University of Texas at Dallas

Department of Computer Science

CS6322 – Information Retrieval

Spring 2022

Instructor: Dr. Sanda Harabagiu

Mid-Term Exam

Issued: March 10th 2022 at Noon

Due: March 12th 2022 – before Midnight

**Instructions**: Do not communicate with anyone in any shape or form. This is an independent exam. Do not delete any problem formulation, just write your answer immediately after each question in the space provided. **You are required to type your answers**. If the problem is deleted and you send only the answer, you shall receive ZERO points. If you do not write your name and netid, it will be considered that you did not submit your midterm exam, and will obtain ZERO points.

Copy and paste the Midterm Exam into a Word document, enter your answers (either by typing in MS Word) and transform the file of the exam into a PDF format. Submit the PDF file with the name **MidTerm\_Exam\_netID.pdf**, where netID is your unique netid provided by UTD. If you submit your exam in any other format your will receive ZERO points. If you handwrite your answers you shall receive ZERO points.

The MidTerm exam shall be submitted in eLearning before the deadline. No late submissions shall be graded! Any cheating attempt will determine the ENTIRE grade of the final exam to become ZERO.

**Submit in eLEarning as PDF file**

**DO NOT DELETE ANY PROBLEM, Simply add your answers!!!!!**

If you submit only the solutions with no problems, you will receive 0 points!!!!

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Problem 1 : (Stemming) (2 points)

Compute the measure required by the Porter Stemmer for the following strings:

1. enormously (**1 point**)
2. year (**1 point**)

Hint: show all details of how you obtained the measure!

SOLUTION 1.(a):

**V**-String of Vowels. Note: Letter ‘Y’ followed by a consonant is considered a vowel

**C-** String of Consonant.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| E | N | O | R | M | O | U | S | L | Y |
| **v** | **C** | **V** | **C** | **C** | **V** | **V** | **C** | **C** | **V** |

We are looking for string of Vowels followed by Consonants, and we can find 3 such combinations of VC.

🡪

This is how I reduced the string

VCVCCVVCCV

VCVCVCV (Considering consecutive C’s as C, and consecutive V’s as V)

[VC]3 V

We can reduce this to **[VC]3 V**

So, the measure ‘m’, require by the porter stemmer is **3**

SOLUTION 1.(b):

**V**-String of Vowels, **C-** String of Consonant. Note: Letter ‘Y’ followed by a consonant is considered a vowel

|  |  |  |  |
| --- | --- | --- | --- |
| Y | E | A | R |
| **C** | **V** | **V** | **C** |

We are looking for string of Vowels followed by Consonants, and we can find 3 such combinations of VC.

🡪

This is how I reduced the string

CVVC

CVC (Considering consecutive C’s as C, and consecutive V’s as V)

C[VC]1

We can reduce this to **C[VC]1**

So, the measure ‘m’, require by the porter stemmer is **1**

# Problem 2 : (Tolerant Retrieval) (10 points)

You have created the index of a collection of documents resulting in:

allow → 2 → 4 → 6 figuratively → 5 →6

friendly → 1 → 3 → 5

show → 1 → 2 → 4 → 6 shower → 2 → 5

showroom → 3 → 4 → 5

1. You would like to allow tolerant retrieval for searching this collection. What permuterms should you consider?

Detail your solution. (**6 points**)

Hint: Show which permuterms were obtained for each term.

1. What documents are retrieved when processing the following queries:

Q1: \*ow OR \*ly

Q2: sho\* AND \*er

Q3: f\*ly

Q4:\*ra\*

Provide ALL the details that allowed you to find the retrieved documents (**4 points**)

SOLUTION 2.(a):

Steps to obtain permuterms for a term are:

1. Append ‘$’ at the end of the term
2. Rotate the resultant string left. (It is noticed that the ‘$’ sign moves in a diagonal fashion).

As a result of these rotations we will have,

Number of permuterms = length(term)+1

|  |  |  |
| --- | --- | --- |
| allow:  allow$  llow$a  low$al  ow$all  w$allo  $allow | figuratively:  figuratively$  iguratively$f  guratively$fi  uratively$fig  ratively$figu  atively$figur  tively$figura  ively$figurat  vely$figurati  ely$figurativ  ly$figurative  y$figurativel  $figuratively | friendly:  friendly$  riendly$f  iendly$fr  endly$fri  ndly$frie  dly$frien  ly$friend  y$friendl  $friendly |
| show:  show$  how$s  ow$sh  w$sho  $show | shower:  shower$  hower$s  ower$sh  wer$sho  er$show  r$showe  $shower | showroom:  showroom$  howroom$s  owroom$sh  wroom$sho  room$show  oom$showr  om$showro  m$showroo  $showroom |

4

6

2

|  |
| --- |
| allow$ |
| llow$a |
| low$al |
| ow$all |
| w$allo |
| $allow |

|  |
| --- |
| allow |

SOLUTION 2.(b):

Q1.

\*ow 🡪ow$\* (Using rule: \*X lookup on X$\*)

ow$all🡺allow🡺 2🡪4🡪6

ow$sh🡺show🡺 1🡪2🡪4🡪6

**We take an ‘OR’ between the two, because we find the occurrence of our query in both the terms of the dictionary**

**Documents retrieved from query “\*ow” are: 1,2,4,6**

\*ly 🡪ly$\* (Using rule: \*X lookup on X$\*)

ly$friend🡺friendly🡺 1🡪3🡪5

ly$figurative🡺figuratively🡺 5🡪6

**We take an ‘OR’ between the two, because we find the occurrence of our query in both the terms of the dictionary**

**Documents retrieved from query “\*ly” are: 1,3,5,6**

\*ow OR \*ly: 🡺(1,2,4,6) OR (1,3,5,6)

🡺 **1,2,3,4,5,6**

Q2.

sho\* 🡪$sho\* (Using rule: \*X lookup on $X\*)

$show🡺show🡺 1🡪2🡪4🡪6

$shower-->shower🡺 2🡪5

$showroom-->showroom-->3🡪4🡪5

**We take an ‘OR’ between these 3, because we find the occurrence of our query in both the terms of the dictionary**

**Documents retrieved from query “sho\*” are: 1,2,3,4,5,6**

\*er 🡪er$\* (Using rule: \*X lookup on X$\*)

er$show-->shower--> 2🡪5

**Documents retrieved from query “\*ly” are: 2,5**

sho\* AND \*er: 🡺(1,2,3,4,5,6) AND (2,5)

🡺 **2,5**

Q3.

f\*ly 🡪ly$f\* (Using rule: X\*Y lookup on Y$X\*)

ly$figurative-->figuratively--> 5🡪6

ly$friend-->friendly--> 1🡪3🡪5

**We take an ‘OR’ between the two, because we find the occurrence of our query in both the terms of the dictionary**

**Documents retrieved from query “f\*ly” are: 1,3,5,6**

f\*ly : 🡺(1,3,5,6)

🡺 **1,3,5,6**

Q4.

\*ra\* 🡪ra\* (Using rule: \*X\* lookup on X\*)

ratuvely$figu-->figuratively--> 5🡪6

**We take an ‘OR’ between these 3, because we find the occurrence of our query in both the terms of the dictionary**

**Documents retrieved from query “\*ra\*” are: 5,6**

\*ra\* : 🡺(2,5)

🡺 **2,5**

# Problem 3 : (Spelling Correction) (10 points)

1. Compute the edit distance between “commander” and “commotion”. Give all the details of the computation.

(**9 points**)

1. What is the SOUNDEX code for “Spencer”. Show who you have obtained the code! (1 **point**)

SOLUTION 3.(a): **Answer- edit distance = 5**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | C | O | M | M | A | N | D | E | R |
|  | |  |  | | --- | --- | |  |  | |  | 0 | | |  |  | | --- | --- | |  |  | | 1 | 1 | | |  |  | | --- | --- | |  |  | | 2 | 2 | | |  |  | | --- | --- | |  |  | | 3 | 3 | | |  |  | | --- | --- | |  |  | | 4 | 4 | | |  |  | | --- | --- | |  |  | | 5 | 5 | | |  |  | | --- | --- | |  |  | | 6 | 6 | | |  |  | | --- | --- | |  |  | | 7 | 7 | | |  |  | | --- | --- | |  |  | | 8 | 8 | | |  |  | | --- | --- | |  |  | | 9 | 9 | |
| C | |  |  | | --- | --- | |  | 1 | |  | 1 | | |  |  | | --- | --- | | 0 | 2 | | 2 | 0 | | |  |  | | --- | --- | | 2 | 3 | | 1 | 1 | | |  |  | | --- | --- | | 3 | 4 | | 2 | 2 | | |  |  | | --- | --- | | 4 | 5 | | 3 | 3 | | |  |  | | --- | --- | | 5 | 6 | | 4 | 4 | | |  |  | | --- | --- | | 6 | 7 | | 5 | 5 | | |  |  | | --- | --- | | 7 | 8 | | 6 | 6 | | |  |  | | --- | --- | | 8 | 9 | | 7 | 7 | | |  |  | | --- | --- | | 9 | 10 | | 8 | 8 | |
| O | |  |  | | --- | --- | |  | 2 | |  | 2 | | |  |  | | --- | --- | | 2 | 1 | | 3 | 1 | | |  |  | | --- | --- | | 0 | 2 | | 2 | 0 | | |  |  | | --- | --- | | 2 | 3 | | 1 | 1 | | |  |  | | --- | --- | | 3 | 4 | | 2 | 2 | | |  |  | | --- | --- | | 4 | 5 | | 3 | 3 | | |  |  | | --- | --- | | 5 | 6 | | 4 | 4 | | |  |  | | --- | --- | | 6 | 7 | | 5 | 5 | | |  |  | | --- | --- | | 7 | 8 | | 6 | 6 | | |  |  | | --- | --- | | 8 | 9 | | 7 | 7 | |
| M | |  |  | | --- | --- | |  | 3 | |  | 3 | | |  |  | | --- | --- | | 3 | 2 | | 4 | 2 | | |  |  | | --- | --- | | 2 | 1 | | 3 | 1 | | |  |  | | --- | --- | | 0 | 2 | | 2 | 0 | | |  |  | | --- | --- | | 1 | 3 | | 1 | 1 | | |  |  | | --- | --- | | 3 | 4 | | 2 | 2 | | |  |  | | --- | --- | | 4 | 5 | | 3 | 3 | | |  |  | | --- | --- | | 5 | 6 | | 4 | 4 | | |  |  | | --- | --- | | 6 | 7 | | 5 | 5 | | |  |  | | --- | --- | | 7 | 8 | | 6 | 6 | |
| M | |  |  | | --- | --- | |  | 4 | |  | 4 | | |  |  | | --- | --- | | 4 | 3 | | 5 | 3 | | |  |  | | --- | --- | | 3 | 2 | | 4 | 2 | | |  |  | | --- | --- | | 1 | 1 | | 3 | 1 | | |  |  | | --- | --- | | 0 | 2 | | 2 | 0 | | |  |  | | --- | --- | | 2 | 3 | | 1 | 1 | | |  |  | | --- | --- | | 3 | 4 | | 2 | 2 | | |  |  | | --- | --- | | 4 | 5 | | 3 | 3 | | |  |  | | --- | --- | | 5 | 6 | | 4 | 4 | | |  |  | | --- | --- | | 6 | 7 | | 5 | 5 | |
| O | |  |  | | --- | --- | |  | 5 | |  | 5 | | |  |  | | --- | --- | | 5 | 4 | | 6 | 4 | | |  |  | | --- | --- | | 3 | 3 | | 5 | 3 | | |  |  | | --- | --- | | 3 | 2 | | 4 | 2 | | |  |  | | --- | --- | | 2 | 1 | | 3 | 1 | | |  |  | | --- | --- | | 1 | 2 | | 2 | 1 | | |  |  | | --- | --- | | 2 | 3 | | 2 | 2 | | |  |  | | --- | --- | | 3 | 4 | | 3 | 3 | | |  |  | | --- | --- | | 4 | 5 | | 4 | 4 | | |  |  | | --- | --- | | 5 | 6 | | 5 | 5 | |
| T | |  |  | | --- | --- | |  | 6 | |  | 6 | | |  |  | | --- | --- | | 6 | 5 | | 7 | 5 | | |  |  | | --- | --- | | 5 | 4 | | 6 | 4 | | |  |  | | --- | --- | | 4 | 3 | | 5 | 3 | | |  |  | | --- | --- | | 3 | 2 | | 4 | 2 | | |  |  | | --- | --- | | 2 | 2 | | 3 | 2 | | |  |  | | --- | --- | | 2 | 3 | | 2 | 2 | | |  |  | | --- | --- | | 3 | 4 | | 3 | 3 | | |  |  | | --- | --- | | 4 | 5 | | 4 | 4 | | |  |  | | --- | --- | | 5 | 6 | | 5 | 5 | |
| I | |  |  | | --- | --- | |  | 7 | |  | 7 | | |  |  | | --- | --- | | 7 | 6 | | 8 | 6 | | |  |  | | --- | --- | | 6 | 5 | | 7 | 5 | | |  |  | | --- | --- | | 5 | 4 | | 6 | 4 | | |  |  | | --- | --- | | 4 | 3 | | 5 | 3 | | |  |  | | --- | --- | | 3 | 3 | | 4 | 3 | | |  |  | | --- | --- | | 3 | 3 | | 4 | 3 | | |  |  | | --- | --- | | 3 | 4 | | 4 | 3 | | |  |  | | --- | --- | | 4 | 5 | | 4 | 4 | | |  |  | | --- | --- | | 5 | 6 | | 5 | 5 | |
| O | |  |  | | --- | --- | |  | 8 | |  | 8 | | |  |  | | --- | --- | | 8 | 7 | | 9 | 7 | | |  |  | | --- | --- | | 6 | 6 | | 8 | 6 | | |  |  | | --- | --- | | 6 | 5 | | 7 | 5 | | |  |  | | --- | --- | | 5 | 4 | | 6 | 4 | | |  |  | | --- | --- | | 4 | 4 | | 5 | 4 | | |  |  | | --- | --- | | 4 | 4 | | 5 | 4 | | |  |  | | --- | --- | | 4 | 4 | | 5 | 4 | | |  |  | | --- | --- | | 4 | 5 | | 5 | 4 | | |  |  | | --- | --- | | 5 | 6 | | 5 | 5 | |
| N | |  |  | | --- | --- | |  | 9 | |  | 9 | | |  |  | | --- | --- | | 9 | 8 | | 10 | 8 | | |  |  | | --- | --- | | 8 | 7 | | 9 | 7 | | |  |  | | --- | --- | | 7 | 6 | | 8 | 6 | | |  |  | | --- | --- | | 6 | 5 | | 7 | 5 | | |  |  | | --- | --- | | 5 | 5 | | 6 | 5 | | |  |  | | --- | --- | | 4 | 5 | | 6 | 4 | | |  |  | | --- | --- | | 5 | 5 | | 5 | 5 | | |  |  | | --- | --- | | 5 | 5 | | 6 | 5 | | |  |  | | --- | --- | | 5 | 6 | | 6 | **5** | |

SOLUTION 3.(b):

Following these steps for Soundex

1. Retain the first letter of the word.

2. Change all occurrences of the following letters to '0'

(zero):

'A', E', 'I', 'O', 'U', 'H', 'W', 'Y'.

3. Change letters to digits as follows:

B, F, P, V → 1

C, G, J, K, Q, S, X, Z → 2

D,T → 3

L → 4

M, N → 5

R → 6

4. Remove all pairs of consecutive digits.

5. Remove all zeros from the resulting string.

6. Pad the resulting string with trailing zeros and

return the first four positions, which will be of the

form <uppercase letter> <digit> <digit> <digit>.

SPENCER

**Using Rule 1, we will retain the first letter ‘S’**

P becomes 1 (Using Rule 3)

E becomes 0 (Using Rule 2)

N becomes 5 (Using Rule 3)

C becomes 2 (Using Rule 3)

E becomes 0 (Using Rule 2)

R becomes 6 (Using Rule 3)

* **So the word becomes:** S105206

**Now, we will remove all Zeros (Rule 5)**

**So the word becomes:** S1526

**There are no consecutive pair of digits, so return first 4 vales**

**So our word will become:** S152

# Problem 4 : (54 points)

Consider the following three short documents:

Doc #1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Probiotics help digest the milk sugar lactose.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Doc #2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Probiotics in yogurt may boost the functioning of the immune system.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Doc #3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

If you want to give your yogurt a nutritional upgrade, go for Greek yogurt.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (**18 points**) Tokenize manually the document collection, and identify the terms after you remove the following stop words: “the”, “a”, “in”, “of”, “may”, “to”, “for”, “your” . In this way “probiotics” becomes “probiotic”, “functioning” becomes “function” etc.
   1. Generate the dictionary of the collection and list it, adding also the document frequency (**5 points**) Use the format:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Index | 1 | 2 | … |  |  |  |
| Term | boost | … |  |  |  |  |
| DF | 1 | … |  |  |  |  |

* 1. Represent the 3 documents as document vectors by computing three weights:

1. binary weights (**3 points**);
2. raw weights (**3 points**); and
3. TF-IDF weights (**7 points**).

Use the format: Doc# ?

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Index | 1 | 2 | … |  |  |  |
| Weight | ??? | ??? |  |  |  |  |

For each form of weighting list the document vectors in the following format:

SOLUTION 4.A:

The Dictionary:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Term | boost | digest | function | give | go | greek | help | if | immune | lactose | milk | nutrition |
| DF | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| probiotic | sugar | system | upgrade | want | yogurt | you |
| 2 | 1 | 1 | 1 | 1 | 3 | 1 |

The document vectors:

(i)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| Doc#1  weight | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| Doc#2  Weight | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| Doc#3  Weight | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

(ii)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| Doc#1  Weight | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| Doc#2  Weight | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| Doc#3  Weight | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2 | 1 |

(iii)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Doc#1  Weight | 0 | 0.4771 | 0 | 0 | 0 | 0 | 0.4771 | 0 | 0 | 0.4771 |
| Doc#2  Weight | 0.4771 | 0 | 0.4771 | 0 | 0 | 0 | 0 | 0 | 0.4771 | 0 |
| Doc#3  Weight | 0 | 0 | 0 | 0.4771 | 0.4771 | 0.4771 | 0 | 0.4771 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 0.4771 | 0 | 0.1761 | 0.4771 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0.1761 | 0 | 0.4771 | 0 | 0 | 0.1761 | 0 |
| 0 | 0.4771 | 0 | 0 | 0 | 0.4771 | 0.4771 | 0.2291 | 0.4771 |

1. (**15 points**) What are the hit lists for the following Boolean queries (in each case explain how you obtained them from the inverted index):

Q1. yogurt AND immune AND system (5 points)

Q2. (yogurt AND nutrition) OR (probiotic AND milk) (5 points) Q3. (sugar AND milk AND probiotic) OR nutrition (5 points)

SOLUTION 4.B:

Q1.

Yogurt: 2, 3

Immune: 2

System: 2

(yogurt and (immune and system)) => ((2,3) and (**(2) and (2)**))

=> (**(2,3) and (2)**)

=> **2**

Q2.

Yogurt: 2, 3

Nutrition: 3

Probiotic: 1,2

Milk: 1

(yogurt AND nutrition) OR (probiotic AND milk) => (**((2,3) AND (3))** OR **((1,2) AND (1))**)

=> (3 OR 1)

=> **1,3**

Q3.

Sugar: 1

Milk: 1

Probiotic: 1,2

Nutrition: 3

(sugar AND milk AND probiotic) OR nutrition => (**(1) AND (1) AND (1,2)**) OR (3)

=> (1 OR 3)

=> **1,3**

1. (**21 points**) Compute the similarity between Q3 from 4.B and each of the three documents from 4.A using: (i) the **lnc.ltc** scoring (5 points for each document); (ii) the Jaccard coefficient similarity (2 points for document).

SOLUTION 1.D:

(i)

A picture containing text, clock

Description automatically generateds

For Cos(Q3,D1):

Query (Q3) = (Sugar AND milk AND probiotic) OR (Nutrition)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | tf\_row | tf\_wt | df | idf | tf\_idf | Normalize |
| Milk | 1 | 1 | 1 | 0.4721 | 0.4721 | 0.5646 |
| Nutrition | 1 | 1 | 1 | 0.4771 | 0.4771 | 0.5646 |
| Probiotic | 1 | 1 | 2 | 0.1761 | 0.1761 | 0.2084 |
| Sugar | 1 | 1 | 1 | 0.4771 | 0.4771 | 0.5646 |

Document 1 (D1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | tf\_row | tf\_wt | wt | normalize | product |
| Milk | 1 | 1 | 1 | 0.5774 | 0.326 |
| Nutrition | 0 | 0 | 0 | 0 | 0 |
| Probiotic | 1 | 1 | 2 | 0.5774 | 0.1203 |
| Sugar | 1 | 1 | 1 | 0.5774 | 0.326 |

Cos(Q3,D1)= 0.7723

For Cos(Q3,D2):

Query (Q3) = (Sugar AND milk AND probiotic) OR (Nutrition)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | tf\_row | tf\_wt | df | idf | tf\_idf | normalize |
| Milk | 1 | 1 | 1 | 0.4721 | 0.4721 | 0.5646 |
| Nutrition | 1 | 1 | 1 | 0.4771 | 0.4771 | 0.5646 |
| Probiotic | 1 | 1 | 2 | 0.1761 | 0.1761 | 0.2084 |
| Sugar | 1 | 1 | 1 | 0.4771 | 0.4771 | 0.5646 |

Document 2 (D2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | tf\_row | tf\_wt | wt | normalize | product |
| Milk | 0 | 0 | 0 | 0 | 0 |
| Nutrition | 0 | 0 | 0 | 0 | 0 |
| Probiotic | 1 | 1 | 2 | 1 | 0.2084 |
| Sugar | 0 | 0 | 0 | 0 | 0 |

Cos(Q3,D2) = 0.2084

For Cos(Q3,D3):

Query (Q3) = (Sugar AND milk AND probiotic) OR (Nutrition)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | tf\_row | tf\_wt | df | idf | tf\_idf | normalize |
| Milk | 1 | 1 | 1 | 0.4721 | 0.4721 | 0.5646 |
| Nutrition | 1 | 1 | 1 | 0.4771 | 0.4771 | 0.5646 |
| Probiotic | 1 | 1 | 2 | 0.1761 | 0.1761 | 0.2084 |
| Sugar | 1 | 1 | 1 | 0.4771 | 0.4771 | 0.5646 |

Document 3 (D3)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | tf\_row | tf\_wt | wt | normalize | product |
| Milk | 0 | 0 | 0 | 0 | 0 |
| Nutrition | 1 | 1 | 1 | 1 | 0.5646 |
| Probiotic | 0 | 0 | 0 | 0 | 0 |
| Sugar | 0 | 0 | 0 | 0 | 0 |

Cos(Q3, D3) = 0.5646

(ii) Jaccard Coefficients:

JC(Q3,D1)= (**Q3:** Sugar, Milk, Probiotic, Nutrition)

(**D1:** Probiotic, Help, Digest, Milk, Sugar, Lactose)

= (Q3 ∩ D1)/ (Q3 ∪ D1)

= 3/7

= 0.428

JC(Q3,D2)= (**Q3:** Sugar, Milk, Probiotic, Nutrition)

(D2: Probiotic, Yogurt, Boost, Function, Immune, System)

= (Q3 ∩ D2)/(Q3 ∪ D2)

= 1/9

= 0.111

JC(Q3,D3)= (**Q3:** Sugar, Milk, Probiotic, Nutrition)

(**D3:** If, You, Want, Give, You, Yogurt, Nutrition, Upgrade, go, Greek, Yogurt)

= (Q3 ∩ D3)/(Q3 ∪ D3)  
 =1/14  
 =0.071

# Problem 5 : (24 points);

Suppose you have a collection of 5 documents, and only 10 terms are used:

Term1 Term2 Term3 Term4 Term5 Term6 Term7 Term8 Term9 Term10

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DOC1 0 3 2 4 0 5 0 0 4 2

DOC2 3 0 1 4 3 0 0 5 1 6

DOC3 6 0 5 1 2 0 2 5 0 7

DOC4 1 8 0 2 0 1 6 0 2 1

DOC5 2 7 0 0 0 3 0 2 3 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

List the values of the gaps for the last three terms in your index computed for this collection. (**1 point**)

Encode these gaps with (i) unary codes (**3 points**); (ii) Gamma codes (**10 points**); and (iii) Delta codes (**10 points**). You are allowed to write a program to enable you computing the codes. Please add to the exam the code of the program if you chose to use one.

SOLUTION 2.I:

For Index:

*term->DOCid*

term1-> [2, 3, 4, 5]

term2-> [1, 4, 5]

term3-> [1, 2, 3]

term4-> [1, 2, 3, 4]

term5-> [2, 3]

term6-> [1, 4, 5]

term7-> [3, 4]

term8-> [2, 3, 5]

term9-> [1, 2, 4, 5]

term10-> [1, 2, 3, 4]

Gaps for Term 8 and the unary codes for the gaps:

Gaps for term 8: [2,1,2]

We find these gaps by calculating the difference of other values from previous index term

Their respective Unary codes for gaps will have ‘n’ one’s followed by a zero

term 8: [110,10,110]

Gaps for Term 9 and the unary codes for the gaps:

Gaps for Term 9 and the unary codes for the gaps:

Gaps for term 9: [1,1,2,1]

We find these gaps by calculating the difference of other values from previous index term

Their respective Unary codes for gaps will have ‘n’ one’s followed by a zero

term 9: [10,10,110,10]

Gaps for Term 10 and the unary codes for the gaps:

Gaps for term 10: [1,1,1,1]

We find these gaps by calculating the difference of other values from previous index term

Their respective Unary codes for gaps will have ‘n’ one’s followed by a zero

term 10: [10,10,10,10]

SOLUTION 2.II:

Gamma codes for the gaps in the posting files of Term 8:

Gamma code of Gap 2 :

a. Binary form of Gap = 10

Offset = 0 < get it by removing the most significant 1>

b. Length of offset = 1

length in unary form = 10

Gamma code of Gap 2 = “unary length” + “offset”

= 100

Gamma code of Gap 1 :

a. Binary form of Gap = 1

Offset = 0 < get it by removing the most significant 1>

b. Length of offset = 0

length in unary form = 0

Gamma code of Gap 1 = “unary length” + “offset”

= 0

Gamma code of Gap 2:

a. Binary form of Gap = 10

Offset = 0 < get it by removing the most significant 1>

b. Length of offset = 1

length in unary form = 10

Gamma code of Gap 2 = “unary length” + “offset”

= 100

For Term 8: Gamma Code **[100,0,100]**

Gamma codes for the gaps in the posting files of Term 9:

Gamma code of Gap 1 :

a. Binary form of Gap = 1

Offset = 0 < get it by removing the most significant 1>

b. Length of offset = 0

length in unary form = 0

Gamma code of Gap 1 = “unary length” + “offset”

= 0

Gamma code of Gap 1 :

a. Binary form of Gap = 1

Offset = 0 < get it by removing the most significant 1>

b. Length of offset = 0

length in unary form = 0

Gamma code of Gap 1 = “unary length” + “offset”

= 0

Gamma code of Gap 2 :

a. Binary form of Gap = 10

Offset = 0 < get it by removing the most significant 1>

b. Length of offset = 1

length in unary form = 10

Gamma code of Gap 2 = “unary length” + “offset”

= 100

Gamma code of Gap 1 :

a. Binary form of Gap = 1

Offset = 0 < get it by removing the most significant 1>

b. Length of offset = 0

length in unary form = 0

Gamma code of Gap 1 = “unary length” + “offset”

= 0

For Term 9: Gamma Code **[0,0,100,0]**

Gamma codes for the gaps in the posting files of Term 10:

Gamma code of Gap 1 :

a. Binary form of Gap = 1

Offset = 0 < get it by removing the most significant 1>

b. Length of offset = 0

length in unary form = 0

Gamma code of Gap 1 = “unary length” + “offset”

= 0

Gamma code of Gap 1 :

a. Binary form of Gap = 1

Offset = 0 < get it by removing the most significant 1>

b. Length of offset = 0

length in unary form = 0

Gamma code of Gap 1 = “unary length” + “offset”

= 0

Gamma code of Gap 1 :

a. Binary form of Gap = 1

Offset = 0 < get it by removing the most significant 1>

b. Length of offset = 0

length in unary form = 0

Gamma code of Gap 1 = “unary length” + “offset”

= 0

Gamma code of Gap 1 :

Binary form of Gap = 1

Offset = 0 < get it by removing the most significant 1>

Length of offset = 0

length in unary form = 0

Gamma code of Gap 1 = “unary length” + “offset”

= 0

For Term 9: Gamma Code **[0,0,0,0]**

SOLUTION 2.III:

Delta codes for the gaps in the posting files for Term 8:

For Gap 2:

Binary form of Gap: 10

Length of binary gap: 2

Length = gamma of length of binary form of gap : 100

Offset : 0 <after removing the most significant 1>

Delta code = Length, offset

= (100,0)

For Gap 1:

Binary form of Gap: 1

Length of binary gap: 1

Length = gamma of length of binary form of gap : 0

Offset : “ ” <after removing the most significant 1>

Delta code = Length, offset

= (0)

For Gap 2:

Binary form of Gap: 10

Length of binary gap: 2

Length = gamma of length of binary form of gap : 100

Offset : 0 <after removing the most significant 1>

Delta code = Length, offset

= (100,0)

Delta codes for term 8: **[1000, 0, 1000]**

Delta codes for the gaps in the posting files of Term 9:

For Gap 1:

Binary form of Gap: 1

Length of binary gap: 1

Length = gamma of length of binary form of gap : 0

Offset : “ ” <after removing the most significant 1>

Delta code = Length, offset

= (0)

For Gap 1:

Binary form of Gap: 1

Length of binary gap: 1

Length = gamma of length of binary form of gap : 0

Offset : “ ” <after removing the most significant 1>

Delta code = Length, offset

= (0)

For Gap 2:

Binary form of Gap: 10

Length of binary gap: 2

Length = gamma of length of binary form of gap : 100

Offset : 0 <after removing the most significant 1>

Delta code = Length, offset

= (100,0)

For Gap 1:

Binary form of Gap: 1

Length of binary gap: 1

Length = gamma of length of binary form of gap : 0

Offset : “ ” <after removing the most significant 1>

Delta code = Length, offset

= (0)

Delta codes for term 9: **[0,0,1000,0]**

Delta codes for the gaps in the posting files of Term 10:

For Gap 1:

Binary form of Gap: 1

Length of binary gap: 1

Length = gamma of length of binary form of gap : 0

Offset : “ ” <after removing the most significant 1>

Delta code = Length, offset

= (0)

For Gap 1:

Binary form of Gap: 1

Length of binary gap: 1

Length = gamma of length of binary form of gap : 0

Offset : “ ” <after removing the most significant 1>

Delta code = Length, offset

= (0)

For Gap 1:

Binary form of Gap: 1

Length of binary gap: 1

Length = gamma of length of binary form of gap : 0

Offset : “ ” <after removing the most significant 1>

Delta code = Length, offset

= (0)

For Gap 1:

Binary form of Gap: 1

Length of binary gap: 1

Length = gamma of length of binary form of gap : 0

Offset : “ ” <after removing the most significant 1>

Delta code = Length, offset

= (0)

Delta codes for term 9: **[0,0,0,0]**