Regăsirea informațiilor pe web: indexare & căutare

Gabriel Răileanu
1409A

aprilie 2020

1 Descrierea problemei

Proiectul de față își propune indexarea unui set de documente și implementarea unei interfețe care să permită utilizatorului căutarea de termeni în colecția de intrare. Pe parcursul proiectului au fost urmărite următoarele scopuri:

- obținerea unei procesări adecvate a cuvintelor determinate în cadrul unui text
- studiul bazelor de date non-relaționale și a diferitelor mecanisme de stocare
- implementarea de metode folosite în operațiile de căutare

Pentru a executa o căutare a noțiunilor ce prezintă interes, setul de documente a necesitat procesări anterioare, la finalul cărora a avut loc operația de căutare. Etape:

1. filtrarea termenilor de interes din colecția inițială
2. aducerea cuvintelor la forma normală
3. calcul index direct cantitativ & index invers cantitativ
4. căutarea efectivă

2 Soluția propusă

2.1 Filtrarea termenilor

Pentru filtrarea cuvintelor s-au folosit liste de excepții și de *stopword*-uri specifice limbii engleze.

2.2 Forma normală

Pentru a ajunge la forma normală a cuvintelor, s-au avut în vedere doi algoritmi specifici: Porter și Lovins. După o analiză comparativă a celor doi algoritmi, s-a folosit o implementare bazată pe algoritmul lui Lovins. Acesta este mai rapid, făcând un compromis pentru spațiul necesar. În linii mari, algoritmul folosește seturi de sufixe, aplicând pe cuvintele de bază reguli (conține 294 de sufixe, 29 de condiții și 35 reguli de transformare).

2.3 Indecși

Pentru a ajunge la scopul final al proiectului (realizarea de operații de căutare pe colecția inițială), s-a realizat 2 tipuri de indecși:

- index direct cantitativ
- index invers cantitativ

Fiecare astfel de index este persistat într-o bază de date non-relațională (în acest caz alegându-se o bază de date orientată document: MongoDB).

2.3.1 Index direct cantitativ

Fiecare document din colecția de indexat este parcurs caracter cu caracter. Caracterele luate în considerare pentru contorizat sunt alcătuite fie din litere, fie din cifre. Astfel, la sfârșitul procesului de indexare directă, pentru fiecare document, în baza de date vom avea persistate un număr de intrări egal cu numărul de documente din colecția de indexat, documentele de forma:

$$(k, \{w : n\}) \longrightarrow < (w_1, v_1), (w_2, v_2), \ldots, (w_n, v_n) >$$

k: numele documentului $w_{1..n}$: cuvânt
v: nr. apariții n: nr. cuvinte din document

Pentru varianta **paralelă** a implementării, s-a folosit un *pool* (bazin) de workeri, thread-uri ale procesului master. Procesul master trimite către workeri câte un fișier, pe care acesta îl indexează. La finalul procesului, dacă mai există documente care nu au fost indexate, worker-ul mai primește un fișier pentru indexare.

2.3.2 Index invers cantitativ

În implementarea indexului invers cantitativ s-au folosit rezultatele indexului direct cantitativ, concatenându-se rezultatele. Indexul invers cantitativ are forma:

$$(w, \{doc : n\}) \longrightarrow < (w, [\{doc_1, n_1\}, \{doc_2, n_2\}, \ldots, \{doc_m, n_m\}]) >$$

$w_{1..n}$: cuvânt doc_i: numele documentului
n_i: nr. apariții m: nr. documente

Pentru varianta **paralelă** a implementării, fiecare worker va primi un document (a cărui index direct este deja creat în etapa precedentă) și va calcula indexul invers, pe care îl persistă în baza de date cu o funcționalitate de *update or set* (upsert). Astfel, se evită *data race*-ul și concurența asupra aceleeași intrări în baza de date.

În cazul în care în indexul invers apar mai multe intrări pentru aceeași cheie (aici, cuvântul de căutat), căutarea se efectuează tot cu o complexitate redusă, datorită indexului pe text pe care îl pune la dispoziție MongoDB. Astfel, problema se reduce la regăsirea tuturor intrărilor cu acea cheie, lucru care nu presupune decât adăugarea de logică în aplicație.

2.4 Căutarea

2.4.1 Căutarea booleană

Căutarea booleană returnează adevărat sau fals, în funcție de găsirea sau nu a termenului într-o colecție de documente.

Se pot forma, astfel și propoziții pentru căutarea cuvintelor care apar împreună sau a unor grupuri de termeni în care să apară cel puțin unul sau în care să nu apară niciunul dintre termenii căutați.

2.4.2 Căutarea vectorială

Căutarea vectorială își propune inducerea unei ordini în rezultatele oferite de căutarea booleană. În implementarea de față s-a utilizat o distanță de tip cosinus.

Astfel a fost calculat vectorul asociat *query*-ului introdus de utilizator, apoi vectorii asociați fiecărui document. Distanța cosinus a fost calculată pentru fiecare pereche $< query, document >$, iar apoi documentele ce constituie rezultatul căutării au fost returnate în ordinea dată de distanța cosinus.

3 Concluzii

Proiectul a prezentat principalele componente ale unui motor de căutare și etapele prin care un text este procesat pentru a realiza o căutare a cunoștințelor optimă.

4 Bibliografie

[1] Regăsirea informațiilor pe web: notițe de curs.

[2] Regăsirea informațiilor pe web: lucrări de laborator.