COMP 6000I Search Engines and Applications
Fall 2019 Homework 1

Due: Oct 14, 2019 11:59pm

Submission method: Submit your homework to Canvas. Submit a PDF file containing the written answer and the source code of your program for Q.2.

1. This question is about the case study conducted by Blair and Maron, which has been covered in the lecture. The paper can be downloaded from:

   https://home.cse.ust.hk/~dlee/csit6000I/Password_Only/papers/cacm-85-blair-maron-evaluation.pdf

   Read this paper to answer the questions.

   a. [10 points] In the experiment, the lawyers and paralegals are allowed to revise a query as many time as they want. Can you figure out why unlimited query formulation is allowed?

   b. [20 points] State which of the following statements are true from the conclusion of the experiment (briefly justify your answer):

      (i) The precision and recall obtained by the two lawyers are consistent with each other (i.e., the figures do not lead to contradictory conclusions).

      (ii) After the lawyers and paralegals have gone through rounds to formulate an information request into STAIRS' query, the precision and recall of the reformulated information request are significantly improved.

      (iii) If the lawyers are to write their own queries directly on STAIRS (not using the paralegals), the experiment found that the lawyers can get significantly better precision and recall than the paralegals.

      (iv) Recall is time consuming to evaluate. Did the lawyers actually examine all un-retrieved documents to find out how many relevant documents have been missed for a query?

      (v) The Blair and Maron paper is more than 30 years ago. The issues it studies have been completely solved in today's search engines.

   c. [10 points] In the paper, the authors gave a reason for the low recall of the experiment and some examples drawn from the document collection to illustrate the difficulty of

using keywords to retrieve documents. Give a brief summary in, say, 2-3 sentences, to describe the reason and one example given by the authors. Do not copy the whole paragraph(s) from the paper.

d. [10 points] How much time and dollar cost did it take to complete the evaluation?

2. [50] **Write a program** to complete this question:
   - 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.

   - 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

   Functional requirements:

   (a) Preprocess the documents using the following stemming and stopword removal rules:
      i. Discard all spaces, punctuation marks, and words that have less than 4 characters.
      ii. Remove ending "s" from a word (this could make a word have less than 4 characters, and this is fine for this homework).

   (b) Create an index (inverted file) of the preprocessed documents. Each posting (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 postings 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
First 20 words of the document
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: …

Additional information required:

(a) In addition to the inverted index. Describe the other data structures you need to support search and ranking.
(b) If the text passages are not updated, how would you design your program to speed up the computation of the similarity values?

Implementation Notes:

(a) You can use any programming language (C, C++, Java, Python, JavaScript, etc.).
(b) Most modern programming languages provide data structures for inserting name:value pairs into the data structure, and retrieve the value given the name. You should use such data structure in this question (i.e., do not use an array which would require you to do insertion and searching all by yourself). An example data structure for Java is Hashmap. You should use the corresponding data structure in your chosen language.
(c) You are not required to use external files to store any data. The whole program runs in main memory.
(d) Since this is a small dataset, you do not need to map words into word IDs.

Bonus: Maximum 10 points
Bonus is given for Q.2 if you spend significant effort in implementing some good and significant features. You can suggest any features, but here is some examples:
1) Implement a sentence retrieval system which returns sentences as the result while utilizing the document-sentence relations to improve search quality.
2) Implement a better data structure for fast insertion and searching that can scale up to web-scale data.
3) A good summary (snippet) generator that works well with long documents.
4) Good way to handle sematic matching (e.g., handling of synonym and polysemy)
5) …

Note:
Describe your new feature(s) and explain why it/they should be given bonus points. Since the example data is small, any "improvement" may not be observable using the small dataset. You should focus on the **design** that you can argue to work well when the data scales up.