Resume Parser

Author: Sid Tewari | go/LLcPython

## Executive Summary

Using Python, build an application to help job seekers improve their resume for ATS parsing, essentially:

1. Parsing resume and job description
2. Creating a word-cloud, and comparing the two for keywords
3. Presenting action items on how job seeker can improve their resume to increase chances of surfacing it in an ATS

## Background

Beginning Friday July 27th, the GCO LinkedIn Learning Collaboration (LLC) meeting was used to learn Python via a free [Udacity](https://www.udacity.com/course/programming-foundations-with-python--ud036) course. To apply the skills we have learned, and to upskill our Python knowledge, we’ve decided to build an application.

## Core Team

Amir, Aris, Cathy, Clinton, Conrad, Hideki, Sid, Tria

## Product Features & Complexity

|  |  |  |  |
| --- | --- | --- | --- |
| **FEATURE** | **COMPLEXITY** | | |
| **Simple** | **Medium** | **Hard** |
| Accept Resume | Text file | Word Document / copy-paste | PDF, URL |
| Accept Job Description | Text file | Word Document / copy-paste | PDF, URL |
| Parse Input | Count frequency of all words in the input, barring certain hard-coded words | Give user option to exclude words in the input | 1. Intelligently exclude filler words that are not important 2. Figure out root-word for count |
| Generate & Display Word Cloud | Show simple text based output of top ‘n’ keywords | Use an existing word cloud generator to generate word cloud and get it’s output | Show output similar to wordle or other word cloud generators |
| Create Match Score | Calculate match score based on top n words for each input. | Allow users to select words in job description which we can use to calculate score | Calculate score based on relevant keywords in job description |
| Give Feedback on match score | Text based feedback to increase frequency by ‘x, y, z...’ times for each word that is ‘1,2,3...n’ in job description | ... | ... |
| Security | N/a |  |  |
| Application | Local machine, text based output, no user interaction, runs from command-line | Executable file for Windows or OS X. | Published on a website |

## Next Steps / Lesson Plan

1. How to collaborate on code?
   1. We are using GitHub to collaborate.
   2. Cathy did a tutorial on GitHub, which can be accessed here: <https://bluejeans.com/s/bj9Y1/>
   3. The related deck can be found [here](https://docs.google.com/presentation/d/1zoo57s9xC4LrEb4CEeOKorJNG-hdxATM4Zv55W_aoMA/edit#slide=id.p).
2. How to start?
3. Vectorization library (suggested by Ben for word cloud)

## Appendix

* Word cloud generator: <http://wordle.net>
* Company that already has this as a product: <http://jobscan.co>