README

Name: Wenjia Zhang
Columbia UNI: wz2647

List of files:

main.py
relevance_feedback.py
README.pdf
transcript.txt

How to run. Note that your project must run in a Google Cloud VM that you set up exactly following our instructions. Provide all commands necessary to install the required software and dependencies for your program.

Answer:

1. Upload code to VM on local machine : (please replace '/Users/zhang/PycharmProjects/cs6111-hw1/main.py' with your source)
gcloud compute scp /Users/zhang/PycharmProjects/cs6111-hw1/main.py wz2647@cs6111-instance:~/
gcloud compute scp /Users/zhang/PycharmProjects/cs6111-hw1/relevance_feedback.py wz2647@cs6111-instance:~/

2. Install on VM:
sudo apt-get -y update
sudo apt-get install python3-pip
sudo apt install python3-testresources
pip3 install --upgrade google-api-python-client
pip3 install numpy
pip3 install scikit-learn

3. Run on VM:
python3 main.py --query "per se"
python3 main.py --query "wojcicki"
python3 main.py --query "cases"
python3 main.py --query "Information Retrieval pdf"
(I use it to test if checking html and non-html file works well)

Description of the internal design of your project, explaining the general structure of your code (i.e., what its main high-level components are and what they do), as well as acknowledging and describing all external libraries that you use in your code
A detailed description of your query-modification method (this is the core component of the project); this description should cover all important details of how you select the new

keywords to add in each round, as well as of how you determine the query word order in each round.

Answer:

My project first uses api key, engine id, precision and query as input. The project is divided into two main parts, one is user search and the other is relevance feedback. The first part is simple. I use the four inputs to retrieve top 10 results from Google.

Then I enter feedback part. I use fileformat to classify it is html or not. If it returns nothing in fileformat, I can ignore it. Every time one result I don't ignore is retrieved, I need to check it is related to my query or not. If they are relevant, then I will mark the result as 'relevant'. Meanwhile, I count the number of the result I don't ignore and the number of relevant.

Current precision is the number of relevant divided by the number of the result I don't ignore. If Current precision is larger than input precision or equals to 0, the code breaks here. Importantly, if current precision is between 0 and input precision, I make input query, result title and result description together. I use TfidfVectorizer to remove stop words in English. (Sorry, I don't use stop words Professor provided on the webpage).

Then, I get document-term matrix and a mapping of terms to feature indices, and calculate a mapping of feature indices to terms. Then I calculate tf-idf vector using Rocchio algorithm. Reasonable values are 1, 0.75, 0.15, provided by the textbook on page 183. I choose two terms with first and second biggest numbers from matrix after Rocchio algorithm. Next, I combine the two terms with my input query, then make these in the same matrix above to find the descending order of them. If one origin query is not in the matrix, add it to the end of new query.

With the descending order, I make them as a new query to search again. And check if the result is relevant with the new query or not. The project will end, if current precision reaches input precision or 0.

Google API Key: AIzaSyBEGzhnRJLxeoW_JowF0_BXrowPsOSf7Yk
Engine ID : 811402363ffc54fc3

Additional information

References:

https://www.cs.columbia.edu/~gravano/cs6111/Readings/singhal.pdf
https://nlp.stanford.edu/IR-book/
https://docs.python.org/3/library/argparse.html
https://github.com/googleapis/google-api-python-client/blob/main/samples/customsearch/main.py
https://developers.google.com/custom-search/v1/reference/rest/v1/Search

https://edstem.org/us/courses/53443/discussion/4311405
https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/
https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
https://www.geeksforgeeks.org/how-to-convert-numpy-matrix-to-array/
https://www.geeksforgeeks.org/how-to-use-numpy-argsort-in-descending-order-in-python/