Name: _____ Std # _____

National University of Computer & Emerging Sciences

(Karachi Campus)

Information Retrieval & Text Mining (CS567)

Final Examination – Fall 2016 (Sol)

Course: IR&TM (CS567) Time Allowed: 180 Min.

Date: December 22, 2016 Max. Points: 100

Note: Attempt all questions. *Start each question on a new page of the answer book; answer all queries of the question in consecutive order. Answer to the point. Return this paper along with the answer book.*

Boolean Model +Term Vocabulary+ Posting List +Tolerant IR

Question No. 1 [Time:25 min] [Points: 5x4=20]

Answer the following questions briefly in 4-5 lines, answer to the point with importance of the real reasoning behind the concepts.

a. What are the few limitations of Boolean Retrieval Model?

-The Boolean model predicts that each document is either relevant or irrelevant. There is no notation for a partial match to the query.
- User must be aware of Boolean model and their connective in Boolean logic sense.
- Exact matching may leads to retrieval of too few or too many documents.
- It is difficult to rank the output, since all matched documents logically satisfy the query.
- It is difficult to perform relevance feedback.

b. Why term frequency* inverse document frequency (TF*IDF) is considered a good weighting scheme in vector space model?

The term frequency* inverse document frequency (TF*IDF) is considered the best weighting scheme in vector space model, because it assigns term "t" a weight in document "d" that is

1. Highest when occurs many times within a small number of documents (thus lending high discriminating power to those documents);
2. Lower when the term occurs fewer times in a document, or occurs in many documents (thus offering a less pronounced relevance signal);
3. Lowest when the term occurs in virtually all documents.

c. What are the benefits of stemming in information retrieval?

A stemming algorithm, or stemmer, has three main purposes in IR. The first one consists of clustering words according to their topic. Many words are derivations from the same stem and we can consider that they belong to the same concept (e.g., drive, driven, driver). The second purpose of a stemmer is directly related to the information retrieval process, as having the stems of the words allows some phases of the information retrieval process to be improved, among which we can highlight the ability to index the documents according to their topics, as their terms are grouped by stems (that are similar to concepts) or the expansion of a query to obtain more and more precise results. Finally, the conflation of the words sharing the same stem leads to a reduction of the dictionary.

d. How can positional index be used for phrase queries? Give an example.

Positional Index is a generalized data structure to support phrase queries. To process a phrase query, you still need to access the inverted index entries for each distinct term, now their positions are also stored. In the merge operation, the same general technique is used, but rather than simply checking that both terms are in a document, you also need to check that their positions of appearance in the document are compatible with the phrase query being evaluated. Consider the text "to be or not to be", for inverted index term to-> 1,5 in the given document and be-> 2,6 and if we search "to be" as a phrase it appears in the given document as at position 1-2 or 5-6.

Vector Space Model

Question No. 2 [Time:25 min] [Points: 10]

Consider the following set of documents:

D1: If you love life, life will love you back.
D2: If you love life, do not waste time.
D3: I love life and find fun all time.

Using the Stop-word-list = {If, you, will, do, not, I, and, all} answer the following questions:

a. Compute term frequency (tf) for each term in each document. [2]

Terms->term frequency	D1-tf	D2-tf	D3-tf
back	1	0	0
find	0	0	1
fun	0	0	1
life	2	1	1
love	2	1	1
time	0	1	1
waste	0	1	0

b. Computer simple document frequency (df) for each term in the collection. [2]

Terms->doc. frequency	Df
back	1
find	1
fun	1
life	3
love	3
time	2
waste	1

c. With a weighting scheme tf*df provide the document vectors of each document in the collection. [6]

Terms->term frequency	D1-tf*df	D2-tf*df	D3-tf*df
back	1	0	0
find	0	0	1
fun	0	0	1
life	6	3	3
love	6	3	3
time	0	2	2
waste	0	1	0

D1: <1,0,0,6,6,0,0>
D2: <0,0,0,3,3,2,1>
D3: <0,1,1,3,3,2,0>

IR Evaluation & Relevance Feedback

Question No. 3 [Time:20 min] [Points: 5+5]

a. What do we mean by relevance feedback? Why it is not possible for web scale information retrieval? [5]

The idea of relevance feedback (RF) is to involve the user in the retrieval process, using feedback on relevant or irrelevant result-set items for a query, use this knowledge to improve the next round of the retrieval against the same query from an IR system. Web users are often reluctant to provide feedback, it also prolong the search interaction, and the feedback effects on new results are not easy to comprehend, due to these reason RF often not possible at web scale.

b. Suppose that for a query q, there are total of 100 documents in the collection in which 8 are relevant. The search engine returns the following ordered list of documents as the retrieval result (where R means relevant and N means non-relevant document in the given order).

Result-set: R N R N N N N R

i. How much is the Mean Average Precision (MAP) for this query?

MAP= 1/8 (1/1+2/3+3/8) = 0.255

ii. What can be the maximum Mean Average Precision (MAP) for this query?

Maximum possible MAP will be when the system return the five left over relevant documents next to these 8, which is relevant documents at 9^{th}, 10^{th}, 11^{th}, 12th and 13th positions. Hence the MAP will be

MAP max = 1/8 X [1/1+2/3+3/8+4/9+5/10+6/11+7/12+8/13]

MAP max = 0.591

iii. What can be the minimum Mean Average Precision (MAP) for this query?

Minimum possible MAP will be when the system return the five left over relevant documents in the last, which is relevant documents at 96^{th}, 97^{th}, 98^{th}, 99^{th} and 100^{th} positions. Hence the MAP will be

MAP min= 1/8 X [1/1+2/3+3/8+4/96+5/97+6/98+7/99+8/100]

MAP min = 0.293

Probabilistic & Language Model

Question No. 4	[Time:30 min] [Points: 15]

a. How Probabilistic and Language Model for Information retrieval differ from vector space model? Illustrate with an example.[5]

Vector Space Model is a de facto of IR, where we used term frequency and document frequency in high-dimensional vector space to compute IR related mathematical functions. The document written in human natural languages may follow some probabilistic criteria, motivating on this ground the Probabilistic IR exploits the collection frequencies to measure and associated probability with the IR. The language model is more restricted in the sense that in a given language, words are coming from an alphabet set; hence some word has higher probability to appear in a certain language then other.

b. Suppose we have a collection that consists of the 4 documents given below:
D1: click go the shears boys click click click
D2: click click
D3: metal here
D4: metal shears click here

Build a query likelihood language model for this document collection. Assume a mixture model between the documents and the collection, with both weighted at 0.5 Maximum likelihood estimation (mle) is used to estimate both as unigram models. Calculate the model probabilities of the queries click, shears, and hence click shears for each document, and use those probabilities to rank the documents returned by each query. [10]

	click	**go**	**the**	**shears**	**boys**	**metal**	**here**
Mc	7/16	1/16	1/16	2/16	1/16	2/16	2/16
P(q/D1)	4/8	1/8	1/8	1/8	1/8	0	0
P(q/D2)	2/2	0	0	1	0	0	0
P(q/D3)	0	0	0	1	0	1/2	1/2
P(q/D4)	1/4	0	0	1/4	0	1/4	1/4

λ= 0.5

	D1	D2	D3	D4
click	1/2*(4/8+7/16)	1/2*(2/2+7/16)	1/2*(0+7/16)	1/2*(1/4+7/16)
shears	1/2*(1/8+2/16)	1/2*(0+2/16)	1/2*(0+2/16)	1/2*(1/8+2/16)
click shears	0.059	0.045	0.014	0.043

Click → D2, D1, D4, D3
Shears → D4, D1 D2 D3
Click Shear → D4, D1, D2, D3

Text Classification

Question No. 5 [Time:20 min] [Points: 15]

a. Define the vector space classification scheme called k-nearest neighbor classification. What are the advantages of this method? [5]

In k-nearest neighbor classification the documents are represented in vector space. The kNN classification assigns each document to the majority class of its k closest neighbors where k is a parameter, it is generally a small odd positive integer. The rationale of kNN classification is that, based on the contiguity hypothesis, we expect a test document d to have the same label as the training documents located in the local region surrounding d.

k-NN is robust to noisy training data, it is a method of choice for large classification datasets.

b. Consider the following examples for the task of text classification

	docID	words in document	in c = *China*?
training set	1	Taipei Taiwan	yes
	2	Macao Taiwan Shanghai	yes
	3	Japan Sapporo	no
	4	Sapporo Osaka Taiwan	no
test set	5	Taiwan Taiwan Sapporo	?

Using Bernoulli Naïve Bayes to estimate the probabilities of each term (feature) and hence classify the test set document. [10]

P(c) = 1/2 = 0.5
P(~c) = 1/2 =0.5

P(Taipei/c)=1/2	P(Taipei/~c)=1/4
P(Taiwan/c)=3/4	P(Taiwan/~c)=1/2
P(Macao/c)=1/2	P(Macao/~c)=1/4
P(Shanghai/c)=1/2	P(Shanghai/~c)=1/4
P(Japan/c)=1/4	P(Japan/~c)=1/2
P(Sapporo/c)=1/4	P(Sapporo/~c)=3/4
P(Osaka/c)=1/4	P(Osaka/~c)=1/2

P(c/D5) = p(c) X [p(Taiwan/c) x p(Sapporo/c) x (1-p(macao/c)x..x(1-p(Osaka/c)]
= 0.00329
P(~c/D5)= 0.00585

P(~c/D5) > P(c/D5) Hence D5 belong to class ~c

Text Clustering

Question No. 6	[Time:20 min] [Points: 10]

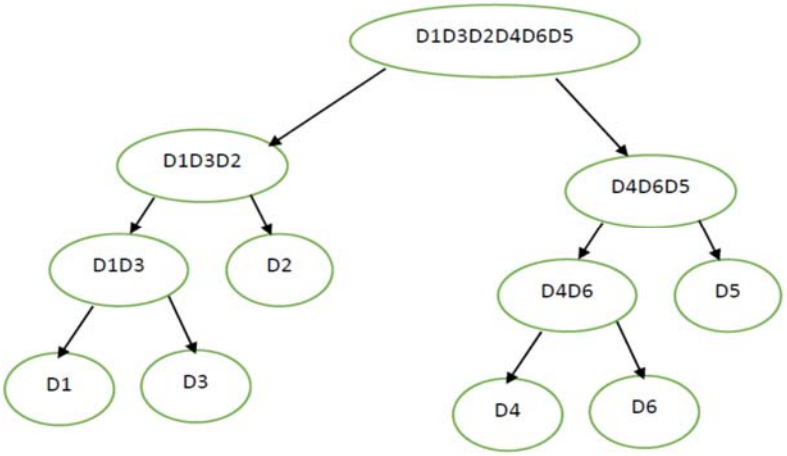
a. Differentiate between k-Mean and Agglomerative Hierarchical clustering algorithms.

1. Hierarchical and Partitional(k-Mean) Clustering have key differences in running time, assumptions, input parameters and resultant clusters.
2. k-Mean clustering is faster than hierarchical clustering.
3. Hierarchical clustering requires only a similarity measure, while k-Mean clustering requires stronger assumptions such as number of clusters and the initial centers.
4. Hierarchical clustering does not require any input parameters, while k-Mean clustering algorithms require the number of clusters to start running.
5. Hierarchical clustering returns a much more meaningful and subjective division of clusters but k-Mean clustering results in exactly k clusters.
6. Hierarchical clustering algorithms are more suitable for categorical data as long as a similarity measure can be defined accordingly.

b. Consider the following 6- documents in 2D space

$D_1(1,1)$, $D_2(1,3)$,$D_3(2,1)$, $D_4(4,3)$,$D_5(4,1)$ and $D_6(3,3)$

Using single-link bottom-up HAC clustering, draw the complete clustering hierarchy of the given dataset.



Web Search + Link Analysis

Question No. 7 [Time:30 min] [Points: 20]

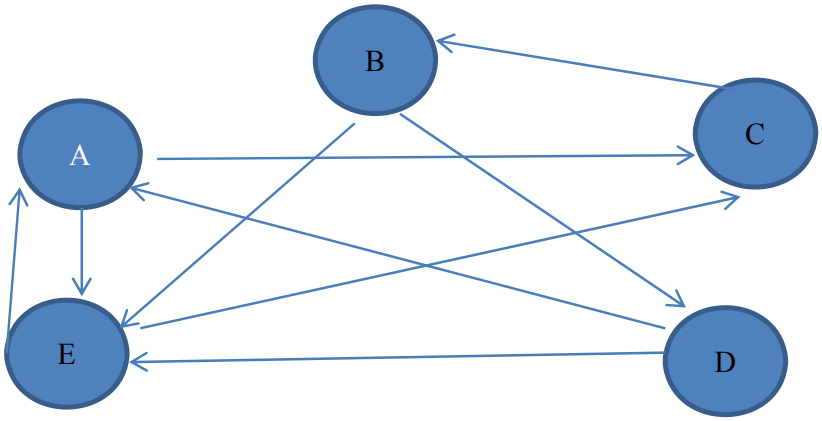
a. Illustrate at least four differences between HITS and PageRank algorithms for link analysis. [5]

HITS	PageRank
It gives 2 scores Hub and Authority for each page.	It gives one score e.g. PageRank.
It is executed at query time	It is executed at indexing time.
Not robust against spams.	Robust against web-spams.
Never favor pages, but can be manipulated.	Favor old pages.
It is query dependent	It is query independent

b. Consider the following segment of web graph for link analysis based on PageRank: Page A points to Page C and E ; Page B points to D and E ; Page C points to B Page D points to A and E; Page E points to A and C

Answer the following questions:

i. Give a pictorial representation of the given graph [2]



ii. Compute the adjacency matrix of the given graph [2]

	A	B	C	D	E
A	0	0	1	0	1
B	0	0	0	1	1
C	0	1	0	0	0
D	1	0	0	0	1
E	1	0	1	0	0

iii. Computer the probability matrix of (b) using $\alpha = .4$ [2]

P(A)= 0.4+0.5(P(D)/2 +P(E/2))=0.9, similarly
P(B)=1
P(C) =1
P(D)=0.75
P(E)=1.25
Hence Probability matrix will be
M=[A B C D E]=[0.9 1 1 0.75 1.25]

iv. Let a surfer start from page C, taking its vector Vc= <0 0 1 0 0> , compute an approximation of the PageRank scores of the pages of the given graph, using three iterations [9]

Using these values for three iterations we will get:

Iteration # 1

P(A)= 0.891
P(B)= 1.031
P(C)= 1.062
P(D)=0.7578
P(E)=1.197

Iteration # 2

P(A)=0.912
P(B)=1.023
P(C)=1.046
P(D)=0.755
P(E)=1.191

Iteration # 3

P(A)=0.912
P(B)=1.022
P(C)=1.044
P(D)=0.755
P(E)=1.191

<The End.>