Name: _____

CSE 435/535 Information Retrieval Fall 2015
Midterm (90 minutes, 100 points)

Question 1: State whether the following statements are True or False (10 points) – You do not have to give reasons.

a) If the Jaccard distance between two strings is small, their edit distance will also always be small (False)

If strings are reverse of each other, jacquard distance is 0 but not edit distance

b) Term-at-a-time scoring results in better search results than document-at-a-time scoring. (False) these are used for computational efficiency, not for better results

c) In VSM, the similarity scores can never be negative. (True)

All vectors are always positive

d) Heaps' law helps in estimating the size of the dictionary for a given collection (True)

e) Creating a champion list helps in processing Boolean queries faster (False)
Boolean queries don't care about tf s.

f) Lowercasing all tokens during indexing can never hurt recall (True)

g) Using pivoted length normalization for project 1 would help improve precision. (False)
Project 1 had all documents of comparable length

h) Forward indexes can be helpful in generating dynamic summaries (False)
Forward indexes don't store positional information

i) For a term partitioned index, addition of a new term does not affect all index partitions (True)
Only the affected partition needs to be changed

j) If each term is assigned a rank based on the frequency of its occurrence in the corpus, then the ratio of ranks of two such terms is equal to the ratio of their IDFs. (True)
If the most frequent term occurs N times, term at rank k occurs N/k times by Zipf's law. Thus for ranks m and n, (m/n) = (m/N) / (n/n)

Question 2: Multiple Choice (10 points)

(i) Which of the following evaluation metrics do commercial search engines most often use: (a) Mean Average Precision (b) NDCG (c) R-precision.

Answer: B

(ii) Which of the following is not a reason to do stemming: (a) increase precision (b) increase recall (c) decrease index size

Answer: A

(iii) Which of the following is not a type of IR model: (a) Boolean (b) Probabilistic (c) Vector Space (d) Term-at-a-time

Answer: D

(iv) Which of the following is not a part of a TREC query: (a) topic (b) title (c) description (d) narrative

Answer: B

(v) Consider the starting few bits of a gamma-encoded index as follows: 1111101001111111101111111.... The reconstructed postings list looks like: (a) 51, 63, … (b) 19, 31, … (c) 51, 114, … (d) 19, 50, …

(Ans: C) Decode gamma code only until end of offset of second gamma code - works out to 51,63. Since thats the encoded gap - posting list is 51,114,..

(vi) Which of the following is not an example of a variable length encoding scheme: (a) Gamma codes (b) Unicode (c) Soundex (d) Metaphone

Answer: C

(vii) The edit distance between the strings "live" and "evil" is: (a) 1 (b) 2 (c) 3 (d) 4

(Ans : D) Normal levenshtein computation

(viii) Which of the following tf formulations is incorrect: (a) 1 - log (tf) (b) sqrt(tf) (c) tf / max(tf) (d) 0.5 + (log(tf) / log(avg(tf))

(Ans : A) All others are strictly increasing with increasing tf except (a)

(ix) For the query ***je suis Charlie*** (in French) on your index for Project 1, which approach would guaranteed give you poor precision assuming everything else is unchanged (a) Hashtag tokenization based on case changes (#TestHashtag == Test, Hashtag) (b) Word by word translation of query into each target language (c) Changing default operator to AND (d) Using quote characters to determine and index phrases

(Ans : b) By word by word translation, a lot of false positives would show up. All other approaches use some sort of phrase search

(x) MapReduce would be helpful in speeding up processing in all of the following cases except: (a) Document indexing (b) Query processing (c) Document Clustering (d) Distributed Index merging

(Ans : D) Index merging requires iteration over the full index, can't be parallelized.

Question 3: Dictionaries and Tokenization (10 points)

(i) (3 Points) : Whitespace segmentation works for English language tokenization - what is a technique that would work for languages with little or no whitespace?

Answer: Character N/K-grams: treat overlapping sequences of n characters as tokens.

(ii) (2 Points) : How does stemming affect recall? What about precision?

Answer: Improves recall, hurts precision.

(iii) (5 Points) : Why should we store the document frequency with the term in the term-dictionary? Hint: Think about what happens when we do Boolean retrieval like "(Calpurnia and Brutus) and Caesar"

Answer: If we're performing boolean retrieval we only want documents matching the boolean expression, in the example above, documents that have calpurnia, burtus and caesar. So we know if Caesar has a small number of documents, we can check only those documents for the other two terms. As the dictionary is typically stored in memory, this should be much faster then going to disk.

Question 4: Indexing and SOLR (20 points)

(Part A) (6 Points) : A key element of the inverted index is the postings list. Answer the following questions about the posting list:

(i) In big "O" notation, how long does it take to get the intersection of a m length list with an n length list? (2 Points)

O(m + n)

(ii) What benefit does a skip list give you? (2 Points)

A skip list can decrease the lookup speed for finding a particular document in the posting list by skipping over unecessary documents.

(iii) Suppose the number of terms in the skip list is N. Compare the performance of a skip list that has pointers to every other term, e.g. at location n there is a skip to n + 2, to one that is roughly the square root of N. In general, which is faster? (2 Points)

For the two element skip: while we may skip a lot we will also be increasing storage space and comparisons. Obviously we'll skip less for the square root skip, but the skips should be more effective and take up less space. The heursitic given in class was square root of the length of the list.

(Part B) (6 Points) : Given a query phrase, "Pluto is still a planet", show an example positional index where "is" and "a" are stopwords (not indexed) and the remaining terms have 3 documents each. One document should have the phrase, the other two should not (they could be separate documents). For brevity, assume the term id is just the term itself, do not include the dictionary. Use the form <-----> as an entry in the postings list.

Answer:
<pluto:3, 1: 15, 2: 10, 3: 15>,
<still:3, 1: 17, 4: 10, 5: 15>,
<planet:3, 1: 19, 6: 10, 7: 15>,

(Part C) (8 Points) : With regard to Solr, answer the following questions.

(i). What does the stored flag do on a field?

If true, nothing, if false, it might change the format of the data. It's still searchable, just not presentable.

(ii) . If you didn't use managed indexing, how would you cope with fields that have a fixed suffix, say '_t', but whose prefix you're not aware of ahead of time?

Dynamic fields.

(iii) What flag should you set for a field to have multiple entries?

multivalued

(iv). What happens if the index flag is set to false?

Solr data is stored but not available for search.

Question 5: Vector Space Model (15 points)

Consider the following document collection:

Doc1: You say goodbye, I say hello
Doc2: You say stop, I say go
Doc3: Hello, hello, you say goodbye

Part A (5 points) Generate a table containing (i) the vocabulary of tokens/terms assuming no stemming (ignore capitalization and punctuation), in alphabetical order, and (ii): the idf for each of the terms in the collection.

(log10(3/3)=0, log10(3/2)=0.18, log10(3/1)=0.48)

terms	df	idf
you	3	0
say	3	0
goodbye	2	0.18
i	2	0.18
hello	2	0.18
stop	1	0.48
go	1	0.48

Part B (5 points) Generate the document term matrix based on tf-idf weighting

	you	say	goodbye	i	hello	stop	go
Doc1	0	0	0.18	0.18	0.18	0	0
Doc2	0	0	0	0.18	0	0.48	0.48
Doc3	0	0	0.18	0	0.36	0	0

Part C (5 points): List three distinct limitations of vector space models.

- Long documents are poorly represented because they have poor similarity values (a small scalar product and a large dimensionality)
- Search keywords must precisely match document terms; word substrings might result in a "false positive match"
- Semantic sensitivity; documents with similar context but different term vocabulary won't be associated, resulting in a "false negative match".
- The order in which the terms appear in the document is lost in the vector space representation.
- Theoretically assumes terms are statistically independent.
- Weighting is intuitive but not very formal.

Question 6: Query Processing and Scoring (15 points)

Given the same collection as in the previous question, and the query "I hello"

Part A: rank the documents by similarity to the query. As for the term weighting in the query, it is 1 if present in the query and 0 otherwise.

score(Doc1, q) = 0.36
score(Doc2, q) = 0.18
score(Doc3, q) = 0.36
Rank: Doc3 = Doc1>Doc2

Part B: Let's say now we also have static scores for each of the documents, as follows:

g(Doc1)=0.7, g(Doc2)=0.4, g(Doc3)=0.3

Recalculate the scores for each document with the query "say hello", and re-rank them accordingly.

score(Doc1, q) = 1.06
score(Doc2, q) = 0.58
score(Doc3, q) = 0.66
Rank: Doc1> Doc3>Doc2

Part C: In real retrieval systems, we will have significantly more documents, not just 3! . Indicate why ordering the postings lists by decreasing order of TF would reduce the query time for term-at-a-time query.

when traversing the postings list for a query term t, we stop after considering a prefix of the postings list – either after a fixed number of documents r have been seen, or after the value of tft,d has dropped below a threshold

Question 7: Evaluation (20 points)

Part A (12 points) Consider an information need for which there are 10 relevant documents in the collection, The top 20 results are judged for relevance as follows, Rs and Ns represent Relevant (R) and Non-relevant (N) document returned by this system. The leftmost document is the top ranked searched result.

R N R N N R N N N R N N N N R N N N N R

(1) What is the precision of the system on top 10 documents?
4/10 = 0.4

(2) What is the interpolated precision at 25% recall?
0.5

(3) What is the MAP of this system for this query if the total number of relevant documents is 6 instead?

1/6 * (1 + 2/3 + 3/6 + 4/10 + 5/15 + 6/20) = 8/15 = 0.53

(4) What is the R-precision of this system if the total number of relevant documents is 6 instead?
3/6 = 0.5

Part B (2 points) Give one reason why one would prefer using precision at K measure?

The advantage of precision at K measure is that the total number of relevant document doesn't need to be known, which appeals to applications such as web search.

Part C (5 points) Given a query, the system returns the following ranked document list as follows.

d1, d2, d3, d4

The relevancy score for each of these documents are given as

d1: 3, d2: 2, d3: 3, d4: 0

If the ideal relevancy score is

3, 3, 2, 0

What is the nDCG of this system?
(given: $log_2 1 = 0, log_2 2 = 1, log_2 3 = 1.585, log_2 4 = 2$)

DCG = 3 + (2/1 + 3/1.585 + 0/2) = 6.8927
iDCG = 3 + (3/1 + 2/1.585 + 0/2) = 7.2618
nDCG = DCG/iDCG = 0.9491

Part D (1 point) Give an example of a search system where recall is more important than precision.

professional searches such as paralegals and intelligent analysts.