README - 590R Assignment 2

Submitted By :- Ankita Mehta

October 20, 2017

This document will take you through the implementation of Retrieval Models for shakespeare dataset written in Python language. This part has been build over the previous submission. To run the project, change the current working directory to Assignment/src. Paths mentioned in the commands written below are according to the above current working directory.

NOTE: python RetrievalAPI.py -h commands will give you the information about the command line parameters of the wrapper.

1 Code Dependencies

Python 2.7.12 :: Anaconda has been used for this project.
Pip version 9.0.1
Installing JSON : pip install json
Installing trecrun: pip install trectools : Refer this github link for more details: trecrun
Installing argparse: pip install argparse

2 Build and Run the Code

Code structure :
The code structure has been divided into three main folders: Assignment2/data, Assignment2/src, Assignment2/results.

*Assignment2/data:

1. shakespeare.json: It is the input data file for creating compressed/uncompressed index
2. query.txt: It is the query file having 10 queries
3. unc_manifest: : It is the manifest file having the names of files and indexes created for uncompressed file.
4. comp_manifest: It is the manifest file having the names of files and indexes created for compressed file.

*Assignment2/src:
This folder has some codes from the previous assignment :

1. main.py: It is the entry point for finding the indexes, generating query files with unique random words and dice coefficient.
2. indices_creation.py: It is the class having functions for finding the indexes, generating query files with unique random words and dice coefficient.
3. Encoding_decoding.py: It is the class having functions for performing delta and Vbyte encoding-decoding.
4. API_extract_statistics.py : It is the class to extract vocabulary, Collection Term frequency and document frequency for the query word.

Codes which have been amended/written specifically for this assignment are :

1. RetrievalAPI.py: : It is the entry point for API Retrieval.

2. Calculate_probabilistic_score.py : It is the class to calculate doc scores for probabilistic models viz : BM25 , QL-JM , QL-DIR, using document-at-a-time evaluation method for uncompressed indices.
3. prob_scores.py: It is the class in which BM25 , QL-JM , QL-DIR scores have been implemented.
4. Calculate_VSM_score.py : It is the class to calculate doc scores for Vector space model using document-at-a-time evaluation method for uncompressed indices.
5. VSM.py : It is the class in which VSM score calculation has been implemented.

*Assignment2/results: It has the some previous results (6 for each) for compressed and uncompressed index to be used for this assignment.

1. xxx_docNo_playId: It is the mapping from doc_No to playId
2. xxx_docNo_sceneId: It is the mapping from doc_No to scene_Id
3. xxx_lookup_table: It is the lookup table having mappings from term to doc_No , count , Collection_term_frequency , document_frequency
4. xxx_sceneId_docNo: It is the mapping from scene_Id to doc_No
5. xxx_Inverted_list: It is the binary file having stored inverted lists.
6. xxx_docNo_length: It is the mapping from doc_No to its length.

Note: Here xxx is either 'unc' or 'comp' for uncompressed and compressed index respectively.

Other results relevant to this assignment are:

1. bm25.trecrun , ql_jm.trecrun , ql_dir.trecrun , vsm.trecrun are the files written in TREC format for each model. One more file is present in results folder : judgements.txt which has the judgements for top 10 queries in the order of **BM25,QL-JM , QL-DIR**

The other files present in the Assignment2 folder are : README, report.pdf

One wrapper have been written for this project i.e. - **RetrievalAPI.py** having location: Assignment2/src

2.1 RetrievalAPI.py

There are 4 variants to run this code using different command line parameters :

2.1.1 Create scores for all the queries using BM25

Command that will be used to test the uncompressed indexer on one-word query file.

```
python RetrievalAPI.py -q ../data/query.txt -m ../data/unc_manifest -no 1
```

-q specifies the oneword query path ;
-m specifies the manifest file path - the file having all paths written

2.1.2 Create scores for all the queries using QL-JM

Command that will be used to test the uncompressed indexer on one-word query file

```
python RetrievalAPI.py -q ../data/query.txt -m ../data/unc_manifest -no 2
```

2.1.3 Create scores for all the queries using QL-DIR

Command that will be used to test the uncompressed indexer on one-word query file

```
python RetrievalAPI.py -q ../data/query.txt -m ../data/unc_manifest -no 3
```

2.1.4 Create scores for all the queries using VSM

Command that will be used to test the uncompressed indexer on one-word query file

```
python RetrievalAPI.py -q ../data/query.txt -m ../data/unc_manifest -no 4
```