UNIVERSITATEA TEHNICĂ „Gheorghe Asachi” din IAȘI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DOMENIUL: Calculatoare şi Tehnologia Informaţiei
SPECIALIZAREA: Tehnologia Informaţiei

Indexare și căutare

PROIECT LA DISCIPLINA
REGĂSIREA INFORMAȚIILOR PE WEB

Studentă: Frențescu Maria
Grupa: 1409A
Profesor îndrumător: Ș.l. dr. ing. Alexandru Archip

Iaşi, 2020

Cuprins

Capitolul 1. Prezentarea generală

Prima componentă a proiectului disciplinei „Regăsirea informaţiilor pe WEB” propune realizarea unui program pentru a obţine o procesare adecvată a cuvintelor din mai multe texte. Aplicaţia propusă a fost scrisă în Python iar interfaţa cu utilizatorul este prezentată în linia de comandă. Proiectul conţine un singur fişier sursă „*Proiect*.py”, suplimentar fiind nevoie de o structura de directoare şi fişiere ce conţin excepţiile şi stop-word-urile.

În cadrul proiectului am abordat următoarele cerinţe:

1. Crearea indexului direct
2. Crearea indexului indirect
3. Căutarea booleană
4. Conectarea şi lucrul cu baza de date MongoDB
5. Căutarea vectorială

1.1. Index Direct

Acest modul creează index-ul direct al tuturor fişierelor de tip *txt* găsite în structura de directoare şi subdirectoare existentă. Pentru a parcurge această structură în mod secvenţial se foloseşte o coadă pentru extragerea cuvintelor din fişiere. După extragerea cuvintelor, urmează procesarea acestora în 3 paşi. Iniţial, cuvintele sunt testate contra unei liste de excepţii, aceasta conţinând cuvinte ce nu se găsesc în dicţionar. În a doua fază, cuvintele se testează contra unei liste de stop-word-uri, aceasta conţinând cuvinte ce nu prezentă interes pentru relevanţa documentelor rezultate din căutare. În final, cuvintele sunt trecute printr-un proces de *stemming*, în care se elimină terminaţiile.

După toate aceste procesări, cuvintele astfel obţinute se stochează în fişiere de tip *.json*. Am creat 2 fişiere *.json*, unul în care sunt stocate cuvintele şi valorile asociate cuvintelor iar în altul sunt stocate căile de acces ale subdirectoarelor.

1.2. Index invers

Acest modul creează index-ul invers, după index-ul direct creat anterior. Am folosit fişierul .json ce conţine căile de acces ale directoarelor şi fiecărui cuvânt i se adaugă perechi de tipul <document, număr de apariţii>. Se creează şi pentru acesta un fişier *.json* în care se găsesc perechile formate de tipul <cuvânt <document, număr de apariţii>>

1.3. Căutarea booleană

Pentru a crea funcţia de căutare booleană trebuie creaţi înainte index-ul direct şi cel indirect şi memorarea fişierelor *.json* cu datele dobândite.

Funcţia de căutare booleană reprezintă o funcţie de căutare a unui sistem de regăsire de informaţii conform modelului boolean. Trebuie să permită operaţiile AND, OR şi NOT. Aplicaţia poate procesa interogări ce conţin 2 sau mai multe chei de căutare.

Iniţial se face citirea interogării, după care şi aceasta trebuie procesată aplicându-se aceleaşi reguli, şi anume eliminarea cuvintelor ce sunt excepţii, stop-word-uri şi modificarea acestora pentru a ajunge la forma de bază. Cuvintele se pun într-o listă de operanzi, iar operaţiile AND (&), OR (|) şi NOT (!) se pun într-o listă de operatori.

După procesarea datelor şi stocarea acestora în cele 2 liste se efectuează căutarea cuvintelor din lista de operanzi în *.json*-ul index-ului indirect. În caz ca aceste cuvinte sunt găsite, li se aplică operaţiile din lista de operatori. şi se afişează rezultatul.

1.4. Conectarea şi lucrul cu baza de date MongoDB

Pe lângă stocarea datelor în *.json* s-a încercat şi stocarea datelor într-o bază de date MongoDB (localhost, port 270017). Numele bazei de date folosite este *mydatabase* şi conţine o

singură colecție în care se găsesc datele în urma execuției index-ului direct.

1.5. Căutarea vectorială

S-a încercat rezolvarea problemei căutării vectoriale, programul conținând și 3 funcții pentru această problemă.

O primă funcție conține calculul a tf (term frequency) și a idf (inverse document frequency) în care s-au aplicat funcțiile din Figura 1.1:

- **P1** – frecvența de apariție a cuvântului k în cadrul documentului d (în eng.: *term frequency*)

$$tf(k,d) = \frac{count(k)}{|d|} \tag{2}$$

- **P2** – frecvența inversă de apariție a cuvântului k în cadrul mulțimii de documente D (în eng.: *inverse document frequency*)

$$idf(k) = log\frac{|D|}{1 + |\{d : k \in d\}|} \tag{3}$$

Figura 1.1: Formule calcul tf și idf

O altă funcție este pentru calculul vectorului asociat fiecărui element după formula din Figura 1.2:

Vectorul asociat fiecărui document va fi:

$$\vec{d} = \{key : tf(key,d) \cdot idf(key)\} \tag{4}$$

Figura 1.2: Formule calcul vector

A treia funcție este pentru calculul normei euclidiene pentru fiecare fișier în parte.

Capitolul 2. Exemple de execuţiei

Pentru interfața cu utilizatorul am creat un meniu uşor de utilizat.

```
D:\AN4\sem2\riw\RIW\Proiect_ETAPA1>python Proiect.py
----------Meniu:----------
1.Creati index direct
2.Creati index indirect
3.Cautare booleana
4.Conectare la baza de date
5.Adaugare index direct in mongo
-->Introduceti optiunea:
```

Figura 2.1: Meniul utilizatorului

Pentru opțiunea 1 se va crea indexul direct:

```
----------Meniu:----------
1.Creati index direct
2.Creati index indirect
3.Cautare booleana
4.Conectare la baza de date
5.Adaugare index direct in mongo
-->Introduceti optiunea: 1
-S-au creat cele 2 json-uri cu Indexul Direct
Doriti alta optiune? (1=da 2=nu)
```

Figura 2.2: Creare index direct

Pentru opțiunea 2 se va crea indexul indirect:

```
----------Meniu:----------
1.Creati index direct
2.Creati index indirect
3.Cautare booleana
4.Conectare la baza de date
5.Adaugare index direct in mongo
-->Introduceti optiunea: 2
-S-au creat cele 2 json-uri cu Indexul Direct
--S-a creat json-ul pentru Indexul Indirect
Doriti alta optiune? (1=da 2=nu)
```

Figura 2.3: Creare index indirect

Pentru opțiunea 3 se va efectua operația de căutare booleană:

```
----------Meniu:----------
1.Creati index direct
2.Creati index indirect
3.Cautare booleana
4.Conectare la baza de date
5.Adaugare index direct in mongo
-->Introduceti optiunea: 3
-S-au creat cele 2 json-uri cu Indexul Direct
--S-a creat json-ul pentru Indexul Indirect
->Introduceti interogarea pentru cautarea booleana: Game&throne
-->Rezultatul cautarii:
{'fisiere_lucru\\fisierul_2\\9', 'fisiere_lucru\\fisierul_2\\fisierul_5\\7', 'f
isiere_lucru\\12', 'fisiere_lucru\\fisierul_1\\6', 'fisiere_lucru\\fisierul_1\\
fisierul_3\\3', 'fisiere_lucru\\11', 'fisiere_lucru\\fisierul_1\\fisierul_3\\2'
, 'fisiere_lucru\\fisierul_2\\fisierul_5\\8'}
Doriti alta optiune? (1=da 2=nu)
```

Figura 2.4: Căutare booleană

Şi opţiunea 5 în care se crează conexiunea la baza de date şi adăugarea index-ului direct în colecţia *mydatabase*:

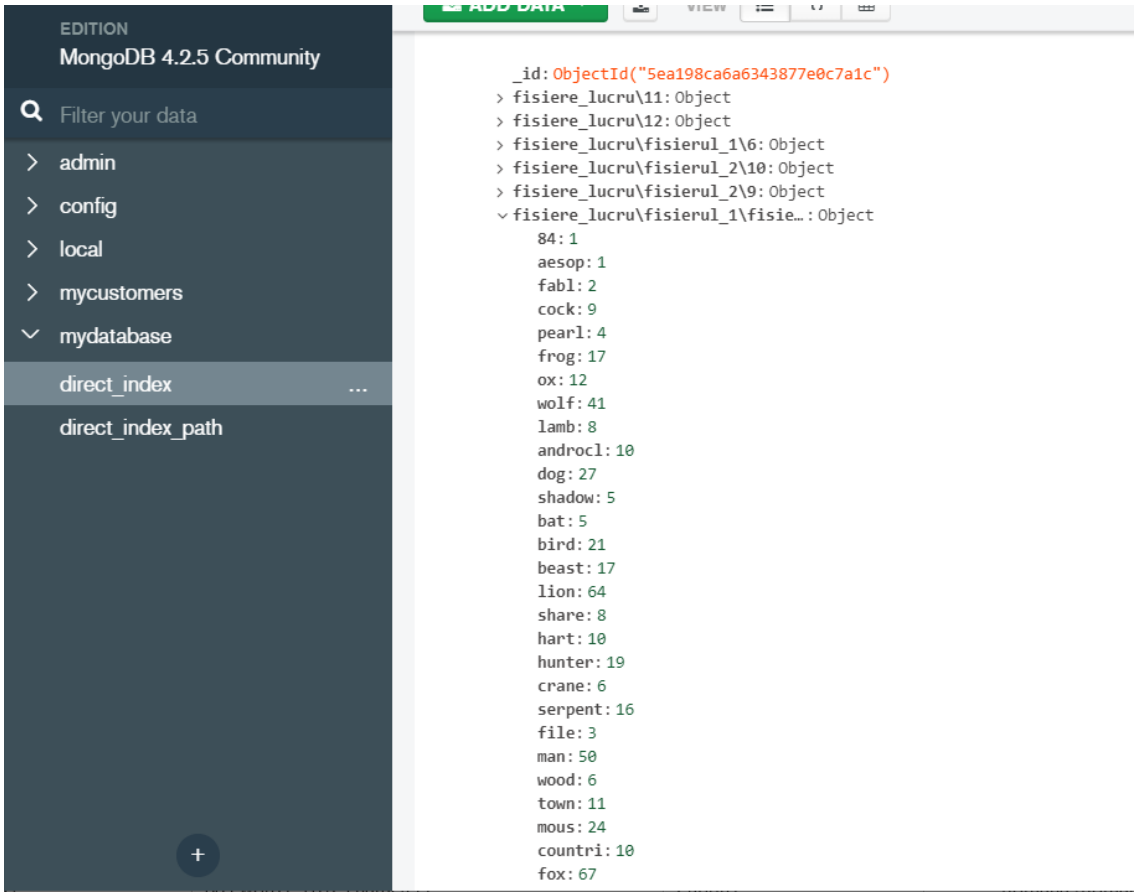

Figura 2.5: MongoDB