Overview

This application aims to extract skills from jobs description. Given a job description, the systems will first recognize skills from the text and then find implicit skills which are not mentioned in the requirement. The output will consist of extracted skills, related skills, and trending skills.

The system includes two modules, called skill extractor and implicit skills extractor.

Skill extractor

(To be done by TGN)

Implicit skill extractor.

1. Overview.

Implicit skills are skills that are not mentioned in the job description. For example, a job description for technical consultant may not mentioned specific technical skills which are helpful for this job. Determent implicit skills will help suggesting related skills or recommending related jobs to users. This can be helpful for matching resumes and job's descriptions as well. Besides, implicit skills are needed in inference systems to infer useful skills for students in educational systems.

Topic model is chosen because of the nature of a job description. Nowadays, numerous jobs require cross functional skills which belong to different fields. A job itself can be belong to multiple industrial sectors. For example, an engineer in fintech industry should acquire both technical and financial skills. Topic model can satisfy this requirement because it represent the coverage of different topics in text documents. This is similar to field coverage in a job description.

2. Implicit skill extractor module.

This module depends on skill extractor module to find implicit skills in a job description. Usually, those skills will be listed in similar job requirements which belongs to the same industrial sector or researching fields.

Based on that, LDA model is used to transform and determent the topic coverage of job descriptions. Topics can be working areas or research domains such as engineering, software, or finance. The number of topics is chosen by grid search algorithm to yield the most optimal parameters.

The top ̀ most similar jobs of the query will be retrieved based on the distance of topic distribution vectors.

Implicit skills are then extracted from those jobs and added to the query.

The model is built using Scikit Learn, a python library. Firstly, job's descriptions are loaded from dataset and then cleaned by removing punctuation. Those descriptions are then transformed to vectors of term frequencies. In this process, words are converted to lowercase and stop words are filtered based on the existing English stop words list in Scikit Learn. After that, the vectorized documents are fed to a Scikit Learn module to train the model.

The model is tuned using grid search, an algorithm used in Scikit Learn. It receives the data and a grid of parameters and then builds multiple models using all combination of given parameters. The best one is then selected by comparing the score of the models. This score is calculated based on the log likelihood of LDA models. The grid of parameters is a dictionary containing parameter's name and lists of values. Below is an example:

{

'n\_component’: [5, 0, 15, 20],

'learning\_rate’: [0.5, 0.75]

}

In this project, the number of topics, 'n\_components' is the only parameter for tuning purpose.

After finding the best model, it will be used for transforming the given job description.

The given description will go through the similar process consists of vectorization and transformation. After that, similar descriptions are retrieved based on Euclidean distance. The smaller the distance, the more similar the documents. The top five similar texts are then chosen and used for extracting implicit skills.

Skills extractor module are called to get the skills from the texts. Those skills will be added to the list of skills found in the given job description.