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

Sub No: **CS60092** **Sub Name: Information Retrieval**
Department/Centre/School : Computer Science and Engineering
Duration 3 hrs **Marks = 80**

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. All calculations must be shown. No marks will be awarded for sketchy answers and answers/results without proper reasoning/calculations. In case of reasonable doubt, make assumptions and state them upfront. You can keep probability values in fractional forms.

1. The following questions concern evaluation of IR systems using different metrics.

   (a) Consider the following search results for two queries Q1 and Q2 (the documents are ranked in the given order, the relevant documents are shown in bold).
   Q1: **D1**, **D2**, D3, **D4**, D5, **D6**, D7, D8, D9, D10
   Q2: **D1**, D2, **D3**, D4, D5, **D6**, D7, D8, **D9**, D10
   For Q1 and Q2 the total number of relevant documents are, respectively, 5 and 8. Find the MAP over these queries. **[5]**

   (b) Let us consider that a search engine outputs 5 documents named $(D1, D2, D3, D4, D5)$ in that order. Define the relevance scale (0-3) where:
   0 : not relevant
   1-2 : somewhat relevant
   3 : completely relevant
   Suppose these documents have relevance scores:
   $D1$: 3
   $D2$: 2
   $D3$: 0
   $D4$: 0
   $D5$: 1
   Compute the (i) CG_5, (ii) DCG_5 (iii) $IDCG_5$ and (iv) $nDCG_5$. **[6]**

2. Suppose that a user's initial query is *cheap CDs cheap DVDs extremely cheap CDs*. The user examines two documents, d_1 and d_2. She judges d_1, with the content *CDs cheap software cheap CDs* relevant and d_2 with content *cheap thrills DVDs* non-relevant. Assume that we are using direct term frequency (with no scaling and no document frequency) for our vector space representations. There is no need to length-normalize vectors. Using Rocchio relevance feedback what would the modified query vector q_m be after relevance feedback? Assume $\alpha = 1, \beta = 0.75, \gamma = 0.25$. **[5]**

3. For the web graph in Figure 1, compute PageRank, hub and authority scores for each of the three pages. Also give the relative ordering of the 3 nodes for each of these scores, indicating any ties.

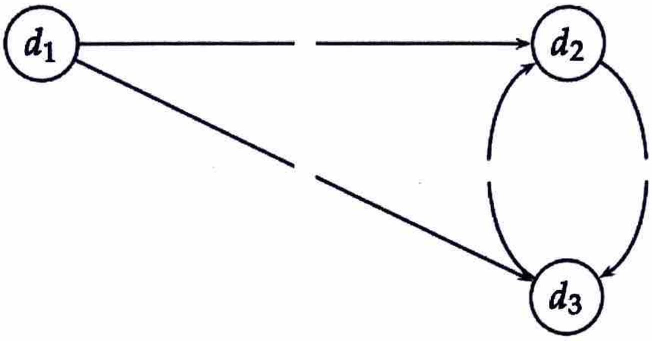

Figure 1: The three node web graph.

PageRank: Assume that at each step of the PageRank random walk, we teleport to a random page with probability 0.1, with a uniform distribution over which particular page we teleport to. **Hubs/Authorities**: Normalize the hub (authority) scores so that the maximum hub (authority) score is 1. **[3+5=8]**

4. A directed network D has vertex set $V = \{1, 2, 3, 4\}$ and edge set $E = \{(1,2), (2,1), (1,3), (2,3), (3,4)\}$.

   (a) Starting with hub score 1 at each vertex, make two complete iterations in the calculation of hub and authority scores for the network D. **[5]**

   (b) Comment on the answer you obtained in terms of the HITS ranking of the vertices. **[1]**

5. This question is about document summarization.

   (a) Consider the following lexical chains. Each chain has a number of terms. Each term is associated with its number of occurrences in the chain (read a chain entry term: f as follows – the term occurs f times in the chain).
   Chain 1 microsoft: 10 → concern: 1 → company: 6 → entertainment-service: 1 → enterprise: 1 → massachusetts-institute: 1
   Chain 2 computer: 4
   Chain 3 ai: 2 → artificial-intelligence: 1 → field: 7 → technology: 1 → science: 1
   Chain 4 bayesian-technique: 1 → condition: 1 → datum: 2 → model: 1 → information: 3 → area: 1 → knowledge: 3

      i. Score each chain using the method described in class. Show the computation. **[2]**
      ii. Which among the above are strong chains? Show the computation. **[3]**

   (b) Consider the following candidate and reference summaries.
   Candidate: Transformers Transformers are fast plus efficient
   Reference: HuggingFace Transformers are fast efficient plus awesome

      i. Compute ROUGE-1 precision and recall.
      ii. Compute ROUGE-L precision and recall.
      iii. In the context of this example which measure do you think is more reasonable.

   [2 + 2 + 1 = 5]

6. Consider a document $\mathcal{D}$ consisting of multiple sentences (after lower-casing, removing punctuations and stopwords, and lemmatization):
`<s> michael bob go school </s>`
`<s> school playground </s>`
`<s> michael bob zack friend </s>`
Here, the markers `<s>` and `</s>` denote beginning and end of sentence respectively.

Also, consider the query $\mathcal{Q}$: `<s> bob go playground </s>`.

A language model $\mathcal{L}^{(\mathcal{D})}$, trained from the document $\mathcal{D}$, can learn to estimate the probability of generating a sentence $s = [t_0, t_1, \ldots, t_n]$, where $t_0 =$ `<s>` and $t_n =$ `</s>`, as

$$P(s|\mathcal{L}^{(S)}) = \prod_{i=1}^{n} P(t_i|t_0 t_1 \ldots t_{i-1})$$

(a) A bi-gram language model $\mathcal{L}_{bi}^{(\mathcal{D})}$ approximates the terms $P_{bi}(t_i|t_0 t_1 \ldots t_{i-1}) \approx P(t_i|t_{i-1})$. Calculate $P(\mathcal{Q}|\mathcal{L}_{bi}^{(\mathcal{D})})$. *Note that `</s><s>` is not a valid bigram.*

(b) *Backoff* is a smoothing technique that can solve the issue of zero probabilities, for bigrams that have not been observed in the training set. The approximation is adjusted as

$$P_{bo}(t_i|t_0 t_1 \ldots t_{i-1}) \approx \hat{P}(t_i|t_{i-1}) \quad \text{if count}(t_{i-1}t_i) > 0 \quad \text{else } \lambda(t_{i-1})\hat{P}(t_i)$$

where $\hat{P}(x) = P(x) - k$, and $\lambda(x)$ is a parameter based on the term x. Calculate $P(\mathcal{Q}|\mathcal{L}_{bo}^{(\mathcal{D})})$, given that $k = 1/20$.
Hint: To calculate $\lambda(x)$, you can make use of the fact that $\sum_z P_{bo}(z|x) = 1 \; \forall z$. For example, to calculate $\lambda($`<s>`$)$:

$$\begin{aligned} P_{bo}(\texttt{michael}|\texttt{<s>}) + P_{bo}(\texttt{bob}|\texttt{<s>}) + P_{bo}(\texttt{<go>}|\texttt{<s>}) + \ldots &= 1 (\text{over all terms in the corpus}) \\ \text{or, } \hat{P}(\texttt{michael}|\texttt{<s>}) + \lambda(\texttt{<s>})\hat{P}(\texttt{bob}) + \lambda(\texttt{<s>})\hat{P}(\texttt{<go>}) + \ldots &= 1 \end{aligned}$$

You can further assume that $\forall_z P_{bo}(\texttt{<s>}|z) = \forall_z P_{bo}(z|\texttt{</s>}) = 0$. *That is, `<s>` cannot follow any token, and no token can follow `</s>`.*

(c) What is the role of k in the above equation? What happens when $k = 0$?

For all calculations above, *show all the steps*. You must retain *at least 3 decimal places* at every step in your calculation. **[3 + 7 + 2 = 12]**

7. Consider a corpus of documents $\mathcal{D}$ and a query q. Let X_t be a boolean random variable (r.v.) indicating if the term t occurs in a document d, and let R be a boolean r.v. indicating if the document d is relevant w.r.t. q. Suppose that there are a total of n relevant documents w.r.t. q, and out of those, m contain the term t. *We have observations of X_t only for the relevant documents, no other observations are available.*

The probability distribution of the data can be characterized by the parameter $p_t = P(X_t = 1|R = 1, d, q)$. Show that the Maximum Likelihood Estimate (MLE) of the parameter p_t is m/n. In other words, show that $p_t = m/n$ maximizes the probability for the *observed data only*.

Hint: The probability of the data is given by $P(X_t|R, \mathcal{D}, q) = \prod_{d \in \mathcal{D}} P(X_t|R, d, q)$. **[8]**

$P(X_t = 1 | R = 1, d, q)$

$n p^{n-1} (1-p)^{m-n} - (1-p)^{m-n-1} (m-n) . p^n = 0$

$p_t^n (1-p_t)^{m-n}$

$p^{n-1} (1-p)^{m-n-1} [n(1-p) - (m-n)p] = 0$

8. Answer the following questions:

   (a) Consider the following postings list of docIDs:
   `64 → 192 → 196`

      i. Represent the *entire* postings list with Variable Byte Code compression.

      ii. Represent the *entire* postings list with Gamma Code compression.

   (b) If a dictionary (stored as an array sorted according to docIDs) contains 11 terms, and each term can occur with the same probability for any query, what is the average number of comparisons required for any term when the dictionary is uncompressed? What is the average number of comparisons required when the same is compressed using Blocking (Block size = 4)? Note that during blocking only the last block can contain less than 4 entries.

[(3 + 3) + (2 + 2) = 10]

9. Consider a set of 3 documents with docIDs {`1, 2, 3`} with a vocabulary of 6 terms {`a, b, c, d, e, f`}. Note that each symbol `a`, `b`, `c`, `d`, `e` and `f` denotes a term. The posting lists are shown below (numbers inside parentheses represent term frequencies within the documents, e.g., term 'a' appears 3 times in doc 1 and 2 times in doc 3):

```
a → 1(3) → 3(2)
b → 1(1) → 2(2)
c → 3(2)
d → 1(1) → 2(1) → 3(2)
e → 1(2) → 2(1)
f → 1(1) → 2(3) → 3(1)
```

Also consider a **query:** `b c d e`

   (a) Represent each document and the query as a 6-dimensional vector, under the **lnc.ltc** scheme. (Take base of logarithm as 10)

   (b) Calculate *cosine similarity* scores between each document and the query.

[7 + 3 = 10]