Labo 2 : Evaluation of results obtained from *Apache Lucene*

1 Introduction

The goal of this lab is to compare different analyzers by calculating a set of the metrics introduced in the course.

We will use the same corpus as the first lab: the CACM publication list.

1.1 Organization

The lab should be realised in a group of maximum 2 students.

1.2 Deliverables

You will return a zip archive with:

- The report (PDF only) containing both the answers to the questions asked in the different sections,
- The source code of your implementation.
- The file metrics.txt (see below)

1.3 Deadline

See deadline on *Moodle*.

2 What you have

Start by downloading the lab 2 archive on Moodle.

In this lab, you will make a comparative evaluation of using four different analyzers:

1. `WhiteSpaceAnalyzer`
2. `StandardAnalyzer`
3. `EnglishAnalyzer` (it already provides default stopwords), and
4. `EnglishAnalyzer` with `common_words.txt`

You will apply one-by-one each of the four mentioned analyzers on the CACM collection and compare different metrics which are detailed below.

To be able to evaluate results you need a benchmark. The CACM collection provides you such a benchmark. It provides you a set of queries and their respective relevant results/documents. This way you can compare how your analyzers perform against the ground truth i.e. the set of documents returned by each queries from CACM.

2.1 CACM Queries

The CACM collection contains 64 queries given in query.txt. The file contains the query id and the query text separated by tabulation. There is an empty line between each query.

The parsing of the file is already done for you.

2.2 CACM List of relevant results

The CACM collection has also a list of relevant results per query provided in qrels.txt, which correspond to the 64 queries. For example, query 1 has 5 relevant results that include publications 1410, 1572, 1605, 2020 and 2358. Note that there are queries with no relevant documents (for example queries 50-56).

The parsing of the file is already done for you.

2.3 Structure of the lab

There are 2 classes:

- `Lab2Index` it will create an index for you given an analyzer. Also it can perform queries and return to you the results/documents. You don't need to modify this class.
- `Evaluation` it uses the `Lab2Index` class, computes the asked metrics (i.e. compare the returned documents from the analyzers vs the true documents from the CACM queries) and finally displays these metrics. Your job is to implement the metrics. More on this below.

3 What you need to do

Basically these are the steps to do the evaluation of an analyzer. You have to repeat (i.e. change the analyzer class) for each analyzer mentioned in section 2 of this document.

3.1 Indexing

Create an index with two fields: publication id and publication content. Publication content should contain both the title and the summary of the publication.

This is already done for you. You just have to provide an instance of an Analyzer.

3.2 Querying

For each query, use the Lab2Index.search() method to get the list of retrieved documents. Let's call this list queryResults.

Alongside , retrieve the true results of that query using qrels.get(). Let's call this list qrelResults.

See the given source code to have more information.

3.3 Evaluation

Compute the following metrics using mainly queryResults and qrelResults:

1. **Summary Statistics :**

   a. total number of documents,
   b. total number retrieved documents for all queries
   c. total number of relevant documents for all queries and
   d. total number of relevant documents retrieved for all queries.
   e. Average precision for all queries
   f. Average recall for all queries
   g. F-Measure using (e.) and (f.)
2. **Average precision for each query (AP) :** As an example, consider a query that has four relevant documents which are retrieved at ranks 1, 2, 4, and 7. The actual precision obtained when each relevant document is retrieved is 1, 1, 0.75, and 0.57, respectively, the average precision of which is 0.83.
3. **Mean Average precision (MAP) :** MAP is calculated by averaging the AP values (from the previous point) over all queries.
4. **R-Precision for each query :** R-Precision is the precision at the position where R documents have been retrieved, R being the number of relevant documents for the query in the benchmark collection.
5. **Average Precision at Standard Recall Levels :** Precision at 11 standard recall levels (0, 0.1, 0.2, . . . , 0.9, 1.0). Each precision average is computed by summing the interpolated precisions at the specified recall value and then dividing by the number of queries.

3.4 Compare the results from the differents analyzers

a) **metrics.txt :** Save the displayed metrics from each analyzers into a text file called `metrics.txt`. You can do it just by copy-pasting the console output after each run. There is no need to code something to print into a file.
b) **Recall-Precision Graph :** It is created using the 11 recall values obtained in the step 5. Use whatever program/library you want to produce a graph that shows one curve per analyzer in the graph. You can use a spreadsheet to do that for example.

3.5 Reporting the results

Compare the analyzers using the graph you made at the previous step. Make a conclusion on your results. To submit (as described in section 1.2): source code, report and `metrics.txt` file