BIRZEIT UNIVERSITY

Faculty of Engineering and Technology

Electrical and Computer Engineering Department

First Semester 2021/2022

Course 539: Special Topics: Information Retrieval and Web Search

Instructor: Dr Adnan Yahya. **Midterm Exam**

Time: 80 minutes max

Please answer the following questions using the exam sheets only.

Question	Q1	Q2	Q3	Q4	Q5	Total
ABET outcome	e	a	e	a		
Max grade	21	20	21	20	21	103
Earned						

Question 1 (21%): Consider that our document collection S has the following 3 documents (after stop word removal): Work with English only except for item 3.

D1: "large fast program" "برنامج سريع كبير"

D2: "large large fast train" قطار سريع كبير كبير

D3: "slow program" برنامج بطيء

For all the documents, calculate the tf scores for all the terms in S. Order the terms in the vectors alphabetically. Compute idf values with no log and ignore normalization for terms and values. Given the following query:

Q: "fast train program", برنامج قطار سريع

1. 9%: Complete the following table for these documents and Query. Sort Terms alphabetically:

Answer:

Word df	Idf= N/df=3/df	D1 tf↓ tf.idf	D2 tf↓ tf.idf	D3 tf↓ tf.idf	Q tf↓ tf.idf
W1 fast 2	3/2	1:3/2	1:3/2	0:0	1:3/2
W2 large 2	3/2	1:3/2	2:3	0:0	0:0
W3 program 2	3/2	1:3/2	0:0	1:3/2	1:3/2
W4 slow 1	3/1	0:0	0:0	1:3	0:0
W5 train 1	3/1	0:0	1:3	0:0	1:3

2. 9%: Rank D1, D2 and D3 with respect to the query **Q** according to the cosine metric and tf.idf. Explain why.

Answer:

|Q|=SQRT(9/4+9/4+9)= SQRT(27/2)=3.67;
|D1|=SQRT(9/4+9/4+9/4)= SQRT(27/4)=2.6
|D2|=SQRT(9/4+9+9)= SQRT(81/4)=9/2=4.5
|D3|=SQRT(9/4+9)= SQRT(45/4)=3.35

Answer:

Sim(Q,D1)/(|Q|*|D1|)=(9/4+9/4)/(3.67*2.6)=(9/2)/(3.67*2.6)=0.47-------(Rank:2)
Sim(Q,D2)/(|Q|*D2|)=(9/4+9)/(3.67*4.5)=11.25/3.67*4.5=0.68-----(Rank:1)
Sim(Q,D3)/(|Q|*D3|)=(9/4)/(3.67*3.35)=4.5/3.67*1.5=0.18----------(Rank:3)

3. 3% Give three possible words to extend the query **برنامج قطار سريع for better recall.**

Example: جدول تراين اكبرس [example: many others possible: add terms at small distance from any of the words of the query, from sysnsets, ….]

Question 2 (20%):

a. A search engine has a collection of 7,000,000 documents with 700 words per document, on average.

(i) 3% What is the minimal length for document IDs for the postings? In bits and in full bytes. Why?

Answer:

2**22 <7,000,000<2**23 so we need **23 bits and 3 bytes**.

Document length is irrelevant here!

(ii) 3% If the vocabulary size is 300,000 terms and the average dictionary word length is 10 characters How many **bits** do you need for pointers if one is to store the dictionary as a single string with pointers to the start of each **word** (what is the length of each pointer). In bits and in full bytes. Why?

Answer: 300,000 terms of 10 character/bytes on average: 3,000,000. Each pointer points to one of these 3,000,000 characters and for that we need **22 bits and 3 bytes**.

(iii) 3% Compute the γ-code and the variable byte code for the decimal number 514. Explain.

Answer: γ-code: 514 = 512+2 =100000000+10=100000010 Offset = **00000010** of length 0 so the γ-code is **11111111000000010**

variable byte code: 10000000 00000101 [rightmost bit is continuation if 0, stop if 1] Can be reverse.

(iv) 4% Recover the gap values (in decimal) for the following string representing γ-encoding of a sequence of gaps in a posting list.

111110101001110111111101010101

Answer:

111110101001110111111101010101

110100 1111 11010 11

52 15 26 3

(v) 3% Consider a letter bigram index for wildcard queries. Give an example of a string that falsely matches the wildcard query **mon*hs** if search is simply using a conjunction (ANDing) of bigrams.

Answer: -m, mo, on, + hs, s- ➔ moremonths

Has all the bigrams but is not correct.

(vi) 4% Represent the binary number 10101011101010110 as a variable length code with a continuation bit.

Answer:

10101011101010110➔ divide into 7 bits and add continuation/end bits ➔101010111101010100001100

1 means continue, 0 means end. [other options possible]

Question 3 (21%):

1. 10% Shown below is a portion of a positional index in the format:
term: doc1: <position1, position2, . . . >; doc2: <position1, position2, . . . >; etc.

angels: 2: <36,174,252,651>; 4: <12,22,102,432>; 7: <17>;
fools: 2: <1,17,74,222>; 4: <8,78,108,458>; 7: <3,13,23,193>;
fear: 2: <87,704,722,901>; 4: <13,43,113,433>; 7: <18,328,528>;
in: 2: <3,37,76,444,851>; 4: <10,20,110,470,500>; 7: <5,15,25,195>;
rush: 2: <2,66,194,321,702>; 4: <9,69,149,429,569>; 7: <4,14,404>;
to: 2: <47,86,234,999>; 4: <14,24,774,944>; 7: <199,319,599,709>;
tread: 2: <57,94,333>; 4: <15,35,155>; 7: <20,320>;
where: 2: <67,124,393,1001>; 4: <11,41,101,421,431>; 7: <16,36,736>;

Which document(s) if any match each of the following queries, where each expression within quotes is a phrase query?

–3% "fools rush in" –Order important: phrase queries- **Answer: {D2, D4, D7}**

– 3%"fools rush in" AND "angels fear to tread" **Answer: {D2, D4, D7} ∩ {D4}={D4}**

{D2, D4, D7} **{D4}**

-2% List the last known 4 words of document 7: **Answer:**

fear[528]➔**to**[599]➔ **to**[709]➔ **where**[736] **(order important, numbers not)**

-2% If $|s_i|$ denotes the length of string s_i, show that the edit distance between s_1 and s_2 is never more than $\max(|s_1|, |s_2|)$.

Answer: we can always replace the characters of the shorter by those of the longer then add the rest.

Or we can always replace the characters of the longer by those of the shorter then delete the rest. In both cases we have the max as the limit.

2. Select the **BEST** Match: 11%

?	**Text**		**Defines**
P	The vocabulary size of a text can be estimated using	A	Similarity Measure
F	A metric used to measure the importance of a term in a text document collection	B	Distance Measure
I	# of changes needed to convert one string into another	C	Tokenization
D	Levenshtein distance:	D	Insert-Delete-replace one step
L	Damerau-Levenshtein Distance	E	Insert-Exchange 1 step
K	Removing most frequent words in the collection	F	IDF
J	Removing least frequent words in the collection	G	TF
C	Dividing a string into words	H	Normalization
A	A measure of how close documents are to each other	I	Levenshtein Distance
M	Removing word affixes: suffixes and prefixes	J	Typo (error) Cleaning
N	Number of non-positional postings for a term	K	Stop word removal
		L	Insert-Delete-replace-exchange 1 step
		M	Stemming
		N	DF
		O	Zipf 's Law
		P	Heap's Law

		Q	Recall
		R	Inverted Index

Question 4 (20%)

1- 4% Assuming Zipf's law with a corpus independent constant A= 0.1, what is the fewest number of most common words that together account for more than 19% of word occurrences (i.e. the minimum value of m such that at least 17% of word occurrences are one of them most common words).

PR=0.1➔P=0.1/R: Top 3 words: 0.1/1 +0.1/2 +0.1/3=0.1+0.05+0.033➔18.3%Top 4 words: 0.1/1 +0.1/2 +0.1/3+01/4=0.1+0.05+0.033+0.025➔20.8%So the min number is 4

2- 4% Assume the 4th most frequent word has frequency 16000. Can you estimate the number of distinct words (types) in the dictionary? Explain

PR =0.1➔at rank 4: P=16000/N; (16000/N)*4=0.1➔N=640,000 (total number of Tokens).

Assume the highest ranking term (at rank D where D is the number of types) occurs only once (reasonable assumption for this size): (1/N)*D = 0.1=➔D=0.1*N=64,000: this is the number of types.

3- 3% Assume the 4th most frequent word has frequency 1600. Can you estimate the frequency of the most frequent word in the dictionary? Explain.

F4=F1/4; F1=F4*4=6400

4- If we add another collection to this one with similar characteristics: exactly the same **number** of words and the same **vocabulary size** (but no duplicate documents): In your opinion:

4-1. 3% What will happen to the new vocabulary size?

By Heap's law the dictionary size should increase: though not double!

4-2. 3% What will happen to the frequency of the 10 most frequent words? Will they change? Remain the same? Explain.

Yes. The frequencies are most likely to increase as they are generally stop words frequent in most documents. Relative frequency need not change, though

5- 3% Prove that text power laws have a linear log-log graph.

Power laws have the form (X,Y: variables; K,Z are constants): $Y=K.X^{Z}$ ➔logY=logK+Z.logX=K1+K2.logX

Linear in terms of logX, LogY.

Question 5 (21%) True or False: Place √ in the right square and fill the table at the end (-3% if not filled):

1- □ **True** □ **False** Inverted index join (intersection) should start from the query term with highest IDF.

2- □ **True** □ **False** With Positional indexing it is possible to recover the original document from the index something not possible for non-positional index.

3- □ **True** □ **False** Positional indexing can double or even triple the space needs for an inverted index.

4- □ **True** □ **False** Boolean search requires more advanced skills on part of the user compared to state space model search (Google style search).

5- □ **True** □ **False** The phrase “أيا جارتا ما أنصف الدهر بيننا تعالي أقاسمك الهموم تعالي” has more tokens than types/terms (no pre-processing beyond tokenization).

6- □ **True** □ **False** Using skip pointers requires more space for the posting lists (compared to no skips).

7- □ **True** □ **False** In the “bag of words” model of the document word order and word co-occurrence patterns are NOT important.

8- □ **True** □ **False** Pseudo-relevance feedback is based on user judgement on relevance to revise the query while Relevance feedback blindly assumes that the first n documents are relevant for some n.

9- □ **True** □ **False** We can get the number of unique terms in a document from an inverted index.

10- □ **True** □ **False** Two documents D1 and D2 both have the word “Palestine” 2 times. D1 and D2 will always have the same rank in any web search.

11- □ **True** □ **False** Relevance quality of a document is judged against the terms present in the given query.

12- □ **True** □ **False** We usually prefer cosine similarity over Euclidean distance in vector space models because the former is computationally more efficient.

13- □ **True** □ **False** Two English words with different SOUNDEX codes can NEVER be the same.

14- □ **True** □ **False** Probabilistic ranking principle is based on the sequential examination assumption.

15- □ **True** □ **False** N-gram based bag-of-word retrieval model is used to handle phrase queries.

16- □ **True** □ **False** Normalization for NL text results in smaller dictionary size and less ambiguity

17- □ **True** □ **False** We can directly get the number of unique terms in a particular document from an inverted index.

18- □ **True** □ **False** Pseudo-Relevance feedback always increases precision and recall.

19- □ **True** □ **False** Precision and recall always trade off with each other: if one increases the other decreases and vice versa.

20- □ **True** □ **False** Current search engines are good at question answering (for simple questions).

21- □ **True** □ **False** For a collection: Phrase index has more terms (vocabulary) and more postings than single word index.

Q	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
□T	□T	□T	□T	□T	□T	□T	□T	□T	□T	□T	□T	□T	□T	□T	□T	□T	□T	□T	□T	□T	□T
□F	□F	□F	□F	□F	□F	□F	□F	□F	□F	□F	□F	□F	□F	□F	□F	□F	□F	□F	□F	□F	□F