COMP6714 (16S2) ASSIGNMENT 2

DUE ON 23:59 31 OCT, 2016 (MON)

Q1. (*25 marks*)

Consider using the maxscore algorithm to find top-2 results for a query with three different terms $\{A, B, C\}$. The scoring function is the BM25 function with $k_1 = k_3 = 2.0$ and $b = 0$.

$$score(d, Q) = \sum_{t \in Q} idf_t \cdot \frac{(k_1 + 1)tf_{t,d}}{k_1((1 - b) + b\frac{L_d}{L_{ave}}) + tf_{t,d}} \cdot \frac{(k_3 + 1)tf_{t,Q}}{k_3 + tf_{t,Q}}$$

Answer the following questions. You need to show major steps.
The posting lists are shown below. Each posting consists of document ID and tf.

term	**idf**	**postings**
A	6	$(D_1 : 1), (D_2 : 8), (D_5 : 3), (D_8 : 10)$
B	2	$(D_1 : 1), (D_5 : 4), (D_6 : 1), (D_7 : 4)$
C	1	$(D_1 : 1), (D_2 : 2), (D_4 : 1), (D_5 : 2), (D_6 : 3), (D_8 : 1), (D_9 : 1), (D_{10} : 3), (D_{11} : 7)$

TABLE 1. Posting Lists

(1) Show that the maxscore for each keyword can be computed *without* examining the postings list.
(2) Using the maxscore obtained above, determine the postings that are accessed *for scoring* by the algorithm. You need to assume that each `skipTo(x)` call "magically" moves the cursor **directly** to the first posting with document ID at least x (i.e., it does *not* access any other postings).

Hint 1. *Calculate the maxscore if you know that the maximum tf is 1, 10, 100, and 1000, respectively.*

Q2. (*25 marks*)

The *cluster pruning* method is introduced in Chap 7.1.6 of [MRS08].

(1) Consider the basic method (i.e., only using the closest leader to the query q). Justify the choice of choosing $\sqrt{N}$ leaders in the preprocessing step. (Hint: try to design a simple model to estimate the query processing cost)

(2) Find a minimal example where the basic method fails to return the closest document vector to the query q. You only need to give the document vectors and the query vector, and list the document returned by the cluster pruning method and the correct answer. Will the variation of the basic method (i.e., $b_1, b_2 > 1$) eliminate such problem (and guaranteed to return the correct answer)?
(3) Can you propose some modification to this method such that it guarantees returning the closest vector for any query? Describe your method and illustrate it with a small example.

Q3. (*25 marks*)

The following list of Rs and Ns represents relevant (R) and nonrelevant (N) returned documents in a ranked list of 20 documents retrieved in response to a query from a collection of 10,000 documents. The top of the ranked list is on the left of the list. This list shows 6 relevant documents. Assume that there are 8 relevant documents in total in the collection.

```
R R N N N     N N N R N     R N N N R     N N N N R
```

(Note that spaces above are just added to make the list easier to read)

(1) What is the precision of the system on the top-20?
(2) What is the F_1 on the top-20?
(3) What is/are the uninterpolated precision(s) of the system at 25% recall?
(4) What is the interpolated precision at 33% recall?
(5) Assume that these 20 documents are the complete result set of the system. What is the MAP for the query?

Assume, now, instead, that the system returned the entire 10,000 documents in a ranked list, and these are the first 20 results returned.

(6) What is the largest possible MAP that this system could have?
(7) What is the smallest possible MAP that this system could have?
(8) In a set of experiments, only the top-20 results are evaluated by hand. The result in (5) is used to approximate the range (6) to (7). For this example, how large (in absolute terms) can the error for the MAP be by calculating (5) instead of (6) and (7) for this query?

Q4. (*25 marks*)

Consider the documents below.

docID	**document text**
D_1	I don't want to go A groovy king of love You can't hurry love This must be love Take me with you
D_2	All out of love Here i am I remember love Love is all Don't tell me

(1) build a unigram query likelihood language model (LM) for each document. Assume that (i) the only preprocessing done before tokenization is to transform all letters to lower cases, and (ii) we use the Jelinek-Mercer smoothing method with $\lambda = 0.5$.
(2) show which document will be ranked first for the queries:

- Q_1: `i remember you`
- Q_2: `don't want you to love me`

(3) assume that we have a prior probability distribution over the two documents as $p(D_1) = 0.7$ and $p(D_2) = 0.3$. Will this change the ranking results of the two previous queries?

Submission Instructions

You need to write your solutions to the questions in a pdf file named `ass2.pdf`. You **must**

- include your **name and student ID** in the file, and
- the file can be opened correctly on CSE machines.

You need to show the key steps to get the full mark.

Note: Collaboration is allowed. However, each person must independently write up his/her own solution.

You can then submit the file by `give cs6714 ass2 ass2.pdf`.

Late Penalty: -10% for the first two days, and **-30%** for the following days.