**Assignment 3: Feedback and Query Expansion**

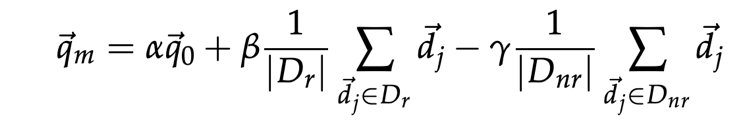
In the assignment 1 and 2, you learned information retrieval indexation (offline computation) and search + ranking (online computation). As assignment 2 shows, different ranking algorithms’ performance can be somehow different, and query quality (like short query or long query) plays a vital role in information retrieval. Don’t forget one of the essential tasks of IR – **help users to generate better query**, computationally, and to better characterize users’ information needs. Feedback is an important approach to achieve this goal.

Before you start this assignment, you should carefully read the feedback book chapter: <https://nlp.stanford.edu/IR-book/pdf/09expand.pdf>

Then, you will have a chance to verify these algorithms via the following tasks:

**Task 1: Rocchio algorithm for relevance feedback**

As the book said, “The Rocchio Algorithm is the classic algorithm for implementing relevance feedback. It models a way of incorporating relevance feedback information into the vector space model”. (section 9.1.1) You will need to test this algorithm with the following function:



Where is the original query vector, and are the set of known relevant and nonrelevant documents respectively, and α, β, and γ are weights attached to each term. These control the balance between trusting the judged document set versus the query: if we have a lot of judged documents, we would like a higher β and γ. Starting from , the new query moves you some distance toward the centroid of the relevant documents and some distance away from the centroid of the nonrelevant documents.

For this task, you will need some relevant and non-relevance documents to estimate the centroids of positive/negative documents. You can access [those documents from here](https://iu.box.com/s/f10v1gpovz35pl5oha7pnkk59x6y9ziu). Meanwhile, please read this tutorial for Lucene query generation: <https://lucene.apache.org/core/2_9_4/queryparsersyntax.html> which enables complex queries like (title:foo OR title:bar)^1.5 (body:foo OR body:bar)

For this assignment, we ONLY USE THE SHORT QUERIES for experiment.

In the reading, “Reasonable values might be α = 1, β = 0.75, and γ = 0.15. In fact, many systems, such as the image search system in Figure 9.1, allow only positive feedback, which is equivalent to setting γ = 0”. In this task, please tune the value of those hyperparameters. When you fix α = 1, please test different values of β and γ. Then, in the report, please use F1 to tell which parameter values can optimize the retrieval performance for this dataset. (Note that, in the reading “if we have a lot of judged documents, we would like a higher β and γ”)

Note that you are generating a computational query, which, hopefully, can be statistically better than user’s original query. From vector space perspective, the query can be an updated vector. Unfortunately, we didn’t have a chance to implement language model feedback (KL-divergence), but you are welcome to try it on your own.

For task 1, please fill the following tables:

Table 1: Rocchio algorithm parameter value comparison (we fix α = 1, and please use F1 score):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | β = 0.2 | 0.4 | 0.6 | 0.8 | 1 |
| γ = 0 |  |  |  |  |  |
| 0.2 |  |  |  |  |  |
| 0.4 |  |  |  |  |  |
| 0.6 |  |  |  |  |  |
| 0.8 |  |  |  |  |  |
| 1 |  |  |  |  |  |

Please highlight the best performed value setting in this table, and explain the reason why: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Table 2: Compared Rocchio algorithm performance (with the best parameter setting) with vector space model w.r.t. precision, recall, F1, MAP, and NDCG. Please explain the reason why.

**Task 2: Choose the feedback terms**

In task 1, we calculate the centroids of positive and negative documents, and all the document terms were employed to enhance the original query. Tan et al., 2007 found that “feedback has the disadvantage that irrelevant terms, which occur along with relevant ones in the judged content, may be erroneously used for query expansion, causing undesired effects” <http://sifaka.cs.uiuc.edu/czhai/pub/sigir07-term.pdf> So, in this task you will need to investigate an enhanced feedback approach to find the most important subset of terms for feedback. For above formula, instead of using all the terms in the documents, please propose another solution to filter out the noisy terms. I provide two candidate solutions here:

Solution 1: using IDF to filter out the noisy terms. Intuitively, terms with low IDF can be somehow noisy, e.g., “the”, “of”, “in”. We can easily select the top ranked terms with the highest IDF scores for expansion.

Solution 2: we can find a subset of words similar to user’s original query . You have different choices to calculate the word to query similarity. For instance, you can use the word2vec approach to calculate the average distance between the target word to the word in the original query. Of course, you can bring your own similarity function.

For this task, please first propose your own solution (with proper math formula). Then, please implement your idea, and give me evaluation results here (you may need to compare the outcome with task 1).