Sistem de Regăsirea Informației pentru Limba Romana

Information Retrieval & Text Mining

Student: Lupașcu Marian

CLASELE

Proiectul conține următoarele clase:

- Indexer – clasa care pornește de la un set de documente pe care îl preia ca parametru in funcția main (args[0]) si creează un “inverted index” pe care îl salvează in folderul ".\index". Aceasta clasa citește documentele cu ajutorul clasei DocumentReader (care folosește Tika) apoi se salvează ca documente txt (care conțin fix aceiași informație) intr-un folder temporar pe baza căruia se construiește “inverted index-ul”, apoi folderul temporar este stres si informația este salvata pe disc in “inverted index”.
- Searcher – clasa care pornește de la “inverted index-ul” creat anterior si de la un sting de căutare numit in continuare query. Aceasta clasa returnează documentele care sunt relevate pentru stringul de căutare pe baza unui scor de confidenta.
- DocumentReader – clasa care parează diferite tipuri de documente cu Tika (txt, doc/docx sau pdf) si returnează informația ca plain-text.
- MyRomanianAnalyzer – clasa de procesare a informațiilor salvate de Indexer si a query-ului de căutare. Aceasta clasa este o extensie a clasei RomanianAnalyzer din Lucene (nu a putut fi moștenită din RomanianAnalyzer întrucât aceasta este clasa final in Lucene), peste care au fost adăugate câteva features: metoda modifyStopWords(care îmbogățește lista default de stopwords din Lucene cu stopwords fără diacritice – daca exista stopword-ul câteva acum există si cateva) și in funcția de baza createComponents se mai adaugă cateva procesări. Acesta sunt eliminarea de diacritice a token-ilor apoi eliminarea de stopwords cu si fără diacritice, de unde rezulta doar tokeni fără stopwords, apoi ca stemming se folosește SnowballFilter si alți cateva filtri de normalizare din Lucene.
- RemoveDiacriticalsFilter – o clasa filtru de eliminare de diacritice din tokeni, necesara pentru o preprocesare in MyRomanianAnalyzer.

RULARE

In continuare toate operațiile vor fi făcute din folderul de baza si anume P1.

1. Se creează un folder in P1 in care se adaugă documentele dorite a fi indexate. By default există folderul docs care conține 13 documente doc/x, 6 documente pdf, 4 documente txt si 2 imagini (deci in total 23 de documente ce vor fi analizate din 25 – cele 2 imagini nu vor fi analizate).
2. Se creează folderul dependencies in care se adaugă dependințele proiectului:
    - lucene-analyzers-common-8.6.3.jar,
    - lucene-core-8.6.3.jar
    - lucene-queryparser-8.6.3.jar
    - pdfbox-app-2.0.21.jar
    - tika-app-1.24.1.jar
3. Se pornește un terminal in P1 apoi se adaugă comanda de indexare a documentelor

```
java -Dfile.encoding=UTF-8 -classpath ".\out\production\P1;.\dependencies\lucene-core-8.6.3.jar;.\dependencies\tika-app-1.24.1.jar;.\dependencies\pdfbox-app-2.0.21.jar;.\dependencies\lucene-queryparser-8.6.3.jar;.\dependencies\lucene-analyzers-common-8.6.3.jar" com.main.Indexer ".\docs"
```

4. După indexare se poate caută diverse informații cu comanda. Proiectul a fost rulat cu Java 15.

```
java -Dfile.encoding=UTF-8 -classpath ".\out\production\P1;.\dependencies\lucene-core-8.6.3.jar;.\dependencies\tika-app-1.24.1.jar;.\dependencies\pdfbox-app-2.0.21.jar;.\dependencies\lucene-queryparser-8.6.3.jar;.\dependencies\lucene-analyzers-common-8.6.3.jar" com.main.Searcher "de modificat"
```