Ans2 (a)

Sahil Jindal
18CS10048

Ans2 (a) Advantage: Lookup is faster

Disadvantage: Cannot handle wild card queries easily.

(b) There can only be one original vocabulary term from which it was derived because of the $ terminal character.

For eg. book & okbo would have given same permuterm vocabulary if we don't use $.

(c) $red

(d) For trigram, we would search for terms:

$re

red

& take intersection.

The term "recurred" would match the boolean query but not permuterm query.

Date...../...../.....

Page.....

(e) Jaccard coeff (A, B) $= \frac{|A \cap B|}{|A \cup B|}$.

For a bigram index :-

bar :- ba, ar

barbaric :- ba, ar, rb, ba, ar, ri, ic

Removing repetition,

(bar) ∩ (barbaric) ⇒ ba, ar

(bar ∪ barbaric) ⇒ ba, ar, rb, ri, ic

∴ Jaccard coeff $= \frac{2}{5} = 0.4$.

A3

(a) Reading time for collection of data from disk:

= (no. of tokens) * (bytes per token) * (time for transfer each byte)

+ disk seek time.

~~where disk seek time~~

$= (4 \times 10^7)(8)(2 \times 10^{-8}) + 5 \times 10^{-3}$

$= 6.4 + 0.005 = 6.4058$

Date...../...../.....

Page.....

(b) Total searching time = (no of blocks) * (N.logN) * (time for low level operation)

= (20)(N log N)(10^{-8})

= $2 \times 10^{-7} \times N \log N$

where N = no of documents (not given in ques)

(c) Total time for writing sorted blocks to disk

= (No of blocks) * (time to write single block)

where

time to write single block = (No of term per block + No of posting in each block) * (size of termid) * (time to transfer 1 byte data).

$= \left(20\right)\left(\frac{250000}{20} + \frac{4 \times 10^7}{20}\right)(4)(2 \times 10^{-8})$

= (25 + 4000)(4)(2).

= 32200 seconds.

Date...../...../.....

Page.....

(d) Assuming buffer for each block to contain $\frac{1}{10^m}$ of the block content.

then, required time

= disk seek time + time for reading part of sorted posix lists from disk + time for writing final index.

Now, disk seek time = (No of blocks)* (time to read full block given we have $\frac{1}{10^m}$ of block)* (avg seek time)* 2

&

time for reading part of sorted posix list = time for writing final index = (no of terms + total posting) * (size of term id) * (time to transfer single byte).

$$\therefore \text{repl time} = 20 * \left(\frac{4 \times 10^7 \times 8 \times 2 \times 10^{-8} \times 10}{20} \right) * (5 \times 10^{-3}) * 2 + (250000 +$$

Ans 4

Judge 1

Judge 2	R	N
R	2	5
N	5	3

(a) $P_r(a) = P(\text{Agreement}) = \frac{2+3}{15} = \frac{1}{3}$.

$P(NR) = \frac{5+5+3+3}{30} = \frac{16}{30} = \frac{8}{15}$.

$P(R) = \frac{5+5+2+2}{30} = \frac{14}{30} = \frac{7}{15}$

$P_r(e) = P(\text{Agree (hasard)}) = P(NR)^2 + P(R)^2$

$= \frac{113}{225}$.

$K = \frac{P_r(a) - P_r(e)}{1 - P_r(e)} = \frac{0.33 - 0.502}{1 - 0.502}$

$K = -0.3453$

Date...../...../.....

Page.....

(b) Retrieved = 4, 5, 6, 7, 3, 8

Relevant IDs: 3, 4

Precision = $\frac{\text{Relevant}}{\text{Retrieved}}$

$= \frac{2}{6} = \frac{1}{3}$.

Recall = $\frac{\text{Retrieved}}{\text{Relevant}}$

$= \frac{2}{2} = 1$.

(c) Relevant IDs: 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14.

Precision = $\frac{6}{6} = 1$

Recall = $\frac{6}{12} = \frac{1}{2}$

(d) (iv)

Bonus : We can show results for different queries to users & crowd source their opinions about the relevance of the result.

We will show them various results & use the K-rank of that product in the result by our IR system.

In otherwords, we will get the relevance score of a product & then use it to find score of our IR system