Information Retrieval (CS317)

Programming Assignment No. 1

Spring 2019

Submission Date: March 12, 2019

Assignment Objective

The objective of this assignment is to make you understand how different indexes work in retrieving different query from a collection. You will create Inverted index and positional index for a set of collection to facilitate Boolean Model of IR. Inverted files and Positional files are the primary data structure to support the efficient determination of which documents contain specified terms and at which proximity. You also learn to process simple Boolean expression queries through this assignment.

Datasets

You are given a collection of Short Stories (File name: ShortStories) for implementing inverted index and positional index. A single file contains 50 pen crafts of stories for children. You also need to implement a pre-processing pipeline. It is recommended to first review the given text files for indexing. Each file contains a single story; the first line contains the name of the story. The second line contains the author-name You need to treat each as a unique document. If there is any extra header and footers for each pen craft, it need to be filter and only get the content of it will be challenging.

Query Processing

In this assignment, all you need to implement an information retrieval model called Boolean Information Retrieval Model with some simplified assumptions. You need to treat each story as a document and need to index it content separately. There are 50 documents. you need to implement a simplified Boolean user queries that can only be formed by joining three terms (t1, t2 and t3) with (AND, OR and NOT) Boolean operators. For example a user query may be of the form (t1 AND t2 AND t3). For positional queries, the query text contains "/" along with a k intended to return all documents that contains t1 and t2, k words apart on either side of the text.

Basic Assumption for Boolean Retrieval Model

1. An index term (word) is either present (1) or absent (0) in the document. A dictionary contains all index terms.
2. All index terms provide equal evidence with respect to information needs. (No frequency count necessary, but in next assignment it can be)
3. Queries are Boolean combinations of index terms (at max 3).
4. Boolean Operators (AND, OR and NOT) are allowed. For examples:
   X AND Y: represents doc that contains both X and Y
   X OR Y: represents doc that contains either X or Y
   NOT X: represents the doc that do not contain X
5. Queries of the type X AND Y / 3 represents doc that contains both X and Y and 3 words apart.

As we discussed during the lectures, we will implement a Boolean Model by creating a posting list of all the terms present in the documents. You are free to implement a posting list with your choice of data structures; you are only allowed to preprocess the text from the documents in term of tokenization in which you can do case folding and stop-words removal but no-stemming. The stop word list is also provided to you with assignments files. Your query processing routine must address a query parsing, evaluation of the cost, and through executing it to fetch the required list of documents. A command line interface is simply required to demonstrate the working model. You are also provided by a set of 10 queries, for evaluating your implementation.
Coding can be done in either Java, C/C++, Python, or C# programming language. There is additional marks for intuitive GUI for demonstrating the working Boolean Model along with phrase query search.

Files Provided with this Assignment:

1. ShortStories
2. Stop-words list as a single file
3. Queries in a single file. (Some test queries- result set will be shared soon)

Evaluation/ Grading Criteria

The grading will be done as per the scheme of implementations, query responses and matching with a gold standard.

<The End>