Advanced Database Systems

Project 1

Team members: PING-FENG LIN, pl2730, Weihan Chen, wc2681

- List of the file: ISE.py, Readme.pdf, transcript.pdf
- Description of how to run the program:

1. Under the directory enter: python3 ISE.py <1 or 2or 3or 4> <Threshold > <Query><k>
2. Install all the dependent package (all the package mentioned in the instruction file, and bs4)
3. See the result

- Internal Design of the project:

First we set up the two pipelines(pipeline1 with parameters(`'tokenize', 'ssplit', 'pos', 'lemma', 'ner'`),pipeline2 with parameters(`'tokenize', 'ssplit', 'pos', 'lemma', 'ner', 'depparse', 'coref', 'kbp'`),and then use the Google Search Engine to obtain 10 URLs.

We get the text of the 10 URLs by using bs4 package and we use the pipeline1 to select the sentences with the right composition. And then use pipeline2 to extract the required relations from these sentences.(More details will be instructed in next part.)

- Details of step 3:

1.We use bs4 package to retrieve the corresponding webpage, and use “try,except” to avoid timeout.

2.We use pipeline1 to select the sentences with the right composition and store them in a list. i.e. “per:schools attended” relation need “PERSON” and “ORGANIZAITON”, so we can check every sentences to find out if the tokens in this sentence contained “PERSON” and “ORGANIZAITON”. If it contains, we will reconstruct the sentence and store it in a list.

3.Then we use pipeline2 to parse all the sentences in the list and we will get kbpTriple attribute of each sentences. We check if these kbpTriples satisfied our requirements(relation and confidence) and store them in a dictionary.

4.We can sort the dictionary by its confidence value and then check if the length of the dictionary is longer than k. If not, we can do it again by replacing the query with the first element of the dictionary.

- Search Engine ID: 000568897140034501999:kyspcqzrszq

- JSON API key: AIzaSyBKlE390a0krI3_Oe-zk-_Pdf7J0J7Oo1I