Advanced Database System - Project 1

Group 2

Name	UNI
Li-Chieh Liu	ll3123
Qing Lan	ql2282

Files List:

proj1-stop.txt - Our stopwords requirement.txt - Dependency package Search.py - Our main program
DataClean.py - Remove stop words and symbols
Rocchio.py - Modified Rocchio algorithms

Google Search API

Engine ID: 014170202143592210537:4zb34sjofuu
JSON API key: AIzaSyBz-iFhhFx_sQSBMxKBMh9d5ZjD2nyQtLw

How to run our program

do `pip3 install -r requirement.txt`
do `python3 Search.py <JsonAPIKey> <CSEKey> <precision> <query>`

Internal design of your project

The overall design can be conclude in the following steps:

The input

The search title and snippets are chosen as the input. The requests library are used to do the HTTP GET call to obtain the search result. Query sentences would be used here to find the proper result. We also store the user's decision on whether the reuslt is relevant or not which would further used in the query modification method.

Step 1 Sentence Cleaning and Stopwords Removal

We use a regular expression to remove all Symbols, which including . , ? * ... After this step, we use the provided stopwords list to remove all stop words. After finishing up the cleaning part, a list would be provided to do the vectorization.

Step 2 Word Dictionary and Sentence Vectorization

A word dictionary would be created to do the vectorization after the cleaning step. Thes words

would be filtered to remove duplicates and then sort in an order. We use a list with that order to store the occurance of each word in the documents. After this step, we successfully obtained a sentence vector.

Step 3 Query Modification and Search Again

Apply the Query Modification Method to the vectors using the sentence vectors and relevant labels. The method would provide the next word in the query and then we do the search again and move back to step 1. If the precision meet the requirement, the program would simply return.

Our Query Modification Method

We basically use the following formula - a moodified Rocchio algorithms

$$\overrightarrow{Q_m} = (a \cdot \overrightarrow{Q_o}) + \left(b \cdot \frac{1}{|D_r|} \cdot \sum_{\overrightarrow{D_j} \in D_r} \overrightarrow{D_j}\right) - \left(c \cdot \frac{1}{|D_{nr}|} \cdot \sum_{\overrightarrow{D_k} \in D_{nr}} \overrightarrow{D_k}\right)$$

Dj: Relevant Document Vector
Dk: Irrelevant Document Vector
Qo: Original Query Vector
Qm: Voted Query Vector

Here we set:
a = 0 (since we cannot delete words in the original query)
b = 1
c = 0.5

We add one word at a time, the new word would be the highest voted word in vector Qm.
We create a long vector, each entry is a word appear in the top 10 search, if a document is marked as relevantm, we upvote the words in this document by weight b = 1, if it is irrelevant, we downvote the the word in this document by weight c = 0.5, and therefore comes out an vector Qm, we sort Qm to get the highest voted word and add to our new query.