

## Project 1 Documentation

The 2 main classes of the project are Indexer and Searcher. Other classes present in the project are Utils, Constants and Tester. The main library used in this project is Apache Lucene.

The Indexer class is the one which deals with the inverted indexes that must be built. It has a constructor that gets the directory path where the inverted indexes must be saved. It has an instance of the IndexWriter class from Lucene which it uses to create these inverted indexes to the directory. The Indexer has a function called create Index which receives the data directory path and a FileFilter from the Utils class to keep only the relevant files. It indexes all the files in the data directory by calling indexFile for each one. The indexFile function gets the contents of the file using an appropriate function based on its extension. There are 3 such functions, one for .txt documents which uses the basic Scanner class, one for .doc and .docx documents which uses the Apache POI library and one for .pdf files which uses the Apache PDFBox library. After the function indexFile gets the file documents based on its extension, it uses the removeDiacritics function from the Utils class to remove diacritics and the RomanianAnalyzer from Lucene and the analyze function from Utils to remove stopwords, stem and lower the result. The resulting tokens are rejoined in a single String and added as a Document field. The IndexWriter instance then adds the document. The Indexer class also has a main function to start indexing the contents of the data directory.

The Searcher class is the one which deals with searching for documents. It has a constructor which gets the index directory path and it also has an instance of IndexSearcher and of QueryParser from Lucene. The QueryParser uses the StandardAnalyzer from Lucene. The function used to search in the documents is called search and gets the search string

which is fed to the QueryParser instance which is then used by the IndexSearcher instance to search the documents. The getDocument function is used to obtain a Document instance from each ScoreDoc resulted from the search. The main function of the Searcher class deals with reading the query from the console, removing diacritics through the removeDiacritics function from the Utils class, and uses the instance of RomainianAnalyzer from Lucene and the analyze function from the Utils class to remove stopwords, stem and lower the words. The resulting tokens are rejoined in a String and used for the search function. The results are then displayed to the System.out together with some useful information.

Other classes of the project are Utils, Constants, CustomFileFilter and Tester. Utils contains functions used by both Indexer and Searcher. The first is removeDiacritics which is used to replace the diacritics of a String with standard characters. The second is analyze which is used to get tokens after removing stopwords, lowering and stemming a String. The Constants class includes some useful constants such as index directory, data directory and the name of the content field of Document objects. The final class is Tester, which just has a main function which calls the main functions of Indexer and then Searcher.

After testing, it seems to correctly handle situations when words with diacritics are used in a query but not in documents, when they are used in documents but not in a query. I could not find any unacceptable edge case that's not working resulting from the combination of stemming, diacritics removal and stopwords removal.