Homework–7

Course Name: Information Retrieval/Web Search(COMP 8130)

Course Instructor: Professor Vasile Rus

Submitted By

Student Name: Anjana Tiha
UID: U00619942
Date: 11/22/2017

Problem 1 [30 points].

Develop a retrieval program that takes as input an user query in the form of a set of keywords, uses the inverted index to retrieve documents containing at least one of the keywords, and then ranks these documents based on cosine values between query vector and document vectors. The output should be a ranked list of documents with links to the original documents, i.e. URLs to the original documents on the web.

Answer 1:

Functionality :

Program takes query search string as input and returns most relevant url from analyzing previously collected 10,000 unique web documents from "memphis.edu" domain where each collected document contains more than 50 tokens after preprocessing.

Method :

Query Processing

Added Query processing functionality along with TF-IDF vector generation of collected 10000 documents. For query Processing: Preprocessed query simmilar to document corpus processing:

1. Takes input string for query.
2. Preprocesses the query string by removing the following:
    - digits
    - punctuation
    - stop words (used the generic list available at ...ir- websearch/papers/english.stopwords.txt)
    - uppercase
    - morphological variations
3. Tokenized query string.
4. Generetaed TF-IDF vector from query.
5. Calculate cosine similarity between TF-DF vector of document corpus and query string.
6. Ranks the cosine similarity of document and query in descending order.
7. Show the corresponding urls for matching ranked documents from most similar to least.

TF-IDF of Document Corpus

Generated TF-IDF of document corpus from text collected from 10,000 web documents. Generated hashmap in format map{word}{file name}= current word count} in current file. Also maintained fields for maximum word count for a word in a file along with total word count. Another hashmap had (file, word: single word count in this file). This hashmaps were saved for later query similarity calculation.

Problem 2 [20 points].

Develop a web interface to the program above.

Web Interface

Developed web interface using Django open-source web framework.
To see search using the web interface, please install Django web framework.

Steps:

1. Install pip for python if not installed already.
2. Move to python directory or scripts directory in Anaconda.
3. Please enter -"pip install Django" for installing Django.

To open project in web interface:

1. To runserver for current project go to project folder "search_engine_website" where manage.py file is located.
2. Open command prompt in the directory of manage.py and type manage.py preceded by python.exe location and python in the follwoing manner:
3. `C:\Users\Anjana\Anaconda3\pythonmanage.pyrunserverserver`
4. `(format->locationforpython.exe+python+manage.py)`
5. To view web interface for search engine go to http://127.0.0.1:8000/

Files:

- search engine website contains all the documents after preprocessing, and hashmaps from doc to url map and other tfidf vector for document and query.
- search_engine.py is the main search engine file.

Document link:

Google Drive Links:
`https://drive.google.com/open?id=1JOHK4UHuzLB4a6CaN8eAookPj48v-fJJ`

Figures

Figure 1: Search Engine Interface

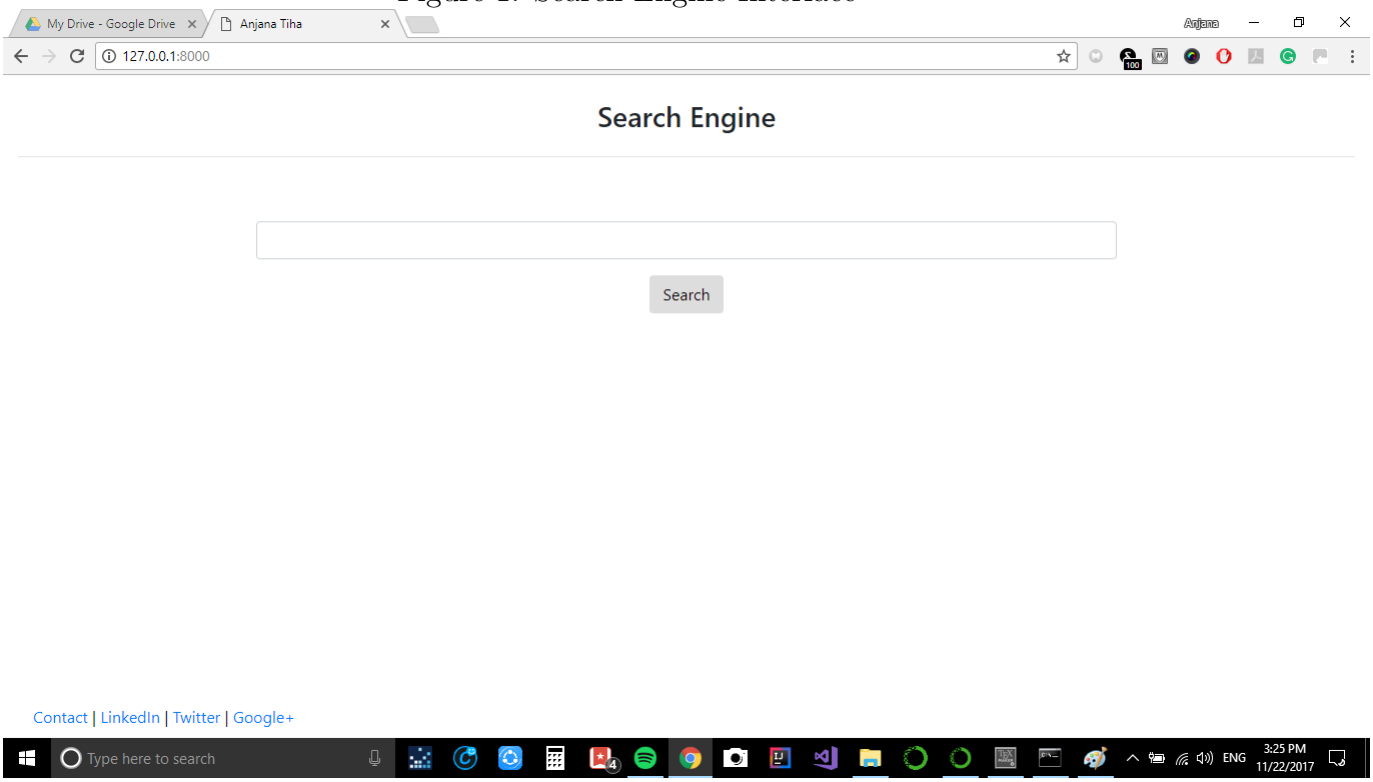

Figure 2: Search Engine Interface With Results