Student Name: _____

Andrew ID: _____

Seat Number: _____

Midterm Exam
Search Engines (11-442 / 11-642)
October 20, 2015

Answer all of the following questions. Each answer should be thorough, complete, and relevant. **Points will be deducted for irrelevant details.** Use the back of the pages if you need more room for your answer.

Calculators, phones, and other computational devices are not permitted. If a calculation is required, write fractions, and show your work so that it is clear that you know how to do the calculation.

Advice about exam answers...

- Sometimes an answer says "I would use <technique> to do <x>". That answer shows that you remember a name, but it does not show that you remember how the technique works, or why it is the right tool for this problem. Give a brief description of how the technique works and why it is the right tool for this job. If the technique needs other information, explain where the information comes from.

.

1 Zipfs Law

Write the formula for Zipf's Law (define your terms). Give one practical example of its use (i.e., a situation where it would be useful). **[10 points]**

Answer

The typical expression of Zipf's Law is

$$Rank_t \times ctf_t = A \times N$$

where:

- $Rank_t$: The rank of term t, when terms are sorted by ctf.
- ctf_t: The collection term frequency of term t
- A: A constant, often about 0.1
- N: The number of word occurrences in the corpus

Variations that are mathematically equivalent are acceptable answers.

Zipf's Law is useful for predicting the amount of storage required for terms that occur more than n times. (We accepted a variety of answers for practical examples.)

2 Indexing

Describe the use of multiple inverted lists for the same index term to improve retrieval efficiency. Provide two examples of additional types of lists that a system might store, and explain how they improve efficiency. What is an advantage and a disadvantage of maintaining multiple inverted lists per term? Would all index terms have multiple lists, or just some of them? Why? (Hint: This question is not about Top-docs / Champion lists.) **[14 points]**

Answer:

When disk space is not a problem, it makes sense to store different inverted lists that contain different amounts of information so that I/O and computation are minimized whenever possible. For example, a system might have a list that contains only document ids for unranked Boolean operators, a list that contains only document ids and term frequency for score operators, and a list that contains document ids, term frequency, and positions for positional operators such as NEAR/n and WINDOW/n. The advantage is that each operator does only the I/O and decompression that it absolutely needs. The disadvantage is an increase in disk space and term dictionary (pointers to inverted lists on disk). All index terms would have multiple lists because any index term can be used by any query operator.

3 Document Representation

a. What kind of words are considered stopwords? How are stopwords identified? Give examples and describe the advantages and disadvantages of removing stop words. Explain when it might be wise to leave stopwords in a query. **[6 points]**

Answer

Stopwords are words such as 'the', 'of', 'a' that are very frequent in a corpus and unlikely to have much meaning on their own.

- They are usually identified through a combination of frequency analysis and manual review. For example, consider all frequent terms as stopwords, but examine a query log to see which frequent terms might be important. If a word is frequent and important in the query log, don't make it a stopword.
- Advantage: Discards meaningless words, reduce index size, improve accuracy.
- Disadvantage: "to be or not to be" after removal, it makes queries difficult to satisfy.
- If there is no disk limit, we could store stop words in our index. If stop words are more than half the terms in the query, keep them in the query.

b. What is stemming, and how is it usually performed? Give examples and describe its advantages and disadvantages. Is stemming used primarily to improve Recall or Precision, and why? Describe the differences between the Porter and Krovetz (K-Stem) stemmers. **[9 points]**

Answer

Stemming is the conflation of lexical variants such as ‘apple’ and ‘apples’ into a single index term.

- Stemming is usually performed by morphological analysis software (‘stemmers’ in English) written by computational linguists and NLP researchers.
- Advantage: Improves recall, matches lexical variants that have the same meaning.
- Disadvantage: May conflate words that a person would consider different (e.g., ‘execute’ and ‘executive’) or different in a particular situation (e.g., the company ‘apple’ and ‘apples’).disadvantage: too expensive to do when processing many docs.
- Stemming improves Recall, because it allows a query to match documents that have minor lexical differences with the query.
- Porter: Aggressive stemmer that produces large conflation classes. The stems are not necessarily English words. Krovetz stemmer: Conservative stemmer that produces smaller conflation classes. The stems are very likely to be English words. Porter and Krovetz produce equally accurate search results.

4 Pseudo Relevance Feedback

Describe Indri's pseudo relevance feedback algorithm. Be clear about the parameters that affect the behavior of the algorithm. Include the formula for the later version of the term weighting algorithm (define your terms). Show what the final combination of the original query and expanded query looks like. **[16 points]**

Answer:

Algorithm: Run the original query. Select the top-ranked M documents. Every term in those documents is a candidate expansion term. Calculate a score for each term using the following formula:

$$p(t \mid I) \propto \sum_d p(t \mid d) p(I \mid d) \log \frac{1}{p(t \mid C)}$$

$$\propto \sum_d p(t \mid d) p(I \mid d) \log \frac{length(C)}{ctf_t}$$

where:

- *d*: a top-ranked document
- *p(t|d)*: the MLE score of *t* in d, defined as *tf / |d|*
- *p(I|d)*: the score of document *d* for the original query
- *p(t|C)*: the MLE score of *t* in the entire collection, defined as *ctf/|C|*.

Select the top N terms. Form a #wand query from the top N terms using their *p(t|I)* weights. The final query is #wand (w original_query (1-w) expanded_query), where w is a parameter (often about 0.5).

5 Retrieval Models

Describe what document priors are, and why they are useful to a retrieval model. Given two examples of document priors. Show how document priors are used with the BM25, query likelihood, and KL divergence retrieval models. **[15 points]**

Answer

Document priors are query-independent estimates of the value of each document, typically based on document characteristics. Examples include priors based on a spam score, PageRank, or the length of the url. They are useful because they are a measure of the general quality or value of a document that is unrelated to the details of the query.

BM25:
$$\begin{aligned} p(R \mid d) &= p(R \mid d_T, d_F) \\ &\propto \text{BM25}(d_T) + \sum_{d_i \in d_F} \log \frac{p(d_i \mid R)}{p(d_i \mid \bar{R})} \\ &\propto \text{BM25}(d_T) + \sum_i w_i F_i(d_i) \end{aligned}$$

Query likelihood:
$$p(q \mid d) \propto \log p(d) + \sum_{q_i \in Q} \log p(q_i \mid d)$$

KL divergence:
$$p(q \mid d) \propto \log p(d) + \frac{1}{|Q|} \sum_{q_i \in Q} \log p(q_i \mid d)$$

6 Document Structure

Write the formula for BM25F (and define the variables). Explain the intuition that motivates BM25F, i.e., whether it views a document as multiple bags-of-words or a single bag-of-words. Explain the purpose of each BM25F parameter. Given Query q, under what conditions would the documents below look the same to BM25F? Explain your answer. **[15 points]**

Query: #sum (a b c)

$Document_1$		$Document_2$
a a a	$field_1$	a b c
b b b	$field_2$	a b c
c c c	$field_3$	a b c

Answer:

$$BM25F(q,d) = \sum_{t \in q} \left(\log \frac{N - df_t + 0.5}{df_t + 0.5} \right) \frac{tf_t}{k_1 + tf_t}$$

$$tf_t = \sum_{f \in F} w_f \frac{tf_{t,f,d}}{(1 - b_f) + b_f \frac{length_{f,d}}{avglength_f}}$$

N: number of documents in the corpus
df_t: number of documents that contain query term t
k_1: the saturation parameter for the entire document
w_f: a weight indicating the importance of field f
$tf_{t,f,d}$: number of times query term t occurs in field f of document d
b_f: a parameter that controls the length normalization for field f
$length_{f,d}$: length of field f in document d
$avglength_f$: the average of all fields of type f in the corpus

BM25F views a document as a single bag-of-words that is constructed by sampling each document field. Some fields are more important or more relevant than other fields, so the degree of sampling is based on the relative importance of each field f, indicated by a parameter w_f. A single saturation parameter is k_1 is used for the single, document-level bag-of-words. The length normalization of each field f is controlled independently by a parameter b_f:

The documents would look the same if the three fields have the same average field lengths, and if the search engine uses the same w_f and b_f for each field.

7 Evaluation

For different search scenarios, we use different evaluation metrics to judge a ranked list. For each of the scenarios below, choose the most appropriate metric. Justify your choice. **[15 points]**

a) A student searching for the 11-642 course page.

b) A customer who wants to buy a laptop case from an online store. The search results have different sizes, looks, ratings, and prices. The customer likes or dislikes them in varying degrees.

c) A lawyer who is working on finding related documents from the U.S. Court of Federal Claims cases for her upcoming trial. She has a staff to look through many documents if necessary.

Answer:

a) Reciprocal Rank, because there is exactly one relevant document. If the relevant document is not ranked first on the search results page, the penalty should be strong.

b) NDCG, because the relevance value is multi-valued, there may be multiple relevant documents, and the probability that a person views a page depends upon its rank.

c) This user cares about Recall, so a Recall-oriented metric is best. Set-based Recall would be a good choice if the lawyer uses Boolean queries. Otherwise, Mean Average Precision (MAP) would be a good choice, because it considers all of the known relevant documents.