UNIVERSITY OF OSLO

Faculty of Mathematics and Natural Sciences

Exam in INF3800/INF4800 Search Technology
Day of exam: June 2nd, 2017
Exam hours: 14:30-18:30 (4 hours)
This examination paper consists of 3 page(s)
Appendices: None
Permitted materials: None

Make sure that your copy of this examination paper is complete before answering.

You can answer in Norwegian or English. Please use the language that you are most comfortable with.

TERMS OF ENDEARMENT (10%)

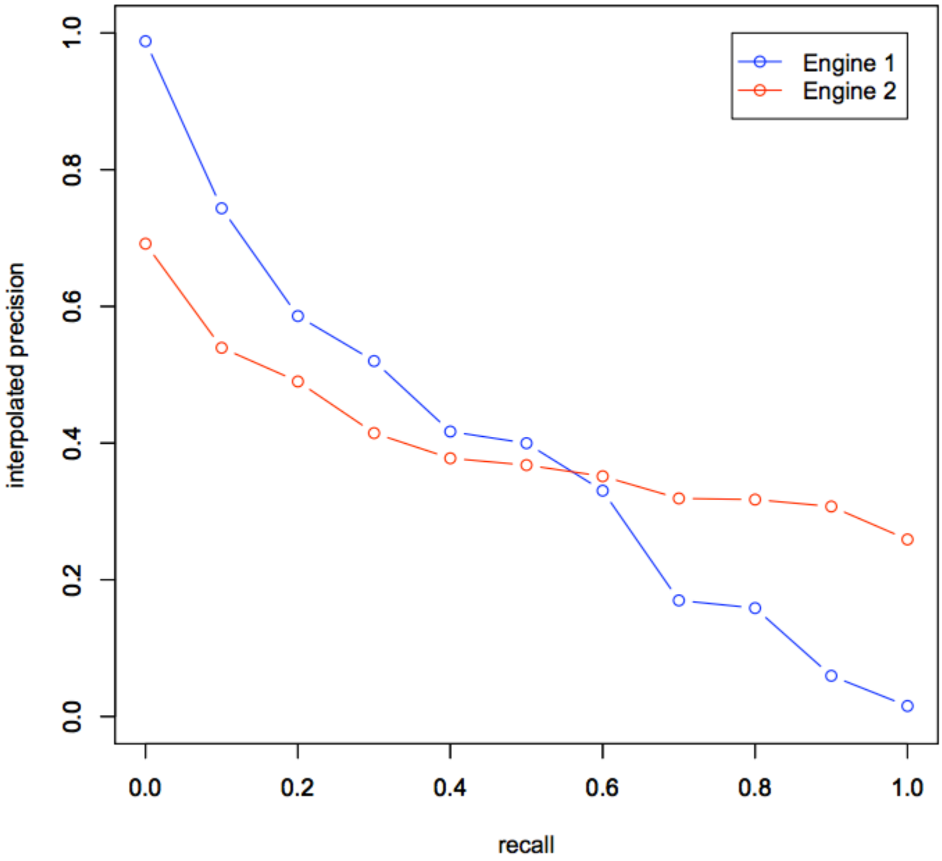
a) [5 points] Why are the notions of term frequency and inverse document frequency used so often in document scoring functions?

b) [5 points] How do stopword removal and stemming reduce the size of an inverted index?

TOTAL RECALL (15%)

a) [5 points] For a given query there are 16 relevant documents in the collection. The precision for the query is 0.40, and the recall for the query is 0.25. How many documents are in the result set?

b) [10 points] The figure below depicts interpolated precision-recall curves for two search engines that index research articles. There is no difference between the engines except in how they score documents. Imagine you're a scientist looking for all published work on some topic. You don't want to miss any citation. Which engine would you prefer and why?



THE HOBBIT (15%)

a) [10 points] A term-document incidence matrix with Boolean entries (indicating the presence or absence of a term in a document) is sometimes referred to as a bit vector index. With 5000 documents and 10000 unique vocabulary terms, a bit vector requires 5×10^7 bits of storage. Suppose documents have 200 terms on average. If we added 2200 more documents to the collection, roughly how big would the bit vector index become? Use Heaps' law with k = 10 and β = 0.5.

b) [5 points] Show how to γ-encode the integer 24.

NO STRINGS ATTACHED (20%)

a) [6 points] Provide a short and informal description of the Aho-Corasick algorithm, and what can it be used for.

b) [7 points] Briefly outline how edit distance between two strings is defined, and how you can use dynamic programming and a 2D table to compute the edit distance between two string *s* and *t*.

c) [7 points] Now assume that you need to efficiently find all strings in a big dictionary *D* that have an edit distance smaller than *k* from *s*. You can assume that *k* is a small number. Which properties of the abovementioned dynamic programming algorithm and the 2D table can you exploit to make this search efficient?

MY NEIGHBOR TOTORO (20%)

a) [7 points] Explain briefly the principles behind the Rocchio algorithm and the *k* nearest neighbor (*k*-NN) algorithm, respectively.

b) [7 points] Discuss their similarities and differences.

c) [6 points] Show how the two algorithms may produce different classification results.

THE POSTMAN ALWAYS RINGS TWICE (20%)

a) [10 points] When traversing multiple posting lists and scoring documents, there are several strategies available to us. Briefly explain the difference between document-at-a-time and term-at-a-time scoring, and discuss some of their relative merits.

b) [10 points] Assume that you have two posting lists with *n* and *m* postings, respectively. You can compute their intersection in $O(n + m)$ time, but can you do better in some cases? Explain how, or explain why not.