COMP479 Project 2

Report

Project 2

Submitted by

Student ID	Name of Student
40058103	Guillaume Rochefort-Mathieu

Department of Computer Science and Engineering

Concordia University

Montreal

Fall 2019

Contents

Chapter 1: Implementation

The implementation of the project was made keeping in mind the usage of the least amount of memory possible within the constraint of the project.

1.1 Preprocessing

Most of the preprocessing was kept from the original implementation of Project 1. What mostly change is the addition of the term frequency to the postings.

1.1.1 Term Frequency

After applying preprocessing of the document, the tokens were inserted in a FreqDist from the NLTK. The FreqDist is a dictionary which kepts count of the number of occurence for each token and also the number of tokens sample.

The sample occurence of the tokens was then added to a dictionary containing all the length of docuemnts which are used later for the BM25. Each token is then yield with its term frequency as a tuple.

The remaining process of the SPIMI algorithm remains the same but in this case postings are tuples of docID and tf.

1.2 Search Engine

The search engine did change. Since the search engine is now using a ranking algorithm the queries needed to be ranked with regard to the BM25 algorithm.

1.2.1 Query

The query language was removed from the Search Engine. As previously mentionned, the queries is evaluated with regard to the results calculated by the BM25.

1.2.2 Simple Ranked Search Engine

The process how queries are now handle has changed. The postings for the compressed queries term are now retrieved from the index.

Next, docID, term a term frequency are inverted. That is for each document a list of term and term frequency tuple is created. That is since not all terms are in every document multiple evaluation of BM25 would result in 0. (See 1.2.3 for more information on BM25)

Once all the postings list are retrived the query operation is executed. If the query is an AND, then the postings list is first sorted by ascending length and intersect the with the result starting with the smallest listi s the result. If the operation is an OR then the postings list are inversely map that is the docID retuns the number of terms that the document have. The inverse map is then return sorted in descending order of number of terms a document contains.

Note that single keyword may either use AND or OR as the search engine checks if only one postings list is return then it returns it as the results.

1.2.3 BM25

As previously mentionned, BM25 is a non-binary model. Hence, the introduction of the term frequencies were necessary to be able to calculate a ranking of given documents retrieved for a given query.

The particular derivation of the BM25 equation that was implemented in this project is equation (11.32) from the IR book. [1]

Since the indexer was originally implemented as a binary model, using the suggested values for $1.2 \leq k_1 \leq 2$ and $b = 0.75$ to have optimal result.

The equation was then executed for each document and the documents where then sorted in descending order for their respective result.

The ranking

Chapter 2: Queries Output

In this chapter the results of the queries given by Dr. Sabine Bergler and other students. Unfortunately, the documents extra tags PLACES, TOPICS and CATEGORY were to broad to make a golden ratio out the test data.

Instead, queries were run with different values for the hyperparameter k_1 and b

2.1 Challenge Queries

2.1.1 Democrats' welfare and healthcare reform policies

The query "Democrats' welfare and healthcare reform policies" resulted in the following documents. (Note only top ten where inserted in the report as the list is exhaustive)

With the hyperparameter $k_1 = 1.2$ and $b = 0.75$

[(6940, 30.826782687139662), (4006, 30.349621147373888), (11204, 28.677240336726285), (18271, 27.169468260658117), (1999, 26.78649749270545), (21577, 26.70714646695537), (19173, 26.351621960796944), (7375, 26.128307771174303), (18731, 26.044327401676465), (20805, 25.871190678891946), . . .]

With the hyperparameter $k_1 = 2.0$ and $b = 0.5$

[(4006, 36.4642759307857), (6940, 36.45534493246048), (11204, 35.941339710232015), (1999, 33.4777071695746), (21577, 33.17183211833669), (18731, 32.260501601622266), (18271, 31.57656369413789), (12806, 31.498242925776317), (17940, 31.471905497700465), (2417, 31.39360574309009)]

2.1.2 Drug company bankruptcies

The query ”Democrats’ welfare and healthcare reform policies” resulted in the following documents. (Note only top ten where inserted in the report as the list is exhaustive)

With the hyperparameter $k_1 = 1.2$ and $b = 0.75$

[(8384, 8.387725769404485), (19640, 8.179057807749686), (9542, 7.825796949089394), (9342, 7.598398317442999), (192, 7.533576775217546), (7940, 7.359970480240968), (1391, 7.3204944662731), (3739, 7.298517394446336), (1998, 7.168606330774399), (21348, 7.168606330774399), . . .]

With the hyperparameter $k_1 = 2.0$ and $b = 0.5$

[(8384, 10.703154578267146), (9342, 9.34049680226152), (19640, 9.253239222112047), (192, 9.167837111363596), (7940, 9.027253030867518), (730, 8.714635927761218), (9542, 8.196056441640353), (1391, 8.181945782766997), (17680, 8.161021415173431), (6544, 7.923393047903912)]

Note that the first value of the tuples are the docID and the second value are their BM25 value.

2.1.3 George Bush

The or query ”George Bush” resulted in retrieving these documents. (Note only top ten where inserted in the report as the list is exhaustive)

With the hyperparameter $k_1 = 2.0$ and $b = 0.5$

[(8593, 1.7664459669287513), (20891, 1.678145553017121), (4008, 1.5163313599010013), (16780, 1.4352788122417348), (20719, 1.2559456931524726), (3560, 1.185803639953226), (10400, 1.0299220969312544), (2766, 1.025766610876294), (4853, 1.0216445228267386), (7525, 1.013668562603789), . . .]

With the hyperparameter $k_1 = 2.0$ and $b = 0.5$

[(20891, 1.8707189729793066), (8593, 1.7255741341431026), (4008, 1.7085198854543973), (16780, 1.564786014681132), (3560, 1.3195993124294256), (20719, 1.3037645320085105), (7525, 1.1440976703543277), (8500, 1.0784466571085476), (20860, 1.0282562091697118), (10400, 0.9731132751960619)]

2.2 Conclusion on Hyperparameter

After, these small test cases, as the query is smaller the scale of the discrepencies is smaller. That actually follow the theory in the book where b is used for the normalization of the length.

In the case of the query "George Bush" the change in the hyperparameter didn't really affect the output. With more optimization for this search engine we could determine the best value for the hyperparameter k_1 and b but it seems the values suggested in the IR book [2] were a good starting point.

For optimization, we could establish a hypothesis with regard which files would be return. Unfortunately, the corpus would be need to be annonated to be able to calculate metrics such as recall and precision on the BM25.

References

[1] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008, p. 233.

[2] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008, p. 233.