**PSG COLLEGE OF TECHNOLOGY**

**Department of Applied Mathematics and Computational Sciences**

**IX Semester M.Sc. DS 2020 - 2021**

**15XD96 Information Retrieval Lab**

**Package Abstract**

**Phase 3**

**Title:** Resume Short-listing

**Roll No:** 16PD22

**Name:** Nehal M

**Problem Statement:** Many institutions and companies use resumes as a way of short-listing candidates in preliminary rounds. Going through every resume and short-listing is an arduous process and in many instances, it is prone to human errors. This project aims to automate this short-listing process by retrieval of relevant resumes to a given job description(JD).

**Dataset:** Resumes and Job descriptions are collected from indeed.com and linkedIn. Also job reviews are scraped from ambitionbox.com. Also data related to the selected candidates are available which is used for further analysis.

**Tools:**

Django framework is used to build the user interface.

BeautifulSoup in python is used for scrapping documents from websites.

NLTk and Spacy libraries are use to process documents.

PDFMiner.six is used to convert pdf to text.

**Research Implementation Details:**

In phase 1 and phase 2 we build a basic search engine to retrieve the resumes which matches the JD.

In Phase 3 we improved the model by working on 3 major aspects: Vectorization, Information extraction and an ensemble ranking approach.

**Vectorization:**

* Words are vectorized using GloVe vectorization.
* GloVe model is built by taking both global statistics and concurent word similarities.
* So it is a better fit than the tf-idf approach.

**Information Extraction:**

* For the information extraction part we used Named Entity Recognition to extract skillset info from resume.
* For this we built a custom corpus and trained our own model so as to suit the context of our dataset.
* This will help us to capture better features.

**Ranking:**

* Finally for the ranking we used a combined strategy of classification and ranking.
* Classification is sone using Naive Bayes, SVM and RNN models and for ranking BM25 and cosine similarity measures were taken.
* For a ranking, documents are chosen based on max voting of the classifiers and then the selected documents are ranking based on the average of the 2 ranking scores.

**Evaluation:**

* Models are evaluated using precision @R
* For the base model we got precision@R value of 32%
* After adapting research paper and implementing them, we improved precision@R to 71%.

**Paper Links:**

* GloVe: Global Vectors for Word Representation

<https://nlp.stanford.edu/pubs/glove.pdf>

* Intelligent Hiring with Resume Parser and Ranking using Natural Language Processing and Machine Learning

<http://www.ijircce.com/upload/2016/april/218_Intelligent.pdf>

* Neural Architectures for Named Entity Recognition

<https://arxiv.org/pdf/1603.01360.pdf>

* Improvements to BM25 and Language Models Examined

<http://www.cs.otago.ac.nz/homepages/andrew/papers/2014-2.pdf>