Information Retrieval (CS60092) Test 3 Final Exam
Apr 11, 2022 Full Marks: 50
Time: 85 minutes (8:00 – 9:10) + 15 minutes for uploading answer-scripts

** Write your name and roll number clearly on the first page. Write the page number on every sheet that you use.*
** Attempt all questions. All parts of the same question are to be answered together.*
** All workings must be shown, without which no marks will be awarded.*
** If you make any assumption, write the assumption clearly. Any assumption made should be logical.*
** Send **a single combined PDF** to assignmentan1998@gmail.com by 9:40 am (according to the Moodle clock).* The format of the file would be **<ROLL_NUMBER.pdf>**
** **Answer-scripts must be submitted by 9:25 AM** (according to the Moodle clock). You MUST submit your scripts to assignmentan1998@gmail.com within 9:25 AM. No submission will be allowed after 9:25 AM. There will be **NO EXCEPTIONS**.*
** This is a **proctored exam**. You need to join the online examination hall allotted to you and be visible on camera through the entire duration of the class test. Otherwise, you will be considered absent even if you upload the answer-script.*

Question 1

Q1. (a) Assume there are N documents. Naively, to find all possible duplicates, you can compare each pair, use a duplicate detection technique to determine if they are duplicates. Can you come up with a faster algorithm? Explain the time complexity. **[3 Marks]**

Solution: Recall the shingles technique. Use the same random permutation. Get min{\Pi_i(f(s))} for each doc. Sort them. Collect the ones for which they are the same. Complexity comes down to N log N * (shingles calculation time)

Q1. (b) What are the cases where the HITS algorithm would take the longest time to converge? Can there be cases where the hub and authority scores start diverging? Give an example. **[1+1 Marks]**

Solution: A long chain of nodes. A-> B-> C-> D …
Dynamic situations. Hyperlinks being modified – added/deleted arbitrarily

Q1 (c) A Markov chain is said to be *ergodic* if there exists a positive integer T_0 such that for all pairs of states i, j, in the Markov chain, if it is started at time 0 in state i then for all $t > T_0$, the probability of being in state j at time t is greater than 0.
Consider our algorithm of converting the adjacency matrix to the transition probability matrix. How would you modify the algorithm to guarantee that the resulting probability matrix does not represent an Ergodic Markov chain? **[1 Marks]**

Solution: you can remove teleportation. But you cannot guarantee, as the original underlying graph itself can be fully connected one.

In that case you need to make one or more random transition probabilities 0.

No marks for writing just teleportation.

Q1 (d) Twitter is a social network, where users tweet short messages which become visible to other members of the network. Many tweets have different hashtags. Clicking on a hashtag can show you all tweets using that hashtag. Each user have some followers, and they can themselves follow others.

Imagine you are designing an IR system for a twitter-users, who are interested in movies and sports related tweets. How would you use the PageRank algorithm (or its variants) to design such a system? Clearly state, what will be the documents and how do you get links between the documents in this case. **[4 Marks]**

Use movies pagerank by setting S=movie related tweets. Use sports pagerank by setting S=sports. Then use personalized pagerank, imagining w/ prob q, the user is interested in movies, vs 1-q interested in politics.

Look at personalized PageRank part IIR Manning book.

Question 2

Q2 (a) Suppose you are given a corpus

the prince is the future king
the princess will be a queen

Consider the CBOW algorithm for learning word vectors. Consider the context size as 2, i.e., given the current word, we need to predict the next word. Assume all elements of initial W1 (inner embeddings) and W2 (outer embeddings) matrices are filled with 0.1. Hidden layer embedding size is 5.
Show us the calculation for $J(\theta)$ considering all windows over full corpus, for the first step. Show each step clearly. Final numerical values are not required as long as the computation steps are correct. **[5 Marks]**

Vocab = 10 size
The prince
- W_1,the = [0.1 0.1 0.1 0.1 0.1]
- W_2,the = [0.5 0.5. 0.5 0.5 …. 0.5]
- P(w_o,prince|w_i,the) = exp(0.5)/10*exp(0.5)

Prince is
Is the
The future
Future king

The princess
Princess will
Will be
Be a
A queen
Same computation for every window.
- J(\theta) = 10 * 1/10 log (exp(0.5)/10*exp(0.5))

For getting the windows correctly: 2, the expression for each window: 2 and last expression: 1.

Q2 (b) For analogy questions such as **a:b :: c:?**, we make an implicit assumption while finding the closest similar vector to **c** while using the vector-offset method. What is the assumption? Give an example of analogy how such an assumption will not hold true. **[1+1 Marks]**

We assume, that we do not consider a, b, and c in the set of vectors; otherwise we will always get one of them as the answer. Problem happens when there is many-to-1 mapping snow:white::sugar:white; or reptile:animal::mammal:animal.

Q2 (c) Word2vec technique compresses a large-dimensional sparse vectors to smaller dimensions. Is it a lossy compression technique or lossless? Explain properly (no marks

without explanation) **[2 Marks]**

It might be lossy or lossless. The objective forces it to preserve information. But, depends on how the learning algorithm converges, and whether the learnt embedding is losing information.

If someone writes lossy, you can give 1.5 marks (as long as they explain).

Q2 (d) For Word2vec last layer, why is softmax function $\left(\frac{e^{z_i}}{\sum_{z \in Z} e^{z}}\right)$ used over normalization using sum of values $\left(\frac{z_i}{\sum_{z \in Z} z}\right)$?. **[1 Marks]**

If the sum is zero, we risk divide by zero error. Also, softmax (especially softmax with CE) is function is well-behaved under differentiation than linear function.
Give 1 marks if someone can say the first.

Question 3

Q3 (a) What is a fundamental difference in assumptions between Heap's Law and Zipf's law? Explain. **[2 marks]**

Zips'f law vocab is infinite. Heap's law states its vocab is finite.

Q3 (b) In the logarithmic merge, multiple indexes are kept on the disk. Imagine a bug that at any point of time may delete any one of the indices (multiple indexes will not be deleted at one point of time). Provide the sketch of two methods (one more efficient than the other) through which you can ensure the retrieval results do not get affected. Compare them in terms of efficiency. **[2 marks]**

Simple backups. 1) Back up all indices in the same machine. 2) back up indices in another machine.

In same machine, disk space competes with the current retrieval system. In the different machine, if connected through network – there will be lag.

If you see better intelligent solutions to this, you can give grace marks of upto 1.

Q3 (c) The following list represent the relevance scores of the top 20 images retrieved in response to a query from a collection of 10,000 images by Google Search .

2 1 0 0 0 0 0 0 2 0 1 0 0 0 1 0 0 0 0 2

The top of the ranked list is on left of the list. Higher the score means high relevance. 0 defines non-relevant. The list shows 6 relevant images. Assume that there are 8 relevant images in total in the collection (other two having relevance scores 2 and 1). Also assume that the system has returned a total 10k images, and these are the first 20 results in the list.

i) What are the largest and smallest possible MAPs that this system could have?
ii) In this system, the labels 0-1-2 come from humans looking at query-document pair. Imagine a setting, where you show users a query and two documents each time. The user marks one of the document to be more relevant. Among precision, recall, and NDCG; which score can be computed given such comparisons? How would you compute the score?
[6 Marks]

i)

2 1 0 0. 0 0 0 0 2 0 1 0 0 0 1 0 0 0 0 2

1/1 2/2 2/3 2/4 2/5 2/6 2/7 2/8 3/9 3/10. 4/11. 4/12. 4/13 4/14. 5/15. 5/16 5/17 5/18. 5/19. 6/20

MAP_largest = (1+1+3/9+4/11+5/15+6/20+7/21+8/22)/8 = 0.5034

MAP_Smallest = (1+1+3/9+4/11+5/15+6/20+ 7/9999 + 8/10000)/8=0.4164

ii)
Can compute NDCG. The ideal list would be the list where no judgement is inverted. No scores are assigned.

This becomes a partially ordered tree. Write a small algorithm to assign grades (0-1-2-3) so that di has a higher grade than dj, if di is ranked more relevant for dj.

Once you do that, NDCG calculation is easy.

P and R calculation is not possible, because no one is directly saying its non-relevant. For example, d1 > d2 and d2> d3 can happen. Its hard to say d2 or d3 is irrelevant.

Question 4

Q4 (a) In the Binary Independence Model, a term x_i is thought of independent of all other terms. Now, assume that each term x_i is only dependent on the previous term x_i-1. In that case, what is the simplified formula of the odds that a document is relevant given a query q and document vector x**. [4 Marks]**

$$O(R|q,x) = O(R|q).\prod_{i=1,x_i=1,x_j=1,\ x_j \text{ is previous word of } x_i \text{ in } d}^{n} \cdot \frac{p(x_i|x_j,R=1,q)}{p(x_i|x_j,R=0,q)}$$

Assume $p_{ij} = p(x_i|x_j,R=1,q)$ and $r_{ij} = p(x_i|x_j,R=0,q)$. The product is only over document terms. Then, it simplifies to

$$O(R|q,x) = O(R|q).\prod_{i=1,x_i=1,x_j=1,\ x_j \text{ is previous word of } x_i \text{ in } d}^{n} \cdot \frac{p_{ij}}{r_{ij}}$$

Q4 (b) Now compare above to the BIM model in terms of parameters. Assume that each probability (say p_i, r_i and equivalent ones above) are independent parameters that you need to estimate. For any query, give an estimate of the number of parameters in BIM and the above model requires. **[2 Marks]**

N = average document size. M= average query size.
For each query, p_i and r_i needs to be estimated from relevant and non-relevant documents for only query terms found in the document. So, 2* (N+M) for query.

In the above its almost ${}^{N}C_2 + {}^{M}C_2$ per query.

Q4 (c) Suppose your corpus contains the following 3 sentences:

S1: **vector space model** makes similar assumptions
S2: language model views document as generative model
S3: has cleaner state of assumptions than vector space

Use a mixture language model to rank these document as per relevance to the query – 'model assumptions'. Determine the range(s) of lambda for which S3 ranks higher than S1 and S2. Do not remove the stop words**. [4 Marks]**

Simply formulate using lambda. You will get expressions like f(\lambda) > g(\lambda). You can solve for that.

Question 5

Q5 (a) In Jaccard Similarity computation, we compute $A \cup B$, over two document A and B. If similarity is 1.0, can we say documents are duplicates? If not, give an example where the meaning would be different, but the set of words are same. If yes, explain how. **[2 Marks]**

No.

John and Mia are not friends. Ashley is a friend of Mia.
Mia and Ashley are not friends. John is a friend of Mia.

Give 1.5 if example is partially correct

Q5 (b) Why do we use inverse document frequency in tf-idf weighting? Why do we use "log(.)" function in IDF computation? Can any other function be used? Give an example **[3 Marks]**

Rare words should be weighted higher.
Log to dampen the effect of N/df_t
Yes.
Anything whose slope is sublinear. \alpha * N/df_t, where alpha <1 , say 0.5

Q5 (c) In computing cosine similarity between two vectors, we normalize both. Say for q and d, similarity is $\frac{\vec{q}.\vec{d}}{|\vec{q}|.|\vec{d}|}$. However, during Word2vec offset calculation, we a:b::c:?, we only normalize (w_b-w_a+w_c), not the w_x for the words we search for. (Hint, check in 2-D, what happens when you normalize vs do not normalize.) **[3 Marks]**

Simply checking in 2-D they will find, that with direction same, and magnitude different – it will be different vectors. We should not collapse them onto the unit-ball by normalizing.

Q5 (d) In Learning to rank, for annotation collection phase, if we show a list to the user to query and ask them to re-rank, would it be sufficient for pairwise loss calculation? What is the tradeoff between doing this or showing users triplets of <query, document-1 and document-2>? **[2 Marks]**

Yes.
But, ranking a full list is a cumbersome job, as users need to take each document a find a perfect place for it in the list. Its much better for users to simply compare two documents wrt a query.