Rohit Rangarajan

April 1, 2019

[rrangara@usc.edu](mailto:rrangara@usc.edu)

**Resume Summarizer Assignment**

**About the project:**

The project is to generate a summary of important information in a candidate’s resume, along with its score.

Another option is to rank a list of resumes, according to their scores.

This has been implemented in python version 2.7 using PyCharm and tested on ubuntu.

**The different stages in the project workflow:**

1. Building a model from annotated resumes. Please note that this logic has been entirely taken from an existing code base

<https://github.com/DataTurks-Engg/Entity-Recognition-In-Resumes-SpaCy>

2. Text extraction from the input resume file. The file format could be pdf, txt or docx.

pdf —> using Tika (or) PyPDF2

docx —> using docxpy

3. Extracting the sections from the text coming from step 2. This uses the model built in step 1.

4. Scoring the resume and summarizing.

**Class Design**

**1. Model building**

This is just refactoring of existing code.

**2. Text extraction**

**TextExtractor:**

1. Txt2TextExtractor —> takes text document and returns its text

2. PDF2TextExtractor —> takes PDF document and returns its text

This is just a wrapper over the below classes

- TikaPdfExtractor

- PyPdfExtractor

3. Docx2TextExtractor —> takes docx document and returns its text

**3. Sections extraction**

**SectionsExtractor:**

Given a model and the resume text, this returns the different section names and the corresponding values

**4. Sections parsing**

**SectionsParser:**

This works on the sections extracted from the resume text, passing the same to spaCy’s NLP model and get noun chunks and entities. This eventually helps in the scoring phase.

**5. Scoring**

**ResumeScorer:**

This is an abstract class and provides the framework for interesting scoring strategies.

**JsonBasedScorer:**

This uses a json file that contains ‘Rules’ for scoring across sections and computes the final score based on the extracted sections and the matching rules. Please note that the parameters in the json file do not cover all sections and combinations. It is only to show how this works.

**6. Resume processing**

**ResumeProcessor:**

This is the main class. It provides the interface for ranking resumes, as well as generating summary for a specific resume. This is the top-level class for this project.

**Limitations:**

1. The current model for extracting sections from resume is very fragile, mainly because of less training data. So it may not work with many resumes. To make your job simple, the code works with the resumes available in the distribution (resume\_format\_samples)
2. The current scoring scheme uses a json-based configuration. Although it is possible to generalise this, I have only used some sections of the resume to illustrate its use.

**Installing and running**

**Module dependencies:**

docxpy

PyPDF2

tika

spacy

The above can be installed using:

pip install <package>

For spaCy, we need the ‘large’ model. Install it using:

python -m spacy download en\_core\_web\_lg

**How to run:**

Extract the zip file. Directory structure should be like this:

ResumeProcessor

resume\_samples

resume\_model

src

<python and json files>

test\_results

‘test\_results’ directory is used only during training and model generation.

Go to “src” folder and do the following:

python resume\_processor.py -summary “../resume\_format\_samples/resume\_text0.txt”

(or)

python resume\_processor.py -rank “../samples\_for\_ranking”

The first command summarizes the given resume.

The second one outputs scored and sorted resumes in descending order.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* THE END \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*