CS6200 Project

There are 9 retrieval systems implemented the project

1. BM25
2. TFIDF
3. Lucene
4. BM25-QueryExpansion
5. TFIDF-QueryExpansion
6. BM25-Stopping
7. TFIDF-Stopping
8. BM25-Stemming
9. TFIDF- Stemming

Lucene is implemented in java and rest of the systems are implemented in python.

Steps to run the project

Project setup:

Prerequisites for running the code:

1. Make sure Python version 2.7 is installed
2. Make sure JAVA is installed along with eclipse IDE
3. Please make sure all the python packages being referred in the code have been installed. Some of the main packages are Scipy, collections.

Step 1: Lucene: - generate documents with document ranks for queries

Lucene Dependencies:

1. lucene-queryparser-4.7.2.jar
2. lucene-core-4.7.2.jar
3. lucene-analyzers-common-4.7.2.jar

HW4.java: Right click and run as java program in eclipse, the user will be prompted to provide the below details.

* Index location: path where the Lucene indexes must be created.

Example: C:\Study\Information\_retrieval\IRProject\lucene\_index

* Data Location – Location of data files which must be indexed

Example: C:\ Study\Information\_retrieval\IRProject\files

* Query file – complete path to file containing CACM query text file

Example: C:\Users\rakesh\PycharmProjects\untitled2\Querylist.txt

* Result files – files for each query containing the top 100 scores with the document ID will be generated in the folder “LUCENE\LUCENE-RESULTS” in the current working directory

Step 2: Running BM25, TFIDF, BM25-QueryExpansion, TFIDF-QueryExpansion, BM25-Stopping, TFIDF-Stopping

* Create a new python project with all the files listed below

1. BM25RetrievalModule.py
2. consts.py
3. evalTask.py
4. evaluation.py
5. genUtils.py
6. indexer.py
7. indexerUtils.py
8. main.py
9. QueryExpansion.py
10. queryutils.py
11. RMBase.py
12. snippetGenerator.py
13. Task1.py
14. Task2.py
15. Task3A.py
16. Task3B.py
17. TFIDFRetrievalModule.py
18. tokenizer.py
19. utils.py

* Copy the “LUCENE” folder generated in Step 1 and paste it in python project directory
* Open command prompt and enter the below command in the command line

Command: - python main.py <<CACM-DATA directory path>>

Example: - python main.py “D:\CACM-DATA”

* “OUTPUT” folder will be generated in the project directory containing all the task results. The task results generated are described below

1. EVALUATION\_RESULTS: This folder contains the results for precision, recall, P@5 and P@20, All results for t-tests, Mean values for systems and average precision values respectively are presented in the below files:
   1. For precision values, files are named as follows: PRECISION\_RECALL\_VALUES\_OF\_\_<<SYSTEM\_NAME>>

Where SYSTEM\_NAME can be any one the first 7 systems.

Example: PRECISION\_RECALL\_VALUES\_OF\_\_BM25-Baseline.csv

* 1. For P@K values: P@K\_VALUES\_OF\_\_<<SYSTEM\_NAME>>
  2. ALL\_SYSTEMS\_T\_TESTS\_RESULTS.csv
  3. ALL\_SYSTEMS\_MEAN\_VALUES.csv
  4. ALL\_SYSTEMS\_AVG\_PRECISION\_VALUES.csv

1. TASK1: This folder contains BM25 results, Snippet generation for BM25 and TFIDF, TFIDF results, Document frequency table for Unigram, Word counts for file for Index, Index.
2. TASK2: This folder contains the results of systems BM25 and TFIDF with query expansion and it contains the following BM25 results, TFIDF results,

Document frequency table for Unigram, Stopped index, word counts by file for stopped index.

1. TASK3A: This folder contains the results for BM25 and TFIDF systems with stopping
2. TASK3B: This folder contains index of the stemmed version of the corpus and retrieved results for queries given for this task.
3. TOKENIZED-DATA
4. Querylist.txt

Citations:

Following things were used while developing this project:

Beautiful Soup package in python

- https://beautiful-soup-4.readthedocs.io/en/latest/

Scipy package in python