Student Name: Student NetID:

University of Texas at Dallas
Department of Computer Science
CS6322 – Information Retrieval
Fall 2018
Instructor: Dr. Sanda Harabagiu

Take-Home Mid-Term Exam

Issued: October 10th 2018
Due: October 17th 2018 – before MidNight
Submit in eLEarning as PDF file
DO NOT DELETE ANY PROBLEM, Simply add your answers!!!!!
If you submit only solution with no problems, you will receive 0 points!!!!

Problem 1 :

Consider the following three short documents:
Doc #1

For a few days this summer, Alexa, the voice assistant who speaks to me through my Amazon Echo Dot, took to ending our interactions with a whisper: *Sweet dreams*

Doc #2

We're all falling for Alexa, unless we're falling for Google Assistant, or Siri, or some other genie in a smart speaker. When I say "smart," I mean the speakers possess artificial intelligence, can conduct basic conversations, and are hooked up to the internet, which allows them to look stuff up and do things for you.

Doc #3

Perhaps you think that talking to Alexa is just a new way to do the things you already do on a screen: shopping, catching up on the news, trying to figure out whether your dog is sick or just depressed. It's not that simple.

A. (**18 points**) First remove stop words and punctuation; parse manually the documents and select the terms which have the lexical roles **only of nouns**, **verbs, adjectives** or **adverbs.** Next, reduce all the terms to their lemmas, e.g. "interactions" becomes "interaction", "ending" becomes "end" etc. With the generation of the lemmas from the 3 documents you are ready to create the dictionary and present it (*5 points if the dictionary is correct*). Generate also the document vectors by computing three weights:
   (i) binary weights (*3 points*);
   (ii) raw weights (*3 points*); and
   (iii) TF-IDF weights (*7 points*).

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

	Term1	Term2	Term3	Term4	Term5	Term6	Term7	Term8 ...
DOC1	0	3	1	0	0	2	1	0 ….
DOC2	5	0	0	0	3	0	0	2 ….
DOC3	3	0	4	3	4	0	0	5 ….

SOLUTION 1.A:

The Dictionary:

The document vectors:
(i)

(ii)

(iii)

B. (**12 points**) Create an inverted (sorted list) index of the three documents, including the dictionary and the postings. The dictionary should also contain (for each term) statistics such as the Document Frequency and the total number of occurrences of the term in the collection (collection frequency). The postings for each term should contain the document id and the term frequencies in the respective document. **List the index for the first 6 terms from the dictionary only**. You do not need to list the entire index. You will get credit only for the first 6 terms from the dictionary and their respective inverted lists.

SOLUTION 1.B:

C. (**12 points**) What are the hit lists for the following Boolean queries (in each case explain how you obtained them from the inverted index):

Q1. Alexa AND simple AND screen (3 points)
Q2. (Alexa AND whisper) OR (speaker AND conversation) (3 points)
Q3. (Alexa AND smart AND intelligence) OR talk (3 points)
Q4. (fall OR think) AND (artificial OR screen) (3 points)

SOLUTION 1.C:

D. (**24 points**) Compute the similarity coefficients for each of the four queries from 1.C and each of the three documents using: (i) the cosine similarity with the lnc.ltc scheme – which stands for the SMART notation: *ddd.qqq* (4 points for each query); (ii) the Jaccard similarity (2 points for each query).

SOLUTION 1.D:

(i)

Cos(Q1,D1)=

Cos(Q1,D2)=

Cos(Q1,D3)=

Cos(Q2,D1)=

Cos(Q2,D2)=

Cos(Q2,D3)=

Cos(Q3,D1)=

Cos(Q3,D2)=

Cos(Q3,D3)=

Cos(Q4,D1)=

Cos(Q4,D2)=

Cos(Q4,D3)=

(ii) Jaccard Coefficients:

JC(Q1,D1)=

JC(Q1,D2)=

JC(Q1,D3)=

JC(Q2,D1)=

JC(Q2,D2)=

JC(Q2,D3)=

JC(Q3,D1)=

JC(Q3,D2)=

JC(Q3,D3)=

JC(Q4,D1)=

JC(Q4,D2)=

JC(Q4,D3)=

Problem 2 *(34 points):*
Suppose you have a collection of 5 documents, and only 10 terms are used in them:

	Term1	Term2	Term3	Term4	Term5	Term6	Term7	Term8	Term9	Term10
DOC1	0	3	5	4	0	5	0	0	4	2
DOC2	3	0	1	4	3	0	0	5	1	6
DOC3	5	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 four terms in your index computed for this collection. Encode these gaps with (i) unary codes (**8 points**); (ii) Gamma codes (**16 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 1:

Gaps for Term 2:

Gaps for Term 3:

Gaps for Term 4:

SOLUTION 2.II:

Gamma codes for Term 1:

Gamma codes for Term 2:

Gamma codes for Term 3:

Gamma codes for Term 4:

SOLUTION 2.III:

Delta codes for Term 1:

Delta codes for Term 2:

Delta codes for Term 3:

Delta codes for Term 4: