CS6200 Information Retrieval
Fall 2019
Instructor: Omar Alonso

Homework #3. Build your own indexing and query a TREC data set

Objective: process a collection of documents, create an inverted index on titles, and run queries. Use the same document collection for HW2 but this time you need to index all documents. Your index should be able to handle large numbers of documents and terms.

This programming assignment involves writing three components:

1. An indexer (50 points)
2. Implement vector space scoring (25 points)
3. Evaluate search results (25 points)

Prerequisites

1. Use the AP89_DATA.zip collection.

Document Indexing

Create an index of the downloaded corpus. The documents are found within the ap89_collection folder in the data .zip file. You will need to write a program to parse the documents and index the content. Note that each document begins with a line containing `<DOC>` and ends with a line containing `</DOC>`.

Ranked Retrieval

Implement vector space ranked retrieval.

Query Execution

Write a program to run the queries in the file query_desc.51-100.short.txt, included in the data .zip file. You should run all queries (omitting the leading number) and output the top 50 results (`DOCNO` and `TITLE`) for each query to an output file.

The file `qrels.adhoc.51-100.AP89.txt` contains the relevance assessments. By looking at the query description file, we know the content of topic 91:

```
91.   Document will identify acquisition by the U.S. Army of specified advanced
weapons systems.
```

The qrels tell us which files are relevant or not:

```
91 0 AP890111-0065 0
91 0 AP890113-0040 1
```

Looking on the files again:

```
Defense Minister Says Short-Range Nuclear Weapons to be Modernized (NR)
Military Buildup At a Glance (R)
```

What you need to submit

1. Your indexer's Java source code

2. Your query program's Java source code
3. A description of your indexing design. How are you approaching the problem, your specific design choices, and other information that you think is important to describe.
4. A description of your scoring implementation for both vector space
5. Compute the precision of the search results according to the qrels.
6. Documentation on how to compile and run the code. I'll be testing your work.