University of Virginia
Department of Computer Science

CS 6501: Information Retrieval
Fall 2014

9:30am-11am, Thursday, November 11th

Name:
ComputingID:

- This is a **closed book** and **closed notes** exam. No electronic aids or cheat sheets are allowed.
- There are 12 pages, 4 parts of questions (the last part is bonus questions), and 115 total points in this exam.
- The questions are printed on **both** sides!
- Please carefully read the instructions and questions before you answer them.
- Please pay special attention on your handwriting; if the answers are not recognizable by the instructor, the grading might be inaccurate (*NO* argument about this after the grading is done).
- Try to keep your answers as concise as possible; grading is *not* by keyword matching.

Total	/115

Academic Integrity Agreement

I, the undersigned, have neither witnessed nor received any external help while taking this exam. I understand that doing so (and not reporting) is a violation of the University's academic integrity policies, and may result in academic sanctions.

Signature: _____

Your exam will not be graded unless the above agreement is signed.

1 True/False Questions (20 pts)

Please choose either True or False for each of the following statements. For the statement you believe it is False, please give your brief explanation of it (you do not need to explain when you believe it is True). One point for each question. *Note the credit can only be granted if your explanation is correct.*

1. Depth-first is a basic visiting strategy in focused crawling.
   ***False**, and Explain*: depth-first crawling is in parallel with focused crawling.

2. We can easily get the number of unique terms in a particular document from an inverted index.
   ***False**, and Explain*: we have to traverse the inverted index to get this count.

3. Invert index join should start from the query term with highest IDF.
   True

4. Document ranking happens after inverted index look up.
   ***False**, and Explain*: they start simultaneously.

5. Stemming helps improve recall of a Boolean retrieval model.
   True

6. Vector Space Model is equivalent to Bag-of-Word model.
   ***False**, and Explain*: VSM is more general than BoW.

7. Increasing the dimension of latent space in Latent Semantic Analysis is helpful for recall.
   ***False**, and Explain*: we should decrease the dimension to increase recall.

8. $p(Q, D|R) = P(Q|R)P(D|Q, R)$ is called query generation model, a.k.a., language model.
   ***False**, and Explain*: query generation model should be $p(Q, D|R) = P(D|R)P(Q|D, R)$.

9. The assumption of words are "independent and identical distributed (i.i.d.)" in documents is the foundation of statistic language models.
   ***False**, and Explain*: language model has no assumption about the documents.

10. When we have indexed all the text documents in the world, there is no need to do smoothing.
    ***False**, and Explain*: there will still be unseen words in individual documents.

11. Smoothing of language models will not affect the relative order of the most probable words in each document.
    ***False**, and Explain*: there is no such guarantee.

12. Relevance quality of a document is judged against all the query terms in the given query.
    ***False**, and Explain*: against the information need.

13. Precision and recall always trade off with each other.
***False**, and Explain*: they can both increase in perfect ranking.

14. P@2=0.2 means in average you will have 20% of chance to find a relevant document at position 2 over all the queries.
***False**, and Explain*: it should be true for the top 2 positions.

15. Mean Average Precision prefers a system to return as many relevant documents as possible.
True

16. Given a very large IR evaluation collection, where System A achieves a MAP of 0.33 and System B achieves a MAP of 0.79, we can safely conclude that System A is significantly better than System B.
***False**, and Explain*: statistic test is needed, though I should say B is better than A (this is a typo in the question).

17. Interleaved test requires less observations than A/B test to reach the same conclusion.

True

18. Given EM iterations guarantee to improve the objective function step by step, starting point does not matter for it.
***False**, and Explain*: EM only guarantees local maximum.

19. Rocchio is not applicable in BM25.
***False**, and Explain*: BM25 can be understood as VSM; therefore Rocchio is applicable.

20. Relevance feedback helps RSJ model improve ranking for new query and unseen documents.
***False**, and Explain*: it is helpful for current query and unseen documents.

2 Short Answer Questions (30 pts)

Most of the following questions can be answered by one or two sentences. Please make your answer concise and to the point. Three points for each question.

1. Name three components in a typical retrieval system, where MapReduce are potentially helpful. (A MapReduce program is composed of a Map() procedure that performs filtering and sorting and a Reduce() procedure that performs a summary operation in parallel.)

   - invert index construction
   - crawling
   - document analyzing or ranking

2. In a given corpus of Spanish documents, the frequency of the most frequent word is 1,270,873. Then what is the estimated frequency for the second most frequent word in this corpus and why?

   by Zipf's law: 1,270,873/2

3. How to support phase query, e.g., "president of the united states", in an inverted index?

   store position in inverted index and check the distance between the words in query and those in documents.

4. In inverted index compression, what part of an inverted index is usually compressed? And why it can be effectively compressed?

   posting list is mostly compressed; and the non-uniform distribution, i.e., Zipf's law, ensures the effective compression.

5. Why cosine similarity is preferred over Euclidian distance in Vector Space Models?

   document length and unseen words have been better handled in cosine similarity.

6. Given a bigram language model θ, how can we use it to compute the probability of of a document d of length n ($n > 1$) containing words $w_1, \ldots, w_n$? That is, finish the expression

   $p(d|\theta) = \prod_{i=1}^{n} p(w_i|w_{i-1}, \theta)$

7. Mathematically define Maximum a Posterior estimation. (just in math equation)
   $\arg\max_\theta p(X|\theta)p(\theta)$

8. Give two reasons why Dirichlet Prior smoothing is preferred than Add-1 smoothing.

   - not all unseen words are equally important

- document length normalization

9. In retrieval evaluation, all the documents are independently labeled against the query and the labels are assumed to be static when being used to evaluate different retrieval systems. List three limitations of such an assumption?

   - cannot reflect various users' distinct result preferences
   - relevance quality is not independent with the other returned documents
   - relevance quality is dynamic respect to the search context

10. For a particular query q, the multi-grade relevance judgements of all documents are $\{(d_1, 1), (d_3, 4), (d_6, 2), (d_9, 3), (d_{11}, 1), (d_{31}, 2)\}$, where each tuple represents a document ID and relevance judgment pair, and all the other documents are judged as irrelevant. Two document search systems return their retrieval results with respect to this query as, System A: $\{d_1, d_2, d_3, d_4, d_5, d_6, d_7\}$ and System B: $\{d_{31}, d_{22}, d_3, d_6, d_{15}\}$ (these are all results they have returned for this query). Compute the following ranking evaluation metrics for System A and B. For each of the metric, you do not need to come up with the precise numbers, just illustrate how you compute each of them. (Since there are two DCG definitions discussed in class, you can choose either one to answer this question.)

Metric	System A	System B
P@5	$2/5$	$3/5$
AP	$(1 + \frac{2}{3} + \frac{3}{6})/6$	$(1 + \frac{2}{3} + \frac{3}{4})/6$
NDCG	$(\frac{2^1-1}{\log_2(1+1)} + \frac{2^4-1}{\log_2(1+3)} + \frac{2^2-1}{\log_2(1+6)})/\text{iDCG}$	$(\frac{2^2-1}{\log_2(1+1)} + \frac{2^4-1}{\log_2(1+3)} + \frac{2^2-1}{\log_2(1+4)})/\text{iDCG}$

$\text{iDCG} = \frac{2^4-1}{\log_2(1+1)} + \frac{2^3-1}{\log_2(1+2)} + \frac{2^2-1}{\log_2(1+3)} + \frac{2^2-1}{\log_2(1+4)} + \frac{2^1-1}{\log_2(1+5)} + \frac{2^1-1}{\log_2(1+6)}$

3 Essay Questions (50 pts)

All the following questions focus on system/algorithm design. Please think about all the methods and concepts we have discussed in class (including those from the students' paper presentations) and try to give your best designs in terms of feasibility, comprehensiveness and novelty. When necessary, you can draw diagrams or write pseudo codes to illustrate your idea. Ten points for each question.

1. LinkedIn is designing a new service called "find my referral". Given a company's name of interest as input query, it identifies top 10 users, who (previously) work(ed) at that company, with 0-degree separation (i.e., friends) or 1-degree separation (i.e., friends of friends) to the current user. In LinkedIn, the social connection is stored in a relational database with two tables: one table for all users with a set of attributes, including current employer and previous employer(s), and one table for friendship (e.g., two columns of user IDs, and each row presents a pair of friends). The ranking team has already designed their retrieval strategy: 1) users who are currently working in the target company will be ranked exclusively higher than those who previously worked in that company; 2) based on this, furthermore, direct friends will be ranked exclusively higher than indirect friends (i.e., friends of friends).

   Based on the graph structure and ranking strategy, you are recruited to implement a component to efficiently prefetch all the candidate users for the ranking team to generate the final ranking. Describe the procedure or any new data structure in your design to fulfill this goal with corresponding time and space complexity analysis.

   Build two inverted indices: from company name to current employees and previous employees (order users by user ID so that binary search is possible in the posting list). Maintain two lists to collect users who currently work in the target company and those who previously worked there, while inverted index lookup. Then 1) join the posting list from current employees with the user's friend list; 2) if we did not collect 10 candidates, join with the user's friends' friend list; 3) if we still did not get enough candidates, repeat 1) and 2) with previous employee list.
   Or you can design a linear scan algorithm without building inverted index (with proper shortcut design since we only need to return 10 candidates). The potential issue of the direct scan approach is that in average one user might have many (e.g., K) different employers, then inverted index lookup will provide K times speedup.
   As a result, it is preferred that you can analyze in what kind of situation, such linear scan will be more efficient, e.g., most users only have a few employers.
   (In fact I should not provide the friendship table, so that only the inverted index option is feasible. In current setting, both solutions will get full credit with proper explanation and analysis.)

2. Twitter is building its next generation of tweet search engine. Engineers are arguing that classical IR techniques are totally sufficient to build it so that there is no need for them to work overtime. As Twitter's chief research scientist in IR, do you agree with them? If not, what are the major technical challenges? What has to be innovated and what has to be adapted in such a system? (Hint: think about the system architecture of a classical IR system and the nature of Twitter.)

   Important aspects include: 1) dynamic indexing, given the emerging new tweets are generated every second; 2) document length normalization might not be essential, given the fixed maximum length of a single tweet; 3) freshness should be emphasized in twitter; 4) collaborative ranking, social connections, social endorsement (e.g., retweet, favorite) needed to considered in ranking; 5) the language on twitter is changing, e.g., jargon, abbreviations and hashtags, and it increases the vocabulary gap.

3. Our CS department plans to automatically construct a text profile (e.g., a list of keywords or phases) for every faculty member's research expertise according to their publications. You are invited as a consultant in this project. Can you suggest at least two different solutions for it?

   Feasible solutions include: 1) LSA; 2) TF*IDF weighting of N-grams; 3) mixture of topic models (we learned this in relevance feedback); 4) parse the publications and extract the keyword section; 5) collect the venues of those publications and extract more keywords there.
   The suggested solutions directly related to this class are the first three; and the other solutions are collected from students' answers.

4. Walmart lab is conducting an important research project to improve *walmart.com*'s product search effectiveness. Multiple teams claim their algorithms are the best and should be deployed. As the manager of this project, what would be your judging criteria? How should you make a reasonable decision accordingly?

   Important aspects to consider include: 1) sample a large collection of user queries, have domain experts to annotated the return results (e.g., using pooling technique to collect ranking results from different ranking systems), and then compute standard IR evaluation metrics, e.g., MAP, MRR and NDCG, to make the comparison; 2) perform online A/B test or interleaved test based on users' search behaviors (implicit feedback), e.g., click, abandonment rate, purchase rate; 3) compare revenue (since this is Walmart); 4) customer survey and questionnaire can also be used.

5. Bing is a keyword-based retrieval system similar as what you have built in MP2 (although it is clearly not ☺). Given a particular user's immediate search history, e.g., queries $\{q_{i-k}, q_{i-k+1}, \ldots, q_{i-1}\}$ ($k > 1$ and queries are ordered by time) and corresponding result clicks in each of those queries, how could you improve Bing to better serve the user's current query q_i, assuming $\{q_{i-k}, q_{i-k+1}, \ldots, q_{i-1}\}$ and q_i are for the same information need. You can specify your solution with respect to any ranking algorithm you have implemented in MP2. (Hint: here you have the query reformulation chain and result clicks.)

   Important aspects to consider: 1) query reformulation chain: removed query terms in previous queries could be considered as negative feedback, while kept terms could be considered as positive feedback; 2) click feedback from previous queries can be treated as relevant feedback for current query: those being skipped can be treated as negative feedback, while those being clicked can be treated as positive feedback, but documents that are nearly duplicated to previously clicked documents should not get too much promotion. 3) similarity between the current query and previous queries can be considered as weight for the click feedback.

4 Bonus Questions (15 pts)

All these questions are supposed to be open research questions. Your answers have to be very specific to convince the instructor that you deserve the bonus (generally mention some broad concepts will not count). Five points for each question.

1. How could you model negative feedback (i.e., irrelevant documents) in language models?

   I do not have an answer for it. I am expecting some idea related to the KL-divergence based language model framework.

2. How could you predict what will be a user's next query?

   Helpful aspects include: 1) current user's search history; 2) all the other users' search history, who share some immediate search history with the current user; 3) more general search context: location, time, current search trend.

3. Your suggestions of how to make our IR class better for next year?

   Anything reasonable will get you the full credit.