Information Retrieval and Text Mining Project 1

Olteanu Fabian Cristian

FMI, AI Master, Year 2

1. Thought process

The program is composed of two main components: an indexer and a searcher (both use the Apache Lucene API).

The indexer uses Apache Tika to read pdf, doc(x) and text files only (by creating a class that implements the FileFilter interface from the java io package).

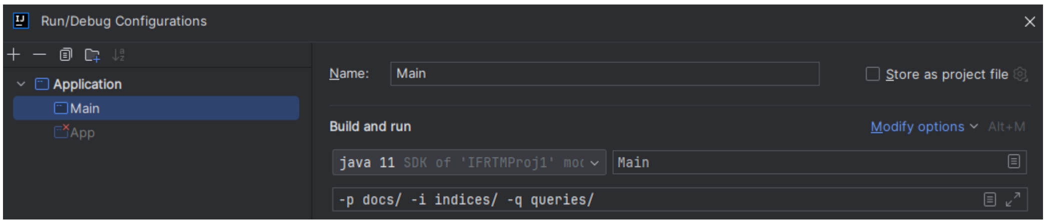
After the documents from a given path have been indexed and a query has been given, the searcher parses the query and searches the top n hits (by default, "n" is equal to the number of the documents in the folder, but it can be changed to any natural value) from the indexed documents.

The searcher uses a modified version of the Romanian Analyzer found in the Apache Lucene API to include an additional list of Romanian stopwords, but also to most importantly add an ASCIIFoldingFilter to convert all diacritics to the equivalent ASCII characters. This is done after applying the LowerCaseFilter, StopFilter and SnowballFilter (with the Romanian stemmer). The effect of this improvement was evident after conducting testing on a number of documents, which yielded much better results than the base RomanianAnalyzer, as I was able to query for phrases containing combinations of words with diacritics and words without diacritics, phrases containing stemmed words etc while getting correct results.

2. Running instructions

Open the project located in IFRTMProj1 with IntelliJ and install JDK 11 if prompted. Afterwards, run the Main class (as in the figure shown below) using the following params: "-p docFolder/ -i folderInWhichToStoreIndices/ -q queryFolder/ -h maxNumberOfHits(optional)". If the query folder is not mentioned, a the CLI will prompt for queries.

The program reads all .txt files from the query directory and provides results for every line (considered as an individual query) of each file in the folder. If a line is empty, it is skipped.



Name: Main

java 11 SDK of 'IFRTMProj1' mo(

Main

```
-p docs/ -i indices/ -q queries/
```

Figure 1. Running the program in IntelliJ