Nirmalendu Prakash

**Personal contribution**

I worked on the NLP engine for the solution, which contains:

* data extraction from resume
* data extraction from job postings
* deriving applicant desired job type and job type from job posting
* Finding out most relevant applicant work experience and extracting most suitable sentences from it to be used in cover letter.
* Creating templates for cover letter
* Generating cover letter

**Learning**

I explored many different ways to approach this problem, namely:

Keyword based-

All job requirements mention skills required. For a technology job, this can be a technology or programming language such as python, java, r etc. For a teaching job, it can be subjects such as maths, science etc. If the job mentions “programming experience on machine learning tools such as python etc.” Here if the applicant has mentioned r or matlab experience, this should still be a match as these technologies are related. An ontology map solves the problem of identifying similar work experiences but when I browsed through a list of softwares online, I realized that a software exists for any English word. So any random word in job requirement can be mistaken as a software.

NLP based approach 1-

After discarding ontology based approach, I considered matching every job posting with user resume and find the most matching sentences and use it to generate letter. This can create noise for the user, as there could be many irrelevant job mails. Also, the solution can evolve to be goal based instead of being reactive i.e. instead of waiting for job mails, the RPA agent can crawl the internet for job posting. This would result in a lot of noise. So a way has to be found for filtering irrelevant jobs. Therefore, each resume and job posting has to be classified.

NLP based approach 2-

A method has to be devised for classifying job types and resume types. For this, a dataset has to be created for each resume type and for each job type. Creating such a dataset manually can be time taking. The automated approach is described in detail in the NLP section. Also, the approach should be extensible without need of a code change.

The approach used only needs addition of new types in job\_type.json and executing the script using a simple command:

Also, after creating a dataset and going through few of them, I realized that a job posting may contain information about the company, business unit etc. which is not useful for filtering and can skew the scores. So the next task is to identify which are the relevant blocks of information in the job post-namely the roles\responsibility section and requirement section.

A separate dataset was created for role and requirement section and each block from a new jo post is matched against each job role and requirement in the dataset and using an average score and a threshold, the irrelevant blocks of information from job post is discarded.

A similar approach is adopted for data extraction from resume.

It has been a great learning exploring various ways to achieve the objective. I came to know about the pros and cons of using 1 approach versus the other. I would learn about generative NLP and perhaps get rid of the need of a template for letter generation.

Also, I enhanced my knowledge about web scraping tools and how to effectively use them.

**How will you apply this knowledge and skills in other situations?**

I will read about generative NLP and knowing the pros and cons of a template based approach will help me. I will also explore statistical models for speech to text conversion and see how I can improve the existing solution.