IR Tutorial -- September 8, 2021

Skip List Merging

Question

Consider the following three postings list:

a) “Sachin”: 2, 4, 8, 16, 19, 23, 28, 43, 64, 78
b) “Ramesh”: 1, 2, 3, 5, 8, 12, 15, 18, 19, 27, 32, 39, 44, 56, 61, 70, 78
c) “Tendulkar”: 8, 19, 24, 39, 78

All lists contain skip pointers, skip length being Floor(√P), where P represents the length of the corresponding lists. Skip pointers start from the first element.

Write down the sequence of postings comparisons for obtaining the list of documents containing all three words. What is the optimal no. of comparisons?

Solution:

First, let’s look at the skip pointers in all three lists:

a) “Sachin”: 2 -> 16, 16 -> 28, 28 -> 78 (Length of list = 10, Skip length = Floor(√10) = 3)
b) “Ramesh”: 1 -> 8, 8 -> 19, 19 -> 44, 44 -> 78 (Length of list = 17, Skip length = Floor(√17) = 4)
c) “Tendulkar”: 8 -> 24, 24 -> 78 (Length of list = 5, Skip length = Floor(√5) = 2)

Now, we need to find documents containing all the three words. Therefore, our query becomes:
“Sachin” AND “Ramesh” AND “Tendulkar”

For optimal comparisons, we need to start with the shortest two postings lists, i.e.
“Sachin”: 2, 4, 8, 16, 19, 23, 28, 43, 64, 78
“Tendulkar”: 8, 19, 24, 39, 78

The sequence of comparisons will be:

1) 2,8
2) 16,8
3) 4,8
4) **8,8**
5) 16,19
6) 28,19
7) **19,19**
8) 23,24
9) 28,24
10) 28,78
11) 28, 39
12) 78, 39
13) 43, 39
14) 43, 78
15) 64, 78
16) **78, 78**

Therefore, there are 3 matches, i.e. 8, 19, 78. The intermediate postings list thus becomes:
“Sachin” AND “Tendulkar”: [8, 19, 78]

Do you note any difference?

This intermediate postings list does not have any skip pointers. Now, we need to merge (intersect) this postings list with that of “Ramesh”

“Ramesh”: 1, 2, 3, 5, 8, 12, 15, 18, 19, 27, 32, 39, 44, 56, 61, 70, 78 (With skip pointers)
“Sachin” AND “Tendulkar”: 8, 19, 78 (No Skip Pointers)

The sequence of comparisons will be:

17) 1,8
18) **8,8**
19) 12,19
20) 15,19
21) 18,19
22) **19,19**
23) 27,78
24) 32,78
25) 39,78
26) 44,78
27) **78,78**

Therefore, the documents with IDs 8, 19, and 78 contain all three words.

Optimal no. of comparisons: 27.

Vector Space Model

Question

Consider a small collection C that consists in the following three documents:
d1: “new york times”
d2: “new york post”
d3: “los angeles times”

Given the query: “new new times” , rank the documents based on the vector space model using **ntc.ntc** scheme.

Solution :

-- For a document collection, we first determine a set of unique terms (vocabulary). Let the no. of unique terms be n
-- The documents are represented as n dimensional vectors, where each dimension corresponds to a term.
-- The terms are weighted using its $tf - idf$ value.
-- The queries are also represented similarly, as n dimensional vectors

N = no. of documents in the collection = 3

term	df	idf [$log_{10}(N/df)$]
new	2	$log_{10}(3/2) = 0.176$
york	2	$log_{10}(3/2) = 0.176$
times	2	$log_{10}(3/2) = 0.176$
post	1	$log_{10}(3/1) = 0.477$
los	1	$log_{10}(3/1) = 0.477$
angeles	1	$log_{10}(3/1) = 0.477$

Each term is weighted by its tf-idf value, where

tf = frequency of the term in the doc or query ; idf = constant across all docs (calculated above)

	new	**york**	**times**	**post**	**los**	**angeles**
d1	0.176	0.176	0.176	0	0	0
d2	0.176	0.176	0	0.477	0	0
d3	0	0	0.176	0	0.477	0.477
q	0.352	0	0.176	0	0	0

cosine-similarity (d1,q) = $\frac{(0.176 * 0.352) + (0.176 * 0.176)}{\sqrt{3*(0.176^2)} * \sqrt{0.352^2 + 0.176^2}}$ = 0.775

cosine-similarity (d2,q) = $\frac{(0.176 * 0.352)}{\sqrt{2*(0.176^2)+0.477^2} * \sqrt{0.352^2 + 0.176^2}}$ = 0.293

cosine-similarity (d3,q) = $\frac{(0.176 * 0.176)}{\sqrt{0.176^2+2*(0.477^2)} * \sqrt{0.352^2 + 0.176^2}}$ = 0.113

Ranking
d1
d2
d3