Universitatea Bucureşti
Facultatea de matematică şi informatică

Lucrare de disertaţie

Metode de ierarhizare în Regăsirea Informaţiei

Autor:
Călin Avasîlcăi

Coordonator:
Lect. Dr. Marius Popescu

MAI 2010

Cuprins

Capitolul 1

Introducere

Singtagma *regăsirea informaţiei*(RI) are un spectru larg de inţelesuri. Scoaterea unui card de credit din portofel pentru a completa numărul cardului într-un formular este o formă de regăsire a informaţiei. Dar, ca domeniu de studii academice, *regăsirea informaţiei* poate fi definita astfel[1]:

> Regăsirea informaţiei se referă la găsirea de material (de obicei documente) de natură nestructurată (de obicei text) care satisface o nevoie de informaţie, din colectii mari de date (eventual stocate pe calculatoare).

Regăsirea informaţiei obişnuia să fie o activitate practicata de caţiva oameni cum ar fi bibliotecarii. Acum lumea s-a schimbat şi sute de milioane de oameni folosesc sisteme de regăsire a informaţiei precum motoate de căutare web în fiecare zi. Regăsirea informatiei devine rapid forma dominantă de accesare a informatiei întrecând modul tradiţional de căutare de tip "baze de date" (căutarea dupa un număr de identificare).

RI este interdisciplinară, bazată pe informatică, matematică, ştiinţa informaţiei, arhitectura informaţiei, psihologie cognitivă, lingvistică, statistică, etc. Sisteme automate de RI au fost folosite pentru a reduce ceea ce se numeste "supraîncărcarea de informaţie". Multe universităţi şi librării publice folosesc sisteme RI pentru a facilita accesul la cărţi, jurnale şi alte tipuri de documente. Cele mai folosite aplicatii RI sunt motoarele de căutare web.

1.1 Scurt istoric

Ideea de a folosi calculatoarele pentru a căuta "bucăti" de informaţie a fost popularizată în articolul *As We May Think* de către *Vannevar Bush* în anul 1945[2]. Primele sisteme automate de regăsire a informaţiei au fost introduse în anii 1950 si 1960. Până in 1970 cateva tehnici au fost testate cu succes pe corpusuri mici de text cum ar fi colecţia *Cranfield* (câteva mii de documente). Sisteme mari de RI cum ar fi *Lockheed Dialog system*, au fost date în folosinţă la începutul anilor 1970.

În 1992 Departamentul de Apărare al Statelor Unite împreună cu Institutul Naţional de Standarde şi Tehnologie (NIST) au sponsorizat *TREC* (**T**ext **RE**trieval **C**onference) ca parte din programul TIPSTER. Scopul a fost să se asigure comunitaţii RI infrastructura necesară pentru testarea şi evaluarea metodelor de regăsire a textului pe colectii foarte mari. Acest program a acţionat ca un catalizator în ceea ce priveşte cercetarea metodelor RI care scaleaza la colectii foarte mari. Aparitia motoarelor de căutare pe web a adus o nevoie si mai mare de sisteme RI care pot face faţa unor colecţii imense de date.

1.2 Descrierea procesului

Un proces de regăsire a informaţiei începe când un utilizator introduce o interogare (*query*) în sistem. Interogările sunt formulări formale ale *nevoilor de informaţie*, de exemplu, cuvinte scrise în caseta de căutare a unui motor de căutare web. În RI o interogare nu identifică în mod unic un obiect din colecţia de obiecte. În schimb, mai multe obiecte pot fi considerate ca răspunsuri la interogare, eventual cu grade de relevanţă diferite.

Un obiect este o entitate stocată în sistemul RI. În funcţie de aplicţie, obiectele pot fi texte, documente, imagini, videoclipuri, etc. În general, documentele propriu-zise nu sunt ţinute în sistem ci sunt reprezentate de surogate(rezultate în urma procesării documentelor) si metadate.

Majoritatea sistemelor RI calculează un scor numeric ce reprezintă cat de bine se potriveste un obiect cu interogarea utilizatorului şi apoi ierarhizează obiectele în funcţie de această valoare. Cele mai bune obiecte sunt apoi afişate utilizatorului.

1.3 Un exemplu

Capitolul 2

Regăsirea informaţiei - generalităţi

2.1 Indexare

2.2 Cautare

2.3 Regăsirea booleană şi ad-hoc

Capitolul 3

Evaluarea în regăsirea informaţiei

Există multe alternative pentru a proiecta un sistem RI. Cum hotărâm care dintre aceste tehnici sunt eficiente în anumite aplicaţii? Este idicată folosirea unui *stop list* sau a unui *stemmer*? Regăsirea informaţiei a evoluat ca o disciplină empirică, necesitând evaluări atente şi complete pentru a demonstra superioritatea performanţelor noilor tehnici apărute pe colecţii reprezentative de documente.

3.1 Evaluarea unui sistem RI

Pentru a măsura eficienţa unui sistem ad-hoc RI este necesară o colecţie de test alcătuită din trei componente:

1. O colecţie de documente
2. O serie de *nevoi de informaţie* exprimate prin interogări
3. Un set de judecăţi de relevanţă, de obicei o valoare binară (*relevant/nerelevant*) pentru fiecare pereche interogare-document.

Abordarea standard în evaluarea unui sistem RI se învârte în jurul noţiunii de documente *relevante* şi *nerelevante*. Cu privire la o nevoie de informaţie, unui document dintr-o colecţie de test i se dă o clasificare binară: relevant sau nerelevant. Colecţia de documente şi setul de interogări trebuie să fie destul de mari: este nevoie de efectuarea unei medii a rezultatelor de performanţă peste seturi mari de test întrucât acestea variază destul de mult de la interogare la interogare. Ca o regulă derivată din experimente, 50 de nevoi de informaţie reprezintă un minim suficient.

Relevanţa este evaluată relativ la nevoia de informaţie, nu la interogare. De exemplu, o nevoie de informaţie ar putea fi:

> Informaţie cu privire la faptul că vinul roşu este mai eficient în prevenirea atacurilor de cord decât vinul alb.

Interogărea asociată acestei nevoi de informaţie ar putea fi:

> vin ŞI roşu ŞI alb ŞI atac ŞI cord ŞI eficient

Un document este relevant dacă se adresează nevoii de informaţie nu doar pentru că se întamplă să conţină cuvintele din interogare. Acest lucru este des neînţeles în practică, deoarece nevoia de informaţie este "ascunsă". În orice caz, ea este prezentă. Pentru o interogare care constă într-un cuvânt, este foarte greu pentru un sistem RI să îşi dea seama care este nevoia de informaţie. Dar utilizatorul are o astfel de nevoie şi va judeca rezultatele pe baza acesteia. Pentru a evalua un sistem este nevoie de o exprimare "deschisă" a nevoii de informaţie, care poate fi folosită la judecarea documentelor întoarse de sistem ca relevante sau nerelevante. Relevanţa poate fi gândită ca o scală, cu unele documente *mai* relevante decât altele. Dar, pentru început, se va folosi o valoare binară a relevanţei.

Multe sisteme conţin diferiţi parametri de calibrare ce pot fi ajustaţi pentru a mări performanţa. Este greşit să se raporteze rezultate pe o colecţie de test obţinute în urma modificării acestor parametrii cu scopul de a maximiza perfrormanţa pe colecţia respectivă deoarece parametrii vor fi setaţi să maximizeze performanţa pe un anumit set de interogări în loc să facă acest lucru pentru un eşantion aleator de interogări.

3.2 Colecţii sandard de test

Aceasta este o listă de colecţii standard de test si evaluare. Majoritatea sunt menite pentru a testa sisteme ad-hoc de RI dar sunt si câteva folosite pentru clasificare de text.

Cranfield Această colecţie a fost prima care a permis măsuri precise de eficientă ale sistemelor de regăsire a informaţiei, dar acum este prea mică pentru un experiment "serios". A fost alcătuită în anii 1950 in Regatul Unit şi conţine 1398 de sumare de articole despre aerodinamică, un set de 225 de interogări şi judecăţi de relevanţă complete pentru toate perechile (interogare, document).

TREC NIST a construit un framework de evaluare pentru RI începând cu 1992. Din acest framework cele mai cunoscute sunt cele folosite la testele *TREC Ad Hoc track* în timpul primelor opt conferinţe TREC (**T**ext **RE**trieval **C**onference) între 1992 şi 1999. În total, aceste colecţii de test însumează 6 CDuri şi conţin 1,89 milioane de documente (compuse cu preponderenţă din articole de ştiri) şi judecăţi de relevanţă pentru 450 de nevoi de informaţie numite subiecte. Colecţiile TREC 6-8 sunt formate din 150 de nevoi de informaţie si 528.000 de articole de ştiri. Acesta este probabil cea mai bună subcolecţie datorită faptului că este cea mai mare şi subiectele sunt mai consistente. Din cauza faptului că aceste colecţii sunt foarte mari, nu există judecăţi exhaustive ci sunt specificate doar pentru un subset de documente (primele k documente întoarse de sistemul pentru care nevoia de informaţie a fost dezvoltată).

GOV2 În ultimii ani, NIST a făcut evaluări pe colecţii mult mai mari, incluzând colecţia de 25 de milioane de pagini web, *GOV2*. GOV2 este acum cea mai mare colecţie de pagini web disponibilă pentru cercetare. Cu toate acestea, GOV2 încă este de cel puţin 2 ori mai mică decât colecţia de documente indexate de companiile mari de căutare web.

NTCIR Proiectul *NTCIR - NII Test Colections for IR Systems* a construit diferite colecţii de test de mărimi similare cu colecţiile TREC. Aceste colecţii sunt concentrate pe limba est-asiatică şi pe *regăsirea informaţiei cross-language* (interogările sunt făcute într-o limbă peste un set de documente scrise în diferite limbi).

CLEF Seria de evaluări CLEF (Cross Language Evaluation Forum) s-a concentrat pe limbile europene şi regăsirea informaţiei de tip cross-language.

Reuters Pentru clasificarea de text, cea mai folosită colecţie de test a fost *Reuters-21578* compusă din 21578 articole de ştiri. Mai recent Reuters a publicat *Reuters Corpus Volume 1 (RCV1)*, o colecţie mult mai mare constând 806.791 documente.

3.3 Evaluarea rezultatelor neordonate

Având aceste ingrediente, cum este măsurată eficienţa unui sistem? Cele mai frecvente si fundamentale două măsuri în RI sunt aşa-numitele *precizia(precision - P)* şi *returnarea(recall - R)*. Acestea sunt mai întâi definite pentru cazul în care sistemu RI întoarce un set neordonat de documente pentru o interogare şi apoi extinse la liste ierarhizate de documente.

Precizia este dată de raportul dintre documentele regăsite care sunt relevante şi documentele regăsite:

$$P = \frac{\#(documente\, regăsite\, si\, relevante)}{\#(documente\, regăsite)} = P(relevante|regăsite) \tag{3.1}$$

Returnarea este raportul dintre documentele regăsite care sunt relevante şi documentele relevante:

$$R = \frac{\#(documente\, regăsite\, si\, relevante)}{\#(documente\, relevante)} = P(regăsite|relevante) \tag{3.2}$$

Aceste noţiuni pot fi explicate mai clar folosind tabelul de contingenţă de mai jos.

Atunci:

$$P = \frac{tp}{tp + fp} \qquad R = \frac{tp}{tp + fn} \tag{3.3}$$

	Relevante	Nerelevante
Regăsite	pozitive adevărate(tp)	pozitive false(fp)
Neregăsite	negative false(fn)	negative adevărate(tn)

O alternativă evidentă este judecarea sistemului după acurateţe, adică, procentul clasificărilor corecte. Folosind tabelul de mai sus, $ac = (tp + tn)/(tp + fp + fn + tn)$. Acest lucru pare plauzibil din moment ce există două clase, relevant şi nerelevant, iar un sistem RI poate fi privit ca un clasificator pe două clase (întoarce documentele etichetate cu *relevant*). Aceasta este, de fapt, măsura de eficienţă folosită de obicei în evaluarea problemelor de clasificare.

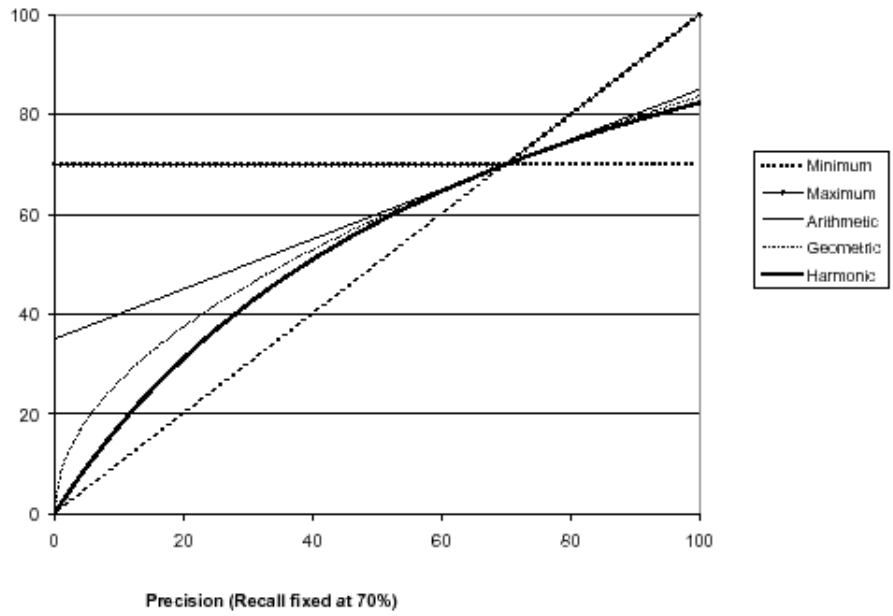

Figura 3.1: Grafic care compară media armonică cu celelalte medii

Există un motiv din cauza căruia acurateţea nu este o masură corespunzătoare pentru regăsirea informaţiei. În aproape toate circumstanţele, datele sunt extrem de "denaturate": de obicei, peste 99,9% din documente sunt în categoria nerelevant. Un sistem configurat să maximizeze acurateţea poate părea că funcţionează foarte bine prin considerearea tuturor documentelor ca fiind nerelevante la orice introgare. Acest lucru este inacceptabil pentru un sistem RI. Un utilizator va dori întotdeauna să vadă niste documente şi poate avea o mică toleranţă pentru pozitive false dacă majoritatea documentelor îi satisfac nevoia de informaţie. Măsurile P şi R concentrează evaluarea pe întoarcerea de pozitive adevărate, întrebând ce procent de documente relevante a fost găsit şi cate pozitive false au fost întoarse.

Avantajul faptului că există două masuri diferite este acela că în cele mai multe circumstanţe una este mai importantă decât cealaltă. Un utilizator web obişnuit doreşte ca fiecare rezultat de pe prima pagină să fie relevant(precizie mare) dar nu are nici cel mai mic interes să ştie şi mai ales să privească fiecare document relevant. Pe de altă parte, un profesionist care caută documente într-un sistem RI intern va încerca să găsească toate documentele relevante la nevoia lui de informaţie(returnare mare) şi va tolera valori mici ale preciziei pentru a îsi atinge scopul. În orice caz, există un compromis între cele două valori: se poate întotdeauna obţine un recall maxim dar precizie mică dacă se întorce o listă cu toate documentele. Recall-ul este o functie non-descrescătoare de numărul de documente regăsite. Pe de altă parte, într-un sistem bun, precizia de obicei scade odată cu cresterea numărului de documente regăsite. În general se doreşte o anumită valoare a recall-ului, tolerându-se un anumit procent de pozitive false.

O măsură care înglobează atât precizia cât şi returnarea este *măsura F*, care se calculează ca media armonică ponderată a celor două valori:

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R} \quad unde \quad \beta^2 = \frac{1-\alpha}{\alpha} \tag{3.4}$$

unde $\alpha \in [0,1]$ şi, ca urmare, $\beta^2 \in [0,\infty)$. Măsura F balansată ţine cont în mod egal de precizie şi recall($\alpha = 1/2, \beta = 1$). Este notată de obicei cu F_1 (de la $F_{\beta=1}$):

$$F_{\beta=1} = \frac{2PR}{P+R} \tag{3.5}$$

Valori ale lui β mai mici decât 1 scot în evidenţă precizia, în timp ce valori mai mari ca 1 pun accentul pe returnare. Valorile discutate până acum sunt măsuri între 0 şi 1, dar se mai scriu si ca procente pe o scală de la 1 la 100. Media armonică este folosită în locul mediei aritmetice deoarece în cazul în care un sistem ar întoarce toate documentele(100% recall), ar avea tot timpul cel puţin $F = 50\%$. Media armonică a două numere ($a << b$) este mai apropiată de minim decât de media lor aritmetică(vezi figura 3.1).

3.4 Evaluarea rezultatelor ordonate

Măsurile *precision*, *recall* şi F sunt bazate pe seturi; sunt calculate folosind seturi neordonate de documente. Este nevoie de o extindere a acestor măsuri (sau de o definire de noi măsuri) dacă se doreşte evaluarea unor rezultate ierarhizate care reprezintă standardul în zilele noastre. Într-un context de regăsire ierarhizată setul care interesează este dat de *primele k* documente. Pentru fiecare astfel de set, valorile preciziei si returnării pot fi reprezentate grafic printr-o *curbă precision-recall*, ca cea din figura 3.2.

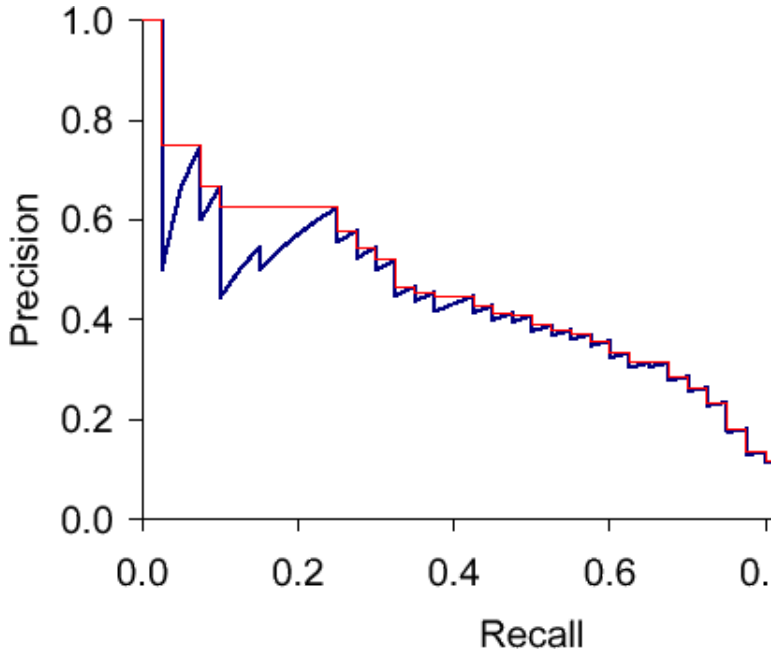

Figura 3.2: Grafic *precision-recall*.

Curbele precision-recall au o formă de "fierăstrău": dacă al $(k+1)$-lea document regăsit este nerelevant, atunci recall-ul este acelaşi ca pentru primele k documente regăsite, dar precizia scade. Dacă documentul este relevant atunci ambele valori cresc. Este de obicei util să se elimine această formă şi modul standard de a face acest lucru este folosind *precizia interpolată* p_{interp}. La un anumit nivel de recall r, p_{interp} este dat de cea mai mare valoare a preciziei găsită pentru orice nivel de recall $r' \geq r$:

$$p_{interp} = \max_{r' \geq r} p(r') \tag{3.6}$$

Precizia interpolată este ilustrată în figura 3.2.

Examinarea întregii curbe precizion-recall este foarte informativă, dar, câteodata este necesar ca informaţia să fie redusă la câteva numere. Modul tradiţional de a realiza acest lucru (folosit, de exemplu,

la primele 8 evaluari TREC Ad Hoc) este *precizia medie interpolată în 11 puncte*. Pentru fiecare nevoie de informaţie, precizia interpolată este măsurată în cele 11 puncte de recall: $0.1, 0.2, ..., 1.0$. Exemplul din figură 3.2 este ilustrat în tabelul 3.1.

Recall	Precizie interpolată
0.0	1.00
0.1	0.67
0.2	0.63
0.3	0.55
0.4	0.45
0.5	0.41
0.6	0.36
0.7	0.29
0.8	0.13
0.9	0.10
1.0	0.08

Tabelul 3.1: Calcularea preciziei medii interpolate în 11 puncte.

Pentru fiecare nivel de recall se calculează media aritmetică a preciziei interpolate la acel nivel de recall pentru fiecare nevoie de informaţie din colecţie. O curbă precizie-recall cu 11 puncte poate fi vizualizată apoi grafic. Un exemplu este prezent în figura 3.3.

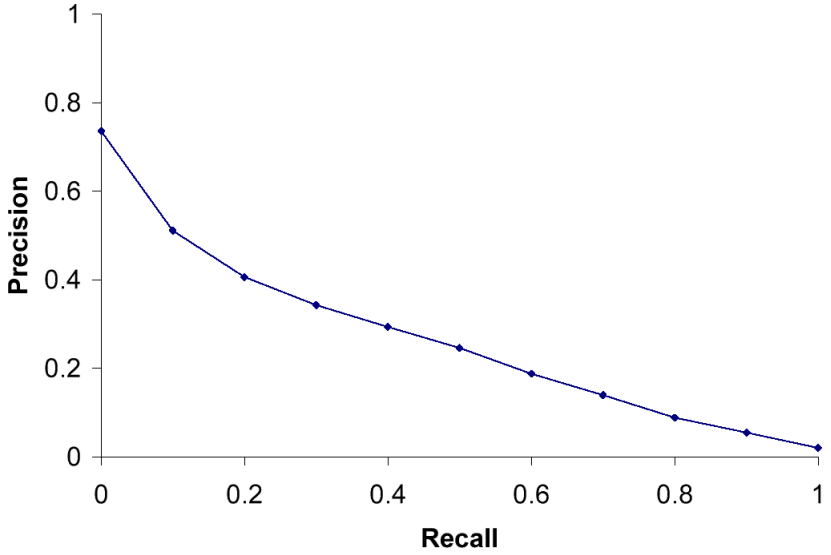

Figura 3.3: Grafic al preciziei medii interpolate în 11 puncte.

În ultimii ani, alte măsuri au devenit mai comune. Cea mai folosită în comunitatea TREC este *Mean Average Precision (MAP)*, care pune la dispoziţie o singură măsură de calitatea de-a lungul nivelelor de recall. Printre tipurile de masuri de evaluare s-a arătat că MAP funcţionează foarte bine la capitolele discriminare şi stabilitate. Pentru o sigură nevoie de informaţie, precizia medie este media valorilor de precizie obţinute pentru primele k documente după regăsirea fiecărui document relevant. Media acestei valori peste toate nevoile de informaţie este MAP. Dacă setul de documente relevante pentru o nevoie de informaţie $q_j \in Q$ este $d_1, ..., d_{m_j}$ şi R_{jk} este setul de rezultate de la primul până la documentul d_k, atunci:

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Prec(R_{jk}) \tag{3.7}$$

Când un document relevant nu este regăsit, precizia este considerată 0. Pentru o singură nevoie de informaţie, precizia medie aproximează aria de sub curba neinterpolată precizie-recall, ăsadar MAP aproximează aria medie de sub curba precizie-recall pentru un set de interogări.

Folosind MAP, nu sunt alese nivele fixe de recall şi nu este nevoie de interpolare. Un set de nevoi de informaţie trebuie sa fie cuprinzător si diversificat pentru a putea măsura cât mai exact eficientă unui sistem.

Măsurile descrise până au în componenţa precizia pentru fiecare nivel de recall. În cazul multor aplicaţii, cum ar fi un motor de căutare web, acest lucru nu reflectă neapărat nevoile unui utilizator. În acest caz contează mai mult câte rezultate bune se gasesc în primele pagini. Acest lucru conduce la nevoia de a măsura precizia pentru primele documente regăsite (10 sau 30). Această tehnică poartă numele de *"Precizia la k"*. Are avantajul că nu necesită o esitmare a dimenisiunii setului de documente relevante(m_j în cazul MAP) şi dezavantajele că este cea mai puţin stabilă dintre metodele de evaluare şi că nu se poate calcula media prea bine pentru ea.

O alternativă care atenuează această problemă este *R-precizia*. Implică existenţa unui set *Rel* de documente despre care se ştie că sunt relevante, din care se calculează apoi precizia pentru primele $|Rel|$ documente regăsite. R-precizia se ajustează la mărimea setului de documente relevante: un sistem perfect ar putea puncta 1 la această măsură pentru fiecare interogare, în timp ce până şi un sistem perfect, ar putea să atingă o precizie la 20 de 0.4 dacă ar exista doar 8 documente în colecţia de documente relevante. Calcularea mediei peste setul de interogări are mai mult sens în acest caz. Dacă există $|Rel|$ documente relevante pentru o interogare, se examinează primele $|Rel|$ rezultate date de un sistem şi se găsesc r documente relevante, atunci precizia (şi deci R-precizia) este $r/|Rel|$, dar şi recall-ul este tot $r/|Rel|$. Aşadar, R-precizia este identică cu o altă măsură folosită câteodată: *break-even point*, măsură definită ca valoarea la care precizia şi recall-ul sunt egale. Ca şi *Precizia la k*, *R-precizia* descrie un singur punct de pe curbă, şi e uneori neclar de ce interesează mai mult nivelul în care cele două valori sunt egale decât nivelul cel mai bun de pe curbă (în care F este maxim) sau un nivelul de interes pentru o anumită aplicaţie (*Precizia la k*). Cu toate acestea, R-precizia pare să fie corelata cu MAP, lucru constatat în urma experimentelor.

Un alt concept folosit uneori în evaluarea unui sistem este *curba ROC* ("ROC" vine de la *Receiver Operating Characteristics*). O curbă ROC reprezintă grafic rata pozitivelor adevărate (*senzitivitate*) ca funcţie de rata pozitivelor false (1 - *specificitate*). Aici senzitivitate este un alt terment pentru recall. Rata de pozitive false este dată de $fp/(fp + tn)$. Figura 3.4 ilustrează cuba ROC corespondentă cubei precizie-recall din figura 3.2.

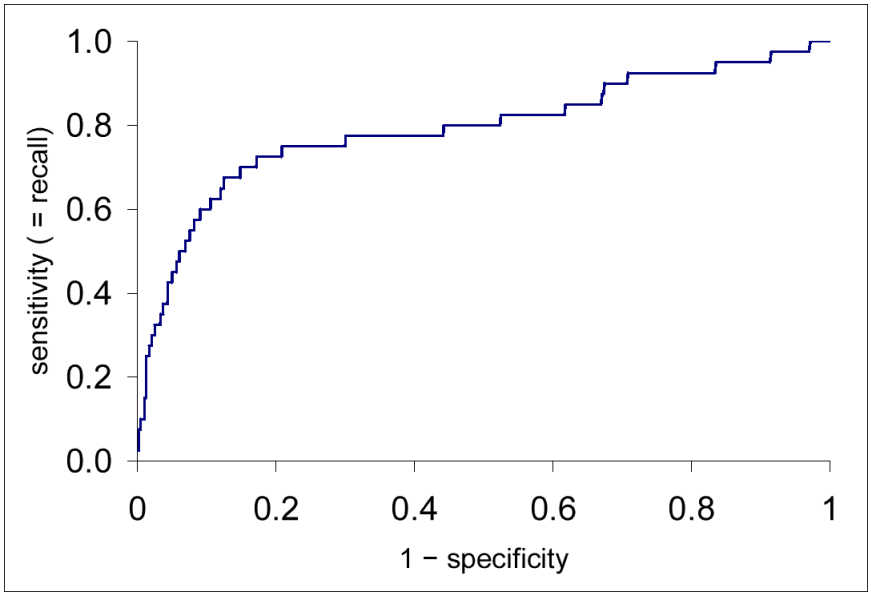

Figura 3.4: Curba ROC

O curbă ROC urcă întotdeauna din stânga jos spre dreapta sus. Pentru un sistem bun, graficul urcă abrupt în partea stângă. Are mai mult sens când se ia în considerare întreg spectrul de regăsire şi pune la dispoziţie o altă perspectivă asupra datelor. În unele cazuri se foloseste estimarea ariei de sub curba ROC, care este o valoare analoagă măsurii MAP.

O altă abordare care a fost adoptată din ce în ce mai des, în special în cazul sistemelor care folosesc

machine learning, este măsura *câştigului cumulativ* (*cumulative gain*) şi, în particular, *normalized discounted cumulative gain(NDCG)*. NDCG este proiectat pentru situaţii în care relevanţa este nebinară. Ca şi *Precizia la k*, este evaluată peste primele k rezultate. Pentru un set de interogări Q, fie $R(j,d)$ scorul de relevanţă pe care evaluatorii l-au dat documentului d pentru interogarea j. Atunci,

$$NDCG(Q,k) = \frac{1}{|Q|}\sum_{j=1}^{|Q|} Z_{kj} \sum_{m=1}^{k} \frac{2^{R(j,m)}-1}{\log_2(1+m)}, \tag{3.8}$$

unde Z_{kj} este un factor de normalizare calculat pentru a face ca NDCG-ul unei ierarhizări perfecte la k pentru interogarea j să fie 1. Pentru interogări pentru care $k' < k$ documente sunt returnate, ultima sumă se face până la k'

Capitolul 4

Metode de ierarhizare

4.1 Modelul spaţiului vectorial (VSM)

4.1.1 Similaritatea cosinus şi td-idf

4.2 Euristici pentru eficientizare

4.3 Cluster pruning

4.4 Modelul axiomatic

4.4.1 F2-EXP

4.5 Modelul probabilistic

Dacă s-ar cunoaşte relevanţa unui subset de documente, s-ar putea estima probabilitatea apariţiei unui termen t într-un document relevant $P(t|R = 1)$ şi, ca urmare, acesta ar putea reprezenta baza unui clasificator care decide dacă un document este relevant sau nu.

Utilizatorii încep cu *nevoi de informaţie* pe care le transformă în *interogări*. În mod similar, documentele sunt transformate în *reprezentări de documente* care diferă de primele cel puţin prin felul în care textul este împărţit în token-i. Bazându-se pe aceste două reprezentări, un sistem încearcă să determine cât de bine satisfac documentele nevoile de informaţii. În modelul Boolean sau VSM, dându-se numai o interogare, pentru un sistem RI nevoia de informaţie este incertă. Dându-se interogarea şi reprezentarea documentelor, un sistem trebuie să "ghicească" daca un document are conţinut relevant pentru respectiva nevoie de informaţie. Teoria probabilităţilor pune la dispoziţie o fundaţie de principii potrivite pentru raţionament în situaţii incerte. Aceste principii pot fi exploatate pentru a estima cât de probabil este ca un document sa fie relevant pentru o nevoie de informaţie.

Există mai multe posibile modele probabilistice de regăsire. În continuare voi discuta despre *principiul probabilistic de ierarhizare* şi despre *modelul binar de independenţă*, care a fost primul model probabilistic de regăsire. În final voi prezenta şi sistemul de ponderare *Okapi BM25*, care a avut un succes destul de mare în practică.

În acest context, este util conceptul de şanse (*odds*), pe lângă cel de probabilitate.

$$Odds : O(A) = \frac{P(A)}{P(\bar{A})} = \frac{P(A)}{1 - P(A)}. \tag{4.1}$$

4.5.1 Principiul probabilistic de ierarhizare

Cazul 1/0 loss

Presupunem că sistemul de RI întoarce ca răspuns la o interogare o listă ordonată de documente şi folosirea unei notaţii binare pentru relevanţă. Pentru o interogare q şi un document d, fie $R_{d,q}$ o vari-

abilă aleatoare care indică dacă d este relevant în contextul interogării q. Variabila ia valoarea 1 când documentul este relevant şi 0 altfel.

Folosind un model probabilistic, ordinea evidentă în care documentele trebuie prezentate utilizatorului este dată de ierarhizarea documentelor după probabilitatea estimată de relevanţă în raport cu nevoia de informaţie: $P(R_{d,q} = 1|d, q)$. Vom scrie R în loc de $R_{d,q}$. Aceasta reprezintă temelia *principiului probabilistic de ierarhizare (PRP)*:

> Dacă răspunsul unui sistem la fiecare cerere este o ierarhizare a documentelor din colecţie în ordinea descrescătoare a probabilitaţii de relevanţă, unde probabilităţile sunt estimate cât mai bine cu putinţă pe baza datelor pe care sistemul le are la îndemână, eficienţa sistemului este cea mai bună care se poate obtine folosind aceste date.

În cel mai simplu caz al PRP, nu există costuri de regăsire sau alte motive de îngrijorare care să valorifice diferit acţiunile sau erorile. Se pierde un punct fie pentru întoarcerea unui document nerelevant, fie pentru lipsa întoarcerii unui document relevant. O astfel de evaluare binară asupra preciziei poartă numele de *1/0 loss*. Scopul este să se întoarcă cele mai bune k rezultate posibile, pentru orice valoare k aleasă de utilizator. PRP spune că documentele trebuie ierarhizate în ordinea descrescătoare a $P(R = 1|d, q)$.

Teoremă 1 *PRP este optim în sensul că minimizează pierderea aştepată (sau riscul Bayes) în cazul 1/0 loss.*

Acestă teoremă este adevărată dacă toate probabilităţile sunt corecte, ceea ce în practică este imposibil. Cu toate acestea, PRP reprezintă o fundaţie pentru contruirea de modele de RI.

Costuri de regăsire

Să presupunem existenţa unui model de costuri de regăsire. Fie C_1 costul de regăsire a unui document relevant şi C_0 costul de regăsire a uni document nerelevant. Atunci pentru un document d şi pentru toate documentele d" neregăsite dacă

$$C_1 \times P(R = 1|d) + C_0 \times P(R = 0|d) \leq C_1 \times P(R = 1|d') + C_0 \times P(R = 0|d') \tag{4.2}$$

atunci d este următorul document care trebuie întors. Acest model asigură un cadru formal în care putem modela costurile diferenţiale ale falselor-pozitive şi falselor-negative.

4.5.2 Modelul binar de independenţă

BIM este modelul care a fost folosit cu PRP. Introduce câteva asumţii simple care permit estimarea funcţiei probabilistice $P(R|d, q)$. Aici binar este echivalent cu boolean: atât documentele cât si interogările sunt reprezentate ca vectori binari de incidentă a termenilor. Un document d este reprezentat de vectorul $\vec{x} = (x_1, ..., x_M)$, unde $x_t = 1$ dacă termenul t este prezent în documentul d şi $x_t = 0$ altfel. În contextul acestei repezentări, multe documente pot avea aceeaşi reprezentare. În mod similar, interogarea q este reprezentată prin vectorul de incidenţă $\vec{q}$. "Independenţă" se referă la faptul că termenii sunt modelaţi asa cum apar în documente în mod independent. Modelul nu recunoaşte niciun tip de asociere între termeni. Această asumpţie este, evident, incorectă, dar, în pofida acestui aspect, oferă rezultate satisfăcătoare în practică. Este asumpţia care stă şi la baza modelului *Bayes Naiv*, şi este într-un fel echivalentă cu asumţia din VSM, în care fiecare termen reprezintă o dimensiune ortogonala faţă de celelalte.

Pentru a face o strategie probabilistică de regăsire precisă, trebuie estimat modul în care termenii din documente contribuie la relevanţă. Cu alte cuvinte, trebuie să specificăm cum frecvenţa termenilor într-un document, numărul de documente care conţin un termen, lungimea unui document şi alte statistici influenţează calculul relevanţei unui document. După acest proces documentele vor fi ordonate în ordinea descrescătoare a acestor probabilităti estimate.

Se porneşte de la asumpţia că relevanţa fiecărui document este independentă de relevanţa celorlalte documente. În practică, acest lucru pune o problemă în momentul în care sunt întoarse documente duplicate sau aproape duplicate. În contextul BIM, probabilitatea că un document este relevant la o

interogare $P(R|d,q)$ este modelată via probabilitatea $P(R|\vec{x},\vec{q})$, folosind vectorii de incidenţă. Apoi, aplicând regula lui Bayes, se obţine:

$$P(R=1|\vec{x},\vec{q}) = \frac{P(\vec{x}|R=1,\vec{q})P(R=1|\vec{q})}{P(\vec{x},\vec{q})} \quad P(R=0|\vec{x},\vec{q}) = \frac{P(\vec{x}|R=0,\vec{q})P(R=0|\vec{q})}{P(\vec{x},\vec{q})} \tag{4.3}$$

Aici, $P(\vec{x}|R=1,\vec{q})$ şi $P(\vec{x}|R=0,\vec{q})$ reprezintă probabilitatea ca dacă un document relevant, respectiv nerelevant, este întors, acesta să aiba reprezentarea $\vec{x}$. Aceste probabilităţi nu se pot calcula exact, aşa că trebuie folosiţi estimatori: statistici ale colectiei de documente sunt folosite pentru a estima aceste probabilităţi. $P(R=1|\vec{q})$ şi $P(R=0|\vec{q})$ reprezintă probabilitatea apriori de a întoarce un document relevant, respectiv nerelevant, dată fiind interogarea $\vec{q}$. Pentru că un document este fie relevant fie nerelevant în contextul unei interogări, avem:

$$P(R=1|\vec{x},\vec{q}) + P(R=0|\vec{x},\vec{q}) = 1. \tag{4.4}$$

Derivarea unei funcţii de ierarhizare

Decât să estimăm $P(R=1|\vec{x},\vec{q})$ direct, deoarece interesează doar ordinea în care sunt întoarse documentele, folosim alte cantităţi care sunt mai usor de calculat şi care au ca rezultat aceeaşi ordine. Putem ordona documentele după şansele (odds) de relevanţă, ceea ce duce la o simplificare a relaţiei:

$$O(R|\vec{x},\vec{q}) = \frac{P(R=1|\vec{x},\vec{q})}{P(R=0|\vec{x},\vec{q})} = \frac{\frac{P(\vec{x}|R=1,\vec{q})P(R=1|\vec{q})}{P(\vec{x},\vec{q})}}{\frac{P(\vec{x}|R=0,\vec{q})P(R=0|\vec{q})}{P(\vec{x},\vec{q})}} = \frac{P(R=1|\vec{q})}{P(R=0|\vec{q})} \cdot \frac{P(\vec{x}|R=1,\vec{q})}{P(\vec{x}|R=0,\vec{q})} \tag{4.5}$$

Termenul stâng al expresiei din dreapta al ecuaţiei 4.5 este constant pentru o interogare dată. Pentru că interesează doar ordinea, nu e nevoie să se estimeze. Trebuie estimat, în schimb, celălalt termen, lucru care pare dificil iniţial: cum poate fie precis estimată probabilitatea unui întreg vector de incidenţă? Pentru a face posibilă estimarea, se face asumţia de *condiţional-independenţă Naive Bayes*: prezenţa sau absenţa unui cuvânt într-un document este independentă de prezenţa sau absenţa oricărui alt cuvânt:

$$\frac{P(\vec{x}|R=1,\vec{q})}{P(\vec{x}|R=0,\vec{q})} = \prod_{t=1}^{M} \frac{P(x_t|R=1,\vec{q})}{P(x_t|R=0,\vec{q})} \tag{4.6}$$

Aşadar:

$$O(R|\vec{x},\vec{q}) = O(R|\vec{q}) \cdot \prod_{t=1}^{M} \frac{P(x_t|R=1,\vec{q})}{P(x_t|R=0,\vec{q})}. \tag{4.7}$$

Pentru că fiecare x_t este fie 0, fie 1, putem separa termenii astfel:

$$O(R|\vec{x},\vec{q}) = O(R|\vec{q}) \cdot \prod_{t:x_t=1} \frac{P(x_t=1|R=1,\vec{q})}{P(x_t=1|R=0,\vec{q})} \cdot \prod_{t:x_t=0} \frac{P(x_t=0|R=1,\vec{q})}{P(x_t=0|R=0,\vec{q})}. \tag{4.8}$$

Fie $p_t = P(x_t=1|R=1,\vec{q})$ probabilitatea ca termenul x_t să apara într-un document relevant la interogarea dată şi $u_t = P(x_t=1|R=0,\vec{q})$, probabilitatea ca termenul să apara într-un document nerelevant. Aceste cantităţi pot fi vizualizate în tabelul de contingenţă următor (suma pe coloane este 1):

	document	relevant (R=1)	nerelevant(R=0)
termen present	$x_t = 1$	p_t	u_t
termen absent	$x_t = 0$	$1-p_t$	$1-u_t$

Facând asumpţia că termenii care nu apar în interogare au probabilităţi egale de apariţie intrn document relevant, respectiv nerelevant: $q_t = 0 \Rightarrow p_t = u_t$, vor trebui luaţi în considerare doar termenii care apar în interogare:

$$O(R|\vec{x},\vec{q}) = O(R|\vec{q}) \cdot \prod_{t:x_t=q_t=1} \frac{p_t}{u_t} \cdot \prod_{t:x_t=0,q_t=1} \frac{1-p_t}{1-u_t}. \tag{4.9}$$

Primul produs este peste termenii interogării care apar în document şi produsul din dreapta peste cei care nu apar.

Expresia poate fi manipulată prin includerea termenilor găsiţi în document în produsul din dreapta, dar, în acelaşi timp, ajustând produsul stâng pentru simplificare:

$$O(R|\vec{x},\vec{q}) = O(R|\vec{q}) \cdot \prod_{t:x_t=q_t=1} \frac{p_t(1-u_t)}{u_t(1-p_t)} \cdot \prod_{t:q_t=1} \frac{1-p_t}{1-u_t}. \tag{4.10}$$

Produsul drept este acum peste toţi termenii interogării, ceea ce înseamnă că e constant pentru o interogare, la fel ca $O(R|\vec{q})$. Aşa dar, singura cantitate care trebuie estimată pentru a ierarhiza documentele este cea din produsul stâng. Putem ordona documentele şi după rezultatul logaritmării produsului, deoarece log este o funcţie monotonă. Cantitatea folosită la ierarhizare se numeşte *valoarea statusului de regăsire (RSV - retrieval status value)*:

$$RSV_d = \log \prod_{t:x_t=q_t=1} \frac{p_t(1-u_t)}{u_t(1-p_t)} = \sum_{t:x_t=q_t=1} \log \frac{p_t(1-u_t)}{u_t(1-p_t)}. \tag{4.11}$$

Totul se reduce la clacularea RSV. Definim c_t:

$$c_t = \log \frac{p_t(1-u_t)}{u_t(1-p_t} = \log \frac{p_t}{1-p_t} + \log \frac{1-u_t}{u_t}. \tag{4.12}$$

Termenii c_t reprezintă proporţiile *log odds* pentru termenii interogării. Valolare va fi 0 dacă un termen are şanse egale să apar intrn document relevant, respectiv nerelevant, şi pozitivă dacă este mai probabil să apară într-un document relevant. Cantităţile c_t funcţionează ca ponderi ale termenilor în model, iar scorul pentru o interogare şi un document este: $RSV_d = \sum_{x_t=q_t=1} c_t$. Problema rămasă este cum să se estimeze cantităţile c_t pentru o colecţie de documente şi o interogare.

Estimări teoretice

Următorul tabel de contingenţă prezintă o serie de statistici ale colecţiei, unde df_t reprezintă numărul de documente care conţin termenul t:

	document	relevante	nerelevante	total
termen present	$x_t = 1$	s	$df_t - s$	df_t
termen absent	$x_t = 0$	$S - s$	$(N - df_t) - (S - s)$	$N - df_t$
	total	S	$N - S$	N

Aşadar, $p_t = s/S$ şi $u_t = (df_t - s)/(N - S)$ şi

$$c_t = K(N, df_t, S, s) = \log \frac{s/(S-s)}{(df_t - s)/((N - df_t) - (S - s))}. \tag{4.13}$$

Pentru a evita posibilitatea apariţiei de zerouri (de exemplu, toate sau niciun document relevan conţine un anumit termen) se adaugă $\frac{1}{2}$ la fiecare dintre cele 4 cantităţi şi apoi se ajustează totalurile (N + 2). Ca urmare avem:

$$\hat{c}_t = K(N, df_t, S, s) = \log \frac{(s+\frac{1}{2})/(S-s+\frac{1}{2})}{(df_t - s + \frac{1}{2})/(N - df_t - S + s + \frac{1}{2})}. \tag{4.14}$$

Adgarea valorii $\frac{1}{2}$ este o formă simplă de *uniformizare.*

Estimări practice

Sub asumţia că documentele relevante reprezintă un procent infim din colecţie, este plauzibilă aproximarea statisticilor pentru documentele nerelevante cu statisticile pe întreaga colecţie. Ca urmare, u_t (probabilitatea ca un document nerelevant să conţină termenul t pentru o interogare) poate fi aproximat cu df_t/N şi:

$$\log \frac{1-u_t}{u_t} = \log \frac{N - df_t}{df_t} \approx \log \frac{N}{df_t} \tag{4.15}$$

Rezultatul este interesant şi prin faptul că furnizează o justificare teoretică a celei mai întâlnite forme de ponderare *idf* folosită în VSM.

Tehnica de aproximare din ecuaţia 4.15 nu poate fi usor extinsă la documente relevante. Cantitatea p_t poate fi estimată în mai multe moduri:

1. Se poate folosi frecvenţa termenilor din documentele cunoscute deja ca fiind relevante(dacă se cunosc - metodă folosită în cadrul *feedback-ului de relevanţă*)

2. Se poate presupune că fiecare termen are sănse egale să aparaă într-un document relevant: $p_t = 0.5$. Această estimare este destul de slabă. Combinând această metodă cu aproximarea lui u_t de mai sus, ierarhizarea documentelor este dată de termenii interogării care apar în documente scalaţi cu ponderea *idf*.

3. O altă aproximare propusă foloseşte statisticile aparitiilor termenilor în colecţie: $p_t = df_t/N$.

4.5.3 Okapi BM25

Metodele probabiliste sunt unele din cele mai vechi modele formale în RI. Încă din 1970 erau privite ca o oportunitate pentru a pune bazele teoretice în RI şi, odată cu "renaşterea" modelelor probabiliste în lingvistica computaţională în anii 1990, această oportunitate să întors iar metodele probabiliste reprezintă unul dintre subiectele cele mai discutate subiecte în materie de RI. Obţinerea unor aproximări rezonabile ale probabilităţilor necesare pentru un model RI probabilistic este posibilă, dar necesită prezumţii majore. În modelul BIM acestea sunt:

- o reprezentare booleană a documentelor, interogărilor, relevanţei
- independenţa termenilor
- termenii care nu apar în interogare nu afectează rezultatul
- valorile de relevanţă ale documentelor sunt independente

Poate că din cauza severităţii asumpţiilor de modelare este dificilă obţinerea unei performanţe mai bune. O problemă generală pare să fie că modelele probabiliste fie necesită informaţii parţiale de relevanţă, fie duc la derivarea unor scheme aparent inferioare de ponderare a termenilor.

Această situaţie s-a schimbat în anii 1990 când schema de ponderare *BM25* a avut rezultate foarte bune şi a început să fie adoptată de multe sisteme RI. Diferenţa dintre sistemele RI bazate pe *spatiul vectorial* şi cele bazate pe modelul probabilistic nu este aşa de mare; în ambele cazuri se construieste un sistem similar, singura diferenţă fiind că scorul documentelor in contextul unei interogări este dat pe de-o parte de *similaritatea cosinus* aplicată pe vectori de ponderi *tf-idf*, iar pe de altă parte de o formulă uşor diferită motivată de teoria probabilităţilor.

Un model nebinar

Modelul BIM a fost iniţial proiectat pentru scurte înregistrări de cataloage şi a funcţionat destul de bine în acest context, dar pentru căutări *full-text* pe colecţii mari este evident că un model trebuie să ia în considerare frecvenţa termenilor şi lungimea documentelor. Schema de ponderare numită Okapi după sistemul în care a fost iniţial implementată, a fost proiectată folosind un model probabilistic sensibil la aceste tipuri de informaţie fără să introducă prea mulţi parametrii adiţionali.

Cel mai simplu scor pentru un document d este dat de adunarea ponderilor *idf* ale termenilor interogării prezenţi în document:

$$RSV_d = \sum_{t \in q} \log \frac{N}{df_t} \tag{4.16}$$

Pornind de la formula din ecuaţia 4.14, şi estimând $S = s = 0$ în absenţa feedback-ului de relevanţă, se obţine o formulare alternativă a idf:

$$RSV_d = \sum_{t \in q} \log \frac{N - df_t + \frac{1}{2}}{df_t + \frac{1}{2}} \tag{4.17}$$

Această variantă are un comportament ciudat: dacă un termen apare în peste jumătate din documentele din colecţie, modelul dă o pondere negativă, lucru care este nedorit. În cazul folosirii unui *stop list* acest lucru nu se întâmplă de obicei.

Ecuaţia 4.16 poate fi îmbunătăţită prin folosirea frecvenţei termenilor şi a lungimii documentului:

$$RSV_d = \sum_{t \in q} \log \left[\frac{N}{df_t} \right] \cdot \frac{(k_1 + 1) tf_{td}}{k_1((1-b) + b \times (L_d / L_{ave})) + tf_{td}} \tag{4.18}$$

Aici, tf_{td} este frecvenţa termenului t în documentul d şi L_d şi L_{ave} sunt lungimea documentului d, respectiv lungimea media a unui document din colecţie. Variabila k_1 este un parametru pozitiv de ajustare care calibrează scalarea frecvenţei tf_td: $k_1 = 0$ corespunde modelului binar, iar o valoare mare corespunde folosirii frecvenţei brute. b este un alt parametru de calibrare ($0 \leq b \leq 1$) care determină scalarea după lungimea documentului: $b = 1$ presupune scalarea completă a ponderii termenului cu lungimea documentului în timp ce $b = 0$ presupune lipsa normalizării cu lungimea.

Dacă interogarea este lungă, atunci am putea folosi o ponderare similară pentru termenii din interogare. Acest lucru este adecvat dacă interogările au lungimi de dimensiunile unui paragraf, alftel este nenecesar:

$$RSV_d = \sum_{t \in q} \log \left[\frac{N}{df_t} \right] \cdot \frac{(k_1 + 1) tf_{td}}{k_1((1-b) + b \times (L_d / L_{ave})) + tf_{td}} \cdot \frac{(k_3 + 1) tf_{tq}}{k_3 + tf_{tq}}. \tag{4.19}$$

Aici tf_{tq} este frecvenţa termenului t în interogarea şi k_3 este un alt parametru pozitiv de calibrare care ajustează scalarea freveţei termenilor din interogare. Parametru b pentru normalizarea lungimii interogării este nenecesar.

Acesti parametri în mod ideal sunt setaţi să optimizeze performanţa pe o colecţie de test. Căutarea valorilor care să maximizeze performanţa poate fi făcută manual sau automat. În absenţa unor astfel de optimizări, experimentele au arătat că valori bune pentru aceşti parametrii sunt $b = 0.75$ şi $1.2 \leq k_1, k_3 \leq 2$.

Formulele BM25 de ponderare a termenilor au fost folosite cu succes pe o varietate de colecţii şi tipuri de căutare. Au avut o performanţă extrem de bună la evaluările TREC şi au fost implementate în multe sisteme RI.

Capitolul 5

Metode de agregare

5.1 Borda

5.2 Agregarea Rank Distance

Capitolul 6

Studiu comparativ

6.1 Unelte

6.1.1 Apache Lucene

Similaritate şi ierarhizare

Lucene combină modelul boolean (BM) cu modelul de spaţiu vectorial (VSM): documentele care trec de BM sunt etichetate cu un scor de către VSM.

În VSM, documentele si interogările sunt reprezentate ca vectori de ponderi într-un spaţiu multidimensional, unde fiecare termen din index este o dimensiune şi ponderile sunt valorile *tf-idf*. VSM nu necesită faptul ca ponderile să fie valori tf-idf, dar aceste ponderi au rezultate foarte bune în practică, şi, ca urmare, Lucene foloşte această abordare. Pentru un termen t şi un document (sau interogare) x, tf(t,x) creţe odată cu numărul de ocurenţe ale lui t în x iar idf(t) descreşte odată cu creşterea numărului de documente din index care il conţin pe t.

Scorul documentului d pentru interogarea q este dat de *similaritatea cosinus* pentru vectorii de ponderi V(q) şi V(d):

$$cos - sim(q, d) = \frac{V(q)V(d)}{|V(q)||V(d)|},$$

unde număratorul reprezintă produsul scalar, iar numitorul, produsul normelor euclidiene. Ecuatia poate fi văzută si ca produsul scalar dintre cei doi vectori normalizaţi.

Lucene perfecţionează VSM atăt în materie de calitate cât şi de uzabilitate.

- Normalizarea lui V(d) la vectorul unitate poate pune unele probleme în sensul că îndepărtează toată informaţia despre lungimea documentului. Pentru unele documente, lucrul acesta poate reprezenta o problemă. Pentru a evita această problemă, Lucene foloseşte un alt factor de normalizare a lungimii documentului, care normalizează vectorul la un vector mai mare sau egal decât vectorul unitate: doc-len-norm(d).

- La indexare utilizatorii pot specifica faptul că unele documente sunt mai importante decât altele prin asignarea unui *boost* respectivelor documente. Ca urmare, scorul fiecărui documente este multiplicat cu această valoare: doc-boost(d).

- Lucene este bazat pe câmpuri (secţiuni ale unui document), şi, ca urmare, fiecare termen al unei interogări se aplică unui singur câmp, normalizarea vectorului se aplică la nivel de câmp, şi se pot specifica şi nivele de boost pentru câmpuri.

- Acelaşi câmp poate fi adăugat unui document în timpul indexării de mai multe ori, iar, ca urmare, nivelul de boost al acelui câmp este dat de înmulţirea nivelelor de boost ale adăugărilor.

- La căutare utilizatorii pot specifica nivele de boost pentru fiecare interogare, sub-interogare şi termen al unei interogări.

- Un document poate fi relevant la o interogare cu mai mulţi termeni fără să conţină toţi termenii prezenţi în interogare, iar documentele în care apar mai mulţi termeni pot fi "răsplătite" printr-un factor de coordonare, care este mai mare când mai mulţi termeni sunt prezenţi: coord-factor(q, d).

Facând asumpţia simplificatoare că există un singur câmp în index, *formula conceptuala de scor* pentru Lucene este următoarea:

$$score(q,d) = coordfactor(q,d) \times queryboost(q) \times \frac{V(q) \times V(d)}{|V(q)|} \times doclennorm(d) \times docboost(d)$$

Din această formulă se derivează *formula practică de scor* care este implementată de Lucene.
Pentru computarea eficientă a scorului, unele componente sunt calculate şi agregate la indexare:

- Nivelul de boost pentru interogare este cunoscut când căutarea începe.
- Norma euclidiană a vectorului interogare poate fi calculată când începe căutarea, dat fiind faptul că e independentă de documentul pentru care se calculează scorul la un moment dat. Din perspectiva optimizării, merită pusă întrebarea: *are rost să se normalizeze vectorul interogării, din moment ce toate scorurile vor fi multiplicate cu aceeaşi valoare?* Ca urmare, ierarhia documentelor pentru o interogare dată nu va fi afectată de normalizare. Există două motive pentru a păstra normalizarea:
  - scorurile unui document pentru interogări distincte trebuie să fie comparabile(într-o anumită măsură)
  - aplicarea normalizării păstrează scorurile "în jurul" vectorului unitate, împiedicând astfel alterarea scorurilor din cauza limitării de precizie ale numerelor în virgulă mobilă
- Norma pentru fiecare document doc-len-norm(d) şi nivelul de boost doc-boost(d) sunt cunoscute la indexare. Sunt calculate şi rezultatul înmulţirii lor este salvat ca o singură valoare în index: norm(d).

În continuare este prezentată formula practică de scor:

$$score(q,d) = coord(q,d) \times queryNorm(q) \times \sum_{t \in q} (tf(t,d) \times idf(t)^2 \times boost(t) \times norm(field(t),d)),$$

unde:

1. *tf(t, d)* este corelat cu frecvenţa termenului in document. Documentele în care un termen apare de mai multe ori primesc un scor mai mare pentru acel termen. Lucene implementează astfel: $tf(t,d) = \sqrt{freq}$, unde *freq* reprezintă de căte ori apare termenul în document.
2. *idf(t)* este inversul frecvenţei termenului la nivel de index. Acest lucru înseamnaă că termenii mai rari au o contribuţie mai mare la scor. Implementeare Lucene este: $idf(t) = 1 + \log(\frac{numDocs}{docFreq+1})$, unde *numDocs* reprezintă numărul de documente din index şi *docFreq* reprezintă numărul de documente în care apare termenul.
3. *coord(q, d)* este o componentă calculată la momentul căutării: $coord(q,d) = \frac{overlap}{maxOverlap}$, unde *overlap* reprezintă numărul de termeni din interogare care se regăsesc în document şi *maxOverlap*, numărul de termeni ai interogării.
4. *queryNorm(q)* este factorul de normalizare folosit pentru a face scorurile pentru diferite interogări comparabile. Acest factor nu afectează ierarhizarea documentelor din moment ce este acelaşi pentru fiecare document. Implementarea implicită Lucene computează norma euclidiană a vectorului ponderilor(ajustate de nivelele de boost):

$$queryNorm(q) = \frac{1}{\sqrt{boost(q) \times \sum_{t \in q} (idf(t) \times boost(t))^2}}$$

5. *boost(t)* reprezintă nivelul de boost al termenului; acesta poate fi setat din sintaxa interogării dacă este folosit parserul de interogări pus la dispoziţie de Lucene, sau prin intermediul api-ului obiectului *Query*.

6.1.2 Pachetul Benchmark

6.1.3 Apache Open Relevance Project

6.2 Construirea framework-ului

6.3 Rezultate

Capitolul 7

Concluzii

Bibliografie

[1] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schutze, *Introduction to Information Retrieval*, Cambridge University Press. 2008.

[2] `http://en.wikipedia.org/wiki/Information_retrieval`