Group Assignment 3

CS 5180 Information Retrieval

Grade: 60 points.

* This is group work. You will work with your team members and make one submission.

**Due Date: 11/04/2022 11:59PM on Canvas**

**PROJECT DESCRIPTION**

In this assignment, you will reimplement the indexes from group assignment 1 using PyLucene[[1]](#footnote-1). You will also reimplement the vector space model with cosine similarity for retrieving the top K documents (ranked retrieval) from the collection of document provided as an attachment. The specifications are:

**Part A: Exact Top K Vector Space Retrieval [10]**

* In part A, you will construct an inverted index for the collection (provided as the attachment) using PyLucene.
* You will then reimplement the vector space model based retrieval system for exact top K retrieval. You will implement this from scratch (i.e. do not use existing vector space libraries ) using PyLucene
* Use the TF-IDF weighting for the terms to construct the vectors for the documents and queries.
* Your index will be of the form
  + ***Term ID: [,(ID1, [pos1,pos2,..]), (ID2, [pos1,pos2,…]),….]***
* As in assignment 1, your program will accept the collection, process the documents and construct the index. Additionally, you will use the stop list (provided as an attachment) to weed out words from your dictionary. Note you should not index the document for each query. Simply do this at the beginning.
* Your program will then accept a free text query, generate the vectors for the documents and the query and compute the cosine similarity score for the documents. You can assume that you will not get single term queries.
* The program will retrieve and display the names of the top K documents for each query in decreasing order of their score.
* You will also note the time taken to retrieve the results for each query.

Part B: Cosine Similarity and Rocchio’s algorithm [40pts]

* Using PyLuece, you will implement a cosine similarity measure with *tf-idf* weighting. Your index should contain the information that you will need to calculate the cosine similarity measure such as *tf* and *idf* values. You may reuse code from the previous assignments as needed
* Using PyLucence, implement the Rocchio algorithm using the inverted indexes and vector space model from Part A for query refinement. Your system should display results and then prompt the user for providing positive and negative feedback. Use 𝛼= 1, 𝛽= 0.75, and 𝛾= 0.15 as parameters for the Rocchio’s algorithm.

Part B: Experimental study [35pts]

* Run your system for at least 3 queries from the test bed. Pick queries that have 5 or more relevant documents (see TIME.REL file). For each query, you will perform a series of 5 relevance feedback and plot the change in precision, recall and MAP
* You will prepare a report on the experimental study where you will provide at least the following details for each queries:
  + Query text and ID (provided in the testbed)
  + Precision, recall and MAP values of the query
  + IDS of document which are Positive and negative feedback provided for each query during each iteration of the Rocchio algorithm.
  + For each interaction of the Rocchio algorithm, provide the terms of the new query and their weights.
* Your report will have 3 plots (precision vs Rocchio iteration, recall vs Rocchio iteration and MAP measure vs Rocchio iteration) that depict the progressive change in the performance values over the iterations of the rocchio algorithm, for the three queries.
* Also discuss any query drift that you may observe in your results.
* Note: The queries provided in the testbed have varying number of relevant documents (see TIME.REL file). This can be a problem when calculating the performance values, if *k* is kept constant during the retrieval. For the experimental study, assume that the number of relevant documents is provided to the system along with the query. In other words, the value of *k* will change with the query.

Extra Credit: Pseudo Relevance Feedback [20pts]

One of the challenges of user feedback is that the user may not be willing to provide feedback. In such cases, pseudo relevance feedback can be used. You will compare the performance of your user feedback based system from Part A against a pseudo feedback mechanism (implemented using LyLucene), where the top 3 results of the system are considered to be relevant. Run this system with the same queries from your experimental study in Part B. Compare its recall, precision and MAP values against the system using user feedback.

**Other instructions:**

* Implement in Python and PyLucene
* Comment your code appropriately
* You may reuse the code from earlier assignments

**Attachments:**

1. Collection of documents

2. Skeleton code – implement the functions in the code. Use additional functions as needed.

3. Sample output

Submission

**Deadline for submission: 11/04/2022 11:59 PM**

* Submit the following files on canvas as a .zip file.
  + A PowerPoint file that provides a description of your implementation, including pseudocode, experimental results. You will also provide the tasks performed by each team member.
  + The source code files including the following
    - index.py
    - README file --- briefly describing your code and how to execute the code.
    - Output.txt: containing 5 queries queries and the output generated by your code. This is for testing your code.
* Each team will submit a single copy on canvas.

**Each team will also schedule a presentation with the instructor (Week of Nov. 7).** Your presentation will last about 10 minutes. The instructor will email you to set up the presentations.

**Anyone who misses the final presentation will not receive a grade for the assignment.**

**Late/re-submission**

* Team will have the option to resubmit or make a late submission with a penalty of 25%. Team who want to make use of this option will make their presentations during the last two weeks of the semester.

1. <https://lucene.apache.org/pylucene/index.html> [↑](#footnote-ref-1)