B58A0050-NLP-Autumn-Homework2 – Boolean Retrieval Model

58119125 JiangZhuouyang

November 2021

1 Boolean Query's Processing Order of Merge

Term	Freq
eyes	213312
kaleidoscope	87009
marmalade	107913
skies	271658
tangerine	46653
trees	316812

1) Recommend a query processing order for:

(***tangerine*** OR ***trees***) AND (***marmalade*** OR ***skies***) AND (***kaleidoscope*** OR ***eyes***)

2) Which two terms should we process first?

Solution **:**

1) The query processing order need be:

$$\boldsymbol{tangerine} \longrightarrow \boldsymbol{kaleidoscope} \longrightarrow \boldsymbol{marmalade} \longrightarrow \boldsymbol{eyes} \longrightarrow \boldsymbol{skies} \longrightarrow \boldsymbol{trees}$$

2) we can calculate that:

$$Freq(\boldsymbol{tangerine} OR \boldsymbol{trees}) = 46653 + 316812 = 363465 \tag{1}$$

$$Freq(\boldsymbol{marmalade} OR \boldsymbol{skies}) = 107913 + 271658 = 379571 \tag{2}$$

$$Freq(\boldsymbol{kaleidoscope} OR \boldsymbol{eyes}) = 87009 + 213312 = 300321 \tag{3}$$

Because:

Freq(***kaleidoscope*** OR ***eyes***) < Freq(***tangerine*** OR ***trees***) < Freq(***marmalade*** OR ***skies***)

So that, we should process (***kaleidoscope*** OR ***eyes***) first.

2 Usage of 'NOT' Item's Frequency

If the query is friends AND romans AND (NOT countrymen), how could we use the freq of countrymen?

Solution **:**

We still use its document frequency to determine the query process order. However, in my opinion, I would like to divide the option into three steps.

(1) We need first do merge work on the 'YES' items(***friends romans*** here) with increasing order. If all 'YES' items' merge operation traverses have been done, we get a 'YES' merged-result-item, I call it $\mathbf{1 - Result}$.

(2) Then we traverse the 'NOT' items to do another kind of merge work. We simply put the docID appearing in all 'NOT' items after a new item without repetition and count its document frequency.

In this question, because we just have one 'NOT' item, so we regard the document frequency of ***countrymen*** as the 'NOT' merged-result-item's frequency.

(3) With the frequency of two kinds of merged-result-items, we still use increasing order as the processing order.

And when our merge operation traverses them according to the processing order, our work is changed to 'exclude'.

It means that we first copy the 'YES' merged-result-item stored in a list structure. And we need exclude the document from the list structure which is appearing both in the postings of two merged-result-items. And the searching traverses option among the two merged-result-items is in increasing order of doc-freq..

3 General Time Complexity of Boolean Query

Extend the merge to an arbitrary Boolean query. Can we always guarantee execution in time linear in the total postings size?

Solution **:**

No we can't.

We has four meta query structure.

(1) If there is a (***A*** AND ***B***) in query, the time complexity is O(Freq(A)+Freq(B)) to do ***merge*** work.

(2) If there is a (***A*** OR ***B***) in query, the time complexity is O(Freq(A)+Freq(B)) to do ***insert*** work.

(3) If there is a (***A*** AND NOT ***B***) in query, the time complexity is O(Freq(A)+Freq(B)) to do ***exclude*** work.

(4) If there is a (***A*** OR NOT ***B***) however, we can not easily search the result of (OR NOT ***B***) just with the doc postings of ***A*** and ***B***. So that we need to put all of other document which has nothing to do with these two items.