Name: ----------------------------------------------------------------

Registration No: --------------------------

Section: --------------

***Attempt the examination on the question paper and write concise answers.*** *You can use extra sheets for rough work. Do not attach extra sheets used for rough work with the question paper. Do not fill the table titled Question/marks.*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Question | a | b | c | d | e | f | g | h | Total |
| Marks | **/** 1 | **/** 2 | **/** 1 | **/** 1 | **/** 2 | **/** 1 | **/** 3 | **/** 2 | /13 |

**Part 1**

**a)** What kind of problems can arise if we convert all text into lower case during text pre processing? Illustrate with example. [1 Point]

Acronyms will be converted to lowercase which will change their meaning e.g, US will become us.

**b)** What proportion of total vocabulary (proportion of total unique words) of a novel you are expected to see if you have read 45% text of the novel. [2 Points]

V = K \* sqrt(N)

xV = K \* sqrt(0.45N)

x = K \* sqrt(0.45N) / V

x = K \* sqrt(0.45) / V \* sqrt(N)

x = K \* sqrt(0.45) /V \* (V/K) since sqrt(N) = V/K

x = sqrt (0.45) = 0.67 Ans: 67%

**c)** Briefly explain steps of BSBI Indexing algorithm. [1 Point]

Step 0: Divide collection into reasonable size blocks that can fit into memory then perform following steps for each block

Step 1: Create a map for mapping terms to termids and keep that map in memory

Step 2: Sort all termid , docid pairs based on termids

Step3: Merge same term ids to create postings

Step 4: Remove redundant docids from each posting

Step 5: Store postings on hard disk

Merge all postings by using merge routine of merge sort. Bring reasonable size chunks of each block in memory and merge all blocks simultaneously based on termids, refill a chunk of block when its entries are merged.

**d)** What is advantage of using positional index over bigram (biword) index? [1 Point]

Ans: Positional index can match queries consisting of long phrases whereas bigram index can only answer 2 word or bigram queries.

**e)** If we have a corpus of 10 million documents, each of length 3,000 words, and a total vocabulary size of 500,000, what is the approximate maximum

1. size of the postings
2. size of the Boolean matrix (which contains a 1 in row *i* and column *j* if word i occurs in document j and 0 otherwise)

[2 Points]

size of postings = 3 \* 1010

Ans: size of matrix = 5 \* 1012

**Part 2**

**f)** If a coin with unknown bias is flipped 10 times and it comes up heads 10 times then what is the likelihood of getting a tail in next coin flip using Laplace estimates. [1 Point]

P (Tail) = (0 + 1) / (10+2) = 1/12 = 0.08 = 8 %

Given the three-document corpus and a stop word list below, answer the following questions (g and h) AFTER removing stopwords.

|  |  |
| --- | --- |
| **d1** | information retrieval is process of index search retrieval |
| **d2** | retrieval is used for evaluation of search results retrieval retrieval |
| **d3** | evaluation in information in evaluation process search |
| **Query** | information retrieval |
| **Stopwords** | is , of, in, for, to |

**g)** Rank documents according to their TF.IDF score. Show all calculations and fill in the table below.

[3 Points]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Terms** | **TF** | | | **IDF** |
| **d1** | **d2** | **d3** |  |
| information | 1 | 0 | 1 | 0.176 |
| retrieval | 1.3 | 1.47 | 0 | 0.176 |
|  |  |  |  |  |
| TF.IDF | 1\*0.176 + 1.3\*0.176  = 0.404 | 0 +1.47\*0.176  = 0.258 | 1\* 0.176 + 0  = 0.176 |  |

Ranking =

**d1**

**d2**

**d3**

**h)** Calculate similarity of each document with the query using maximum likelihood estimate using Witten-Bell smoothing. (use three document corpus given above) [2 Points]

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Maximum Likelihood Estimates** | **Weights** | **Witten-Bell smoothing** |
| **d1** | maximum likelihood estimates  information : 1/6 = 0.167  retrieval : 2/6 = 0.33 | N / N+V = 6 /11 = 0.54 | Information : 0.167 \*0.54 + 0.11\* 0.46 = 0.14  Retrieval : 0.33 \* 0.54 + 0.27 \* 0.46  = 0.3  0.14\*0.3 = **0.042** |
| **d2** | maximum likelihood estimates  information : 0/7 = 0  retrieval : 3/7 = 0.43 | N / N+V = 7 /12 = 0.58 | Information : 0 \*0.58 + 0.11\* 0.42 =  0.046  Retrieval : 0.43 \* 0.58 + 0.27 \* 0.42  =0.36  0.046 \* 0.36 = **0.0165** |
| **d3** | maximum likelihood estimates  information : 1/5 = 0.2  retrieval : 0/5 = 0 | N / N+V = 5 / 9 = 0.55 | Information : 0.2 \*0.55 + 0.11\* 0.45 = 0.16  Retrieval : 0 \* 0.55 + 0.27 \* 0.45  = 0.12  0.16 \* 0.12 = **0.019** |

background probabilities

information : 2/18 = 0.11

retrieval : 5/18 = 0.27

Ranking:

**d1**

**d3**

**d2**