Assignment 2
Ranked Retrieval for Free Text Queries
IR 2020

Deadline: 2nd November

October 19, 2020

This assignment is on building tf-idf based ranked retrieval system to answer free text queries. It is highly recommended that you use python for this assignment as libraries like nltk will make many things easier (stop word removal and lemmatization). However, if you use any other language, you most probably have to design these modules yourselves which might not perform as good as nltk library in python.

- Find your dataset and other required informations.
  - Dataset: The dataset contains 1000 text files of the same dataset you used for last assignment.
  - Static Quality Score: A python list, containing static quality score of 1000 documents (To know more, please follow chapter 7.1.4 of the textbook by Manning) dictionary where key is document and value is its $g(d)$ value.
  - Leaders: A python list, containing index of 30 leader documents. (To know more, please follow chapter 7.1.6 of the textbook by Manning)
- Remove stop words, punctuation marks, make everything to lowercase and perform lemmatization to generate tokens from the document (use nltk library in python).
- Tasks
  - Let $\mathrm{tf\_idf}_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t$ where $\mathrm{tf}_{d,t} = \log_{10}(1 + \tilde{\mathrm{tf}}_{d,t})$ and $\mathrm{idf}_t = \log_{10}(N/\mathrm{df}_t)$, $\tilde{\mathrm{tf}}_{d,t}$ denotes number of times term t appears in document d.
  - Build $\mathit{InvertedPositionalIndex}$, that is, a python dictionary with (t,idf_t) as keys and (d, $\mathrm{tf}_{t,d}$) as postings (consider t as term and d as document).
  - Build $\mathit{ChampionListLocal}$, that is, a python dictionary that contains a list for each term, containing the index of top 50 documents with highest $\mathrm{tf}_{t,d}$ values.
  - Build $\mathit{ChampionListGlobal}$, that is, a python dictionary that contains a list for each term, containing the index of top 50 documents with highest $g(d) + \mathrm{tf\_idf}_{t,d}$ values.

- Answering free text query: The queries to be answered are free text queries. Remove stop words, punctuation marks, make all lowercase and then apply lemmatization on the query text. Let the resulting query after the first step be Q. Now find the top-10 relevant documents according to each of the following scoring schemes.
  - $tf\_idf\_score(Q, d) = \frac{V(Q)\cdot V(d)}{|V(Q)|\cdot|V(d)|}$, while $V(Q)(t) = idf_t$ if $t \in Q$, 0 otherwise, $V(d)(t) = tf\_idf_{t,d}$,$|x|$ denotes euclidean norm.
  - $Local\_Champion\_List\_Score(Q, d) = \frac{V(Q)\cdot V(d)}{|V(Q)|\cdot|V(d)|}$, while $V(Q)(t) = idf_t$ if $t \in Q$, 0 otherwise, $V(d)(t) = tf\_idf_{t,d}$, $|x|$ denotes euclidean norm and we will be scoring only documents in $\mathcal{A} = \{d | d \in LocalChampionList(t), t \in Q\}$
  - $Global\_Champion\_List\_Score(Q, d) = \frac{V(Q)\cdot V(d)}{|V(Q)|\cdot|V(d)|}$, while $V(Q)(t) = idf_t$ if $t \in Q$, 0 otherwise, $V(d)(t) = tf\_idf_{t,d}$, $|x|$ denotes euclidean norm. and we will be scoring only documents in $\mathcal{A} = \{d | d \in GlobalChampionList(t), t \in Q\}$
  - Cluster Prunning Scheme (To know more, please follow chapter 7.1.6 of the textbook by Manning):
    - Index of Leaders contains list of leader file names.
    - Let your query be Q. Let us define $L(Q) = d$ if $tf\_idf\_score(d, Q) = \max_{d \in IndexOfLeaders} tf\_idf\_score(d, Q)$
    - Find $Followers(L) = \{d | tf\_idf\_score(d, L) \geqslant tf\_idf\_score(d, \bar{L}), \bar{L} \in IndexOfLeaders$
    - $Cluster\_Prunning\_Score(Q, d) = \frac{V(Q)\cdot V(d)}{|V(Q)|\cdot|V(d)|}$, while $V(Q)(t) = idf_t$ if $t \in Q$, 0 otherwise, $V(d)(t) = tf\_idf_{t,d}$, $|x|$ denotes euclidean norm and we will be scoring only documents in $\mathcal{A} = \{d | d \in L(Q) \cup Followers(L)\}$

- Instruction for submission
  - Assume the dataset to be in the path "../Dataset", i.e., the dataset folder is just outside your python code folder.
  - Naming the code file: The name of the code file should be in uppercase letters as below.
    ASSIGNMENT2_ < ROLLNO > .py
    e.g. :- For a student with roll no 17CS92R02, the code file name should be
    "ASSIGNMENT2_17CS92R02.py".
  - Reading the queries: Write code which can take "query.txt" file as an argument as below.
    >> python code.py query.txt
    (query.txt file will contain many queries in free text format. For example–
    religion4
    good or bad
    There will be one query in each line. This file will remain unknown to you. Your program will be evaluated based on the results it produces for the queries in the above file.)

– Saving the search results: Your program should read the queries one by one and get the search results. At the end it should create a text file where you will accumulate your finding in following format.

  Query
  result for scoring scheme 1
  ..
  result for scoring scheme N

  Next query
  ...

  For example

  religion4
  $< \text{doc1}, \text{score} >, < \text{doc2}, \text{score} >, ..$
  $< \text{doc1}, \text{score} >, < \text{doc2}, \text{score} >, ..$
  $< \text{doc1}, \text{score} >, < \text{doc2}, \text{score} >, ..$
  $< \text{doc1}, \text{score} >, < \text{doc2}, \text{score} >, ..$

  good or bad
  $< \text{doc1}, \text{score} >, < \text{doc2}, \text{score} >, ..$
  $< \text{doc1}, \text{score} >, < \text{doc2}, \text{score} >, ..$
  $< \text{doc1}, \text{score} >, < \text{doc2}, \text{score} >, ..$
  $< \text{doc1}, \text{score} >, < \text{doc2}, \text{score} >, ..$

  The name of the results file should follow the below convention.
  $\text{RESULTS2\_} < \text{ROLLNO} > .\text{txt}$
  e.g. :- "RESULTS2_17CS92R02.txt"

  Please upload a single code with proper naming convention.

– Python library restrictions: You can use python libraries like nltk, numpy, os, sys, collections, etc. However, you can't use libraries like lucene, elasticsearch, or any other search api. If your code is found to use any of such libraries, you will be awarded with zero marks for this assignment without any evaluation.

– Plagiarism Rules: If your code matches (more than 50%) with another student's code, all those students whose codes match will be awarded with zero marks without any evaluation. Therefore, it is your responsibility to ensure you neither copy anyone's code nor anyone is able to copy your code.

– Code error: If your code doesn't run or gives error while running, you will be awarded with zero mark.