Documentație proiect RIW

1. Parcurgere fișiere

Pentru parcurgerea unui director am utilizat două funcții: CheckIfText și getFile. În cadrul primei funcții am verificat dacă fișierele din cadrul directorului sunt fișiere text. În cadrul celei de a doua funcții am realizat parcurgerea efectivă din cadrul directorului dat ca argument funcției.

2. Extragere cuvinte dintr-un fișier text și aplicare algoritm Porter

Funcția primește un fișier text ca intrare și îl va parcurge linie cu linie. Prima etapa este aceea de a construi cuvântul literă cu literă până întâlnește spațiu. În mometul în care s-a format un cuvânt acesta va fi trecut printr-o listă de stopword-uri urmând ca apoi să i se aplice algoritmul Porter prin intermediul funcției *stripAffixes*. Această funcție are 2 ramuri: *stripSuffixes* și *stripPrefixes*. Prima funcție încadrează toți pașii algoritmului Porter, pași ce cuprind diverse sufixe și reguli ale acestora. Cea de-a doua funcție își propune eliminarea prefixelor.

În cea de a doua etapă pentru fiecare cuvânt se va contoriza numărul de apariții. Rezultatul va fi stocat într-un HashMap<String,Integer>.

3. Indexarea directă și inversă

Pentru *indexarea directă* vom creea un HashMap<String,Map<String,Integer>>. Prima cheie este reprezentată de calea fișierului, iar în cadrul Map<String,Integer> prima cheie va fi cuvântul, iar valoarea numărul de apariții. Se va parcurge directorul introdus și în cadrul lui vom pune un fișier index.html în care va fi indexarea directă a directorului curent.

Pentru *indexarea inversă* vom creea un HashMap<String,Map<String,Integer>>. Vom parcurge indexarea directă. Prima cheie este reprezentată de cuvânt, iar în cadrul Map<String,Integer> prima cheie va fi calea, iar valoarea numărul de apariții.

4. Căutarea booleană

- în cadrul operației OR vom parcurge indexarea inversă. În cazul în care cuvintele ce le căutăm se regăsesc în HashMap-ul de indexare inversă vom adăuga calea lor în lista noastră. Această lista va fi rezultatul căutării OR.
- În cadrul operației AND, la fel, vom parcurge indexarea inversă. În cazul în care cuvintele ce le căutăm se regăsesc în HashMap-ul de indexare inversă și căile lor corespund, vom adăuga în lista noastră calea comună.

5. Utilizare MongoDB

- Clasa MongoDB -> conxiunea la baza de date.

- Clasa MongoSetup -> construcția bazei de date. În cadrul funcției de indexare vom instanția un obiect al clasei MongoSetup. Vom creea o baza de date numita RIW și două colecții, una numită directIndex iar cealaltă inversIndex.

  În cadrul colecției directIndex vom avea documente cu structura de următoare:

  - _id;
  - Fișier;
  - Un vector denumit Cuvinte ce va stoca documente de tipul cuvânt-apariții

```
> db.directIndex.find().pretty()
{
        "_id" : ObjectId("5ea1d7a2d9a76030dca1ac4a"),
        "Fisier" : "D:\\workspace\\labm1\\Folder1\\file1.txt",
        "Cuvinte" : [
                {
                        "cuvant" : "super",
                        "aparitii" : 2
                },
                {
                        "cuvant" : "text",
                        "aparitii" : 1
                },
                {
                        "cuvant" : "ro",
                        "aparitii" : 1
                },
                {
                        "cuvant" : "such",
                        "aparitii" : 1
                },
                {
                        "cuvant" : "ba",
                        "aparitii" : 1
                },
                {
                        "cuvant" : "cine",
                        "aparitii" : 1
                },
                {
                        "cuvant" : "hala",
                        "aparitii" : 1
                },
```

  În cadrul colecției inversIndex vom avea documente cu structura următoare:

  - _id;
  - Cuvânt;
  - Un vector denumit Indexare ce va cuprinde un document de tipul fișier-apariții

```
> db.inversIndex.find().pretty()
{
        "_id" : ObjectId("5ea1d7a2d9a76030dca1ac4f"),
        "cuvant" : "need",
        "indexare" : [
                {
                        "Fisier" : "D:\\workspace\\labm1\\Folder1\\folder2\\file3.txt",
                        "aparitii" : 1
                }
        ]
}
{
        "_id" : ObjectId("5ea1d7a2d9a76030dca1ac50"),
        "cuvant" : "store",
        "indexare" : [
                {
                        "Fisier" : "D:\\workspace\\labm1\\Folder1\\folder2\\file3.txt",
                        "aparitii" : 2
                }
        ]
}
```