INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
Mid Autumn Semester Examination 2023-24

Date of Examination: 22/09/23 **Session:** AN **Duration** 2 hrs, **Marks** = 50
Sub No: CS60092 **Sub Name:** Information Retrieval
Department/Centre/School : Computer Science and Engineering
Specific charts, graph paper, log book, etc. required NO

Special Instructions: ANSWER ALL questions. All parts of a single question should be answered together. Answers should be brief and to-the-point. Marks will be deducted for sketchy answers and claims without proper reasoning. In case of reasonable doubt, make assumptions and state them upfront. You can keep probability values in fractional forms.

---

1. Consider the two postings list A and B in Figure 1. Intersect the two list and answer the following questions.

   (a) How many times is a skip pointer followed during intersection?

   (b) How many comparisons will be made to perform this intersection? List them.

   (c) How many comparisons would have been made if there were no skip pointers in list A?

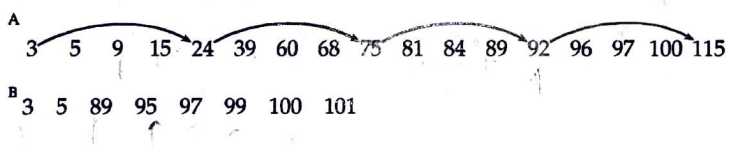

Figure 1: Postings list.

[1 + 3 + 1 = 5]

2. Are skip pointers useful for queries of the form x `OR` y? Justify. **[2]**

3. Consider the following fragment of a positional index with the format:
   word: document: <position, position, . . .>; document: <position, . . .>
   . . .

   Gates: 1: <3>; 2: <6>; 3: <2, 17>; 4: <1>;
   IBM: 4: <3>; 7: <14>;
   Microsoft: 1: <1>; 2: <1, 21>; 3: <3>; 5: <16, 22, 51>;
   The $/k$ operator, word1 $/k$ word2 finds occurrences of word1 within k words of word2 (on either side), where k is a positive integer argument. Thus $k = 1$ demands that word1 be adjacent to word2.

   (a) Describe the set of documents that satisfy the query Gates /2 Microsoft.

   (b) Describe each set of values for k for which the query Gates $/k$ Microsoft returns a different set of documents as the answer. Recall that $\{x\}$ and $\{x, y\}$ are different sets.

```
S1:                                          | referential:
 q1: d01 d02 d03 d04 dXX dXX dXX dXX         |  q1: d01 d02 d03 d04
 q2: d06 dXX dXX dXX dXX                     |  q2: d05 d06
 q3: dXX d07 d09 d11 dXX dXX dXX dXX dXX     |  q3: d07 d08 d09 d10 d11
 q4: d12 dXX dXX d14 d15 dXX dXX dXX dXX     |  q4: d12 d13 d14 d15

S2::                                         | referential:
 q1: dXX dXX dXX dXX d04                     |  q1: d01 d02 d03 d04
 q2: dXX dXX d05 d06                         |  q2: d05 d06
 q3: dXX dXX d07 d08 d09                     |  q3: d07 d08 d09 d10 d11
 q4: dXX d13 dXX d15                         |  q4: d12 d13 d14 d15
```

Figure 2: Results from system S1 and S2.

[2 + 4 = 6]

4. Consider two IR systems S1 and S2 that produced the outputs show in Figure 2 for the 4 reference queries q1, q2, q3, q4. Here dXX refer to the documents that do not appear in the referential (i.e., relevant documents for a given query).

   (a) For each of the two systems, compute the precision and recall for each query (provide the results as fractions). Finally average the precision and recall for all the queries and compare S1 and S2 based on precision. Explain all the steps of your computation.

   (b) Compute the average P@k values for k between 1 and 5 for the IR systems S1 and S2 above. Based on these results, what is your relative evaluation of the two systems? How does it compare to your previous observation based on binary precision. **[3 + 1 + 4 + 1 = 9]**

5. List three situations where an IR system's MAP and MRR performances will be equal. **[3]**

6. Consider a corpus of documents $\mathcal{S}$, having the following documents $D_1, D_2, \ldots, D_4$ (see Table 1):

Documents	Words
D_1	Computer Science and Engineering
D_2	Largest Computing Frameworks for Large Data
D_3	Good Technologies: Large Language Models
D_4	Good Technologies for Better Models
Q	Large Computers and Frameworks for the Larger Good

Table 1: Corpus $\mathcal{S}$ and Query Q for Question 6

We apply pre-processing steps such as lower-casing, stopword removal and lemmatization. Due to lemmatization, the following words get converted:

`{computing,computer,computers}` $\rightarrow$ `compute` `{largest,larger}` $\rightarrow$ `large`

`better` $\rightarrow$ `good` `engineering` $\rightarrow$ `engineer`

Apart from these, all words occuring in *plural* form are converted to *singular*. This gives us a final vocabulary V of the following 10 tempswords:

$\{w_1 =$ `compute`, $w_2 =$ `science`, $w_3 =$ `engineer`, $w_4 =$ `large`, $w_5 =$ `framework`, $w_6 =$ `data`, $w_7 =$ `good`, $w_8 =$ `technology`, $w_9 =$ `language`, $w_{10} =$ `model`$\}$

Apart from the corpus S, consider the query Q as shown in Table 1. You need to implement a vector space retrieval model with the **"lnc.Ltc"** scheme. To do this, you need to start by representing each document $d \in S$ and the query Q using 10-dimensional term-frequency vectors.
$[tf(w,d) = \text{nos. of times term } w \text{ occurs in } d, \forall d \in \{D_1, \ldots, D4, Q\}, \forall w \in V]$

Then, you need to follow the steps described below:

(a) **TF Normalization:** Normalize the above vectors with the specified TF normalization scheme. Note that different normalization schemes are to be used for the documents in S and for Q.
$[tf(w,d) = 1 + \log_{10}(tf(w,d))$ if $d \in S$; $\quad tf(w,d) = \frac{1+\log_{10}(tf(w,d))}{1+\log_{10}(\text{avg}_{w' \in d}\ tf(w',d))}$ if $d = Q]$

(b) **IDF Normalization and TF-IDF computation:** Normalize the above vectors with the specified IDF normalization scheme, and compute TF-IDF values as described below.
df(w) = Nos. of documents $d \in S$ where w occurs, $\forall w \in V$
$[tfidf(w,d) = tf(w,d)$ if $d \in S$; $\quad tfidf(w,d) = tf(w,d) \cdot \log_{10} \frac{|S|}{df(w)}$ if $d = Q]$

(c) **Unit Vector Conversion:** Convert all the obtained TF-IDF vectors into unit vectors using the cosine normalization scheme.
$[tfidf(w,d) = \frac{tfidf(w,d)}{\sqrt{\sum_{w' \in V}\ tfidf(w',d)^2}}\ \forall\ d \in \{D1, \ldots, D4, Q\}]$

(d) **Scoring:** Now, obtain the score of each document $d \in S$ w.r.t. query Q by calculating the dot product of the respective unit vectors.
$[score(d,Q) = \sum_{w' \in V} tfidf(w',d) \cdot tfidf(w',Q)\ \forall\ d \in S]$

[*Use exactly 3 decimal points for every step of your calculation.*] **[5 + 1 + 5 + 2 = 13]**

7. Answer the following questions:

(a) Write down all the permuterm indices for the word `'tattoo'`.

(b) What will be the permuterm keys to perform lookup on, for the wildcard search queries `s*ng` and `man*`?

(c) How would you perform the permuterm retrieval for the wildcard query `to*r*nce`? State all steps briefly.

(d) Calculate the Edit Distance between the words `'sleep'` and `'seek'`. Consider only the three operations insertion, deletion and substitution of a character, each having cost 1. Show the entire table of calculations.

(e) Answer the following only with *"True"/"False"*:
   i. SPIMI is a form of distributed indexing. F
   ii. Map-reduce indexing requires the use of two separate sets of machines for parsers and inverters. F

(f) Fill in the blanks
   i. In BSBI indexing, we first collect *all* termID-docID pairs and sort them before creating the postings list.
   ii. To deal with the issue of frequent merges in dynamic indexing, we can use logar indexing.

[2 + 2 + 2 + 4 + 1 + 1 = 12]