Georgetown University

COSC 488: Introduction to Information Retrieval
Spring 2018

Nazli Goharian (nazli@ir. cs.georgetown.edu)

Project Part 2: Query Processing
Date: February 15
Due Date: March 20

Grading: This assignment is 18 points out of total points (40 points) allocated for projects in the semester. It will be graded on the scale of 100.

Objective:

It is the time to work on the query-processing phase. In this part of the project, you will apply different retrieval strategies/models and similarity measures to perform relevance ranking. You must evaluate how your engine does. For this, you will be using the same Mini-Trec created from the TREC benchmark data provided by NIST, along with Trec *title* queries. The *qrels* file has "presumably" the relevant documents to each of the queries in the query file. Using *treceval*, (provided by Trec) you will evaluate the accuracy of your IR engine.

Requirements:

Inverted Index: Use your search engine that has single-term index, proximity index, phrase index, and stem index. (It is your choice as to the issue of memory requirements for this project !)

Pre-processing of Queries: Enable your application to read a list of queries from the query file. These queries are also tagged and need to be pre-processed same as you did for the documents to identify the query terms. For the experimentations that you search the stem-index, you need to stem query terms at the time of query processing. Naturally, the stemming rules for both documents and queries should be the same! For example if you used Porter stemmer to stem the collection, use Porter stemmer to stem the query terms. Note that the queries in the TREC query file are identified by their unique numbers. You should only use the *title* part of the queries for retrieving documents. Note that you need to identify the *special terms* in queries in the same way that you have identified them in the documents. Similarly, in the same way that phrases are generated in the collection, create phrases from the query terms.

Relevance Ranking: Using different information retrieval strategies and similarity measures, perform the query processing to identify the relevant documents and obtain relevance ranking.

a) Vector Space Model using *Cosine* measure -- use normalized tf-idf
b) Probabilistic model - Use BM25
c) Language model: (Your choice of query likelihood with Dirichlet smoothing, KL-Divergence)

Evaluation: You are asked to do experimentations and gather statistics and provide reports 1 &2, as specified below.

Identify the top 100 retrieved documents with their relevance ranking scores for each query. These top retrieved documents for each query are used as an input to *treceval* software to generate Average Precision over all queries. A description of the format and how to use treceval is given on the blackboard. Then fill out the tables for each specified case, as given for each report:

Report 1: Perform query processing and fill out the table. Include the MAP and running time of your system when using the single term and the stem index, and compare it to the MAP of **Elasticsearch**. Tune the preprocessing step and the scoring function of Elasticsearch and analyze how they affect its performance.

Retrieval Model	**MAP** single term index		**Query Processing Time (sec)**		**MAP** stem index		**Query Processing Time (sec)**	
	Your engine	**Elastic Search**	**Your engine**	**Elastic Search**	**Your engine**	**Elastic Search**	**Your engine**	**Elastic Search**
a) Cosine								
b) BM25								
c) LM								

Report 2: For each query, if the phrase terms are common (high document frequency) send the query to the **phrase index**, otherwise send it to the **proximity index** **(Note:** make sure you are demonstrating that both Phrase index and Proximity index are utilized for query processing during this project and they are functional). **If not enough documents found** then use **single term index** or **stem index** (your choice). Make sure you configuration uses at least two index if not all the four for each query. Provide your analysis. You can set some threshold for the number of retrieved documents.

Retrieval model	**MAP**	**Query Processing Time (sec)**
a, b, or c (your choice)		

Deliverables

Cover page (1 pt): should contain the following in the exact order as specified:

a. Status of this assignment: Complete or Incomplete. If incomplete, state clearly what is incomplete.
b.Time spent on this assignment. Number of hours.
c. Things you wish you had been told prior to being given the assignment.

Design Document (10 pts): The design document should be written prior to coding. There should **not** be any code in your design document. No specific template is provided to you for your design. You may draw a diagram to show the architecture and the flow of the software components, and/or to provide the write-up of your design decisions.

A Functional System & Reports & Analysis (89 pts): Results should be used to provide a good analysis of your engine. Thus, you are expected to provide a good analysis along with your results. (The **ElasticSearch** results will receive a total of **10 points**)

Demo: You may be asked to give a demo of a working system, satisfying the requirements. This includes explaining your design, demonstrating that all requirements are implemented and are functional, and answering the questions.