BIRZEIT UNIVERSITY

Faculty of Engineering and Technology

Electrical and Computer Engineering Department

First Semester 2016/2017

Course ENCS539: Special Topics: Information Retrieval and NLP

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

Please answer the following questions using the exam sheets only.

Question	Q1	Q2	Q3	Q4	Q5	Total
Max grade	24	20	20	20	24	108/100
Earned						

Question 1 (24%): Consider a collection made of the 4 following documents (one document per line)

```
d1. Ali borrows a book for Hind
d2. Ali  reads the book aloud for Hind
d3. Ali and Hind think  aloud
d4. Ali thinks a book is a good gift
```

1- These documents are pre-processed using a stop word list and light stemmer. The resulting index is built to allow to apply vector-based queries. Fill in the following table representing of this index.

(Please note that other methods for calculating idf are tolerated)

	Term *t*	N/d f_t	Idf=$1+\log_2(n)$		d1:$tf_{t,d1}$	tf.idf	d2:$tf_{t,d2}$	tf.idf	d3:$tf_{t,d3}$	tf.idf	d4:$tf_{t,d4}$	tf.idf	Q:$tf_{t,d4}$	Q:tf.idf
1	ali	4/4	1	→	1		1		1		1			
2	aloud	4/2	2	→	0	0	1	2	1	2	0	0	1	2
3	book	4/3	1.4	→	1	1.4	1	1.4	0	0	1	1.4	0	0
4	borrow	4/1	3	→	1		0		0		0			
5	gift	4/1	3	→	0		0		0		1			
6	good	4/1	3	→	0		0		0		1			
7	hind	4/3	1.4	→	1	1.4	1	1.4	1	1.4	0	0	1	1.4
8	read	4/1	3	→	0		1		0		0			
9	think	4/2	2	→	0		0		1		1			

2- Determine the answer set if we issue a Boolean query: hind AND aloud AND NOT book.
{d3}: the documents that have both "hind" and "aloud" but not "book"

3- Given the Boolean query **Ali \3 Hind** find the answer set. X \n Y means X and Y are words in the same document with distance at most n. Consecutive words have distance 1.
{d1, d3}: the documents that have both "hind" and "ali" with no more than 2 words separating them.
{d3} is also accepted if you take into account sop words. d2 cannot be in the answer set.

***We now focus on 3 terms from the dictionary, namely {**book, aloud, hind}**.

4- Compute the tf−idf-based vector representation for the 4 documents in the collection (these vectors are normalized using the euclidian normalization).

d=<aloud,book,hind>

d1=<0, 1.4, 1.4>; all divided by : $|v(d1)| = \sqrt{0+1.4^2+1.4^2} = \sqrt{2+2} = 2$

d2=<2, 1.4, 1.4> all divided by : $|v(d2)| = \sqrt{2^2+1.4^2+1.4^2} = \sqrt{8} = 2.8$

d3=<2, 0, 1.4> all divided by : $|v(d3)| = \sqrt{2^2+0+1.4^2} = \sqrt{6} = 2.45$

d4=<0, 1.4, 0> all divided by : $|v(d4)| = \sqrt{0+1.4^2+0} = \sqrt{2} = 1.4$

5- Consider the query **aloud Hind**. Give the results of a ranked retrieval for this query (still limiting yourself to the 3 selected terms). Which document is considered to be the most relevant to the query?

Q= <2, 0, 1.4> all divided by : $|v(Q)| = \sqrt{2^2+0+1.4^2} = \sqrt{6} = 2.45$

Sim (d1,Q)= (0x2+1.4x0+1.4x1.4)/(2x2.45)= 2/4.9=0.41 ---→ Rank 3

Sim (d2,Q)= (2x2+1.4x0+1.4x1.4)/(2.8x2.45)=6/6.86=.87 ---→ Rank 2

Sim (d3,Q)= (2x2+0x0+1.4x1.4)/(2.45x2.45)=6/6=1-----→ Rank 1

Sim (d4,Q)= (0x0+1.4x0+0x1.4)/(1.4x2.45)=0--------→Rank 4

That d3 is the best ranked looks natural: it has both query words and it has less words and thus a larger intersection with the query than others.

Question 2 (20%): Consider a collection (corpus) made of 500,000 documents, each containing on average 800 words (average: 800 words or tokens per document).

The number of different words (i.e. terms, not taking duplicates into account) for the corpus is estimated to 700,000. For all questions, show your computations.

1- What is the size (mega or giga bytes) of the collection when stored (uncompressed) on disc ?

$500000 \times 800 \times WL_{AV} = 400{,}000{,}000 \times WL_{AV}$ bytes; where WL_{AV} =is average word length in bytes.
$500000 \times 800 \times 6 = 240000000$ bytes = 2.4 GB If average word length is different so is the size; Word length of less than 3 bytes are quite problematic.

2- From your past readings/experience, with the best reduction rate of the dictionary achieved when using linguistic preprocessing (noise words removal, stemming, stop word processing, downcasing), what is the expected size (number of terms) of the dictionary?

Applying this all should result in reduction rate of about 50%: resulting in 700000/2 = 350000 keywords. Other percentages could work, but below 30% is problematic as this is the general proportion of stop words (can reach 40%).

3- Consider an index where the average length of a non-positional posting list is 200. What is the estimation of the total number of postings of this index ?
700000×200 = 140000000 postings

4- How many bytes do you allow respectively for encoding (without compression) a dictionary term? a non-positional posting?
About 20 bytes for the term (max length); 3-4 bytes for document frequency and 3-4 bytes for a pointer to the postings: 20+4+3=27 bytes per term

Could be a little different

5- What are the size (mega or giga bytes) of the resulting dictionary and posting lists?

Dictionary: 700,000x27=16MB
Postings: 140000000x3=410MB

6- If you compress your dictionary using the dictionary-as-a-string method, what is the new size of the dictionary?
700000x(4+3+3+8) (4 bytes for the term frequency, 3 bytes for the pointer to the posting list, 3 bytes for the pointer into the string, and 8 characters per word (term) on average.

700000x(4+3+3+ **TL_{AV}**): **TL_{AV}** is the average term length,should be larger than the average **WL_{AV}**.

Question 3 (18%):

1- What is the largest gap that can be encoded in 2 bytes using the variable-byte encoding?

Out of the 2 bytes 2 bits are used for continuation (one of each 8 bits) leaving only 14 bits for the number:

2**14= 2^{14}=16K=16*1024=16384

2- If we use nipples (4 bits) instead of bytes (8 bits) for variable nipple coding then what is the largest number that can be coded in 2 bytes?

Out of the 2 bytes 4 bits are used for continuation (one of each 4 bits) leaving only 12 bits for the number:
2**12=2^{12}=4K=4*1024=4096

What is the min waste in this case?

4/16=1/4=25% (one bit out of each 4).

3- What is the posting list that can be decoded from the variable byte-code (1 in the most significant bit is NO continuation, 0 continuation). Move from lift to right with the left more significant. Give the values in BOTH binary and decimal.

10001001 00000001 10000010 11111111?
0001001 00000010000010 1111111
9 130 127

4- What would be the following Gamma code represent in Decimal?

1110101111111100000011111110011111
1101 10000011 1011111
13 131 95

5- How many bits does the Gamma code for the number 2**13 (2 to the power 13) take? Show how you arrived at the answer.

213=1000000000000 (1 and 13 zeros), the offset is13 zeros thus we need 13 ones followed by a zero for length and the 13 zeros: overall we need 14+13 bits = 27 bits**

Question 4 (20%):

1. (10%) Assume that an IR system returns a ranked list of 10 total documents for a given query. Assume that according to a gold-standard labelling there are 5 relevant documents for this query, and that the only relevant documents in the ranked list are in the 2nd, 3rd, 4th, and 8th positions in the ranked results. Calculate and clearly show the interpolated precision value for each of the following standard recall levels: {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0} for this individual query

Doc	1	2	3	4	5	6	7	8	9	10	
Relevance		x	x	x				x			
Recall	0	.2	.4	.6	.6	.6	.6	.8	.8	.8	
Precision	0	.5	.667	.75	.6	.5	.44	.5	.44	.40	
RLevel	0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
IRLevel	.75	.75	.75	.75	.75	.75	.75	0.5	0.5	0	0

You may replace the table by a graph (Rlevel vs. IRLevel):

2. (10%) The table below shows the final ranked list of results for an IR search together with their continuous human-rated relevance values. Assume the table contains all documents with non-zero relevance. Compute the values of the DCG and NDCG evaluation metrics for each value of n and add them to the table. Complete the second table to show the idealized DCG (IDCG) values.

Items marked are not a must or reproduced from elsewhere (no grades)

n	Doc	Relevance gain	CG	$\log_2(n)$	DCG	IDCG	NDCG
1	D23	0.6	0.6	-	0.6	1	0.60
2	D78	1.0	1.6	1	1.6	1.9	0.84
3	D90	0.0	1.6	1.59	1.6	2.277	0.70
4	D17	0.5	2.1	2	1.85	2.527	0.73
5	D88	0.9	3.0	2.32	2.23	2.527	0.88

Computing IDCG then copied to previous table.

n	Doc	Relevance gain	CG	$\log_2(n)$	IDCG
1	D78	1.0	1.0	-	1
2	D88	0.9	1.9	1	1.9
3	D23	0.6	2.5	1.59	2.277
4	D17	0.5	3.0	2	2.527
5	D90	0.0	3.0	2.32	2.527

Question 5 (24%) True or False: Place √ in the right square then summarize in table:

1- □ **True** □ **False** Precision at 5 (P@5) is always better than precision at10 (P@10).

2- □ **True** □ **False** Zipf's law implies that stop words have higher ranks than other words (higher rank means closer to rank 1: top ranked).

3- □ **True** □ **False** Boolean search can be done with nonpositional index only.

4- □ **True** □ **False** The sentence "العلم يبني بيوت لا عماد لها والجهل يهدم بيوت العز والكرم" has more tokens than types/terms.

5- □ **True** □ **False** Using skip pointers requires more space for the postings.

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

7- □ **True** □ **False** Suppose your collection contained the document "We hold these truths truths to be self-evident…" You later discovered that somebody had written "truths" twice, which you corrected. To fix the df.idf index, you need to recalculate all the vectors of every document in the collection. Document frequency is not affected

8- □ **True** □ **False** phrase queries can be processed in positional index but cannot be processed in nonpositional index.

9- □ **True** □ **False** Relevance feedback may try to reformulate the user query to minimize the distance to relevant documents and maximize the distance to non-relevant documents.

10- □ **True** □ **False** Huge **Query Logs** is one of the **most important** assets search engine companies have and its size may give **major** advantage to one company over another.

11- □ **True** □ **False** Document normalization by length tends to favor shorter documents and the fix is to use **pivot normalization**.

12- □ **True** □ **False** Pseudo-relevance feedback requires **no user intervention** to modify the query

13- □ **True** □ **False** Search companies tend to place the results of highly paying content at the top of their search results. This model is the reason for search companies wealth

14- □ **True** □ **False** Google snippets are examples of dynamic rather than static summaries.

15- □ **True** □ **False** Stemming primarily improves recall.

16- □ **True** □ **False** If I search for term *X*, and term *X* has many meanings, precision is more likely to be a problem than recall.

#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
True	□	X	□	X	X	X	□	X	X	X	X	X	□	X	X	X
False	X	□	X	□	□	□	X	□	□	□	□	□	X	□	□	□