Assignment 4: Relevance Feedback Model

**Instruction**

In this assignment, you will implement the relevance feedback model presented on the course slides “relevance feedback and query expansion”.

### Tasks: Implement relevance feedback for language model

Your assignment 3 is able to return top N documents for a given query. We give a version of answer code implemented with Lucene API in assignment2-lucene-answer.rar. You can use the code to generate the list of top relevant documents for each query. However, its scores cannot be used into the calculation of the Relevance feedback model.

In this new assignment, what you need to do is to enhance the retrieval model with pseudo relevance feedback. That is, for a given query, you need to implement at least the following steps:

1. Obtain feedback documents: conduct the initial search using the query likelihood retrieval model with Dirichlet smoothing (as required by assignment 3, you can use your own assignment 3 codes), and obtain top K documents where K is a parameter set by the system. These K documents are treated as the relevant documents;
2. For each query term qi in the query, calculate the probability of the feedback documents generating this term, i.e., P(qi | feedback documents). Here all feedback documents are treated as one big pseudo document;
3. Then for each query term qi, the probability of one document D generating it based on relevance feedback is a linear combination of the original probability P(qi | D) and P(qi | feedback documents), where parameter α is used as the coefficient of P(qi | D) and 1-α is used as the coefficient for P(qi | feedback documents);
4. The probability of the query generated by the document is all the probabilities of each query term multiplying together;
5. Sort top N documents based on the probability generated in step 4.

NOTE: when you calculate the probability of feedback documents generating a query term, you should also use Dirichlet smoothing here with the same parameter like your document model

In this task, you should implement:

* **PseudoRFRetrievalModel. RetrieveQuery**

Given the query, this method can return the TopN most relevant documents. Different from assignment 3, this retrieval model is enhanced with the pseudo relevance feedback.

* **PseudoRFRetrievalModel. GetTokenRFScore**

For each token in the query, the probability of a term generated by the feedback documents is calculated.

Returned document list will be printed in the same way as in Assignment 3.

**Requirements and Reminders**

* You CAN ONLY use JAVA or Python to finish this assignment.
* In your own code, you CANNOT use any JAVA API library other than the standard JDK (for example, you cannot use apache commons, apache Lucene, indri etc. in this assignment).
* Feel free to use IDE tools such as Eclipse, Netbeans, PyCharm and etc.
* Do tell us the JAVA version you used for writing your assignment, e.g. JDK 1.6 or 1.7.
* Please only use Oracle JDK or openjdk.
* You CANNOT modify anything in class HW4Main.

**Grading**

Your submission will be graded based on:

* Correctness of the implementation on the required functions (70%)
* Efficiency of your implementation, make sure your code finish 4 query search within 3 minutes (20%)
* Necessary program annotation and commentaries (10%)

## Submission Requirements

A zipped file package with the naming convention as “pittids\_2140a3”. For example, suppose the CSSD id is jud1, then the submission package should be jud1\_2140a3.zip.

You can submit your assignment in **CourseWeb>>Assginments>>Assginment 4**.

The file package should contain:

1. src folder, which contain all your codes.
2. A txt file describing:
   1. how long it takes to finish 4 queries
   2. retrieval results
   3. Java version you used for writing your assignment, e.g. JDK 1.6 or 1.7.