Indian Institute of Information Technology, Sri City, Chittoor

Information Retrieval Duration: 90 Minutes. Max Marks: 35.

Roll No.: _____ Room No.: _____ Seat No.: _____

Name: _____ Invigilator's Signature: _____

Instructions

- This is a closed book test. You are not allowed to carry any printed material with you.
- You are allowed to carry one page (A4 or smaller) hand written notes. Write your name and roll number on these notes.
- Please switch off your mobile phones and any other digital equipment you may have (like smart watches). You may carry a calculator.
- Wherever you see <yourrollno>, replace this text with your 12 char/digit roll number.
- A negative mark of -1 applies to all questions when answered incorrectly.
- Write all answers correct up to two decimal places.
- Put down your final answer in this sheet. Give a succinct explanation for your answer. You may give your explanation in a separate sheet (optional). Explanation will not be graded.

Having heard of the IR Class at IIIT Sri City - Chittoor, top search engine companies have started scouting for talent to fill key technical positions. This is the story of your interview experience with a top search engine company. You easily cleared the first three rounds of interview. Now, you are talking to the architect of the indexing division. He wants to ensure that you understand the basic principles of building a simple search engine.

Question 1: The indexing team is supporting a simple boolean retrieval system using an (n+1)-dimensional vector space model where each distinct character or digit in <yourrollno> represents a dimension where n represents the number of unique digits in <yourrollno>. As an additional dimension, the character X is included. You index the following 10 documents:

S20160010021
S20160010022
S20160010023
S20160010024
S20160010025
S20160010026
S20160010027
S20160010028

S20160010029
S20160010030

Note that tokenization for both query and documents is done character by character. No stop words removal and no stemming is performed apart from the fact that if a character in the query does not exist in your dictionary, you drop it from the query. In this context, answer the following questions.

Question 1.1: If the query is 0026, hoping to retrieve S20160010026, what is the precision? [1 Mark] Answer: _____.

Solution: *Precision refers to the fraction of relevant documents in the retrieved documents. The only relevant document here is S20160010026. Your roll number has at least one of these characters if not all of them. In any case, we retrieve all 10 documents. Therefore, the precision is* $\frac{1}{10} = 0.1$ *or* 10%.

Question 1.2: If the query is S30, and the information need is to match all documents from our collection that contains all of S, 3 and 0, what is the recall? [1 Mark] Answer: _____.

Solution: *Recall refers to the fraction of relevant documents retrieved over the overall number of relevant documents. Since we hope to retrieve all documents containing S, 3 and 0, the relevant documents are S20160010023 and S20160010030. As long as at least there is one character indexed, we will match both these documents. Therefore, recall is* $\frac{2}{2} = 1$ *or* 100%.

Question 1.3: If the query is S30, and the information need is to match all documents from our collection that contains all of S, 3 and 0, how many false positives did you find? [1 Mark] Answer: _____.

Solution: *False positives refer to the retrieved documents (positives) that are irrelevant (false). S20160010023 and S20160010030 are true positives. Now, your roll number plays a role in determining the answer. If your roll number has S, 3 and 0, only two documents namely S20160010023 and S20160010030 are retrieved. This will lead to 0 false positives. If your roll number does not have 3 but has S or 0, we retrieve all documents. This leads to 8 false positives.*

Question 1.4: You know about the set of words and bag of words models. Here, we apply that knowledge in the setting of your roll number. In the set of characters model, what is the vector representation for <yourrollno> in $(x_1, x_2, ...x_{n+1})$ format where x_i represents a dimension? [1 Mark] Answer: _____.

Solution: *Let us say, your roll number is S20160010023. The set of characters would be 0,1,2,3,6,S,X. Remember, we added X to our index. As per the question setting, X forms the* $(n+1)^{th}$ *dimension. We usually write it in a sorted fashion. This question asks for "vector representation" and not the list of dimensions. Therefore, the answer is (1,1,1,1,1,1,0). If your roll number is S20120010012, the answer would have been*

(1,1,1,1,0). Similarly, we can list all the unique characters from your roll number in ascending order and designate a 1 in our vector space. Since X does not exist, the corresponding dimension will always be set to 0.

Question 1.5: In the bag of characters model, what is the vector representation for <yourrollno> in $(x_1, x_2, ...x_{n+1})$ format where x_i represents a dimension? [1 Mark] Answer: _____.

Solution: *Now, we switch to the "bag" of characters model. So, we keep the character frequency. Therefore, if your roll number is S20160010023, the vector would be (5,2,2,1,1,1,0). Note that the answer varies depending on the frequency of each character in your roll number.*

Question 1.6: Assuming you have to index millions of documents, the term-document matrix is likely to contain several zeros. Which data structure is relatively better to store the postings? [1 Mark]

Tick the best answer:

(1) Single-Dimensional Array
(2) Multi-Dimensional Array
(3) Linked List
(4) Binary Search Tree

Solution: *The answer is Linked List. Clearly, arrays are ruled out since the associated problem is about the sparsity of the term-document matrix. It contains several zeros unnecessarily. A linked list solves this problem by only maintaining the document IDs. Hence, we maintain "postings list" instead of a postings array. We are interested in merging the postings to answer boolean queries. Unless the search tree is balanced, the costs of operations are not anymore logarithmic. Since, we keep indexing all the time for freshness sake, a linked list scores over a binary search tree.*

Question 1.7: For the following three sentences, draw the positional inverted index. [3 Marks]

(1) My roll number is <yourrollno>
(2) I like information retrieval
(3) I like to roll when I study

Assume no stop word removal and no stemming are applied. No other content processing is done. Don't forget to replace your roll number in sentence 1.

Solution Hint: *Your roll number has no relevance to this question. The essential elements of a positional inverted index are a) dictionary terms, b) the term frequency, c) the document IDs containing each term, and d) the position of the term in each document. There are different styles of drawing this. As long as your style has all the above mentioned information, your solution is correct. Additionally, it is also desirable to maintain the term frequency in each document. In this case, we prefer storing the document frequency in our dictionary.*

A common mistake is to mention "document frequency" in the dictionary but put down the "term frequency" in the dictionary. Remember, term frequency and document frequency are different. In an inverted index, we use document frequency in deriving an effective query processing order.

One of the project teams of your IR class developed a new stemmer for an Indian language. This search engine company is about to buy this algorithm for stemming. Using this opportunity to discuss with you, the architect wants to know something in general about stemming.

Question 2: Answer the following questions on stemming. [1 Mark Each]

Question 2.1: Does stemming always improve precision? Tick the best answer:
(1) True
(2) False

Solution: *False. Stemming may contribute to loss of precision. Consider a case where two different tokens stem to the same term. Stemming can increase the retrieved set without increasing the number of relevant documents in the retrieved set.*

Question 2.2: If 'is' is converted to 'be', are we stemming or lemmatizing? Tick the best answer:
(1) Stemming
(2) Lemmatizing

Solution: *Lemmatizing leads to meaningful dictionary words. We are lemmatizing.*

The architect is impressed with your answers to the above questions. He is convinced that you can tokenize documents, visualize them on a vector space and evaluate how good your system is to a reasonable extent. So, he invites his friend Ms. Thompson who is the architect of the ranking and relevance team. Ranking and relevance team is typically concerned with matching and scoring documents.

Question 3: First, she is exploring if you will fit her spelling correction team. So, she asks the following questions. [3 Marks Each]

Question 3.1: As computer engineers, we know that algorithm and analysis are related but different. What is the minimum edit distance (Levenshtein Distance) between *algorithm* and *analysis*? Answer: 8.

Question 3.2: What is the minimum edit distance (Levenshtein Distance) between *kangaroo* and *rangoon*? Answer: 4.

Question 3.3: What is the minimum edit distance (Levenshtein Distance) between *first* and *frost*? Answer: 2.

Solution: *You may check with any of the online calculators for a more detailed explanation.*

Since you did extremely well, she continued to see if you can even answer more advanced questions related to relevance.

Question 4: Given below are pairs of sentences. Find the cosine similarity between them after performing case folding. Apply bag of words assumption. No stemming or stop word removal is done. [3 Marks Each]

Question 4.1: "IIIT Sri City Students Rock" and "Rare Rock found at Sri City". Answer: 0.55.

Solution: *Let us start by listing out the terms in ascending order and construct the document vectors. The dimensions would be "at city found iiit race rock sri students". The vector representation of the first document,* $d_1 = (0, 1, 0, 1, 0, 1, 1, 1)$. *Similarly,* $d_2 = (1, 1, 1, 0, 0, 1, 1, 0)$. *The cosine similarity is* $\frac{d_1.d_2}{||d_1||||d_2||} = \frac{3}{\sqrt{5}\sqrt{6}} = 0.547$ *or 0.55 (rounded).*

Question 4.2: "One gone one came one won" and "one came gone and won". Answer: 0.78

Solution: *Note that this question tests your understanding of the bag of words assumption. We have repeating words here. So, the resulting vector will carry the frequencies. We follow a procedure similar to the previous answer. The vectors dimensions are: andcamegoneonewon. The vectors are:* $d_1 = (0, 1, 1, 3, 1)$, *and* $d_2 = (1, 1, 1, 1, 1)$ *Therefore, cosine similarity =* $\frac{1+1+3+1}{\sqrt{12}\sqrt{5}} = 0.775$ *or 0.78 (rounded).*

She is amazed by your excellent answers. However, she wanted to know that your knowledge is deep enough. Since you demonstrated your knowledge of l_2 norm before, she asks you the following questions.

Question 5: Let a n-dimensional vector, $X = (1, 1, 1, 1, ...1)_{1xn}$. Answer the following questions on vector norms. [1 Mark Each]

Question 5.1: What is l_0 norm of X? Answer: n

Solution: l_0 *norm represents the number of non-zero entries in the vector. Therefore, it is* n.

Question 5.2: What is l_1 norm of X? Answer: n

Solution: *l_1 norm represents the absolute value or sum of entries in the vector. Therefore, it is again n.*

Question 5.3: What is l_3 norm of X? Answer: $\sqrt[3]{n}$

Solution: *l_3 norm of $X = \sqrt[3]{1^3 + 1^3 + ... + 1^3}$.*

Question 5.4: What is $||X||_\infty$? Answer: 1

$||X||_\infty$ is the max value in any vector dimension.

Note: The original intent was to ask why would we select l_2 norm for cosine similarity. Expect that for final exams!

You showed how to compare a pair of sentences. You can apply this in a query-document setting. However, Ms. Thompson's team knows that not all terms carry equal weight while scoring. This knowledge is important in scoring documents better against a given query. You will get the job if you answer these final questions correctly.

Question 6: Answer the following questions on IDF.

Question 6.1: If we triple the number of times a word appears in each document, how does the inverse document frequency (IDF) of this word change? [1 Mark]

1. IDF becomes $IDF/3$
2. IDF becomes $3 * IDF$
3. IDF becomes $IDF * (log(N) + log3)$
4. IDF does not change.

Answer: 4. IDF does not change.

Solution: *The only variable in IDF is the document frequency. Document frequency is the number of documents that contain the wprd. This would not change irrespective of tripling the word inside the document where it was already present.*

Question 6.2: If we add one document containing the word of interest to every three documents in the collection, how does the inverse document frequency (IDF) of this word change? [1 Mark]

Answer: IDF becomes $\log(\frac{N+\lfloor N/3 \rfloor}{df+\lfloor N/3 \rfloor})$.

Solution: *Sometimes, easiest of all questions sounds trickiest of all. There was no trick in this except that this does not nicely reduce to a fraction. We are adding one document to every three documents. Therefore if our collection had N documents, we will now have $N + \lfloor \frac{N}{3} \rfloor$ documents. If the document frequency was df, now it changes to $df + \lfloor \frac{N}{3} \rfloor$. Therefore, the IDF = $\log(\frac{N+\lfloor N/3 \rfloor}{df+\lfloor N/3 \rfloor})$.*

Question 6.3: If the following three documents make a collection, can you compute the IDF of all the tokens? [3 Marks]

(1) My roll number is <yourrollno>
(2) I like information retrieval
(3) I like to roll when I study

Solution: *The words,* ***roll****,* ***I*** *and* ***like*** *appear in two documents each. Rest of the words appear in only one document. IDF* $= \log \frac{N}{df}$*. Therefore, IDF of all those terms that appear in only one document* $= \log \frac{3}{1}$ *= 0.477 or 0.48. IDF of* ***roll****,* ***I*** *and* ***like*** *would be* $\log \frac{3}{2}$ *= 0.176 or 0.18.There are different variants of IDF available. You could use any of them. If you use a variant which was not discussed in the class, it is worth writing down the formula.*

Keep your fingers crossed. The result will be declared soon.