Căutarea paginilor WEB

Cristian Damian,
Coordonator: Ștefan Stăncescu

8 februarie 2016

Program Master IISC an II,
Facultatea de Electronică Telecomunicații și Tehnologia Informației,
Universitatea „Politehnica” București

Cuprins

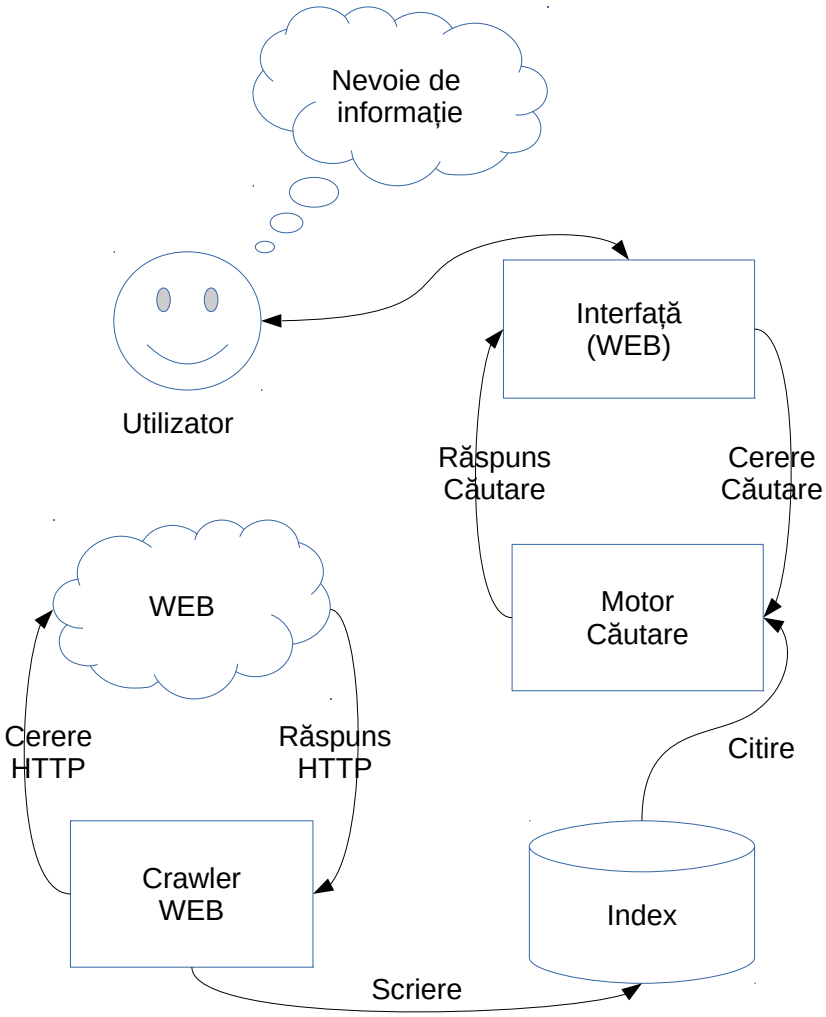
1 Structura unui motor de căutare web

WEB-ul este un mediu de comunicare fără precedent în special prin volumul de informații care îl poate oferii dar prin lipsa de coordonare a conținutului, participanții săi având tot spectrul de motivații și stiluri de exprimare. Tocmai din cauza acestor caracteristici el ar fi inutil fără un mijloc automat de căutare eficient. Majoritatea utilizatorilor se așteaptă să introducă câteva cuvinte într-un câmp și să obțină paginile cu site-urile dorite într-o secundă însă efortul necesar pentru a întreține un motor de căutare este considerabil și proiectarea unor astfel de sisteme constituie o disciplină întreagă.

Scopul acestei lucrări este să se realizeze o scurtă introducere în disciplina sistemelor de căutare punându-se accent pe algoritmi că căutare și ordonare a rezultatelor (ranking) spre deosebire de metodele de explorare a site-urilor web (crawing).

În primul rând, însă, vom discuta de structura generală a unui motor de căutare. Schema de principiu este ilustrată în figura 1.1. O căutare tipică implică un utilizator care are o „nevoie de informație" după cum se spune în domeniul de specialitate. Utilizatorul își exprimă nevoia formulând o cerere de căutare folosind o interfață, de obicei o pagină web dinamică. Cererea (sau interogarea) are de regulă forma unei liste de termeni dar poate avea altă formă (vedeți secțiunea 3.3). Motorul de căutare citește un index, adună paginile ce se potrivesc cu cererea, le ordonează conform unui scor numit ranking generează un răspuns. Răspunsul este afișat apoi de interfață într-o formă inteligibilă de utilizator. Între timp, index-ul este actualizat de către un crawler web care explorează web-ul. Crawler-ul analizează paginile web pentru indexare și urmează link-urile de pe pagini pentru a descoperi pagini noi.

Figura 1.1: Arhitectura unui motor de căutare web.



2 Particularitățile web-ului

Caracteristica esențială care a dus la creșterea explozivă a Web este descentralizarea conținutului, publicarea fără nici un control central al autorului. Aceasta sa dovedit a fi cea mai mare provocare pentru motoarele de căutare web în încercarea lor de a indexa și a prelua acest conținut. Autorii paginilor web au creat conținut în sute de limbi (naturale) și mii de dialecte, solicitând astfel multe diferite forme de operațiuni lingvistice.

Pentru ca publicarea a fost deschisă acum la zeci de milioane, paginile web au prezentat eterogenitate la o scară descurajatoare, în multe aspecte cruciale. În primul rând, conținutul de creație nu mai era rezervat de scriitori editorial pregătiți; în timp ce acest lucru a reprezentat o imensă democratizare, de asemenea, a dus la o variație extraordinară în gramatică și stil (și în multe cazuri, nu gramatică recunoscut sau stil). Unele pagini web, inclusiv paginile create profesional ale unor mari corporații, a constat în întregime din imagini (care, când faceți clic pe, a condus la un conținut mai bogat textual) - Și, prin urmare, nici un text indexabil.

În schimb paginile web au adus ceva nou în disciplina căutării, link-urile. Web-ul constă într-o colecție de documente legate între ele prin link-uri. Acesta poate fi văzut ca un graf direcționat unde documentele sunt noduri și link-urile sunt arce direcționate în sensul parcurgerii linkurilor. Arcele care se duc spre un nod se numesc in-link-urile lui și arcele care ies dintr-un nod se numesc out-link-urile acestuia. Link-urile au de asemenea și un text asociat ce poate fi folosit pentru caracterizarea unui document prin textul asociat in-link-urilor sale.

Graful web-ului nu este puternic conectat și link-urile nu sunt distribuite aleator. Conectivitatea unei pagini poate fi exploatată pentru a genera un ranking ca în secțiunea 4.1. De mute ori se folosește doar un mic subset al web-ului pentru a se studia graful web. Un exemplu de graf este ilustrat în figura 2.1.

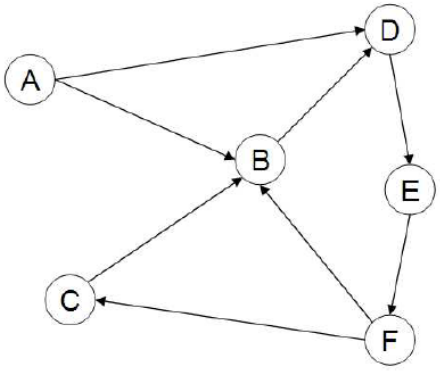
De asemenea, utilizatorii motoarelor de căutare web au anumite caracteristici deosebite. Ei formulează cereri scurte de câteva termeni și cele mai multe au doar unul dau doi termeni, media este undeva între 2 și 3 termeni. Temele cererilor acoperă tot spectrul intereselor umane dar se înclină spre sănătate, comerț și distracție.

Distribuția cererilor urmează o distribuție Zipf însemnând că probabilitatea de emitere a unei cereri urmează următoarea lege:

$$P(q) \propto \frac{1}{k^{\alpha}}$$

unde q este cererea k este rangul cererii ca probabilitate și α este in jurul valorii 1. Cea mai frecventă cerere se realizează până la 1% din cazuri când jumatate din cereri se realizează doar o dată. Asta înseamnă că totuși căutările puțin fecvente sunt totuși destul de importante.

Figura 2.1: Exemplu de graf web [Büttcher-2010].



Cererile se pot clasifica după intenția utilizatorilor astfel:

Cereri navigaționale Intenția utilizatorilor este de a localiza un site anumede pe Web. Ca exemplu un utilizator va scrie cererea „CNN" pentru a localiza site-ul de știri „www.cnn.com". Site-ul poate să difere însă de la utilizator la utilizator. Un utilizator ce știe spaniola ar putea căuta versiunea site-ului in spaniolă de enemplu.

Cereri informaționale Intenția utilizatorilor este de afla ceva despre un subiect dar fără să fie interesat de sursa informației. Utilizatorul poate căuta informații în general cum ar fi o biografie sau poate vrea să afle doar răspunsul la o întrebare (de exemplu când s-au unit principatele române)

Cereri tranzacționale Utilizatorul are intenția de a interacționa cu un site o dată ce îl găsește. Această interacție poate fi o altă căutare, jocuri sau tranzacții comerciale. Exemple de astfel de cereri sunt „meteo", „mașini rulate", „hărți".

Trebuie avut totuși în vedere că intenția poate să difere de la utilizator la utilizator pentru aceeași cerere. De exemplu dacă cererea este „UPS" utilizatorul poate fie să aibă o intenție informativă vrând să știe cum funcționează o sursă sigură de alimentare („uninteruptiblie power supply"), fie o inenție tranzacțională vrând să comande o astfel de unitate, fie o intenție navigațională vrând să intre pe site-ul firmei de curierat UPS sau a Universității Puget Sound.

Se pot afla multe lucruri despre intenția utilizatorului urmărind feedback-ul implicit al acestuia. Cea mai comună metodă este monitorizarea clicuri-lor utilizatorilor prin curbele clickthough. Curbele clickthrough sunt reprezentarea grafică a numărului de click-uri pe un rezultat în funcție de poziția rezultatului returnat pentru o cerere anume. Click-ul utilizatorului este o indicație că utilizatorul crede că un site este relevant pentru cerere.

Dacă au loc inversiuni, o pozitie mai mare are un număr de click-uri mai mic decît o poziție mai mică, aceasta poate indica că ranking-ul nu este corect și se poate îmbunătăți.

Dacă clicurile se adună la mai puține poziții se poate deduce că cererea este de tip navigațional. Dacă, în schimb, clicurile se adună relativ egal pe multe poziții se poate presupune că cererea este de natură informațională.

3 Indexarea

Cum am zis înainte indexarea înseamnă memorarea colecției de pagini web într-o formă ce permite căutarea rapidă după termenii cererii de căutare.

Abordarea naivă este să memorezi tot conținutul și la căutare să îl parcurgi secvențial listând pozițiile unde literele din colecție se aliniază cu literele din cerere. Această abordare se mai numește „grepping" de la comanda Unix „grep", care permite și mai multe opțiuni prin expresii regulate, și este cea mai eficientă pentru corpuri mici de date. Editoarele de text și de documente folosesc în mare parte numai acestă metodă. Problema este că volumul de date de pe web este mult prea mare pentru o astfel de abordare. Astfel conținutul trebuie indexat.

Indexarea are două etape. În prima etapă, textul este procesat pentru a afla care sunt termenii din interiorul lui. În a doua etapă se construiește indexul ce va fi căutat pentru a genera răspunsul numit în literatură „indexul invers" sau „fișierul inversat".

3.1 Procesarea textului

Prima problemă din procesarea textului este cum se definește un termen. În vocabularul de specialitate se definește un token ca o instanță a unui un grup de caractere cu sens, un tip de tokenuri se numește termen. Astfel termeni sun păstrați în index și token-urile sunt instanțele corespunzătoare din colecția de pagini.

Procesarea textului are patru etape: extragerea textului, tokenizarea (separarea textului în tokenuri), procesarea lingvistică a tokenurilor, și indexarea lor la termenul corespunzător. Primele trei etape de procesare a textului se fac atât pentru documentele HTML cât și pentru cererea de căutare.

Extragerea textului dintr-o pagină web se poate face eliminând mark-up-ul HTML și codul dintr-un document. Trebuie reținut însă că textul asociat unei pagini web nu se află numai în interiorul paginii. Link-urile care duc spre pagina respectivă conțin de obicei un text asociat, „anchor text". Acest text asociat poate să fie char mai important pentru indexarea paginii decât textul din pagină deoarece poate conține un sumar sau cuvinte cheie legate de subiectul paginii.

Tokenizarea diferă de la limbă la limbă. Tokenurile corespund de obicei cu cuvintele din limba folosită dar există și alte secvențe de caractere ce ar trebui considerate ca tokene unice cum ar fi abrevieri canonizate ca „I.B.M." sau „UPB", nume compuse ca „San Francisco", nume ce conțin apostroful ca „O'Neil" și „Mc'Donalds" sau alte simboluri ca „C++" sau „M*A*S*H". De asemenea, trebuie separate adrese URL (http://stuff.big.com/new/specials.html), numere de telefon „(+40) 21 555 44" sau coduri alfanumerice. În anumite limbi unde compunerea cuvintelor este des practicată ca

germana sau în care cuvintele nu se separă prin spații ca chineza procesul de tokenizare poate fi foarte complex.

O metodă de a ocoli complexitatea legată de tokeniarea unei limbi este indexarea n-grame-lor adică tuturor grupurilor de n litere dintr-un text. Astfel dacă folosim 5-grame o secvență ca „motor de căutare" ar fi despărțită în „motor"„otor "„tor d" și așa mai departe. Această metodă este ineficientă din cauza volumului de termeni ce trebuie indexați dar este uneori preferabilă unei tokenizări defectuoase.

Metoda naivă de clasificare a tokenurilor în termeni este ca fiecare secventă diferită de litere a unui tokenuri-lor să fie clasificată ca un termen diferit însă acestă metodă este ineficientă și din punct de vedere al numărului de termeni cât și a capacității de căutare. Pentru reducerea termenilor se face o procesare lingvistică a tokenurilor numită normalizare. In acest fel token-urile ce diferă puțin prin formă sunt mapate la același termen. De exemplu, pentru un text în engleză „run"„runs"„running" sunt mapate la termenul „run" deoarece sunt forme diferite ale aceluiași verb. De asemenea, cuvinte cu ortografii diferte pot fi mapate la același termen cum ar fi „anti-democratic" și „antidemocratic". Două tipuri de normalizare se folosesc în literatură „lematizare" și „stemming". Diferența între cele două este că lematiarea face o analiză morfologică riguroasă pentru a aduce cuvintele la formele lor din dicționar, în timp ce prin „stemming" se folosesc euristici simple pentru a reduce tokenurile la cuvinte rădăcină. Este clar că lematizarea este mult mai complexă dar poate da rezultate mai bune în anumite situații.

Uni termeni apar foarte frecvent și în toate textele. Acești termeni se numesc termeni stop, exemple de astfel de cuvinte sunt în engleză „the", „is", „a", „in", etc. În mod tradițional acești termeni nu se indexau deoarece câștigul în performață oferit prin căutarea acestor termeni este foarte mic. Acești termeni trebuie indexați însă în căutarea paginilor Web deoarece temele căutate sunt foarte diverse și cererile de căutare formulate de utilizatori reprezintă de multe ori fraze. Ca exemplu un utilizator ar putea căuta informații despre cunoscuta poezie a lui Edgar Allan Poe scriind titlul „The raven" sau ar putea cauta trupa britanică „The The".

3.2 Indexul invers

Indexul invers este o structură de date ce constă (în principal) dintr-un dicționar de termeni și o listă a aparițiilor în colecția de pagini (in engleză „postings"). Dacă indexul este nepozițional atunci lista conține documentele în care apare termenul și dacă index-ul este pozițional se precizează documentul și poziția exactă. Structura seamănă cu index-ul dintr-o carte. Index-ul poate conține și numărul de documente în care apare termenul și de câte ori apare termenul într-un document. Lucruri ce pot ajuta la ranking.

Implementarea unui index invers este o problemă complexă de informatică ce implică modalități dezvoltarea de algoritmi de creare și citire a indexului eficienți atât din punct de vedere al timpului consumat cât și a gestiunii memoriei. Dicționarul de termeni este de obicei are de obicei termeni în ordine alfabetică iar listele de aparitii sunt aranjate într-o ordine consecventă pentru a ușura operațiile de căutare.

O dată construit index-ul invers se poate considera ca un tip de date abstract cu două

metode:

next(term,current) Returnează prima poziție în care apare termenul „term" după poziția „current".

prev(term,current) Returnează ultima poziție în care apare termenul „term" înainte de poziția „current".

Pentru poziții definim și valorile speciale de început „START" și de sfârșit „END". Metodele returnează „END" respectiv „START" atunci când nu mai găsesc poziții. Valorile pozițiilor pot fi codate în așa fel încât să se țină cont de documentul în care se află tokenul și chiar de structura documentului HTML. Un exemplu de codare ar fi identificatorul de document urmat de identificatorul paragrafului și numărul tokenului în paragraf.

Astfel, realizarea unei căutări în care să listăm locurile unde apare termenul este trivială. Apelăm iterativ metoda „next" reținând poziția curentă până când se returnează „END". Aceasta este baza căutărilor booleene descrise în următoarea secțiune (3.3).

3.3 Căutarea booleană

Căutarea booleană este o căutare în care se returnează toate documentele ce îndeplinesc o condiție indiferent de ordine. Exemplul de căutare din secțiune precedentă (3.2) este cel mai simplu tip de căutare booleană. Cererile căutărilor booleene folosesc fie explicit sau implicit operatori. Operatori de bază sunt operatori logici AND, OR, NOT. Un exemplu de cerere ar fi următorul:

```
(Brutus OR Caesar) AND NOT Calpurnia
```

Un mod în care s-ar procesa acestă cerere este să se obțină listele de documente pentru fiecare „Brutus" și „Caesar" și din rezultat să se excludă documentele care conțin „Calpurnia". Se pot face optimizări modului de căutare. De exemplu, o metoda de a se optimiza căutările cu de tipul:

```
Brutus AND Caesar AND Calpurnia
```

este ca să se sorteze termeni în ordine crescătoare a numărului de apariții să se obțină lista pentru primul termen și să se excludă documentele ce nu conțin următorii termeni. Cererea de mai sus ar fi reformulată astfel:

```
(Calpurina AND (Brutus AND Calpurina))
```

Aceste căutări booleene pot fi extinse cu operatori de proximitate. Acești operatori permit de exemplu căutarea pentru fraze din mai mulți termeni specificând ca doi termeni să se găsească alăturați și în ordinea specificată. O modalitatea de a implementa o astfel de căutare este prezentată în figura 3.1. Se poate specifica ca anumiți termeni să fie în același paragraf sau chiar în aceeași propoziție.

Figura 3.1: Algoritmul de căutare a frazelor [Büttcher-2010].

```
function nextPhrase(terms[], position){
        n = terms.length();
        v = position;
        for(i=0;i<n;i++){
                v = next(terms[i],v);
        }
        if(v == END){return END;}
        u = v;
        for(i=n-1;i>=0;i--){
                u = prev(terms[i],u);
        }
        if(v-u == n-1){return u}
        else {nextPhrase(terms,u);}
}
```

4 Ranking

Ranking-ul este un scor atașat fiecărui rezultat pentru a ordona rezultate unei căutări. Ranking-ul se poate face fie în momentul indexării, caz în care se numește ranking static, fie în momentul căutării pentru o cerere, caz în care se numește ranking dinamic. Ranking-ul se poate folosi doar pentru ordona rezultatele unei căutări booleene sau se poate realiza pentru a putea exclude site-uri ce sunt de calitate slabă sau nerelevante.

Metodele de ranking se bazează în general pe Principiul Ranking-ului Probabilistic („Probability Ranking Principle"):

> Dacă un sistem de căutare răspunde la fiecare interogare cu o ordonare a colecției de documente în ordinea descendentă a probabilității relevanței, atunci se maximixează eficiența sistemului.

4.1 Ranking static

Ranking-ul static se face în momentul indexării și este independent de cereri. Acest ranking realizează operațiuni care ar dura prea mult să fie făcute în timpul cererii cum ar fi analiza grafului web dar se poate analiza și conținutul paginii web în moduri mai sofisticate.

De regulă acest tip de ranking dorește să afle cât de bună, de încredere sau cât de popular este un site web. În continuare vom prezenta două tipuri de analiză a grafului web ce realizează un ranking static.

4.1.1 PageRank

Algoritmul PageRank a fost inventat în ani '90 de Lary Page și Sergey Brin, viitori fondatori ai Google.

Intuiția clasică din spatele PageRank este următoarea. Se consideră o persoană care navighează Web-ul la întâmplare. La fiecare moment el are în față o pagină Web el poate:

- Fie să urmeze un link de pe pagina curentă.
- Fie să sară aleator la o pagină oarecare.

Probabilitatea de a alege un link din pagină este fixat ca δ , implicit probabilitatea se a sări aleator este $1 - \delta$. Valoarea dată lui δ este aleasă între 0.75 și 0.9. În exerciții se folosește 0.75 pentru simplitate. Paginile care nu au linkuri forțează un salt. PageRank $r(\alpha)$ este probabilitatea ca persoana să fie pe o pagină într-un moment oarecare.

Fiecărui link i se oferă o probabilitate de a fi urmat o dată ce persoana s-a hotărât să urmeze un link în funcție de poziția în pagină, textul asociat sau alte criterii. Pentru a

face un PageRank specific pentru o temă putem să modelăm o persoană interesată în acea temă care urmează link-urile cu termenii specifici.

Se poate recunoaște că acest model este echivalent cu un lanț Markov de gradul întâi. Starea următoare a lanțului depinde numai de starea actuală și se știe pentru fiecare stare probabilitatea de a tranzita la orice altă stare. Probabilitatea se poate calcula astfel:

$$P(\alpha|\beta) = \delta w(\alpha|\beta) + (1-\delta)\frac{1}{N}$$

unde N este numărul de site-uri din graf, $w(\alpha|\beta)$ este probabilitatea ca persoana să ajungă la site-ul α de la siteul β o dată ce s-a hotărît să urmeze un link. Se poate afla astfel PageRank prin analiza lanțului Markov.

Motivația din spatele PageRank este că site-urile cu un PageRank mare sunt mai importante sau demne de încredere deoarece sunt recomandate spre vizitare de multe site-uri care la rândul lor sunt recomandate de alte site-uri. Un exemplu de ranking PageRank este ilustrat în figura 4.1 . Se observă că site-ul B are un scor mare fiindca mute site-uri au link-uri catre el fie direct sau indirect. Site-ul C ape un scor mare chiar dacă nu sunt multe link-uri către el findcă singurul link al lui B duce spre C.

4.1.2 Centre și autorități

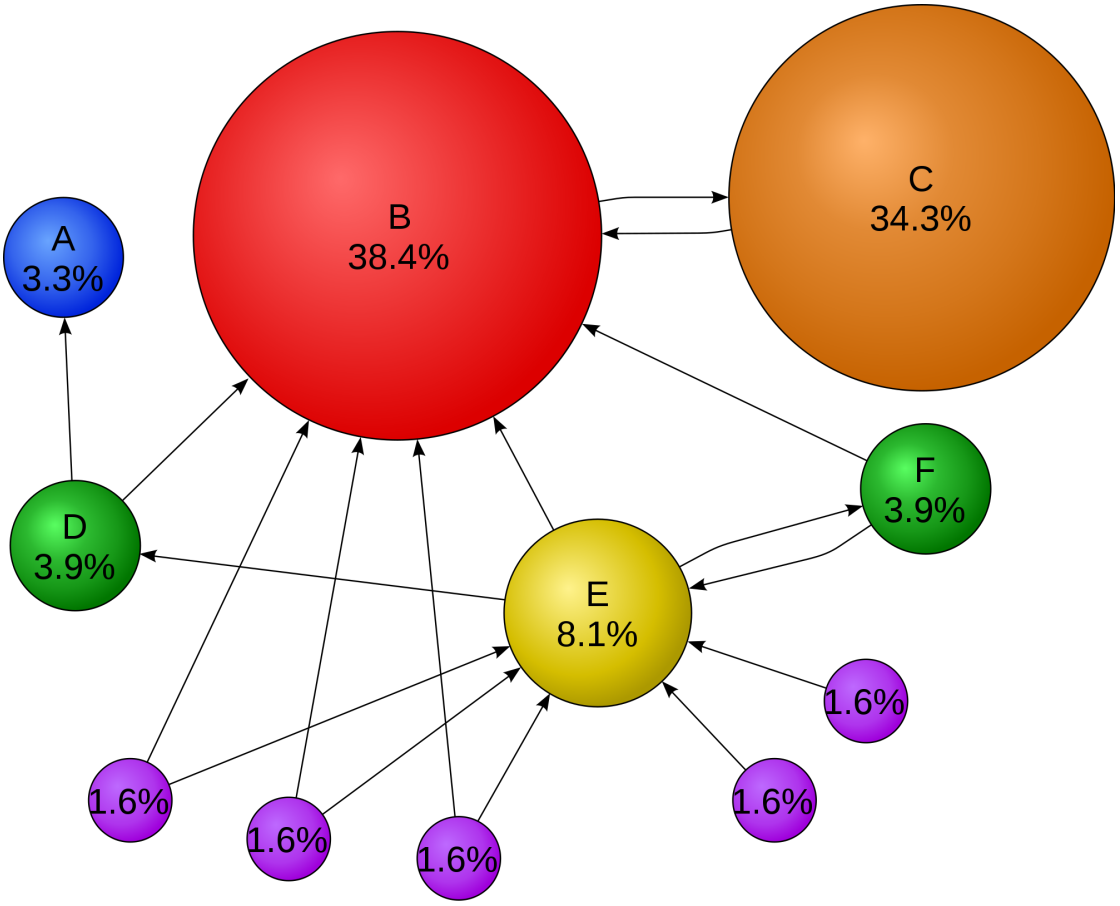
O altă metodă de analiză a linkuilor numită HITS (Hiperlink-Induced Topic Search) se bazează pe observația făcută la începutul web-ului conform căruia paginile de interes de pe web sunt fie centre (hubs) fie autorități (authorities). Autoritățile sunt site-uri de încredere care oferă informații sau servicii. Centrele sunt site-uri care oferă link-uri spre autorități fiind liste compilate de oameni de încredere. Centrele bune oferă link-uri spre autorități bune și autoritățile bune apar in centrele bune. Definiția circulară impune o rezolvare iterativă astfel se definește următoarea iterație:

$$h(\alpha) \leftarrow \sum_{\alpha\to\gamma} a(\beta)$$
$$a(\beta) \leftarrow \sum_{\alpha\to\beta} h(\alpha)$$

unde $h(\alpha)$este scorul de centru al lui α și $a(\beta)$ este scorul de autoritate al lui β și denotă că α are un link spre β. Prin iterarea se ajunge la un scor stabil atât pentru calitatea de hub cât și pentru calitatea de autoritate.

Acestă metodă este utilă îndeosebi pentru găsirea autorităților pentru cereri informative. Autoritățile pot să nu conțină de foarte multe ori termenii căutării. De exemplu site-ul IBM este o autoritate pe tema calculatoarelor dar nu are o frecvență foarte mare pentru termenul calculator. Dar s-a observat că centrele conțin de multe ori termeni ce definesc o temă astfel se face o primă căutare pentru a găsi centrele se urmează linkurile site-urilor pentru a găsi eventualele autorități și se realizează analiza HITS pe subsetul Web-ului. Analiza este prea complicată pentru a se face în timpul căutării dar se poate face în timpu indexării pentru un număr limitat de termeni.

Figura 4.1: Exemplu de ranking PageRank pentru un graf oarecare [Wikipedia-2016].



4.2 Ranking dinamic

Ranking-ul dinamic se realizează în timpul căutării și trebuie să afle cât de relevante sunt site-urile pentru cererea curentă. Asta însemnă generarea unor scoruri proprii și combinarea lor cu scorurile făcute în ranking-ul static pentru a ordona rezultatele la final.

Căutările booleene simple sunt aproape inutile în căutarea web deoarece corpul de documente este atât de mare încât cel mai probabil se vor returna mii de rezultate. Astfel ranking-ul și îndeosebi ranking-ul dinamic este esențial pentru căutarea pe Web.

În special metoda de combinare a scorurilor diferă mult între motoarele de căutare. În secțiunea acesta voi prezenta metode de ranking dinamic bine cunoscute în literatură.

4.2.1 Ponderarea în funcție de zonă

Reamintim că indexul inversat poate memora poziția token-urilor într-o manieră ce ține cont de structura documentului HTML. Acesta poate memora și tipul de text în care a fost găsit: un paragraf normal (adică se găsește în marcaje de tip ...), titlul documentului (între marcajele <title>...</title>), un subtitlu (<h2>...<h2>) sau un text asociat unui link către document.

Dacă dorim ca în contextul unei căutări boolene să facem un ranking al documentelor o primă metodă ar fi să ordonăm documentele în funcție de unde s-a îndeplinit condiția exprimată în interogare. Evident, dacă condiția s-a îndeplinit în titlul documentului sunt mari șanse el să fie relevant utilizatorului.

O metodă de ranking ar fi alegerea unor ponderi g_i pozitive corespunzătoare cu fiecare tip de zonă și în timpul căutării să se calculeze pentru fiecare document scorul:

$$\mathrm{R} = \sum_i g_i s_i$$

unde s_i este 0 dacă condiția nu s-a îndeplinit în tipul respectiv de text și 1 dacă s-a îndeplinit. Găsirea ponderilor g_i optime se face de regulă prin tehnici „Machine Learning" pornindu-se la un set de căutări cu rezultate cunoscute.

4.2.2 Modelul spațiului vectorial

Modelul spațiului vectorial este cel mai bine cunoscut model din domeniu și unul dintre cele mai vechi fiind prezent din ani '60. Modelul este simplu, atât cererile cât și paginile web sunt mapate ca vectori cu toate elementele pozitive într-un spațiu abstract. Pentru a face ranking-ul calculăm o măsură de similaritate între vectorul cerere și vectori paginilor.

Măsura de similaritate canonică este distanța cosinusul unghiului dintre vectori. Acesta se caluleaza astfel:

$$sim(\vec{a}, \vec{b}) = \frac{\vec{a}}{|\vec{a}|} \cdot \frac{\vec{b}}{\left|\vec{b}\right|}$$

Aceasta se numeste măsura de similaritate cosinus și variează de la 0 la 1. Această măsură este convenabilă și fiindcă normele pentru pagini nu se pot calcula la indexare și numai coordonatele nenule contează în calcul. Fiindcă o cerere are destui de puțini termeni vectorul asociat va avea multe coordonate nule după cum vom vedea mai jos. Astfel calculul se simplifică mult.

4.2.3 Term Frequency - Inverse Document Freqency

Pentru a folosi modelul spațiului vectorial ne trebuie o metodă de a trensforma documentele și cererile în vectori. O modalitate populară este folosirea funcțiilor TF-IDF (Term Frequency - Inverse Document Freqency). Coordonatele vectorilor reprezintă termenii indexați și valoarea pentru fiecare coordonată este determinată cu funcți TF ce depind de frecvența termenului adică numărul de tokenuri ai termenului din document sau cerere și funcții IDF ce depind de inversa frecvenței documentelor adică de inversul numărului de documente de conțin un termen. Ecuațiile de formare a vectorilor diferă mult și nu este neobișnuit ca cererea să folosească alte funcții decât documentele. Un exemplu pentru calculul vectorilor este următorul:

$$d(doc, termen) = TF(doc, termen) \cdot IDF(termen)$$

$$IDF(termen) = \log(N/N_{termen})$$

$$TF(doc, termen) = \begin{cases} \log(f_{termen,doc}) + 1 & daca\ f_{termen,doc} > 0 \\ 0 & altfel \end{cases}$$

unde $f_{termen,doc}$ este frecvența termenului, N_{termen} este frecvența documentelor iar N este numărul total de documente.

Intuiția din spatele acestor formulări este că funcțiile TF exprimă relevanța unui document relativ va un termen și funcțiile IDF exprimă importanța temenului. Ca exemplu termenul „the” în engleză este foarte frecvent folosit și nu însemnă nimic dacă un document are termenul dar termenul „Shakespeare” este folosit mai rar și dacă un document îl menționează măcar o dată atunci el va ieși în evidență. Un astfel de model de conversie mai este numit și un model „bag of words” fiindcă nu se ține cont de modul în care sunt aranjate cuvintele ci doar de frecvența lor.

5 Evaluarea unui motor de căutare Web

Pentru evaluarea unui motor de căutare sunt necesare trei elemente:

1. O colecție de documente.
2. O colecție de cereri.
3. Un set de judecăți de de relevanță care să acopere toate perechile cerere-document. Numit de regulă ground truth sau gold standard.

De obicei judecățile sunt binare, adică relevant sau nerelevant. Folosind aceste trei elemente se pot calcula statistici de performanță.

Cele mai cunoscute statistici sunt precizia (P) și reamintirea (R) exprimate mai jos ele nu țin cont de ordinea în care se returnează rezultatele ci doar ce rezultate s-au returnat. Precizia însemnă proporția din documentele returnate ce sunt și relevante.

$$P = \frac{N(\text{relevant } \& \text{ returnat})}{N(\text{returnat})}$$

Reamintirea înseamnă proporția din documentele relevante ce au fost returnate.

$$R = \frac{N(\text{relevant } \& \text{ returnat})}{N(\text{relevant})}$$

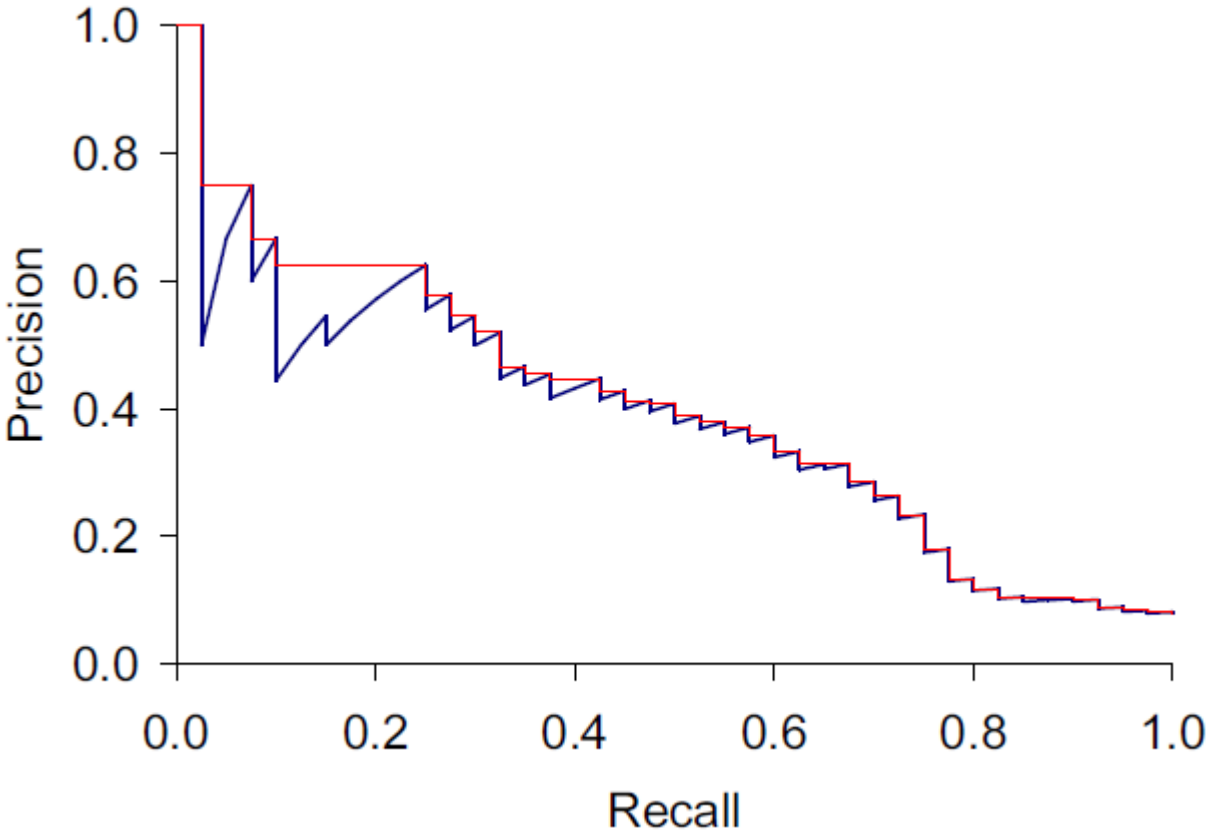
Ambele statistici sunt importante și trebuie echilibrate. Dacă precizia este prea mică însemnă că utilizatorul trebuie el însuși să mai caute în rezultate pentru a determina ce pagini îi sunt de ajutor. Dacă reamintirea este prea mică atunci utilizatorul va primi prea puțină informație din domeniul de interes. Totuși în unele situații se poate favor mai mult precizia sau reamintirea. O măsură ce pe combină pe amândouă se numește „F-measure" și este media armonică ponderată a preciziei și reamintirii exprimată astfel:

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}}$$

unde $\alpha \in [0, 1]$ este factorul de ponderare. Pentru a evalua un motor de căutare fără ranking se calculează precizia și reamintirea medie pentru un set mare de căutări.

Precizia și reamintirea se pot folosi pentru a evalua și ordonarea măsurătorilor. Se pot lua pur și simplu primele k rezultate și se calculează măsurătorile. Utilizatori motoarelor de căutare web se uită foarte rar după primele 10 rezultate. Astfel precizia și reamintirea la 10 rezultate este o statică bună pentru o evaluare rapidă. Dacă se dorește o evaluare mai comprehensivă se poate calcula precizia și reamintirea pentru mai multe k-uri și se

Figura 5.1: Curba precizie reamintire. Albastru: calculată pentru toate valorile lui k. Roșu: Calculată numai în punctele unde se schimbă reamintirea [Manning-2009].



poate afișa graficul preciziei în funcție de reamintire sau curba precizie-reamintire. Un exemplu este dat în figura 5.1.

Dacă totuși se dorește o singură valoare se poate calcula scorul MAP (mean average precision) definită astfel:

$$\text{MAP} = \frac{1}{Q}\sum_{j=1}^{Q}\frac{1}{m_j}\sum_{k=1}^{m_j} Precizie(R_{j,k})$$

unde Q este numărul de cereri din set m_j este număru de documente relevante pentru cererea j și $R_{j,k}$ este setul de rezultate care de la primul rezultat returnat până la ultimul rezultat relevant. Scorul este aproximativ egal cu aria de sub curba precizie-reamintire.

6 Concluzii

Am realizat o prezentare a principiului de funcționare a motoarelor de căutare web începând cu procesarea textului din conținut și până la ordonarea rezultatelor unei cereri la final. Am prezentat structura de bază a unui motor de căutare, index-ul invers și am descris cum se realizează o căutare booleană. Am discutat apoi metodele de ranking static și dinamic. Pentru ranking-ul static am descris metodele PageRank și HITS și pentru ranking-ul dinamic am prezentat modelul spațiului vectorial.

Bibliografie

[Manning-2009] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze. "An Introduction to Information Retrieval",Cambridge University Press, 2009.

[Büttcher-2010] , Stefan, Charles LA Clarke, and Gordon V. Cormack. "Information Retrieval: Implementing and Evaluating Search Engines",MIT Press, 2010.

[Wikipedia-2016] "PageRank". Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 27.01.2016. Online: https://en.wikipedia.org/wiki/PageRank . Accesat: 27.01.2016.