COSC 329

Project option 4:

Scraping indeed.com to gather a list of skills that are in demand for a specific job title.

Name: Jayant Puri

About the project:

I chose to go with project option 4 where I coded a program that scrapped 10000+ job postings on indeed.ca. The job titles I chose were “web developer” and “Data analyst” and the location chosen was Canada.

Files created in the project:

1. extract\_skills.py: This file is where the I scrape Indeed for over 10000 job postings. I also extract the skills from the job posting and save it in 2 files. Those files are:
2. data\_analyst.json: This file contains the skills from over 5000 data analyst jobs stored in the list format.
3. dat.json: This file contains the skills from the job title “web developer”. Scraped over 5000 job postings.
4. analyze\_skills.py: This is the file where the skills that are sored in the 2 json files are read, preprocessed to get cleaner data. Also the top skills in for the each job title is calculated here and the similarity between the job titles is also calculated and displayed in this file.

Original List of tasks to be completed for this project:

1. Get a web response from indeed.com website for a given location and job title.
2. pick the URL of one job posting out of several results
3. Extract the list of skills in that job posting.
4. Use word tokenization on the skills from nltk library.
5. Remove the stop words from the list of skills (done with nltk).
6. Use lemmatization to get the core word
7. Use part-of-speech tagging on the remaining list of words.
8. Isolate the words that are ‘tagged’ as nouns, adjectives (or maybe more tags can be considered or removed after further testing) and label them as ‘skills’
9. Create a dictionary where each ‘skill’ will be the key with the frequency with which that skill appears as the ‘value’.
10. Update the dictionary for over 10,000 different job postings times to get the result.

Modified list of tasks that were completed in this project:

1. Get a web response from indeed.com website for a given location and job title.
2. pick the URL of one job posting out of several results
3. Extract the list of skills in that job posting.
4. Use word tokenization on the skills from nltk library.
5. Use lemmatization to get the core word
6. Use part-of-speech tagging on the remaining list of words.
7. Isolate the words that are ‘tagged’ as nouns (nn) and (nnp) and also attach the word that comes prior to these words to make a bi-gram. Eg: “NodeJs framework”
8. Store all the skills for a particular job title in a json file.
9. Read the json files and create 2 dictionaries (1 for each job title) where each ‘skill’ will be the key with the frequency with which that skill appears as the ‘value’.
10. Calculate the similarity between the dictionaries (2 job titles) with cosine similarity

Working Features:

1. Scraping 10000 job posts
2. Extracting skills from each job posts
3. NLP libraries work as intended to preprocess the skills and remove noise from the data. Tokenization to separate each individual word, lemmatization to get the root word, so that no skill is duplicated and finally pos tag to extract the relevant skills from the list and storing it in s json file.
4. A dictionary of skills for each job title with the frequency with which they occur also works as intended.
5. Cosine similarity to compare 2 different jobs works as intended.

Status of the project:

The project runs as intended, there are no bugs to report. However, there are some limitations with the project: Scraping indeed.com for job postings requires the project to be run multiple times because Indeed has a captcha that’s implemented which activates after scraping 150 jobs. To work around this, I changed my IP address multiple times via a VPN to get all the data. Another limitation of this project is that the skills that are extracted from a job posting need to be in a list format, that is if the skills that are listed is in a paragraph form than my code will not detect those skills.