CS60092: Information Retrieval
Jan 2018, Assignment 1

Motivation: This assignment is to give you a hands on feel about a simple Information Retrieval system.

Task:
You are given a dataset consisting of the following:

- Documents
- Queries
- Documents relevant to the queries.

You have to implement and compare the following systems for boolean query processing:

- Grep based.
- Index based.
- Lucene based (well known indexing tool).

Report performance metrics (Precision and Recall) and total time for searching all queries for both the techniques.

You can issue and queries with all the search terms.

Using the dataset provided :

a) Use grep to find the result to the sample queries given. Time each execution of grep search and make a record.

b) Develop an inverted index - dictionary and postings list using standard data structures in Java (Hashmaps, ArrayList…) or Python(Dictionary, Json Formats, List…). You can choose to tokenize and stem / lemmatize the data. In python use NLTK 3 libraries (http://www.nltk.org/install.html) (NLTK Book -- http://www.nltk.org/book) or CoreNLP libraries 3.6 in Java (http://stanfordnlp.github.io/CoreNLP/download.html). Develop solution for simple conjunctive/disjunctive queries. Run on the queryset given. Tabulate the speedup of search against the aforementioned grep usage. Also calculate precision and recall for the given queryset.

c) Build an inverted index using Lucene (Java) - https://lucene.apache.org/ or PyLucene(Python) -https://lucene.apache.org/pylucene/install.html or Elasticsearch(Python) - https://pypi.python.org/pypi/elasticsearch.

Now again tabulate the speed as well as Precision/Recall and compare with the previous two approaches.

Output expected for submission: Code + document with tabulations of speed, precision/recall, and comparison with the previous two approaches.

Dataset description:

All necessary data is available at:

https://drive.google.com/open?id=1Pvc9MBMc2fF02vTB4BtgaYs4YhW_Pb0-
The folder Assignment1 contains query.txt, output.txt, alldocs.rar.

1. query.txt contains total 82 queries, which has 2 columns query id and query.
2. alldocs.rar contains documents file named with doc id. Each document has set of sentences.
3. output.txt contains top 50 relevant documents (doc id) for each query.