7-Feb-2024

Information Retrieval and Web Search

Assignment - 2

Uday Bhaskar Valapadasu
11696364

1Ans :-

a) Brutas AND NOT Caesar

Here, Rather than gathering documents present in both postings lists, the approach is to gather those appearing in the first list but not in the second.

The time complexity for this query would be same $O(x+y)$.

b) Brutas OR NOT Caesar

Assuming the requirement is to provide a comprehensive list of all documents meeting the query criteria, the time complexity is $O(N)$, where N represents the total number of documents in the collection. This is because the length of the result list can potentially reach N, unrestricted by the length of the postings lists.

2Ans:-

(i) (tangerine OR trees) AND (marmalade OR skies) AND (Kaleidoscope OR eyes)

Estimate the size of each OR operation by adding the frequencies of its disjuncts:

1> (tangerine OR trees) = 316812 + 46653 = 363465

2> (marmalade OR skies) = 107913 + 271658 = 379571

3> (Kaleidoscope OR eyes) = 87009 + 213312 = 300321

To process the query in increasing order of the size of each disjunctive term, ~~is~~ :

The recommend query processing order is:

(Kaleidoscope OR eyes) AND (tangerine OR trees) AND (marmalade OR skies)

③ → ① → ②

(ii) tangerine AND (NOT marmalade) AND (NOT Trees)

Estimate the size of each term taking into account the NOT operation

1> tangerine = 46653

2> NOT marmalade = Total no of documents − frequency of "marmalade" = 500000 − 107913

= 392087

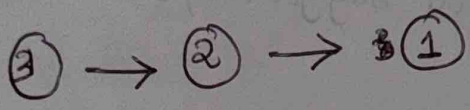
3.> NOT trees = Total no of documents - Frequency of "trees"

= 500000 - 316812

= 183188

To Process the query in increasing order of the size of each term. So, the recommended query processing order is:

(NOT trees) AND (NOT marmalade) AND (tangerine)



3Ans:-

Lets consider a example to understand:

Suppose we have three terms with the following postings list sizes:

- Term 1: Size = 100
- Term 2: Size = [redacted] 105
- Term 3: Size = [redacted] 110

Now, let's say the intersections b/w these postings lists as follows:

1.) intersection of $S1$ and $S2$: 100 steps

2.) intersection of result with $S3$: 100 steps (assuming no intersection with $S3$)

Total steps = 100 + 105 + 100 + 110 = 315 steps

However, if we process the terms in a different order ($S1$, $S3$, $S2$) as you suggested, we would perform the following steps:

1. Intersection of $S1$ and $S3$: 0 steps (assuming no intersection with $S3$)

2. No further intersection needed with $S2$ (since intersection with $S3$ is empty)

Total steps : 100 + 110 + 0 + 0 = 210 steps.

So, processing postings lists in order of size is not always optimal, as demonstrated by this example.

4 Ans :-

postings merge algorithm for x OR Y query.

UNION (x, y)

1.) answer ← ()
2.) while x != NIL and y != NIL
3.) do if docID(x) = docID(y)
4.) ~~then ADD(answer, docID(x))~~
↳ then ADD(answer, docID(x))
5.) x ← next(x)
6.) y ← next(y)
7.) else if docID(x) < docID(y)
8.) x ← next(x)
9.) else ADD(answer, docID(y))
10.) y ← next(y)
11.) return(answer)

- - - - - X - - - - - - -