Endsem

-Avneet Kaur(2014027)

Qm = a*Qo + b*(1/|Dr|)*(Σ Dj) - c(1/|Dnr|)*(Σ Dj)

Qm - modified query, a- original query weight, b - relevant docs weight, c- non relevant docs weight, Dr- set of relevant docs, Dnr- set of non relevant docs

Q1) The idea is to move away from the centroid of non relevant documents. So in this case we can assign a higher value of c. However, assigning b - a lower value might result in the documents moving away from the relevant documents, since we know that the getting positive feedback from a query search results matters more than the negative feedback.

Q2) The idea in this to move the modified query towards the average relevant documents vector (centroid of relevant documents) and away from the average non relevant vector(centroid of non relevant documents), at the same time keeping all the terms of the original query in the modified query. Therefore, we assign a=1. We assign a higher value to beta since we want the modified query to be closer to the relevant documents. Moreover we have learnt from the textbook that positive feedback matters much more than negative feedback. That is, we know that if we have a set of non-relevant documents, for a particular , moving the away from these may not necessarily mean that we are moving the query away from non relevant documents in the entire corpus, since cannot estimate or know the size of the entire corpus. Moreover, the non-relevant documents are distributed over the entire vector space and not clustered together at one place. Therefore we assign a lower value to c and a high value to b. c=0.15 b = 0.75 (in book). Doing this will ensure that high value of beta will return all the relevant documents initially retrieved, and non return any such non-relevant documents of the previous query qo.
On running qo,
RD = {**D2, D3, D5, D6, D9**}, IRD = {D1, D4, D7, D8, D10}
On running modified query,
{**D2, D3, D5, D6, D9**, D11, D13, D14, D15}
Relevant docs match whereas non relevant dont.
A = 1, B = 0.75, C = 0.15

Q3) Kindly note the increasing/decreasing values as highlighted below

| Query | Tfidf rank (In **decreasing** order) (doc_number,cos_sim,index) | Bm25 rank (In **INCREASING** order) |
|---|---|---|
| Portable operating system | ('doc_3127', 0.26784046483618168, 3126)<br>('doc_1930', 0.22072449342793207, 1929)<br>('doc_2246', 0.2089775650950367, 2245)<br>('doc_3196', 0.13240376153998937, 3195)<br>('doc_1033', 0.077829917911029528, | (1235, 5.785863919088389),<br>(3067, 5.795257039511389),<br>(2739, 6.094865863983798),<br>(2423, 6.098237990011075),<br>(1032, 6.13424269194952), |

| | | |
|---|---|---|
| | 1032)<br>('doc_1236', 0.072653741701397573, 1235)<br>('doc_1680', 0.07206692566666098, 1679)<br>('doc_3068', 0.069642790169943375, 3067)<br>('doc_1461', 0.067765641358447565, 1460)<br>('doc_1591', 0.067764441466532432, 1590) | (1590, 6.167706491896828),<br>(1679, 6.46315795727342),<br>(2245, 9.202348029295452),<br>(3195, 9.500338829101363),<br>(1929, 10.949341999438335),<br>(3126, 15.604372324967276) |
| Parallel algorithm | ('doc_2664', 0.2064456296253451, 2663)<br>('doc_950', 0.19803290098910437, 949)<br>('doc_2714', 0.1954210764254968, 2713)<br>('doc_2785', 0.19130654353219742, 2784)<br>('doc_2266', 0.18983165531575324, 2265)<br>('doc_2685', 0.18668780246566377, 2684)<br>('doc_1262', 0.18566458083126838, 1261)<br>('doc_2973', 0.18265441213302877, 2972)<br>('doc_2700', 0.1748574549184061, 2699)<br>('doc_1811', 0.17222159373065898, 1810) | (2699, 6.688985351003363),<br>(2432, 6.7777845069109315),<br>(1261, 6.8056439322709545),<br>(2972, 6.904474725576904),<br>(2663, 6.935236758365553),<br>(1810, 6.98467782007549),<br>(2265, 7.036293820222367),<br>(2784, 7.104692620734937),<br>(949, 7.123323493354117),<br>(2713, 7.934674505222504) |
| Applied stochastic process | ('doc_1696', 0.23303297390085143, 1695)<br>('doc_268', 0.18044876852223399, 267)<br>('doc_2882', 0.1185616409116008, 2881)<br>('doc_1410', 0.10617719193510569, 1409)<br>('doc_1194', 0.094961462225712001, 1193)<br>('doc_2535', 0.094673336030258876, 2534)<br>('doc_1233', 0.092980627567901944, 1232)<br>('doc_1540', 0.091860683082434766, 1539)<br>('doc_20', 0.088232602336684354, 19)<br>('doc_3020', 0.087540062766681123, 3019) | (1440, 4.825546847943579),<br>(3012, 4.953490327830939),<br>(1924, 4.953490327830939),<br>(941, 5.07995442639275),<br>(1179, 5.132366773611164),<br>(1234, 5.14429594159226),<br>(77, 5.158980622969668), (1636, 5.185871922877849), (1215, 5.800708758486163), (1539, 5.881796637334228), (3058, 6.030819362795787), (19, 6.26323038272044), (1358, 6.278026199444438), (3119, 6.6093262826128125), (2534, 6.60979751200548), (1232, 6.746887355521312), (2881, 7.010145340515359), (3019, 7.2044699038401205), (1193, 7.29225706424164), (267, 7.637918058938541), (1409, 7.665194578163053), (1695, |

| | | |
|---|---|---|
| | | 10.930537782005219) |
| Perform evaluation and model of computer system | ('doc_2318', 0.37295557519293532, 2317)<br>('doc_2984', 0.24663351221898883, 2983)<br>('doc_3048', 0.24092058800946875, 3047)<br>('doc_1653', 0.21064584681705353, 1652)<br>('doc_2542', 0.20363518736860356, 2541)<br>('doc_3070', 0.19410429918289801, 3069)<br>('doc_1344', 0.18503816284132293, 1343)<br>('doc_3119', 0.18192973135464438, 3118)<br>('doc_2988', 0.1793759740509534, 2987)<br>('doc_1518', 0.17448141980454629, 1517) | (2740,8.85927334983221),<br>(2451,9.236489346622086),<br>(3069,9.365221397473007),<br>(3135,9.456137100604893),<br>(2570,9.572724442860354),<br>(2811,9.651693759802022),<br>(1652,9.686011259784404),<br>(2987,9.68719177524934),<br>(3118,9.83952506280586), (1517,<br>10.138718096905501), (2710,<br>10.379001378161599), (2541,<br>10.751855702023084), (2983,<br>10.906542534300263), (3047,<br>15.735315345708697), (2317,<br>16.8981492526769) |
| Parallel process in information retrieval | ('doc_2318', 0.37295557519293532, 2317)<br>('doc_2984', 0.24663351221898883, 2983)<br>('doc_3048', 0.24092058800946875, 3047)<br>('doc_1653', 0.21064584681705353, 1652)<br>('doc_2542', 0.20363518736860356, 2541)<br>('doc_3070', 0.19410429918289801, 3069)<br>('doc_1344', 0.18503816284132293, 1343)<br>('doc_3119', 0.18192973135464438, 3118)<br>('doc_2988', 0.1793759740509534, 2987)<br>('doc_1518', 0.17448141980454629, 1517) | (2451, 9.236489346622086), (3069,<br>9.365221397473007), (3135,<br>9.456137100604893), (2570,<br>9.572724442860354), (2811,<br>9.651693759802022), (1652,<br>9.686011259784404), (2987,<br>9.68719177524934), (3118,<br>9.83952506280586), (1517,<br>10.138718096905501), (2710,<br>10.379001378161599), (2541,<br>10.751855702023084), (2983,<br>10.906542534300263), (3047,<br>15.735315345708697), (2317,<br>16.8981492526769) |

Rocchio Implementation:

Rocchio feedback algorithm was implemented in order to modify the query vector for more and more results which the user finds relevant. So initially a query-(existing in the file) is run , then the user is prompted to enter a set of relevant doc indexes by himself evaluating the correctness by opening that particular doc and checking its relevance , then the new set of documents are returned having the documents labelled as relevant but very few or none of the non relevant documents. ……..

……..Initial output →

Doc name, cosine sim, index val1 val2 val 3

…….

…..
…..
….
Enter relevant docs….

- Indexes to be entered in list form -- [23 34 45 ]

New set of relevant docs printed:
Doc name, cosine sim, index  val1 val2 val 3
…….
…..
…..
….

Continued till you break out of the loop by exiting the program