Student Name: Student NetID:

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:
(a) enormously (**1 point**)
(b) year (**1 point**)

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

SOLUTION 1.(a):
1.(b):

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

(a) *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.

(b) *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):

The permuterms of *allow* are:

SOLUTION 2.(b):

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

(a) *Compute the edit distance between* "commander" *and* "commotion". *Give all the details of the computation.*
(**9 points**)

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

SOLUTION 2.(a):

SOLUTION 2.(b):

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.

A. (**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.
   a. 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	…				

   b. Represent the 3 documents as document vectors by computing three weights:
(i) binary weights (**3 points**);
(ii) raw weights (**3 points**); and
(iii) 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:

The document vectors:
(i)

(ii)

(iii)

B. (**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:

C. (**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)

Cos(Q3,D1)=

Cos(Q3,D2)=

Cos(Q3,D3)=

(ii) Jaccard Coefficients:

JC(Q3,D1)=

JC(Q3,D2)=

JC(Q3,D3)=

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:

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

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

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

SOLUTION 2.II:

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

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

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

SOLUTION 2.III:

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

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

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