CS 494 - IR - EXAM 1

SAMPLE EXAM

Name:

University NetID:

This test consists of 4 questions. The number of points for each question is shown below.

- Read all questions carefully before starting to answer them.
- Write all your answers in the space provided in the exam paper.
- The order of the questions is arbitrary, so the difficulty may vary from question to question. Do not get stuck by insisting on doing them in order.
- Show your work. Correct answers without justification will not receive full credit. However, also be concise. Excessively verbose answers may be penalized.
- Clearly state any assumptions you may make when answering a question.
- **Be sure to write your name on the test paper.**

Question	1	2	3	4	total
Points	20	20	20	20	80
Your Points					

Exercise 1 - 20 points. (Boolean Retrieval)
Assume the following collection of short documents:

```
Doc 1:  banking on banks to raise the interest rate
Doc 2:  jogging along the river bank to look at the sailboats
Doc 3:  jogging to the bank to look at the interest rate
Doc 4:  buzzer-beating shot banked in
Doc 5:  scenic outlooks on the banks of the Potomac River
```

i. Construct a term-document matrix that can be used to perform Boolean retrieval. The index terms have already been listed for you in the following table (note that terms have been stemmed and stopwords have been removed):

Term	Doc1	Doc2	Doc3	Doc4	Doc5
bank	1	1	1	1	1
buzzer-beating	0	0	0	1	0
interest	1	0	1	0	0
jog	0	1	1	0	0
look	0	1	1	0	0
outlook	0	0	0	0	1
potomac	0	0	0	0	1
raise	1	0	0	0	0
rate	1	0	1	0	0
river	0	1	0	0	1
sailboat	0	1	0	0	0
scenic	0	0	0	0	1
shot	0	0	0	1	0

ii. What documents would be returned in response to the following queries?

bank AND ¬ interest: Doc 2,4,5

interest AND jog AND ¬ rate: None

bank AND (scenic OR jog): Doc 2,3,5

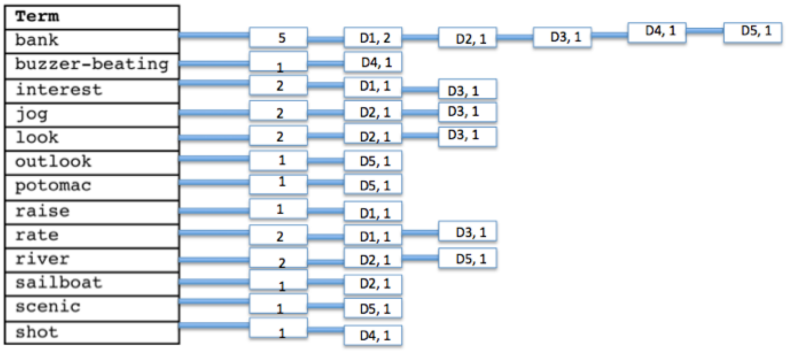
Exercise 2 - 20 points. (Inverted Index and Cosine Similarity)
Consider the collection of documents as follows:

```
Doc 1:  banking on banks to raise the interest rate
Doc 2:  jogging along the river bank to look at the sailboats
Doc 3:  jogging to the bank to look at the interest rate
Doc 4:  buzzer-beating shot banked in
Doc 5:  scenic outlooks on the banks of the Potomac River
```

After performing stemming and removing stop words, the vocabulary is:

$\{bank, buzzer\text{-}beating, interest, jog, look, outlook, potomac, raise, rate, river, sailboat, scenic, shot\}$

i. Construct an inverted index. Show the index graphically with linked lists.



ii. Consider the query "interest jog rate" and simulate the retrieval of documents in response to this query. Show how the inverted index is used to identify relevant documents and how the cosine similarity between the query and the relevant documents is calculated incrementally using a hashtable. (Don't worry about getting the exact numbers, I just want to see how you use the inverted index data to get the cosine similarity).

We need to parse the words in the query. We first find the word "interest" - we search it in the index and we find that the word appears in 2 documents, D1 and D3. We create entries for D1 and D3 in a hashtable and use the information in the index to update the hashtable entries for these documents. We have:
D1 $\rightarrow w_{1,interest} * w_{q,interest}$
D3 $\rightarrow w_{3,interest} * w_{q,interest}$
where $w_{doc,word} = [TF(word, doc)] * \log[N/idf(word)]$
The next word in the query is "jog" - this word appears in documents D2 and D3. We update the hashtable entry for D3, and we create an entry for D2. We have:
D3 $\rightarrow w_{3,interest} * w_{q,interest} + w_{3,jog} * w_{q,jog}$
D2 $\rightarrow w_{2,jog} * w_{q,jog}$
The last word in the query is "rate" - this word appears in documents D1 and D3. We update the hashtable entries for documents D1 and D3. We end up with:
D1 $\rightarrow w_{1,interest} * w_{q,interest} + w_{1,rate} * w_{q,rate}$
D2 $\rightarrow w_{2,jog} * w_{q,jog}$
D3 $\rightarrow w_{3,interest} * w_{q,interest} + w_{3,jog} * w_{q,jog} + w_{3,rate} * w_{q,rate}$
The final values need to be normalized by the product of document and query lengths.

Exercise 3 - 20 points. (Query likelihood language model)
Suppose we have a collection that consists of three documents given below.

```
Doc 1:  you say goodbye
Doc 2:  hello goodbye, hello goodbye, hello
Doc 3:  I say hello
```

Assume that we also have the following query: *hello, goodbye.*

Build the following language models for this collection. In each case, compute the model probabilities for the query, and show the final ranking of the documents.

(i) Estimate unigram models of documents using smoothed MLE, when smoothing is done by adding 0.5 to the observed counts (remember the renormalization).

The table below contains $tf_{(t,d)}$ for all terms t and documents d

	you	say	goodbye	hello	I
Q	0	0	1	1	0
D1	1	1	1	0	0
D2	0	0	2	3	0
D3	0	1	0	1	1

We have the smoothed estimates as below:

	you	say	goodbye	hello	I
D1	(1+0.5)/(3+2.5)	(1+0.5)/(3+2.5)	(1+0.5)/(3+2.5)	(0+0.5)/(3+2.5)	(0+0.5)/(3+2.5)
D2	(0+0.5)/(5+2.5)	(0+0.5)/(5+2.5)	(2+0.5)/(5+2.5)	(3+0.5)/(5+2.5)	(0+0.5)/(5+2.5)
D3	(0+0.5)/(3+2.5)	(1+0.5)/(3+2.5)	(0+0.5)/(3+2.5)	(1+0.5)/(3+2.5)	(1+0.5)/(3+2.5)

$P(Q|D1) = 0.5/5.5 * 1.5/5.5 = 0.02$
$P(Q|D2) = 3.5/7.5 * 2.5/7.5 = 0.16$
$P(Q|D3) = 1.5/5.5 * 0.5/5.5 = 0.02$

So, D2>D1=D3

(ii) Estimate unigram models of documents using a mixture model between the documents and the collection with $\lambda = 0.3$.
Remember that probability of Q given d is:

$$P(Q|d) = \prod_{w \in Q} [(1-\lambda)P(w|M_c) + \lambda P(w|M_d)]$$

We have the following probability estimates for each term from documents and the collection.

	you	say	goodbye	hello	I
D1	1/3	1/3	1/3	0	0
D2	0	0	2/5	3/5	0
D3	0	1/3	0	1/3	1/3

	you	say	goodbye	hello	I
Collection	1/11	2/11	3/11	4/11	1/11

$P(Q|D1) = [0.3 * 0 + 0.7 * 4/11] * [0.3 * 1/3 + 0.7 * 3/11]$
$P(Q|D2) = [0.3 * 3/5 + 0.7 * 4/11] * [0.3 * 2/5 + 0.7 * 3/11]$
$P(Q|D3) = [0.3 * 1/3 + 0.7 * 4/11] * [0.3 * 0 + 0.7 * 3/11]$

$D2 > D1 > D3$

Exercise 4 - 20 points. (IR Evaluation)

i. The table below shows the output of an IR system on two queries. Only top 5 ranks are shown. Crosses correspond to documents which have been judged relevant by a human judge; circles correspond to irrelevant documents. There are no relevant documents in lower ranks (> 5). Compute the Mean Average Precision (MAP).

Rank	Q1	Q2
1	o	x
2	x	o
3	x	o
4	x	o
5	o	x

Average Precision: Average of the precision values at the points at which each relevant document is retrieved.

Mean Average Precision (MAP): Average of the average precision values for a set of queries.

For query 1:
At rank 2, p = 1/2, r = 1/3
At rank 3, p = 2/3, r = 2/3
At rank 4, p = 3/4, r = 3/3 = 1
Avg precision (1/2 + 2/3 + 3/4)/3 = 0.638

For query 2:
At rank 1, p = 1, r = 1/2
At rank 5, p = 2/5, r = 2/2 = 1
Avg precision (1 + 2/5)/2 = 0.7

MAP is [(avg prec for query 1) + (avg prec for query 2)]/2 = 0.669

ii. For query Q1 above, plot an exact recall/precision curve and then overlay it with a graph where the precision values are interpolated to the standard 11 points.

The exact recall/precision points are (1/3,1/2), (2/3,2/3), (1,3/4).

Therefore, the interpolated recall/precision points are: (0,3/4), (0.1,3/4), (0.2,3/4), (0.3,3/4), (0.4,3/4), (0.5,3/4), (0.6,3/4), (0.7,3/4), (0.8,3/4), (0.9,3/4), (1,3/4).