

IR Assignment 1 (Analysis)

Akshaj Patil(MT19111)

1:Query [1]

```
Query = ball and hit and not & trustworthy
preprocessed Query = ['ball', 'and', 'hit', 'and', 'not', 'trustworthy']
TemplListQuery= ['trustworthy'] and ListQuery = ['ball', 'and', 'hit', 'and', 'trustworthy']
Len of not trustworthy = 19982
minLenString= ball
List Query = ['ball', 'and', 'trustworthy']
minLenString= ball
List Query = ['ball']
Total number of comparisons = 35670
Total number of documents = 80
List of documents retrieved are [2908, 2937, 9002, 9007, 9011, 9037, 9052, 9068, 9073, 9088, 9091, 9105, 9131, 9182, 9189, 9201, 9220, 9272, 9346, 9351, 9361, 9368, 9406, 9418, 9423, 9454, 9455, 9459, 9480, 9504, 9520, 9552, 9587, 9588, 9590, 9594, 9602, 9638, 9669, 9674, 9677, 9688, 9689, 9690, 9696, 9723, 9747, 9748, 9751, 9767, 9774, 9781, 9784, 9790, 9799, 9803, 9809, 9844, 9853, 9856, 9901, 9905, 9912, 9914, 9928, 9942, 9948, 9954, 9961, 9962, 9967, 9977, 9987, 10511, 10798, 10884, 10908, 13087, 13379, 14785]
```

Here Query is “ball and hit and not & trustworthy”

After preprocessed punctuation “&” is removed. First not trustworthy is calculated and result is stored in trustworthy. Now here we can see that min posting length is of “ball” so first ball and hit will be calculated and result is stored in “ball” again. Then previous result and trustworthy is performed and final result is stored in ball again. Since here we are using optimized approach so numbers of comparisons are less. It’s 35670 if we do not follow this approach then number of comparisons will increase.

2:Query [2]

```
Query = ball or hit and trustworthy
preprocessed Query = ['ball', 'or', 'hit', 'and', 'trustworthy']
TemplListQuery= [] and ListQuery = ['ball', 'or', 'hit', 'and', 'trustworthy']
minLenString= trustworthy
List Query = ['ball', 'or', 'hit']
minLenString= hit
List Query = ['ball']
Total number of comparisons = 945
Total number of documents = 335
List of documents retrieved are [1, 452, 474, 901, 972, 1202, 1314, 1442, 1549, 1809, 1847, 1993, 2004, 2088, 2089, 2136, 2141, 2142, 2625, 2655, 2908, 2914, 2937, 2972, 3391, 3846, 3913, 4110, 4117, 4130, 4168, 4205, 5039, 5053, 5076, 5781, 6237, 6535, 7170, 7323, 7564, 7726, 8036, 8153, 8265, 8282, 8364, 8389, 8527, 8622, 8634, 8653, 8684, 8698, 8769, 8829, 8857, 8869, 8897, 8915, 8927, 8975, 9002, 9005, 9007, 9009, 9011, 9037, 9041, 9052, 9055, 9065, 9068, 9070, 9072, 9073, 9076, 9088, 9091, 9105, 9131, 9144, 9147, 9159, 9165, 9172, 9181, 9182, 9185, 9189, 9201, 9220, 9233, 9243, 9244, 9245, 9272, 9282, 9290, 9338, 9342, 9345, 9346, 9351, 9361, 9368, 9384, 9406, 9418, 9423, 9454, 9455, 9459, 9480, 9490, 9492, 9497, 9499, 9500, 9501, 9503, 9504, 9508, 9518, 9520, 9552, 9554, 9562, 9583, 9587, 9588, 9589, 9590, 9591, 9592, 9593, 9594, 9597, 9602, 9609, 9615, 9619, 9638, 9658, 9659, 9667, 9669, 9674, 9676, 9677, 9678, 9679, 9680, 9688, 9689, 9690, 9692, 9696, 9698, 9712, 9713, 9716, 9719, 9723, 9727, 9730, 9732, 9745, 9747, 9748, 9751, 9767, 9772, 9774, 9778, 9781, 9782, 9784, 9790, 9791, 9799, 9803, 9809, 9810, 9812, 9822, 9826, 9835, 9841, 9844, 9853, 9856, 9857, 9869, 9870, 9876, 9887, 9893, 9900, 9901, 9905, 9912, 9913, 9914, 9928, 9942, 9944, 9948, 9954, 9961, 9962, 9964, 9967, 9976, 9977, 9982, 9987, 9993, 10079, 10087, 10276, 10438, 10511, 10596, 10624, 10642, 10798, 10800, 10884, 10908, 10974, 11002, 11034, 11285, 11520, 12056, 12064, 12073, 12075, 12076, 12082, 12097, 12107, 12112, 12117, 12552, 13087, 13259, 13316, 13379, 13516, 13570, 13603, 13645, 14010, 14014, 14061, 14062, 14064, 14065, 14075, 14093, 14099, 14112, 14350, 14399, 14426, 14433, 14440, 14441, 14486, 14487, 14537, 14580, 14606, 14607, 14623, 14732, 14740, 14785, 14800, 14824, 14825, 14826, 14827, 14828, 14837, 14878, 14984, 15184, 15367, 15444, 15920, 16024, 16117, 16129, 16191, 16195, 16215, 16345, 16449, 16457, 16459, 16513, 16577, 16607, 16619, 16620, 16681, 16785, 16795, 16843, 16849, 17644, 17649, 17738, 17931, 17938, 18096, 18216, 18426, 18591, 18599, 18662, 18671, 18750, 18753, 18767, 18924, 18969, 19319, 19449, 19451, 19455, 19753]
```

Here Query is “ball or hit and trustworthy”

As length of posting list of “ball” is minimum but it has “or” operator with it so new minimum is “trustworthy”. Now “and” operation between “hit” and “trustworthy” will perform first result will store in “hit”. And then “or” operation with “ball” is performed. Here in above approach if “or” operation is performed first then Query will give wrong result.

3:Query[3]

```
Query = ball and hit and and not trustworthy
preprocessed Query = ['ball', 'and', 'hit', 'and', 'and', 'not', 'trustworthy']
TemplstQuery= ['trustworthy'] and ListQuery = ['ball', 'and', 'hit', 'and', 'trustworthy']
Len of not trustworthy = 19982
minLenString= ball
List Query = ['ball', 'and', 'trustworthy']
minLenString= ball
List Query = ['ball']
Total number of comparisons = 35670
Total number of documents = 80
List of documents retrived are [2908, 2937, 9002, 9007, 9011, 9037, 9052, 9068, 9073, 9088, 9091, 9105, 9131, 9182, 9189, 920
1, 9220, 9272, 9346, 9351, 9361, 9368, 9406, 9418, 9423, 9454, 9455, 9459, 9480, 9504, 9520, 9552, 9587, 9588, 9590, 9594, 960
2, 9638, 9669, 9674, 9677, 9688, 9689, 9690, 9696, 9723, 9747, 9748, 9751, 9767, 9774, 9781, 9784, 9790, 9799, 9803, 9809, 984
4, 9853, 9856, 9901, 9905, 9912, 9914, 9928, 9942, 9948, 9954, 9961, 9962, 9967, 9977, 9987, 10511, 10798, 10884, 10908, 13087,
13379, 14785]
```

Here above query is “ball and hit and and not trustworthy”

As given input is in wrong format due to human error since there are two consecutive “and” so this query is transformed in valid query “ball and hit and not trustworthy” and then operations will be performed accordingly.

4:Query[4]

```
Query = ball and not ball or ball
preprocessed Query = ['ball', 'and', 'not', 'ball', 'or', 'ball']
TemplstQuery= ['2ball'] and ListQuery = ['ball', 'and', '2ball', 'or', '3ball']
Len of not 2ball = 19664
minLenString= ball
List Query = ['ball', 'or', '3ball']
minLenString= ball
List Query = ['ball']
Total number of comparisons = 39750
Total number of documents = 333
List of documents retrived are [1, 452, 474, 901, 972, 1202, 1314, 1442, 1549, 1809, 1847, 1993, 2004, 2088, 2089, 2136, 2141,
2142, 2625, 2655, 2908, 2914, 2937, 2972, 3391, 3846, 3913, 4110, 4117, 4130, 4168, 4205, 5039, 5053, 5076, 5781, 6237, 6535, 7
170, 7323, 7564, 7726, 8036, 8153, 8265, 8282, 8364, 8389, 8527, 8622, 8634, 8653, 8684, 8698, 8769, 8829, 8857, 8869, 8897, 89
15, 8927, 8975, 9002, 9005, 9007, 9009, 9011, 9037, 9041, 9052, 9055, 9065, 9068, 9070, 9072, 9073, 9076, 9088, 9091, 9105, 913
1, 9144, 9147, 9159, 9165, 9172, 9181, 9182, 9185, 9189, 9201, 9220, 9233, 9243, 9244, 9245, 9272, 9282, 9290, 9338, 9342, 934
5, 9346, 9351, 9361, 9368, 9384, 9406, 9418, 9423, 9454, 9455, 9459, 9480, 9490, 9492, 9497, 9499, 9500, 9501, 9503, 9504, 950
8, 9518, 9520, 9552, 9554, 9562, 9583, 9587, 9588, 9589, 9590, 9591, 9592, 9593, 9594, 9597, 9602, 9609, 9615, 9619, 9638, 965
8, 9659, 9667, 9669, 9674, 9676, 9677, 9678, 9679, 9680, 9688, 9689, 9690, 9692, 9696, 9698, 9712, 9713, 9716, 9719, 9723, 972
7, 9730, 9732, 9745, 9747, 9748, 9751, 9767, 9772, 9774, 9778, 9781, 9782, 9784, 9790, 9791, 9799, 9803, 9809, 9810, 9812, 982
2, 9826, 9835, 9841, 9844, 9853, 9856, 9857, 9869, 9870, 9876, 9887, 9893, 9900, 9901, 9905, 9912, 9913, 9914, 9928, 9942, 994
4, 9948, 9954, 9961, 9962, 9964, 9967, 9976, 9977, 9982, 9987, 9993, 10079, 10087, 10276, 10438, 10511, 10596, 10624, 10642, 10
798, 10800, 10884, 10908, 10974, 11002, 11285, 11520, 12056, 12064, 12073, 12075, 12076, 12082, 12097, 12107, 12112, 12117, 125
52, 13087, 13259, 13316, 13379, 13516, 13570, 13603, 13645, 14010, 14014, 14061, 14062, 14064, 14065, 14075, 14093, 14094, 1411
2, 14350, 14399, 14426, 14433, 14440, 14441, 14486, 14487, 14537, 14580, 14606, 14607, 14623, 14732, 14740, 14785, 14805, 1482
4, 14825, 14826, 14827, 14828, 14837, 14878, 14984, 15367, 15444, 15920, 16024, 16117, 16129, 16191, 16195, 16215, 16345, 1644
9, 16457, 16459, 16513, 16577, 16607, 16619, 16620, 16681, 16785, 16795, 16843, 16849, 17644, 17649, 17738, 17931, 17938, 1809
6, 18216, 18426, 18591, 18599, 18662, 18671, 18750, 18753, 18767, 18924, 18969, 19319, 19449, 19451, 19455, 19753]
```

Here in this query “not ball” is calculated first then “ball and not ball” and then previous result “or bal” so basically this query only shows number of documents in which “ball” is present and they are shown above in query image.

5:Query [5](Question 2):

```
['please', 'mail', 'me']  
['please', 'mail', 'me']  
Number of Documents = 4  
Documents retrived are dict_keys([488, 605, 874, 928])
```

Here query is Phrase Query “please mail me”

Combination of documents are performed left to right. So first “please”, “mail” is combined and after that previous result and “me” is combined. In this way we will get final result of phrase query. In the above case number of documents retrieved are 4.