COMP 6000I Search Engines and Applications
Fall 2018 Homework 2

Due: Oct 13, 2018 11:59pm (everybody gets 2 weeks)

Submit your answer in HARDCOPY to Huan ZHAO before the deadline or to me during the lecture on the deadline date. You also need to submit your codes and documents in a zip file. Submission procedure is being worked out; stay tuned.

1. [100] This is a programming assignment. You can use any programming language you want, including scripting languages such as AWK (if you know what it is) or PHP (as suggested by some students).

   - A text collection is available for download at https://www.cse.ust.hk/~dlee/csit6000I/Password_Only/collection-100.txt

     It contains 100 passages separated by a blank line. No document ID (DID) is given in the file. You can assign DIDs to the passages starting from D0001, D0002, etc., sequentially.

   - b) A query file is available for download at: https://www.cse.ust.hk/~dlee/csit6000I/Password_Only/query-10.txt

     Each line contains a query; a query could be a phrase with double quotes enclosing the query terms

   - These files are available now.

Write a program to do the following:

(a) Preprocess the documents using the following rules:
   a. Discard all spaces, punctuation marks, and words that have less than 4 characters.
   b. Remove ending "s" from a word
   c. There is no explicit stopwords to remove.

(b) Create an index (inverted file) of the preprocessed documents. Each postings (an entry in a postings list) contains the DID, and the positions of the indexed word in the document (ref: lecture slides and see example below).

(c) For each of the queries in the query file, retrieve the top 3 documents using cosine similarity and tf/tfmax*idf weighting scheme. For each of the top three results, display (i) the document ID, (ii) five highest weighted keywords of the document and the posting lists, (iii) the number of unique keywords in the document, and (iv) the magnitude (L2 norm) of the document vector, and (v) the similarity score. An example display is shown below.

DID
```
live      -> | D2:1,5 | D3:0   | D6:2   |
never     -> | D5:1   |
only      -> | D6:1   |
tomorrow  -> | D1:2   | D2:2   |
twice     -> | D1:0,4 | D2:0,4 |
```

Number of unique keywords in document
Magnitude of the document vector (L2 norm)
Similarity score

(d) In the above, we only mentioned about the inverted index. Describe the other data structures you need to maintain to support search and ranking.
(e) If the text passages are not updated, how would you design your program to speed up the computation of the similarity values?
(f) Write a report that contains the following:
    (i) Explanation of your design
    (ii) A flowchart describing how the output in (c) is produced
    (iii) Answer to (c), (d) and (e) above; for (c) you can simple do a screen dump
    (iv) Programming language and version, the OS, and libraries/packages used, and how to run the programs.

*** You are not required to use external files to store any data. The whole program runs in main memory and use any data structures available in the programming language you use.

*** Your program(s) must run successfully, and the output must be the same as the results in the report. If the results are different, you will get zero marks for the coding part.

*** As said in the lecture, you must compute the term weights and cosine using your own code and data structures you create. You cannot use existing packages to compute anything involved in the vector space model. However, you can use string processing, sorting, file I/O packages, etc., if you need them.

*** When you encounter situations not described in the question, you can make your decision on how to handle them. Thus, your program, design and output could be different from other students.

*** I mention AWK and PHP as a joke. You should use the language you are familiar with, or else Python.

Grading Scheme:

Report	20%
Coding, correctness, robustness, etc.	70%
Design, good use of data structures, efficiency, etc	10%