COMP6714 (16S2) ASSIGNMENT 2 SAMPLE SOLUTION

Q1. (*25 marks*)

(1) The BM25 formula essentially limits the impact of tfs (the value converges when $tf \to \infty$). In our case, the scoring function is

$$score(d) \leq 6f(tf_1) + 2f(tf_2) + f(tf_3)$$

where $f(x) = \frac{3x}{2+x}$. Since $\lim_{x\to\infty} f(x) = 3$, we can find the maxscores for the terms are 18, 6, and 3.

(2) We first consider D_1, with score

$$score(D_1) = 6f(1) + 2f(1) + f(1) = 9$$

Then we consider D_2

$$score(D_2) = 6f(8) + 2f(0) + f(2) = 15.90$$

At this stage, both of them become the current top-2 results, and $\tau' = 9$. Since $3 + 6 \leq \tau'$, we only need to consider A. (hence no need to score D_4)

Driven by A, the next document to score is D_5. We need to probe the lists of B and C for D_5, and compute its score as

$$score(D_5) = 6f(3) + 2f(4) + f(2) = 16.30$$

Similarly, since now $\tau' = 15.90$.

The next document to consider is D_8

$$score(D_8) = 6f(10) + 2f(0) + f(1) = 16.00$$

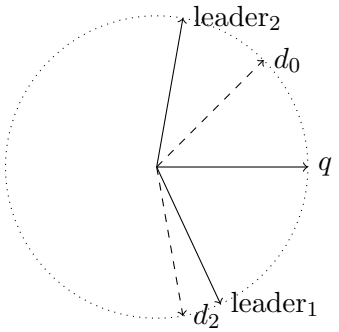
Since A's postings list is now exhausted, we conclude that the final top-2 documents are D_5 and D_8. The algorithm scored 4 documents, and accessed 10 postings.

Q2. (*25 marks*)

The *cluster pruning* method is introduced in Chap 7.1.6 of [MRS08].

(1) Let the number of leaders be x. Each leader has $\frac{b_1 N}{x} - 1$ followers on average. During the query processing, we use linear scan to find the b_2 closest leader and then find at most $b_2(\frac{b_1 N}{x})$ candidates. (We ignore the cost of finding the top-k results from these candidates) The total query processing cost (in terms of distance calculation) is $f(x) = x + b_2(\frac{b_1 N}{x})$. $\frac{\mathrm{d}}{\mathrm{d}\,x} f(x) = 1 - \frac{b_2 b_1 N}{x}$. Hence when $x = \sqrt{b_2 b_1 N}$, the overall query processing cost is minimized. In the basic model, $b_2 = b_1 = 1$, hence $x = \sqrt{N}$.

(2) See the following example where all document vectors are normalized to a unit vector. It is also correct if we remove d_2.



The query q is closer to $leader_1$ than $leader_2$. But the correct answer is d_0 which is a follower of $leader_2$. Even with $b_1, b_2 > 1$, we can find counter-examples (omitted).

(3) Necessary modifications:
- For each cluster c_i, calculate the maximum angle between any of the followers and the leader (denoted as θ_i).
- Assume $k = 1$. In the query processing, we first calculate the angles between all the leaders and the query. We iterate through the clusters identified by the leaders in increasing order of the angle. When visiting a cluster, we explore all its members by calculating the cosine distance to the query (essentially the angle). The stopping criteria is that the current best result has a smaller angle than α_{next}, where $\alpha_{\text{next}} = \text{angle}(c_i, q) - \theta_i$ is a lower bound of the angles between a document in c_i and the query q. This method can be easily extended to deal with top-k queries.

(The performance of the method might be heavily affected by how well documents form clusters)

Note this is just one of the correct modification methods.

Q3. (*25 marks*)

k	1	2	3	4	5	6	7	8	9	10
precision (%)	100.00	100.00	66.67	50.00	40.00	33.33	28.57	25.00	33.33	30.00
recall (%)	12.50	25.00	25.00	25.00	25.00	25.00	25.00	25.00	37.50	37.50
k	11	12	13	14	15	16	17	18	19	20
precision (%)	36.36	33.33	30.77	28.57	33.33	31.25	29.41	27.78	26.32	30.00
recall (%)	50.00	50.00	50.00	50.00	62.50	62.50	62.50	62.50	62.50	75.00

(1) precision@20 is $\frac{6}{20}$.

(2) recall@20 is $\frac{6}{8}$. $F_1 = \frac{2 \cdot \frac{3}{10} \cdot \frac{3}{4}}{\left(\frac{3}{10} + \frac{3}{4}\right)} = 0.4286$

(3) 25% recall corresponds to uninterpolated precisions of 100%, 66.67%, 50.00%, 40.00%, 33.33%, 28.57%, 25.00%.
(4) the interpolated precision for 33% recall is the maximum precision achieved for $k \geq 9$. Obviously, the maximum value is $\frac{4}{11} = 0.3636$.
(5) MAP is $\frac{1}{8} \cdot (\frac{1}{1} + \frac{2}{2} + \frac{3}{9} + \frac{4}{11} + \frac{5}{15} + \frac{6}{20}) = 0.4163$.
(6) The largest possible MAP is $\frac{1}{8} \cdot (\frac{1}{1} + \frac{2}{2} + \frac{3}{9} + \frac{4}{11} + \frac{5}{15} + \frac{6}{20} + \frac{7}{21} + \frac{8}{22}) = 0.5034$.
(7) The smallest possible MAP is $\frac{1}{8} \cdot (\frac{1}{1} + \frac{2}{2} + \frac{3}{9} + \frac{4}{11} + \frac{5}{15} + \frac{6}{20} + \frac{7}{9999} + \frac{8}{10000}) = 0.4165$.
(8) $0.5034 - 0.4163 = 0.0871$

Q4. (*25 marks*)

(1) The probability distributions for each document model and the background model are:

Model		a	all	am	be	can't	don't	go
background		1/38	2/38	1/38	1/38	1/38	2/38	1/38
doc1	raw	1/22	0	0	1/22	1/22	1/22	1/22
	smoothed	30/836	22/836	11/836	30/836	30/836	41/836	30/836
doc2	raw	0	2/16	1/16	0	0	1/16	0
	smoothed	8/608	54/608	27/608	8/608	8/608	35/608	8/608

Model		groovy	here	hurry	i	is	king	love
background		1/38	1/38	1/38	3/38	1/38	1/38	6/38
doc1	raw	1/22	0	1/22	1/22	0	1/22	3/22
	smoothed	30/836	11/836	30/836	52/836	11/836	30/836	123/836
doc2	raw	0	1/16	0	2/16	1/16	0	3/16
	smoothed	8/608	27/608	8/608	62/608	27/608	8/608	105/608

Model		me	must	of	out	remember	take	tell
background		2/38	1/38	2/38	1/38	1/38	1/38	1/38
doc1	raw	1/22	1/22	1/22	0	0	1/22	0
	smoothed	41/836	30/836	41/836	11/836	11/836	30/836	11/836
doc2	raw	1/16	0	1/16	1/16	1/16	0	1/16
	smoothed	35/608	8/608	35/608	27/608	27/608	8/608	27/608

Model		this	to	want	with	you		
background		1/38	1/38	1/38	1/38	2/38		
doc1	raw	1/22	1/22	1/22	1/22	2/22		
	smoothed	30/836	30/836	30/836	30/836	60/836		
doc2	raw	0	0	0	0	0		
	smoothed	8/608	8/608	8/608	8/608	16/608		

(2)

$P(Q_1|D_1) = 52/836 * 11/836 * 60/836 = 0.0000587$

$P(Q_1|D_2) = 62/608 * 27/608 * 16/608 = 0.000119$

$P(Q_2|D_1) = 41/836 * 30/836 * 60/836 * 30/836 * 123/836 * 41/836 = 0.0000000327$

$P(Q_2|D_2) = 35/608 * 8/608 * 16/608 * 8/608 * 105/608 * 35/608 = 0.00000000261$

Thus, D_2 will be ranked first for Q_1 and D_1 will be ranked first for Q_2.

(3)

$$P(Q_1|D_1) * P(D_1) = 0.0000587 * 0.7 = 0.0000411$$

$$P(Q_1|D_2) * P(D_2) = 0.000119 * 0.3 = 0.0000358$$

$$P(Q_2|D_1) * P(D_1) = 0.0000000327 * 0.7 = 0.0000000229$$

$$P(Q_2|D_2) * P(D_2) = 0.00000000261 * 0.3 = 0.000000000782$$

Thus, D_1 will be ranked first for both queries by taking into consideration the prior.