

UNIVERSITATEA TEHNICĂ „Gheorghe Asachi” din IAȘI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DOMENIUL: Calculatoare și tehnologia informației
SPECIALIZAREA: Tehnologia informației

Proiect la disciplina Regăsirea informațiilor pe web

Student
Luchian Bogdan-Ionuț

Iași, 2020

1. Parcurgerea fișierelor

Parcurgerea unui director de fișiere se face prin utilizarea funcției `getFile`. Această funcție presupune parcurgerea directorului de fișiere dat ca argument funcției prin verificarea dacă conține sau nu a unui fișier de tip `txt`.

2. Clasa Porter.java

Clasa Porter este un pachet în care se regăsește algoritmul de procesare a cuvintelor, numit stemmer. Această clasă conține o funcția `stripAffixes` în care sunt apelate cele două funcții principale și anume `stripPrefixes` și `stripSuffixes`. Prima funcție verifică dacă cuvintele din fișierele noastre conține anumite prefixe, precum "kilo", "micro", "milli", "intra", "ultra", "mega", "nano", "pico", "pseudo", iar cea de-a doua verifică dacă cuvântul conține sufixe prin apelarea celor 5 funcții pași. Se folosește pentru a elimina sufixele și prefixele și de a aduce cuvântul la forma de bază. Această clasă se apelează în funcția de `TextSplit` din cadrul proiectul nostru.

3. Indexarea directă și inversă

Pentru indexarea directă vom crea 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, ce va conține calea, cuvântul și numărul de apariții din cadrul fiecărui fișier.

Pentru indexarea inversă vom crea 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

Această căutare se bazează pe criteriul deciziei binare și pe aritmetica mulțimilor. Termenii interogării (sau cheile de căutare) sunt combinate logic utilizând operatorii booleani AND, OR și/sau NOT.

Pașii care se vor aplica sunt următorii:

- verificăm dacă cuvintele căutate sunt diferite de null;
- verificăm dacă cuvintele există în fișiere;
- parcurgem `keySet`-ul `HashMap`-ul indexării inverse prin intermediul cheilor;
- verificăm dacă cheile noastre sunt egale cu cuvintele căutate;
- parcurgem din nou `HashMap`-ul indexării inverse folosind `get`;
- în funcție de operatori vom face:
 - OR – adăugăm cheile din ultima parcurgere în listă;
 - AND – dacă cele două chei sunt egale, adăugăm una din ele în listă;
 - NOT – dacă prima cheie este diferită de a doua, adăugăm prima cheie în listă.

5. MongoDB

MongoDB este o bază de date NoSQL open-source orientată pe documente. Acestă

bază de date beneficiază de suport din partea companiei 10gen. MongoDB face parte din familia de sistemelor de baze de date NoSQL. Diferența principală constă în faptul că stocarea datelor nu se face folosind tabele precum într-o bază de date relațională, MongoDB stochează datele sub formă de documente JSON cu scheme dinamice.

Pentru a folosi MongoDB am create 2 clase:

- MongoSetup.java – în această clasă se fac setările pentru a ne putea conecta la baza de date prin apelarea constructorilor;
- MongoDB.java – se inițializează o variabilă de tip MongoClient și una de tip DB la care se crează o nouă bază de date cu numele “RIW”.

Vom folosi mongodb pentru a stoca căile, cuvintele și numărul de apariții din cadrul indexărilor directă și inversă pe care le adăugăm în două colecții separate. Spre exemplu pentru cele din indexarea directă vom avea un camp cu numele “docs” în care se stochează calea și o listă/obiect (array) numit “temp” în care avem doi parametri: “t” pentru cuvânt și “c” numărul de apariții. În cazul indexării inverse vom avea în “docs” cuvântul, iar în parametru “t” din “temp” vom avea calea acelui cuvânt.

6. Mod de rulare

- Adăugăm calea unui director ce conține un set de fișiere de tip txt;
- Se generează fișierul de indexare inversă și se afișează pe monitor cuvintele și numărul de apariții și se generează pentru fiecare folder în parte un fișier index.html care conține indexarea direct;
- Se introduc două cuvinte de la tastatură;
- Pentru cele două cuvinte introduse se va afișa căutarea booleană prin intermediul operatorilor;
- În baza de date numită “RIW” se vor crea două colecții: “directIndex” și “inversIndex”.

Pentru configurarea proiectului s-au folosit următoarele jar-uri:

- jsoup-1.12.2.jar;
- mongo-java-driver.jar .

Nota de evaluare: 7.5p + 1p mongodb