National University of Computer & Emerging Sciences, Karachi
Spring-2018 CS-Department
Final Examination
May 14, 2018, 9AM – 12Noon

Course Code: CS317	**Course Name: Information Retrieval**
Instructor Name: Dr. Muhammad Rafi	
Student Roll No:	**Section No:**

- Return the question paper.
- Read each question completely before answering it. There are **8 questions and 3 pages.**
- In case of any ambiguity, you may make assumption. But your assumption should not contradict with any statement in the question paper.
- All the answers must be solved according to the sequence given in the question paper.
- Be specific, to the point and illustrate with diagram and code where necessary.

Time: 180 minutes. **Max Marks**: 100 points

Information Retrieval – Basics

Question No. 1 **[Time: 20 Min] [Marks: 10]**

Answer the following questions briefly using 4-5 lines of answer book. Be precise, accurate and to the point, only answer genuine query in the question. Each question is of 2 marks.

a. In your opinion, what is the reason of the popularity and success of vector space model in IR? [2.5]

The Vector Space Model (VSM) is based on rigor mathematical model of vector space (linear algebra). All operations on vector space are well-define. It compactly represents documents in a specified vector space. It allows continuous degree of similarity between queries and documents. It also allows ranking documents according to their possible relevance. Moreover, it allows partial matching. These are main reasons for its popularity and success for last 40 years.

b. Which indexing technique in IR, will be a good choice for proximity query (query of the form [t1 t2 /k])? Explain. [2.5]

The proximity query of the form [t1 t2 /k] where we want a term t1 and t2 should be less than k words apart in the text/document can very efficiently supported by positional indexing as it keeps track of each term and its relative position from the starting of the text/document. The main idea is if a document has both the term t1 and t2 and if the difference of their positions are less than k it is a valid match.

c. A bigram index is used to retrieve document for wildcard query "te*ti*al". Suggest how a Boolean query on a bigram index would look like for this? Give an example of term that may contain in the result-set. [2.5]

The Boolean query using bi-grams will be "$t AND te AND ti AND al AND l$"
Example is: testimonial

d. Differentiate between Block-Sort Based Indexing (BSBI) and Single-Pass In Memory Indexing (SPIMI) scheme. [2.5]

Block-Sort Based Indexing (BSBI)	**Single-Pass In Memory Indexing (SPIMI)**
- BSBI uses continuous disk space to collect all terms from document collections by dividing collection into equal parts, iteratively. - It uses a data structures to collect termID and docID into memory. - The running time is proportional to (T log T) where T is Number of terms in the collection. Dominated by sorting of terms in a collection.	- SPIMI add posting directly to posting list and small posting list are stored into the continuous disk blocks. - There is no need to map termID and docID pairs and hence no sorting is required. Faster and efficient. - The running time is linear in term of T(number of terms in the collection). - SPIMI also support compression of posting lists.

Information Retrieval Models

Question No. 2 **[Time: 30 Min] [Marks: 20]**

a. What are some of the drawbacks of Boolean Model for IR? How vector space model overcome these challenges? [5]

- Boolean Model works on exact matching. Textual documents generally, contains morphological variants and a partial matching is preferred, Vector Space Model (VSM) do partial matching.
- Boolean queries are hard to formulate while VSM works on simple text queries.
- Boolean Model produce flat results while VSM produce raking documents based on tf*idf weighting.

b. Consider the partial document collection D= {***d1***: w4 w5 w6 w1; ***d2:*** w3 w2 w1; ***d3:*** w7 w2 w1} and ***q:*** w4 w3 w7; if the following table gives the **tf** and **idf** score of each term, compute the score of each document against the given query, using cosine of angle between query vector and document vector. Also produce the ranking of the documents against this query. [15]

Word	tf-d1	tf-d2	tf-d3	idf
W1	0.10	0.17	0.12	0.34
W2	0.17	0.21	0.19	0.78
W3	0.23	0.34	0.14	0.81
W4	0.26	0.28	0.29	0.54
W5	0.15	0.65	0.55	0.90
W6	0.31	0.22	0.36	0..62
W7	0.23	0.45	0.27	0.45

First getting the documents vectors with tf*idf weighting as below:

d1 =< (0.10x0.34); 0; 0; (0.26x0.54); (0.15x0.9); (0.31x0.62); 0>
d2 =< (0.17x0.34);(0.21x0.78); (0.34x0.81); 0 ; 0; 0; 0>
d3 =< (0.12x0.34);(0.19x0.78); 0; 0; 0; 0; (0.27x0.45)>
q =(0; 0; 1; 1; 0; 0; 1> no tf*idf weighting for query.

Now,

Cos(d1,q) = d1 x q / (|d1|) x |q| = (0.26x.54) / (0.275x 1.7321) = 0.294
Cos(d2,q) = d2 x q / (|d2|) x |q| = 0.342
Cos(d1,q) = d1 x q / (|d1|) x |q| = 0.261

Hence, Ranking is d2,d3,d1

Evaluation in IR	
Question No. 3	**[Time: 25 Min] [Marks: 15]**

a. There are 20 relevant documents in a collection for a given query "q", The precision of the query is 0.20 and the recall is 0.25 Find How many documents in the results-set retrieved? How many of them would be relevant? [5]

we know,
precision = (relevant-retrieved) / (total-retrieved)
=> 0.2 = (relevant-retrieved) / (result-set) ---------- eq(i)
similarly,
recall = (relevant-retrieved)/ (total-relevant)
=> 0.25 = (relevant-retrieved)/ 20
=> relevant-retrieved= 0.25 * 20 = 5 hence eq(i) => result-set = 5 /0. 2 = 25
Result-set contain 25 documents from which, 5 are relevant.

b. The following list of Rs and Ns represents relevant (R) and non-relevant (N) returned documents in a ranked list of 10 documents retrieved in response to a query from a collection of 1,000 documents. The top of the ranked list (the documents the system thinks are most likely to be relevant) is on the left of the list. This list shows 4 relevant documents. Assume that there are 6 relevant documents in total in the collection for the given query. [10]

R R N N R N R N N N

1. What is the precision of the system on the top 10?

   From the given information, we can see that tp=4; fp=6 and fn=6-4=2 so for precision we have Precision = tp / (tp+fp) = 4/10 = 0.4

2. What is the F1 on the top 10?

   Let’s find recall for F1: we know Recall = tp/ (fn+tp) = 4/6= 0.66 hence
   F1= 2 X (Precision * Recall) / (Precision + Recall)
   F1= (2X0.4X0.66) / (0.4+0.66) = 0.528 / 1.06 = 0.498

3. What is the largest possible MAP that this system could have?

   The maximum MAP possible when the remaining two relevant documents retrieved next to these 10 documents.
   MAP = 1/6 * (1/1+2/2+3/5+4/7+5/11+6/12) = 4.1259/6 = 0.687

4. What is the smallest possible MAP that this system could have?
   The minimum MAP possible when the remaining four relevant documents found as the last documents from the collection.

   MAP = 1/6 * (1/1+2/2+3/5+4/7+5/999+6/1000) = 3.1824/6 = 0.530

Relevance Feedback

Question No. 4 **[Time: 25 Min] [Marks: 15]**

a. What is meant by query expansion? Give an example query that need expansion? When query expansion is very useful? [5]

Query expansion is an autonomous process of reformulating a seed query (q_o) to improve retrieval performance in information retrieval systems. It is generally performed to bridge the gap between user information need and the posed query. A general case is a lexical mismatch for example astronauts or cosmonauts are mean the same thing, the query for astronauts implicitly union with the term cosmonauts to bridge the gap and get the relevant documents from both the terms. It is very useful technique for such a situation.

b. Suppose that a user's initial query is q= w1 w3 w2 and IR systems return four documents. User selected d1= w2 w3 w4 and d4= w1 w3 w4 w1 as relevant. While d2= w3 w3 w4 w5 and d3= w2 w4 w5 w3 as non-relevant to her query. Assume that we are using direct term frequency (with no scaling and no document frequency). There is no need to length-normalize vectors. Using Rocchio relevance feedback algorithm to get modify query vector (optimal) after relevance feedback? Rocchio equation is given below. [10]

$$\vec{q}_m = \alpha \vec{q}_0 + \beta \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j$$

$q= < 1, 1, 1, 0, 0>$
$d_1= < 0, 1, 1, 1, 0>$
$d_2= < 0, 0, 2, 1, 1>$
$d_3= < 0, 1, 1, 1, 1>$
$d_4= < 2, 0, 1, 1, 0>$

Using the given equation, we will get,

$q_m= \alpha * < 1, 1, 1, 0, 0> + \beta * 1/2 * \{ < 0, 1, 1, 1, 0> + < 2, 0, 1, 1, 0>\} - \gamma * 1/2 * \{< 0, 0, 2, 1, 1> + < 0, 1, 1, 1, 1>\}$

$q_m= \alpha * < 1, 1, 1, 0, 0> + \beta * 1/2 * \{ < 2, 1, 2, 2, 0> \} - \gamma * 1/2 * \{< 0, 1, 3, 2, 2> \}$

$q_m= \alpha * < 1, 1, 1, 0, 0> + \beta * <1,1/2,1,1,0> - \gamma * \{< 0, 1/2, 3/2, 1, 1> \}$

$q_m= < \alpha + \beta , \alpha + \beta/2- \gamma/2 , \alpha + \beta-3/2 \gamma, \beta- \gamma, - \gamma>$

We need to put zero on all the dimensions where we have identifiable negative values:

$q_m= < \alpha + \beta , \alpha + \beta/2- \gamma/2 , \alpha + \beta-3/2 \gamma, \beta- \gamma, 0>$

Text Classification	
Question No. 5	**[Time: 20 Min] [Marks: 10]**

a. Why Naïve Bayes is regarded as a baseline classifier for text? [5]

Naive Bayes classifiers is a probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. It is highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. It is a popular baseline for text classification as it offers a standard classification on text, generally improvement can be performed with relaxing the assumptions.

b. How do we treat the words(features) that do not appear in training document for NB Classifiers? [5]

In a Naïve Bayes classifier, the training data never complete and hence there are terms(features/words) that do not appear in training data. The maximum likelihood estimate for such term is zero and hence we cannot be able to the compute probability for a given class. One of the solution to this problem is Laplace smoothing which simply adds one to each count, which is treated as uniform prior.

Text Clustering	
Question No. 6	**[Time: 20 Min] [Marks: 10]**

a. Differentiate between k-mean and HAC approaches to clustering. [5]

K-Mean	**HAC**
It is a partition clustering method. It produces a non-overlapping partition.	It is a hierarchical clustering method. It produces a hierarchy of clusters.
It takes k as number of clusters to be produced.	It performs merging of clusters to produce hierarchy, you can cut at a desire level to get the clusters.
The running time is proportional to number of data points for clustering.	The running time is polynomial.

b. Consider a collection of overly simplified documents d1(1,4); d2(2,4); d3(4,4); d4(1,1); d5(2,1) and d6(4,1). Apply k-means algorithm using seeds d2 and d5. What are the resultant clusters? How do we know that this result is optimal or not? [5]

Let C1=d2 and C2 = d5
Starting C1 and C2 as initial clusters, we need to decide about the membership for each of the documents.
For d1: Dist(C1, d1) < Dist(C2,d1) => d1 belongs to C1.
For d3: Dist(C1, d3) < Dist(C2,d3) => d3 belongs to C1.
For d4: Dist(C1, d4) > Dist(C2,d4) => d4 belongs to C2.
For d6: Dist(C1, d6) > Dist(C2,d6) => d6 belongs to C2.
Hence d1, d2 and d3 are in C1 cluster, and d4, d5 and d6 are in C2 cluster.
K-mean coverage to a local minimum, in order to find optimal clustering one need to produce all possible clustering arrangement.

Web Search & Crawler	
Question No. 7	**[Time: 20 Min] [Marks: 10]**

a. Illustrate the difference between a directory style search(yahoo) and text search (google). [4]

Directory Search (Yahoo)	**Text Search (Google)**
- Yahoo style, directory search maintains a large hierarchical directory of terms, a user need to follow the topic based hierarchical path to get the information.	- Google style, simple text based search, autonomously crawl and index pages on text terms. The query terms are process through index and relevant pages are returned as result.
- It is very challenging to maintain such large directory	- Indexing is easy as compared to directory maintenance
- Relevant results from exploration of the hierarchical path.	- Relevant results as per the retrieval approach (pagerank).

b. What are the different types of users queries on the web? Give example of each type of the query. [3]

Informational queries seek general information on a broad topic, such as leukemia or Provence. There is typically not a single web page that contains all the information sought; indeed, users with informational queries typically try to assimilate information from multiple web pages.

Navigational queries seek the website or home page of a single entity that the user has in mind, say Lufthansa airlines. In such cases, the user's expectation is that the very first search result should be the home page of Lufthansa.

A transactional query is one that is a prelude to the user performing a transaction on the Web – such as purchasing a product, downloading a file or making a reservation. In such cases, the search engine should return results listing services that provide form interfaces for such transactions

c. Outline at least three challenges of a modern web-crawler and suggest one solution to overcome each of them. [3]

The web-sites generally block access through automatic crawlers, the crawler must be polite in accessing these websites.

There can be several issues with physical access, network access and application layer of the web-host, a crawler must be robust to all these problem.

Crawler should be distributed and scalable as it need to access millions of pages per unit time.

Link Analysis

Question No. 8 **[Time: 20 Min] [Marks: 10]**

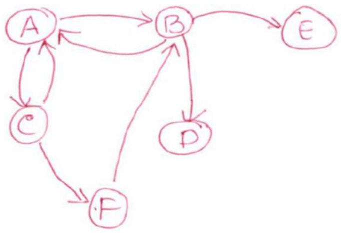
a. Illustrate at least four differences between HITS and PageRank algorithms for link analysis. [5]

HITS	**PageRank**
It gives two scores Hub and Authority for each page.	It gives one score per page.
It is executed at query time.	It is precomputed at indexing time.
It is query dependent.	It is independent from query.
It is not robust against web/link spams	It is robust against web-spams
Never favours pages, but can be manipulated for higher scores.	It favours old pages. It can also be manipulated.

b. Consider the graph information below with 6 pages (A; B; C; D; E; F) where:

A --> B; C
B --> A; D; E
C --> A; F
F --> B
Answer the following questions:

i. Give a pictorial representation of the given graph [1]



ii. Compute the adjacency matrix of the given graph [1]

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

iii. Assume that the PageRank values for any page p_i at iteration 0 is $PR(p_i) = 1$ and that the damping factor for iterations is $d = 0.85$ Perform the PageRank algorithm and determine the rank for every page after 2 iterations. [3]

$$= \begin{bmatrix} 0 & 1/3 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 1 \\ 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0.85 \\ 0.85 \\ 0.85 \\ 0.85 \\ 0.85 \\ 0.85 \end{bmatrix}$$

$$\begin{bmatrix} 0.71 \\ 1.28 \\ 0.43 \\ 6.28 \\ 0.28 \\ 0.43 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.64 \\ 0.79 \\ 0.36 \\ 0.42 \\ 0.42 \\ 0.22 \end{bmatrix}$$