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

Proiect RIW

Student
Andrișan Ionuț-Cosmin

Iași, 2020

1. Cerința proiectului

Implementați o aplicație care să realizeze funcția de căutare a unui sistem de regăsire a informațiilor dintr-un set de documente (txt, html).

Scenariul de test pentru proicet va fi următorul:

- aplicația va primi ca mesaj de intrare numele unui director ce conține un set de fisiere tip txt;
- acest director va trebui parcurs, recursiv, pentru a identifica acele fisire txt și pentru a determina cele două forme de indexare indicate;
- modul de căutare ar trebui expus printr-o pagina HTML simplă (dar nu este obligatoriu – puteți opta pentru a dezvolta orice alta forma de interfață, inclusiv una de tip command line);

Încercați să realizați o implementare paralelă/distribuită pentru componentele ce implementează modulele de intrare și/sau căutare. Ca principal model de lucru, puteți porni de la modelul Map & Reduce.

Încercați să identificați și să analizați și alți algoritmi de stemming.

2. Etapele căutării

2.1. Construirea index-ului direct cantitativ

Aplicația primește ca intrare un director pe care îl parcurgem pentru a identifica fișierele pentru prelucrare. Directorul este parcurs astfel încât să fie identificate și fișierele din subdirectoare (în cazul în care în director avem subdirectoare).

Apoi, lista de fișiere obținută se sparge în cuvinte (pe care le stocăm într-un HashMap), contorizând în același timp și numărul de apariții a fiecarui cuvânt, astfel:

1) cuvântul din text, care a fost determinat în cadrul iterației curente, va fi testat contra unei liste de excepții – un dicționar al unei limbi nu conține, de exemplu, nume proprii; daca acest cuvânt se regăsește în lista de excepții, atunci se va trece la urmatoarele etape de procesare (contorizare număr de apariții, etc.);
2) cuvântul din text, care a fost determinat în cadrul iterației curente, va fi testat contra unei liste de stop-word-uri – cuvintele de legatură care în mod uzual nu aduc informații noi pentru motoarele de cautare; dacă acest cuvânt curent determinat se regasește într-o astfel de listă de stop-word-uri, atunci acesta va fi eliminat din procesarile ulterioare;
3) cuvântului din text, care a fost determinat în cadrul iterației curente, i se va aplica un algoritm de stemming (algoritmul lui Porter) pentru a se ajunge la o forma de bază a cuvântului, numită și formă canonică.

Stocarea indexului direct cantitativ se face astfel: se stochează cheia (documentele indexate), cât și valorile asociate (cuvintele din cadrul documentelor și numărul de aparitii a fiecaruia) cheilor. Acest index se salvează atât într-un fișier txt, cât și într-o bază de date non-relațională (MongoDB).

2.2. Construirea index-ului indirect cantitativ

Index-ul indirect cantitativ se obține pe baza index-ului direct cantitativ și este de forma: cheia (cuvântul indexat) și documentele în care găsim aceasta cheie (aceasta reprezintă fișierele în care se găsește cuvântul, cât și numărul de apariții a acelui cuvânt în documentul respectiv). De asemenea, acest index se salvează atât într-un fișier txt, cât și într-o bază de date non-relațională (MongoDB).

3. Modelul de căutare booleană

Reprezentarea interogării – termenii interogării (sau cheile de căautare) sunt combinate logic utilizând operatorii booleeni AND, OR și/sau NOT.

Regăsirea documentului – se bazează criteriul deciziei binare și pe aritmetica mulțimilor.

Avantaje:

- Este un model de cautare simplu, cu un formalism bine pus la punct, neambiguu.
- Poate fi implementat ușor și poate raspunde rapid pentru interogările uzuale ale utilizatorilor.

De zavantaje:

- Datorită simplității, este un model foarte rigid.
- Interogarile complexe nu pot fi realizate direct.
- Nu poate fi controlata cu exactitate dimensiunea exactă a răspunsului.
- Nu ofera un mecanism direct de feedback din partea utilizatorilor.

Principalii pași implicați:

1. se citește interogarea utilizator;
2. se izolează operanzii (cuvintele) de operatori;
3. cuvintele se proceseaza conform modelului utilizat în construirea index-ului corespunzător;
4. se izolează pe baza index-ului invers, pentru fiecare cuvânt în parte lista de documente ce conțin termenul respectiv;
5. se realizeaza, rând pe rând, operațiile indicate **AND**, **OR** și/sau **NOT**:
   - **AND** echivalează cu intersecția a două mulțimi;
   - **OR** echivaleaza cu reuniunea a două mulțimi;
   - **NOT** echivaleaza cu diferența dintre doua mulțimi;
6. rezultatul obținut este prezentat utilizatorului.

4. Procesarea cuvintelor – algoritmul lui Porter (limba engleza)

Stemmingul lui Porter este un algoritm care este folosit pentru aducerea cuvintelor din limba engleză într-o fromă canonică. Algoritmul este văzut ca un algoritm „înghețat”, care este definit strict, nu poate fi modificat și produce un număr mare de erori.

De multe ori, algoritmul lui Porter pentru stemming nu aduce cuvântul la forma canonica, ci la o forma greșită. Totoușsi, motivul stemmer-ului este de a aduce o serie de cuvinte la aceeași formă.