

## Information Retrieval: Assignment 1

### Problem 1. (10 points)

Try and find a query of the form [query-term-1 query-term-2] (without quotes) that, when run on Google, produces at least one result that either does not contain query-term-1 or does not contain query-term-2. That is, try to find an example where Google does not interpret a two-term query as a conjunction. (If you have difficulty with finding an appropriate query, try one that produces very few hits, say, fewer than 20.) (i) Print out the first page of Google results (or more if you want to) and mark each result with 2 (both terms occur on the page), 1 (one term occurs on the page) or 0 (neither term occurs on the page) (ii) Based on this evidence, does Google interpret all queries as a Boolean conjunction?

### Problem 2. (12 points)

Shown below is a portion of a positional index in the format: term: doc1: ⟨position1, position2, ...⟩; doc2: ⟨position1, position2, ...⟩; etc.

```
ANGELS: 2: ⟨36,174,252,651⟩; 4: ⟨12,22,102,432⟩; 7: ⟨17⟩;
FOOLS: 2: ⟨1,17,74,222⟩; 4: ⟨8,78,108,458⟩; 7: ⟨3,13,23,193⟩;
FEAR: 2: ⟨87,704,722,901⟩; 4: ⟨13,43,113,433⟩; 7: ⟨18,328,528⟩;
IN: 2: ⟨3,37,76,444,851⟩; 4: ⟨10,20,110,470,500⟩; 7: ⟨5,15,25,195⟩;
RUSH: 2: ⟨2,66,194,321,702⟩; 4: ⟨9,69,149,429,569⟩; 7: ⟨4,14,404⟩;
TO: 2: ⟨47,86,234,999⟩; 4: ⟨14,24,774,944⟩; 7: ⟨199,319,599,709⟩;
TREAD: 2: ⟨57,94,333⟩; 4: ⟨15,35,155⟩; 7: ⟨20,320⟩;
WHERE: 2: ⟨67,124,393,1001⟩; 4: ⟨11,41,101,421,431⟩; 7: ⟨15,35,735⟩;
```

Which document(s) (if any) match each of the following queries where each expression within quotes is a phrase query? (i) “fools rush in” (ii) “fools rush in” AND “angels fear to tread”. At which positions do the queries match? (iii) There is something wrong with this positional index. What is the problem?

### Problem 3. (10 points)

Compute the Levenshtein matrix for the distance between the strings “apfel” (input) and “poems” (output). Use this format (as introduced in class):

		f	a	s	t
	0	1 1	2 2	3 3	4 4
c	1 1	1 2 2 1	2 3 2 2	3 4 3 3	4 5 4 4
a	2 2	2 2 3 2	1 3 3 1	3 4 2 2	4 5 3 3
t	3 3	3 3 4 3	3 2 4 2	2 3 3 2	2 4 3 2
s	4 4	4 4 5 4	4 3 5 3	2 3 4 2	3 3 3 3

**Problem 4.** (10 points)

We saw in class that the Levenshtein sequence of operations for converting strings into each other is not unique. For example, “cat” can be transformed into “catcat” either by insert, insert, insert, copy, copy, copy or by copy, copy, copy, insert, insert, insert. In contrast, the minimum number of cost-1 Levenshtein operations for converting one string to another is fixed since the minimum is unique. Possible cost-1 operations are insert, delete and replace. Let  $n_i$ ,  $n_d$ ,  $n_r$  be the number of inserts, deletes and replaces in a sequence of operations. Give an example of a pair of strings and two different sequences of operations  $\sigma_1$  and  $\sigma_2$  that convert the first string into the second such that  $n_i(\sigma_1) \neq n_i(\sigma_2)$  or  $n_d(\sigma_1) \neq n_d(\sigma_2)$  or  $n_r(\sigma_1) \neq n_r(\sigma_2)$ . Or prove that this is not possible.

**Problem 5.** (1 points)

If you wanted to search for **s\*ng** in a permuterm wildcard index, what **key(s)** would one do the lookup on?

**Due date: Thursday, May 2, 2013, 12:00 (noon)**