SOLUTION SKETCH 2014

QUERY PROCESSING

(a) The order is not guaranteed to be optimal. Consider three terms with postings list sizes s1 = 100, s2 = 105 and s3 = 110. Suppose the intersection of s1 and s2 has length 100 and the intersection of s1 and s3 length 0. The ordering s1, s2, s3 requires 100+105+100+110=315 steps through the postings lists. The ordering s1, s3, s2 requires 100+110+0+0=210 steps through the postings lists.

(b) (i) Applying the intersection algorithm on the standard postings list, comparisons will be made unless either of the postings list end, i.e., till we reach 47 in the upper postings list, after which the lower list ends and no more processing needs to be done. Number of comparisons = 11.

(b) (ii) Using skip pointers of length 4 for the longer list and of length 1 for the shorter list, the following comparisons will be made:
1. 4 & 47
2. 14 & 47
3. 22 & 47
4. 120 & 47
5. 81 & 47
6. 47 & 47

Number of comparisons = 6

HEAPS' LAW

(a) Heaps' law is a purely empirical law where:

M = the size of the vocabulary
k, b = empirically determined constants
T = the number of tokens in the collection

(b) In log-log space we have:

log(M) = log(k) + b*log(T)

We have two data points in log-log space we can use to estimate b:

b = (log(3*10^4) - log(3*10^3)) / (log(10^6) - log(10^4))
 = (log3 + 4 - log3 - 3) / (6 - 4)
 = 1 / 2
 = 0.5

Since T^0.5 = sqrt(T) we have:

k = M/sqrt(T)
 = 3000/sqrt(10000)
 = 30000/sqrt(1000000)
 = 30

So, M = 30*sqrt(T).

In our example, we have T = 2*10^10*200 tokens, so:

M = 30*sqrt(2*10^10*2*10^2)
 = 30*sqrt(4*10^12)
 = 30*2*10^6
 = 60*10^6
 = 60000000
 = 60 million

LOOKUP FUN

(a) sting -> sting$, ting$s, ingst, ngsti, g$stin, $sting. Lookup on key ng$s*, i.e., a prefix lookup.

(b) Suffix array (using a 0-based indexing scheme):

10: i
7: ippi
4: issippi
1: ississppi
0: mississippi
9: pi
8: ppi
6: sippi
3: sissippi
5: ssippi
2: ssissippi

Binary search on the prefix is*, then a sequential scan (if match). Can exploit precomputed LCP values.

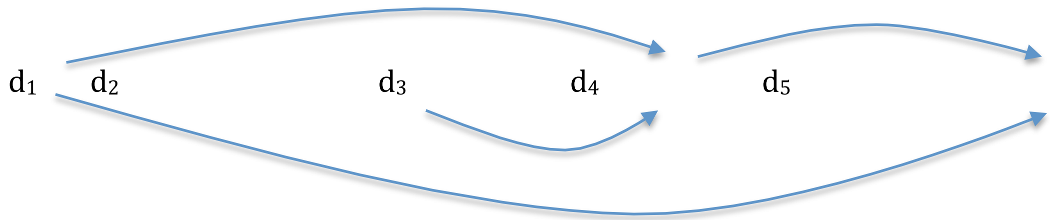
COSINE SCORES, PAGERANK AND CLASSIFICATION

(a) Vectors and inner product score:

	$w_{t,q}$	$w_{t,d1}$	$w_{t,d2}$	$w_{t,d3}$	$w_{t,d4}$	$w_{t,d5}$

speech	¼	2	0	1	1	2
dialogue	¼	2	1	1	1	0
system	¼	2	0	2	1	1
Score:		1.5	0.25	1	0.75	0.75

(b) Links between documents:



Transition matrix:

P =

$$
\begin{matrix}
0.1 & 0.1 & 0.35 & 0.1 & 0.35 \\
0.1 & 0.1 & 0.6 & 0.1 & 0.1 \\
0.1 & 0.1 & 0.1 & 0.1 & 0.6 \\
0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 & 0.2
\end{matrix}
$$

Assuming we start at document 1, the first PageRank estimate is:

$$[1\ 0\ 0\ 0\ 0]\ P = [0.1\ 0.1\ 0.35\ 0.1\ 0.35]$$

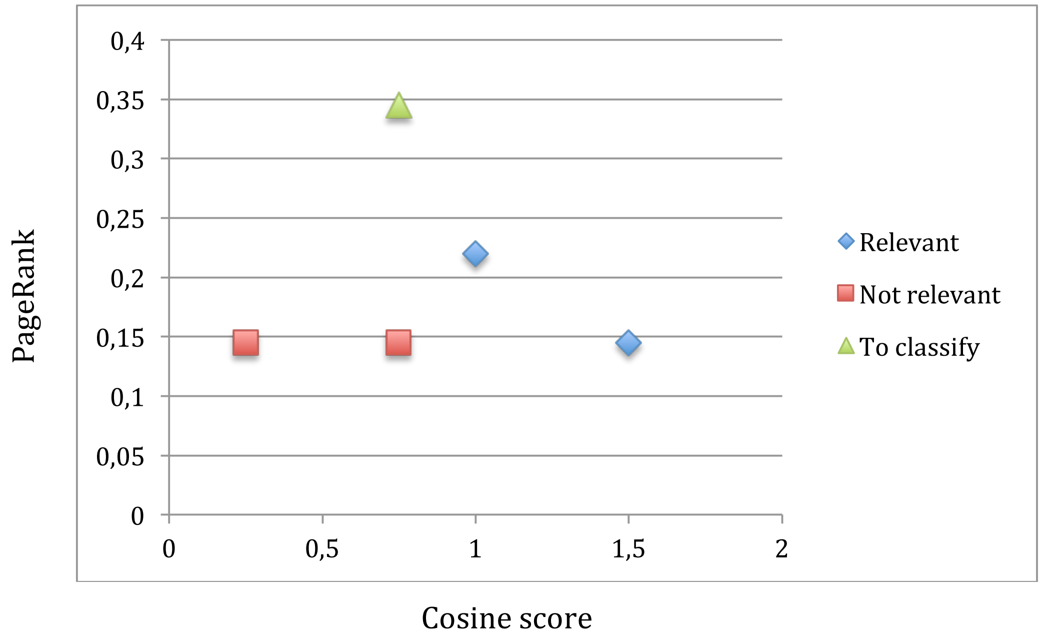
And a second iteration yields:

$$[0.1\ 0.1\ 0.35\ 0.1\ 0.35]\ P = [0.145\ 0.145\ 0.22\ 0.145\ 0.345]$$

(c) The five vectors are:

d1	1.5	0.145
d3	1	0.22
d2	0.25	0.145

d4	0.75	0.145
d5	0.75	0.345



(d) (i) Rocchio classifier:

class	centroid	distance to d5
relevant	[1.25, 0.1825]	0.6625
not-relevant	[0.5, 0.145]	0.45

→ document d_5 classified as not relevant

(d) (ii) 3-NN classifier:

The three closest documents are d_2, d_3 and d_4, which means that d_5 is classified as non-relevant.

(d) (iii) Linear SVM classifier:

classification function: y = sign (w . x + b)
where $w = \sum \alpha_i y_i x_i$ -→ in this case $[20\ 4.4]^T$ - $[15\ 2.9]^T$ = $[5\ 1.5]^T$

for d_5, we have y = sign (5*0.75 + 1.5*0.145 – 4.65) = -1 → not relevant

(e) Examples of features that can be useful in relevance judgments are, e.g., query-term proximity, document age, document length, etc.