University of Virginia
Department of Computer Science

CS 4501: Information Retrieval
Fall 2015

5:00pm-6:15pm, Monday, October 26th

Name:
ComputingID:

- This is a **closed book** and **closed notes** exam. No electronic aids or cheat sheets are allowed.
- There are 7 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 (12×2 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). Two point for each question. *Note: the credit can only be granted if your explanation for the false statement is correct.*

1. Breadth-first crawling helps us avoid duplicated visit of web sites.
   ***False**, and Explain*: given the web is a connected graph, breadth-first crawling cannot avoid circles in the graph. Recording the visited URLs has to be done.

2. Given a well-tuned unigram language model $p(w|\theta)$ estimated based on all the text books about the topic of "information retrieval", we can safely conclude that p("information retrieval"$|\theta$)>p("retrieval information"$|\theta$).
   ***False**, and Explain*: unigram language model cannot capture the order of words.

3. Assume we use Dirichlet Prior smoothing; duplicate the document content multiple times will not change the resulting smoothed document language model.
   ***False**, and Explain*: duplicate the document will increase the document length, which affects Dirichlet Prior smoothing's results.

4. Sublinear TF scaling guarantees that the normalized TF is upper bounded.
   ***False**, and Explain*: the log function is not upper bounded.

5. We do not use a database system to solve information retrieval problems mostly because of efficiency concern.
   ***False**, and Explain*: the major concern is that a database system cannot deal with unstructured text content.

6. Adding an extra word that occurs in the document into a given query will increase the query likelihood of this particular document.
   ***False**, and Explain*: given a term's generation probability is always smaller than one, adding a new term can only decrease the query likelihood score. (If your argument is that the added word could be a stopword and therefore does not affect the query likelihood, it is also considered as correct.)

7. Mean average precision is biased by queries with more relevant documents.
   ***False**, and Explain*: given Average Precision is upper bounded by one and MAP gives equal weight to all the queries, MAP is not biased by queries with more relevant documents.

8. Term independence is a basic assumption in all retrieval algorithms we have learned.
   ***False**, and Explain*: There is no retrieval algorithm we have learned assumes term independence; we make such an assumption is only to reduce computational complexity.

9. N-grams can solve the problem of phrase query.
   ***False**, and Explain*: N-grams cannot solve the problem of phrase query since we do not know the length of query before building the index. We have to store term positions in the inverted index to solve this problem.

10. Latent Semantic Analysis gives us linearly independent basis vectors.
    True

11. Stopword removal will decrease recall.
    True

12. The compression ratio for the posting lists associated with head words will be higher than those associated with tail words.
    True

2 Short Answer Questions (32 pts)

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

1. Let D be a document in a text collection. Suppose we add a copy of D to the collection. How would this affect the IDF values of all the words in the collection? Why? (6pts)

$$IDF(w) = \begin{cases} 1 + log(\frac{N+1}{DF(w)+1}) & \text{if } w \in D \\ 1 + log(\frac{N+1}{DF(w)}) & \text{otherwise} \end{cases}$$

   where N is the original collection size, $DF(w)$ is the original document frequency of word w. Therefore, when w occurs in D, its new IDF decreases (since $N \geq DF(w)$); otherwise, its new IDF increases.

2. In what situation a system's MAP performance will be equal to its MRR performance? (4pts)

   Any of the following situations will lead the same MRR and MAP performance:

   - There is only one relevant document in every query in the test collection.
   - There is no relevant document in every query in the test collection.
   - All documents associated with each query in the test collection are relevant.
   - We have perfect ranking under each query in the test collection.

3. What are the basic assumptions in a query generation model? (6pts)

   - $p(Q|D, R = 0) \approx p(Q|R = 0)$
   - Uniform document prior

4. What are the three basic assumptions in classical IR evaluation? (6pts)

   - query is a proxy of users' information need
   - document relevance is independent from each other
   - sequential browsing from top to bottom

5. For a particular query q, the multi-grade relevance judgements of 50 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. All the other documents are judged as irrelevant. Two ranking systems return their retrieval results with respect to this query as, System A: $\{d_1, d_2, d_3, d_5, d_6, d_7\}$ and System B: $\{d_{21}, d_{22}, d_3, d_6, d_{15}\}$. These are all results they have returned. 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, but 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.) (10pts)

Metric	System A	System B
MRR	1.0	1/3
AP	$(1+\frac{2}{3}+\frac{3}{5})/6$	$(\frac{1}{3}+\frac{2}{4})/6$
NDCG	$(\frac{2^1-1}{\log_2(1+1)}+\frac{2^4-1}{\log_2(1+3)}+\frac{2^2-1}{\log_2(1+5)})/\text{iDCG}$	$(\frac{2^4-1}{\log_2(1+3)}+\frac{2^2-1}{\log_2(1+4)})/\text{iDCG}$

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

What is the agreement rate between these two systems, i.e., the kappa statistic?

	A:1	A:0
B:1	2	3
B:0	4	41

$P(A)=\frac{41+2}{50}$, $P(E)=(\frac{5+6}{100})^2+(\frac{44+45}{100})^2$, $\kappa=\frac{P(A)-P(E)}{1-P(E)}$

3 Essay Questions (44 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. How can we design a new Mean Average Precision metric to incorporate multi-grade relevance judgments, e.g., 0 - irrelevant, 1 - fair, 2 - good, 3 - excellent, 4 - perfect. (12 pts)

   There are many possible solutions.
   One solution is to first compute multiple MAPs with varying relevance thresholds and then calculate the weighted average as the final judgment. For example, in round one, we treat only perfect labels as relevant and the rest as irrelevant to compute the corresponding MAP; and then in round two, we treat excellent and above as relevant to compute the corresponding MAP, etc. In the end, we compute the weighted average of those MAPs, where a similar weighting scheme as used in DCG can be adopted, e.g., weight $\propto 2^i - 1$.
   A simpler solution is to modify the calculation of precision, e.g., compute the accumulated grades on every relevant document, and follow the rest procedures in MAP computation.

2. If we do not assume document independence, how can we design a better ranking algorithm? (12 pts)

   We can combine content diversity and result relevance in result ranking. We can still assume sequential browsing, and then the ranking score of a document to be placed at rank position i can be computed as,

   $$s(d, q, D_i) = rel(q, d) - \alpha \sum_{d_j \in D_i} sim(d, d_j)$$

   where d a candidate document from the whole collection, q is the input query, D_i is the set of documents already displayed above position i, $rel(q, d)$ is the relevance quality of document d to query q, and $sim(d, d_j)$ is the content similarity between document d and d_j. As a result, we assume the dependency between documents can be described by content similarity. Other similarities can also be explored, e.g., hyperlink structure and class/topic categorizations.
   The intuition is that we want to place a document at position i which is both relevant to the query but not very similar to the already displayed documents above position i.

3. Stackoverflow wants to redesign its current search function: it is preferred if it can directly answer questions rather than simple keyword matching in all forum posts. Based on the concepts you have learned in this class, can you design a tailored ranking system for them? *Hint: you can discuss what components we can reuse from a generic search engine, and what components we need to redesign to accommodate documents*

from Stackoverflow (e.g., consider its discussion structure, program snippet in post content, tags, vote count, authors' reputation, etc.). (20 pts)

Given this retrieval is performed on Stackoverflow's own corpus, there is no need to perform crawling. In the inverted index, we need to store more information than term frequency. For example, we can store the position of those terms in a post, so that we can deal with phase questions and we can also know whether such term occur in the document title or body. We should also store the authors of the posts, vote count of the post, tags of the post and authors' reputation. We should store if a post simply contains a question or being labeled as an answer.
Based on such information available in the index, in the retrieval phase, we need to first search against the existing question post (usually the first post of a thread) and return their answers to the new input questions. If there is not enough match, we can go against the other posts in the thread as well. During the ranking process, we can not only match the query term against the actual post content, but also the associated tags (LSA can also be performed to identify potential latent topics or clustering of words). We should give higher weights to the documents receiving more helpfulness vote, or already labeled as an answer (e.g., through non-uniform document prior, or giving extra ranking score to those documents). Since most of questions are related to particular programming language, posts containing code snippets of targeted language should have higher weight.

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).

1. How to utilize user clicks in search evaluations? (10pts)

   We can simply treat clicked documents as relevant and non-clicked documents as irrelevant. We can further consider the click dwell time to further differentiate the quality of those clicked documents, e.g., if a user spends longer time on a clicked document, which should be considered as more helpful than those with shorter dwell time. Pairwise relationship between the clicked and non-clicked documents can also be exploited. For example, by assuming sequential browsing, a non-clicked document followed by an immediately clicked document will be considered as with lower quality than the latter one. Last click in a session usually indicates high likelihood that a user is satisfied by the content in that document, so that higher weight could also be given to such documents.

2. List your favorite (labeled as +) and disliked (labeled as -) aspects of this class. (5pts)