

# UNIVERSITY OF OSLO

## Faculty of Mathematics and Natural Sciences

**Exam in INF3800/INF4800 Search Technology**

**Day of exam: June 10<sup>th</sup>, 2014**

**Exam hours: 14:30-18:30 (4 hours)**

**This examination paper consists of 4 page(s)**

**Appendices: None**

**Permitted materials: None**

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

## QUERY PROCESSING (20%)

- a) [8 points] For a conjunctive query (AND), is processing postings lists in order of size guaranteed to be optimal? Explain why it is, or give an example where it isn't.
- b) [12 points] We have a two-word conjunctive query. For one term the postings list has the 16 entries [4,6,10,12,14,16,18,20,22,32,47,81,120,122,157,180], and for the other it is the one-entry postings list [47]. Work out how many comparisons would be done to intersect the two postings lists with the following two strategies. Briefly justify your answers.
- i. Using standard postings lists.
  - ii. Using postings lists stored with skip pointers, with a skip length of  $\sqrt{P}$  where  $P$  is the length of the postings list.

## HEAPS' LAW (20%)

- a) [5 points] Heaps' law is given as  $M = kT^b$ . Explain what  $M$ ,  $k$ ,  $T$  and  $b$  are.
- b) [15 points] Looking at a collection of web pages, you find that there are 3,000 different terms in the first 10,000 tokens and 30,000 different terms in the first 1,000,000 tokens. Assume a search engine indexes a total of 20,000,000,000 ( $2 \times 10^{10}$ ) pages, containing 200 tokens on average. What is the size of the vocabulary of the indexed collection as predicted by Heaps' law?

## LOOKUP FUN (20%)

- a) [10 points] Write down the entries in a permuterm index that are generated by the term *sting*. If you wanted to search for *s\*ng* in a permuterm wildcard index, what key(s) would one do the lookup on?
- b) [10 points] Consider the term *mississippi*. Write down the suffix array for this term, and explain how you can use this to efficiently locate all occurrences of the substring *is*.

## COSINE SCORES, PAGERANK AND CLASSIFICATION (40%)

You want to automatically classify web documents according to their relevance to a given query. The documents contain both text and links to other documents. The classifier should rely on only two features:

1. The cosine score between a document and a query.
2. The PageRank of a document.

To build this classifier, you are given an example of query  $q$

$q$ : speech dialogue system

as well as four training documents  $\{d_1, d_2, d_3, d_4\}$  manually annotated as relevant or not relevant for the query  $q$ .

		Document content	Relevant for $q$ ?
Training	$d_1$	A spoken dialogue system is a dialogue system [ <a href="#">link to <math>d_3</math></a> ] delivered through voice. It has two components that do not exist in a text-based dialogue architecture: a speech recognizer [ <a href="#">link to <math>d_5</math></a> ] and a text to speech module.	<b>Relevant</b>
	$d_2$	Chatterbots are sometimes referred to as talk bots or chatterboxes. Chatterbots are often integrated into a dialogue agent [ <a href="#">link to <math>d_3</math></a> ] for practical purposes.	<b>Not relevant</b>
	$d_3$	A dialogue system is a computer system intended to converse with a human. They have employed speech [ <a href="#">link to <math>d_5</math></a> ], graphics, haptics, gestures and other modes for communication on both the input and output channel.	<b>Relevant</b>
	$d_4$	The core issue in such a speech system is the dialogue manager which is the element of system that determines what the system should say next.	<b>Not relevant</b>

Based on this training data, the objective is to classify the new document  $d_5$  as being relevant or not to the query  $q$ .

Testing	$d_5$	A speech recognition system may be speaker independent or trained by an individual speaker reading sections of text to the speech recogniser.	?
---------	-------	---	---

**Questions:**

- a) [8 points] Determine the cosine scores between each of the five documents and the query  $q$ . To simplify your calculations, you can ignore the vector normalization and assume the following weighting:

$$w_{t,q} = \text{tf}_{t,q} / \text{df}_t$$

$$w_{t,d} = \text{tf}_{t,d}$$

Take the five documents into account for the document frequency.

- b) [8 points] Determine the PageRank of each document, given the links between the five documents. Use a teleportation rate of 0.5 and assume the random walk starts at  $d_1$ . You can stop the calculations after 2 iterations. Hint: The final result should be [0.145 0.145 0.22 0.145 0.345].
- c) [4 points] Draw a two-dimensional plot with the cosine score on one axis and the PageRank score on the other axis. Place the five documents in this plot.
- d) [15 points] Classify the new document  $d_5$  as relevant or not relevant to  $q$ , using the three following classifiers trained on  $\{d_1, d_2, d_3, d_4\}$ :
- i. Rocchio classifier.
  - ii. 3-nearest neighbour.
  - iii. Linear SVM, given that the optimization solution returns the multipliers  $\alpha_1=0$ ,  $\alpha_2=0$ ,  $\alpha_3=20$ ,  $\alpha_4=20$  and the term  $b = -4.65$ .

To simplify the calculations, use the “Manhattan” metric below as a distance measure between vectors (instead of the Euclidean metric):

$$\|\vec{x} - \vec{y}\| = \sum_i |x_i - y_i| \quad \text{where } |\cdot| \text{ is the absolute value.}$$

- e) [5 points] Can you think of other features (besides the cosine score and the PageRank score) that could be useful for this classification task? Name at least two other features.