

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

Indexare și căutare

PROIECT LA DISCIPLINA
REGĂSIREA INFORMAȚIILOR PE WEB

Studentă: Frentescu Maria

Grupa: 1409A

Profesor îndrumător: Ș.l. dr. ing. Alexandru Archip

Iași, 2020

Cuprins

Capitolul 1. Prezentarea generală.....	II
1.1. Index Direct.....	II
1.2. Index invers.....	II
1.3. Căutarea booleană.....	II
1.4. Conectarea și lucrul cu baza de date MongoDB.....	II
1.5. Căutarea vectorială.....	III
Capitolul 2. Exemple de execuției.....	IV

Capitolul 1. Prezentarea generală

Prima componentă a proiectului disciplinei „Regăsirea informațiilor pe WEB” propune realizarea unui program pentru a obține o procesare adecvată a cuvintelor din mai multe texte. Aplicația propusă a fost scrisă în Python iar interfața cu utilizatorul este prezentată în linia de comandă. Proiectul conține un singur fișier sursă „*Proiect.py*”, suplimentar fiind nevoie de o structura de directoare și fișiere ce conțin excepțiile și stop-word-urile.

În cadrul proiectului am abordat următoarele cerințe:

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

1.1. Index Direct

Acest modul creează index-ul direct al tuturor fișierelor de tip *txt* găsite în structura de directoare și subdirectoare existentă. Pentru a parcurge această structură în mod secvențial se folosește o coadă pentru extragerea cuvintelor din fișiere. După extragerea cuvintelor, urmează procesarea acestora în 3 pași. Inițial, cuvintele sunt testate contra unei liste de excepții, aceasta conținând cuvinte ce nu se găsesc în dicționar. În a doua fază, cuvintele se testează contra unei liste de stop-word-uri, aceasta conținând cuvinte ce nu prezintă interes pentru relevanța documentelor rezultate din căutare. În final, cuvintele sunt trecute printr-un proces de *stemming*, în care se elimină terminațiile.

După toate aceste procesări, cuvintele astfel obținute se stochează în fișiere de tip *.json*. Am creat 2 fișiere *.json*, unul în care sunt stocate cuvintele și 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 fișierul *.json* ce conține căile de acces ale directoarelor și fiecărui cuvânt i se adaugă perechi de tipul <document, număr de apariții>. Se creează și pentru acesta un fișier *.json* în care se găsesc perechile formate de tipul <cuvânt <document, număr de apariții>>

1.3. Căutarea booleană

Pentru a crea funcția de căutare booleană trebuie creați înainte index-ul direct și cel indirect și memorarea fișierelor *.json* cu datele dobândite.

Funcția de căutare booleană reprezintă o funcție de căutare a unui sistem de regăsire de informații conform modelului boolean. Trebuie să permită operațiile AND, OR și NOT. Aplicația poate procesa interogări ce conțin 2 sau mai multe chei de căutare.

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

După procesarea datelor și 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 operațiile din lista de operatori. și se afișează rezultatul.

1.4. Conectarea și lucrul cu baza de date MongoDB

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

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

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

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

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:

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

Figura 1.2: Formule calcul vector

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

Capitolul 2. Exemple de execuției

Pentru interfața cu utilizatorul am creat un meniu ușor 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:

Auto-evaluare:

- Baza – **1 punct**
 - Parcurgerea directorului cu setul de fișiere de indexat, parcurgerea iterativă a unei structuri de directoare (inclusiv sub-directoarele) **0.5 puncte**
 - Preprocesare aplicarea unui algoritm de stemming pentru determinarea formelor canonice ale cuvintelor determinate **2 puncte**
 - Indexarea indirectă – **3.5 puncte**
 - Căutarea în indecșii determinați preîncărcați doar căutarea booleană **1.5 puncte**
 - Bonus: Stocarea indecșilor în colecții de documente de tip MongoDB **1 punct**
- Total: 8.5 + 1(bonus mongo)=9.5**