more reliable and more readable.
2. While Extracting data from pdf, few data points are not possible to extract as, when resume created using some resume creating tools it past the details into resume in form of image which are not possible to extract from the pdf-based resume, to do that we can use "**OPENCV**" python library very powerful for text extraction out of images that is **OCR** (Optical character recognition).
3. Resume parsing can be done for different format like (docx, text, image format, rtf) etc. using different library of python.
4. Furthermore, parameter can be extracted like GitHub link, website, year of experience etc. as these task are similar to other extracted task so not mentioning and extracting.

Limitation

1. For this assignment work on PDF based resume not for all format of resume like (docx, text, image format, rtf). This limitation is not permanent, and it can be moderately achieved and is possible to solve.
2. For given assignment is it not fully functional NLP based resume parsing, but this can be achieved but it needs better configuration of Windows laptop as I have MacBook few libraries not working very well with my laptop like "spacy", "TensorFlow" as M1 chip have limited support as of now and my Window laptop is not that powerful to perform these tasks.
3. If handled with all different format of resume it will work well, but if any new format pops up, it will give result, but its efficiency will reduce as it need to handle those new cases as well. But still most of the possible cases can be achieved with it.
4. Extracting names from resume can never be possible 100% as NLP model is best trained on English world vocab so it will work best on English names but will not be efficient on "Hinglish" names.

Note

1. Here for given assignment I mostly played around regex library and text comparison library followed by Bert model. It can be further enhanced with more powerful NLP library, since this is a prototype of resume parsing and time consuming, I am performing limited task in it.
2. Before running code make sure all Libraries are installed and all directory folders are correctly assigned to fetch data and to save output data.

Thanks

Pawan Kumar Rai