Els models matemàtics de Recuperació de la Informació i la seva implementació en motors de cerca de propòsit general

Jordi Ardanuy

Departament de Biblioteconomia i Documentació

Universitat de Barcelona

Juny 2003

CONTINGUT

1. Presentació i objectius

La Recuperació de la Informació (RI) tracta de la representació, emmagatzematge, organització i recuperació dels objectes informacionals que anomenarem, per simplificar, documents. Aquesta representació i organització ha de proveir a l'usuari d'un accés funcionalment senzill a la informació en la qual està interessat.

Desgraciadament, la tasca de caracteritzar les necessitats informatives de l'usuari és un problema complexe. Cal que primer tradueixi aquesta necessitat en una equació de cerca o consulta que pugui ser processada pel motor de cerca.

De manera general, aquesta traducció procura un conjunt de paraules clau o termes d'indexació que representen, suposadament, la descripció de les necessitats informatives de l'usuari. Donada la cerca, l'objectiu principal del sistema de RI és recuperar la informació rellevant per a l'usuari, que no necessàriament s'ha d'ajustar als documents que contenen tots els termes de la consulta.

Durant molts anys, l'interès d'aquestes qüestions s'ha limitat a bibliotecaris i experts en informació, tot i la ràpida disseminació, gràcies a l'ús dels ordinadors personals, d'eines de RI. Però a principis dels anys 90, un fet va canviar totalment la situació, la introducció del World Wide Web (www) a Internet.

El web s'ha convertit en el lloc de dipòsit dels coneixements i de la cultura humana, cosa que ens ha permès compartir idees i informació amb una rapidesa i volum mai vistos fins al moment. Però, com a conseqüència d'aquesta extraordinària inflació documental, la obtenció de la informació desitjada passà a ser una tasca molt tediosa i complexa. Això va renovar l'interès en la RI.

Una bona part dels esforços esmerçats en aquesta línia s'han dedicat a sistemes de RI de text complet, que són l'objecte del nostre interès. En concret, en el capítol 2 presentem el cocepte de model i una classificació. Els capítols 3, 4 i 5 es dediquen a descriure de manera formal els principals algorismes matemàtics basats en la teoria de conjunts, l'àlgebra lineal i la probabilitat proposats en aquesta àrea de la RI. S'ha intentat mantenir una estructura comuna i una notació coherent entre els diferents models fins a on ha estat possible.

Tanmateix, el nostre estudi no abordarà en detall però, els mecanismes de recuperació per rellevància –programació evolutiva– i algorismes similars com les xarxes neurals ,que quedaran per a tasques ulteriors.

La major part del material bibliogràfic utilitzat en aquests capítols són monografies i, especialment, treballs procedents de publicacions en sèrie.

En el capítol 6, que és el més llarg, s'aborden quins d'aquests models han estat realment incorporats en enginys per a servir com a motors de cerca en col·leccions generals de documents. Estudiarem de manera succinta els seus períodes d'aparició i el predomini en la implantació d'uns esquemes sobre els altres.

L'origen del material d'aquesta part es reparteix entre webs comercials i dels projectes, i treballs de publicacions en sèrie. Una petita part procedeix de la correspondència electrònica personal dels autors amb el desenvolupadors. Desgraciadament, no sempre ha estat possible obtenir les respostes desitjades.

El capítol 7 discuteix breument l'arribada dels models actuals als sistemes de RI, els compara i intenta establir una perspectiva per als propers anys.

Les conclusions sumàries i les tasques futures ocupen el capítol 8, al qual segueix una bibliografia final citada i/o utilitzada.

2. Els Models de Recuperació de la informació

Definició de model

Es pot definir un model de recuperació d'informació [R. Baeza-Yates, B. Ribeiro-Neto, 1999]

(D1) *com un quàdruple conjunt {**D**, **Q**, M, R (q_i, d_j) } on*
***D** és el conjunt de les representacions lògiques dels documents de la col·lecció, anomenats de manera metonímica, simplement documents.*
***Q** es el conjunt format per les representacions lògiques de les necessitats d'informació de l'usuari. Reben el nom de consultes o cerques (queries).*
M es el marc que permet modelar les representacions dels documents, les consultes i les seves relacions.
R(q_i, d_j) és una funció classificadora que anomenarem similitud (Sim) i que associa un nombre real a cada consulta $q_i \in \mathbf{Q}$ i cada document $d_j \in \mathbf{D}$. Tal classificació estableix un ordre jeràrquic (ranking) dels documents respecte la consulta q_i que adopta una forma com:

$$\begin{array}{lll} \textit{Documents} & d_{c_1} & Sim\,(d_{c_1}, q_c) \\ & d_{c_2} & Sim\,(d_{c_2}, q_c) \\ & \ldots & \\ & d_{c_s} & Sim\,(d_{c_s}, q_c) \end{array}$$

$$on \quad Sim\,(d_{c_k}, q_c) \geq Sim\,(d_{c_l}, q_c) \quad sempre\ que\ \ k < l$$

La construcció de tot model de RI implica, en primer lloc, establir les representacions dels documents i de les consultes. Conegudes aquestes es pot procedir a establir el marc on modelar les relacions que, finalment, adopten la forma d'una funció jerarquitzadora de la rellevància d'un document, segons la noció establerta en el marc.

Tradicionalment en els SRI els documents de la col·lecció romanen pràcticament estàtics, mentre que les consultes es van formulant. Aquesta manera d'actuar s'ha vingut a anomenar recuperació *ad hoc* (expressa o *forçada*). Però en el últims anys ha aparegut un nou procediment conegut com *filtrat*. En aquest cas són les consultes les que estan estàtiques – definides normalment a través d'un perfil de preferències de l'usuari –, mentre que els nous documents que entren en el sistema són comparats sistemàticament.

Tanmateix, no estem davant de models de RI diferents, sinó de modes d'operació, ja que en ambdós procediments les funcions de similitud impliquen determinar els mateixos tipus elements de càlcul ($q_i \in \mathbf{Q}$, $d_j \in \mathbf{D}$).

Classificació del models de RI

Si considerem els documents de text des d'una visió lògica, podem distingir entre models de recuperació de textos estructurats – amb parts separades perfectament identificables automàticament i d'importància ponderable – i no estructurats, és a dir, aquells en que tot el text forma un únic element.

Hem de distingir també els Sistemes de RI que es basen en comparar els termes que apareixen en una equació de cerca amb els que contenen els documents, o una selecció d'ells, de les tècniques de classificació que consisteixen en agrupar els documents en categories conceptuals[1]. Tanmateix, cal considerar que és possible la recuperació de documents, fins i tot jerarquitzada per rellevància, a partir de comparar el contingut de les cerques amb cadascuna de les classes obtingudes.

Nosaltres només ens ocuparem dels sistemes basats en la RI en textos no estructurats, mentre que els sistemes de classificació queden fora del nostre abast.

Models conceptuals

[1] Tots basats en espais vectorials. És el cas del mètodes com K veïns més propers (K-NN, K-Nearest Neighbours), K-mitjanes (K-means), vectors suports, xarxes de Hopfield o Kohonen.

En aquest àmbit podem establir una taxonomia a partir dels tres models considerats clàssics: el basat en la lògica de Boole, el model vectorial i el probabilístic d'independència binària (BIR). D'aquesta manera tenim un model de teoria de conjunts, representat pel model de Boole, ja que els documents i consultes son representats per conjunts de termes d'indexació; un algebraic, donat que la representació del model vectorial es basa en un vector n-dimensional; i finalment el probabilístic, ja que el model d'independència binària es basa fonamentalment en el teorema de Bayes.

Aquesta classificació, de totes maneres, no està exempta de problemes, ja que aquestes categories no són, a la pràctica, totalment ortogonals.

En la taula 2.1 distingim les tres categories i els principals models de RI que s'utilitzen habitualment, s'han experimentat o han estat simplement formulats. Tots ells es basen en l'existència d'un índex invertit que conté els termes de la col·lecció i totes les seves ocurrències, manipulats per una taula hashing o una estructura d'arbre, de manera que s'evita una cerca seqüencial. Tanmateix existeixen propostes [U. Manber, S. Wu, 1993] que exploren de manera optimitzada aquesta possibilitat i fins i tot la utilitzen, com en el sistema GLIMPSE.

Model conceptual	**Tipus d'algorismes**
Teoria de conjunts	• *Model de Boole* • *Model de lògica difusa* • *Model estès de Boole*
Algebraics lineals	• Espai vectorial • *Espai vectorial generalitzat* • *Indexació semàntica latent (LSI): SVD*
Probabilístic	• *D'independència binària (BIR)* • *Regressió logística* • *Xarxa d'inferències* • *Xarxa de creences*

Taula 2.1

Si fem una ràpida revisió de la bibliografia que aborda de manera general els models de RI, trobem que a principis de 1960 Maron i Kuhns [1960] ja havien discutit la qüestió de la rellevància i la indexació basada en mètodes probabilístics, en front de les limitacions de la lògica de Boole. L'any 1983, Salton i McGill escrigueren

«*Introduction to Modern Information Retrieval*». que es convertí en un clàssic de la recuperació de la informació durant molts anys tractant el model de Boole, vectorial i probabilístic. Una altra referència important és la monografia de Rijsbergen «*Information Retrieval*» [1975][2] que, a més de cobrir els tres mètodes clàssics, presenta també una discussió notable sobre el model probabilístic. L'any 1992 Frakes i Baeza-Yates [1992] editen una recopilació sobre diverses estructures de dades i algorismes utilitzats en RI. Un dels capítols, escrit per Donna Harman [1992a], inclou una discussió sobre la retroalimentació per rellevància amb una anotacions sobre la història dels procediments en RI des de 1960 fins a 1990.

Spark Jones, Walker i Robertson a «*A probabilistic model of information retrieval: Development and status*» [1998] descriuen algunes de les múltiples influències entre els diferents models.

Finalment «*Modern Information Retrieval*» de Baeza-Yates i Ribeiro-Neto [1999] s'ha convertit en el manual de referència sobre RI tractant, no només els models clàssics, sinó models alternatius com la lògica difusa, o les xarxes d'inferència.

Conceptes generals sobre els models de RI

Els models clàssics de RI consideren que cada document pot ser descrit per un conjunt de paraules clau (keywords) anomenades termes d'indexació. Aquests constitueixen una representació del contingut semàntic del document. Generalment aquests termes són substantius, ja que aquests tenen significat per ells mateixos, mentre que altres partícules sintàctiques serveixen més aviat com a complements. Tanmateix, és del tot possible considerar tots els termes del document com a indexables.

Independentment del camí seguit quant a la selecció del termes a indexar, no és gaire difícil percebre que no tots ells son igual d'útils per descriure el contingut d'un document, ja que alguns són de naturalesa més vaga que d'altres. Decidir sobre la importància d'uns termes sobre els altres per a representar un document és una tasca complexa. Però, tot i aquestes dificultats, hi ha una sèrie de característiques fàcilment mesurables que permeten avaluar el seu potencial. Això és, per exemple, el que passa

[2] Avui en dia disponible en línia [Rijsbergen 1975/79].

amb termes que apareixen en un nombre molt elevat de documents d'una gran col·lecció i que, per tant, no aporten criteris de selecció, o al contrari, en una representació massa minsa i que redueixen insatisfactòriament el resultat d'una consulta. La manera de recollir la influència d'un terme en el document és assignant-li un pes que en quantifiqui la seva importància semàntica.
Podem definir aquests pesos i la seva relació amb els documents i termes de la següent manera:

(D2) *Sigui t el nombre total de termes que indexen una col·lecció i* k_i *un d'aquests termes qualsevol.* $K = \{k_1, k_2, ..., k_t\}$ *és el conjunt de tots els termes indexats. S'associa un pes* $w_{ij} \geq 0$ *a cada terme* k_i *d'un document* d_j*. Per a cada terme que no apareix en el document,* $w_{ij} = 0$*. A cada document* d_j *se li associa un vector de termes indexats* $\vec{\mathbf{d}}_j = (w_{1,j}, w_{2,j}, ..., w_{t,j})$*. A més, sigui* g_i *una funció que retorna el pes associat al terme* k_i *de qualsevol vector, particularment,* $g_i(\vec{\mathbf{d}}_j) = w_{ij}$*.*

La major part de sistemes de RI assumeixen que els pesos no estan correlacionats entre elles, és a dir, són independents. Aquesta simplificació redueix enormement la tasca de càlcul del pesos i accelera el còmput de la classificació jeràrquica de resultats de la cerca. Malgrat el que es podria teòricament pensar, els treballs de recerca realitzats considerant aspectes de correlació no han demostrat una millora en la classificació de resultats que justifiqui la complexitat afegida [R. Baeza-Yates, B. Ribeiro-Neto, 1999].

3. Models basats en la Teoria de Conjunts

Model de Boole

El model de Boole es basa en la teoria de conjunts i l'àlgebra de Boole. El fet que el concepte de conjunt sigui força intuïtiu, que les expressions semàntiques de l'àlgebra de Boole siguin precises i la simplicitat inherent al formalisme motivaren que aquest model rebés força atenció durant anys i que fóra adoptat per la major part dels primers sistemes automàtics de recuperació de la informació.

Aquest model considera que els termes indexats són simplement presents o absents en un document, de tal manera que els pesos són variables binàries amb valors 0 o 1. Una cerca està composada de termes vinculats per tres operadors lògics: *i* (and, $\wedge$), *o inclusiu* (or, $\vee$), i *no* (not, $\neg$); la consulta queda reduïda a una expressió de l'àlgebra de Boole que pot ser representada seguint la forma normal disjuntiva (DNF[3]).

Així tenim **(D3),** *que en un model de Boole, els pesos del termes indexats són variables binàries id est,* $w_{i,j} \in \{0, 1\}$*. Una consulta q consisteix en una expressió de Boole convencional. Sigui* $\vec{\mathbf{q}}_{DNF}$ *la forma disjuntiva normal de la cerca q. A més, siguin* $\vec{\mathbf{q}}_{CC_a}$ *els vectors que componen* $\vec{\mathbf{q}}_{DNF}$*. Llavors la similitud entre el document* d_j *i la cerca q es defineix com*

$$\textbf{(F1)}\ \ Sim(d_j, q) = \begin{cases} 1 \text{ si } \exists\, \vec{\mathbf{q}}_{CC_a} / (\vec{\mathbf{q}}_{CC_a} \in \vec{\mathbf{q}}_{DNF}) \wedge (\forall\, k_i, g_i(\vec{\mathbf{d}}_j) = g_i(\vec{\mathbf{q}}_{CC_a})) \\ 0 \text{ en cas contrari} \end{cases}$$

Si $Sim(d_j, q) = 1$*, llavors el model prediu que el document* d_j *és rellevant per a la cerca q. En cas contrari el considera no rellevant.*

El model de Boole es basa en una decisió binària sobre la rellevància o no d'un document per a una cerca donada, sense contemplar cap noció de graduació, cosa que el fa més apte per recuperar dades que no pas informació.

[3] Acrònim de Disjunctive Normal Form.

A més a més, no és una tasca simple convertir una necessitat d'informació en una expressió de sintaxi rígida.

A nivell bibliogràfic, les restriccions dels operadors de Boole foren identificats fa ja més de 30 anys [Verhoeff, Goffman, Belzer, 1961]. La qüestió d'adaptar el formalisme de Boole a d'altres contextos ha rebut gran atenció. Bookstein discutí el problema de barrejar-lo amb sistemes ponderats [Bookstein, 1978] i basats en estimacions probabilístiques [Bookstein, 1985]. Loose i el mateix Bookstein [1988] es van ocupar de l'ús de cerques amb operadors de Boole en un model probabilístic. Dos anys després [Anick et al, 1990] proposà una interfície funcionant amb llenguatges naturals per utilitzar-se en un model de Boole. Finalment, citem una proposta d'ús d'un sistema de RI seguint el model de Boole basat en tesaurus [Lee, Kim, Lee, 1993].

Model de lògica difusa

Aquest model es basa en la elasticitat de la lògica difusa per resoldre problemes en els quals les relacions entre conceptes són poc definides.

Podem establir el nostre model considerant que cada terme de la cerca defineix un conjunt difús i que cada document hi pertany en més o menys intensitat. Encara que existeixen diferents enfocaments per al mateix problema, nosaltres seguirem el proposat per Ogawa, Morita i Kobayashi [1991].

(D4) *Sigui N el nombre total de documents en la col·lecció, n_i el nombre de documents en els quals apareix el terme k_i i $n_{i,l}$ el nombre en què apareixen els termes k_i i k_l.*

Llavors definim el coeficient de correlació $c_{i,l}$ entre els termes k_i i k_l com:

$$\textbf{(F2)} \qquad c_{i,l} = \frac{n_{i,l}}{n_i + n_l + n_{i,l}}$$

Sigui $u_{i,j}$ una funció de pertinença difusa (i. e. $u_{i,j} \in [0,1]$) del document d_j a la classe descrita per l'índex k_i. i calculada mitjançant:

(F3) $u_{i,j} = 1 - \prod_{k_k \in d_j} (1 - c_{i,l})$

Sigui com abans $\vec{q}_{DNF}$ *la forma disjuntiva normal de la cerca q. A més, siguin* $\vec{\mathbf{q}}_{CC_a}$ *els vectors que composen* $\vec{q}_{DNF}$ *. Llavors la similitud entre el document* d_j *i la cerca q es defineix com*

(F4) $Sim(d_j, q) = 1 - \prod_{\vec{\mathbf{q}}_{CC_a} \in \vec{\mathbf{q}}_{DNF}} (1 - (\prod_{g_l(\vec{\mathbf{q}}_{CC_a})=1} u_{l,j}) \cdot (\prod_{g_l(\vec{\mathbf{q}}_{CC_a})=0} (1-u_{l,j}))$

Els orígens de l'ús de la lògica difusa en RI es remunten als anys 70 amb el treball de Radecki [1976, 1979], Sachs [1976] i Tahani [1976]. Bookstein [1980] va proposar la utilització d'aquests tipus d'operadors per a ponderar els resultats en cerques amb operadors de Boole. Kraft i Buel [1983] van anar més lluny i van generalitzar-ne tot el model de Boole. Miyamoto, Miyake i Nakayama [1983] van discutir la generació de tesaurus usant concurrències i lògica difusa i el seu ús en un SRI [Miyamoto, Miyake, 1986]. Més modernament Ogawa, Morita i Kobayashi [1991] desenvoluparen el model presentat aquí.

Model estès de Boole

El model de Boole és molt simple i elegant, però a més de les dificultats per descriure equacions precises per part de l'usuari no admet una ponderació de termes ni conseqüentment una jerarquització de resultats.
Degut a això aquest model ja no s'acostuma a presentar com a única alternativa, de manera que els nous sistemes incorporen en el seu nucli alguna forma de recuperació vectorial que permet una manipulació simple i relativament ràpida. Un exemple dels models proposats és l'anomenat Model estès de Boole que combina, característiques de la teoria de conjunts amb models algèbrics no necessàriament lineals.

(D5) *Siguin* $w_{i,j} \in [0, 1]$ *el pes del terme* k_i *d'un document* d_j*. Aquests valors es poden obtenir a partir del càlcul normalitzat tf-idf (veure infra p. 17 i 18)*

(F5) $$w_{i,j} = f_{i,j} \cdot \frac{idf_i}{\max_l(idf_l)}$$

on $f_{i,j}$ es la freqüència normalitzada del terme k_i en el document d_j i idf_i és un valor associat a la inversa de la freqüència d'aparició del terme k_i.

Sigui la cerca q agrupada com una reunió d'interseccions, és a dir $q = (k_{a1} \wedge k_{b1} \wedge k_{c1} \wedge \ldots \wedge k_{t1}) \vee (k_{a2} \wedge k_{b2} \wedge k_{c2} \wedge \ldots \wedge k_{t2}) \vee \ldots \vee (k_{as} \wedge k_{bs} \wedge k_{cs} \wedge \ldots \wedge k_{ts})$.

Llavors

(F6) $$Sim(d_j, q) = \left(\frac{\sum_{i=1}^{s} \left[1 - \left(\frac{(1-w_{ai,d})^p + (1-w_{bi,d})^p + \ldots + (1-w_{ti,d})^p}{t_t} \right)^{\frac{1}{p}} \right]^p}{s} \right)^{\frac{1}{p}}$$

P és la norma utilitzada. En el cas que p = 1, el model es redueix al vectorial tf-idf tractat després. En el cas que p = ∞, llavors el model equival al de lògica difusa tractat supra. La norma p = 2 es correspon amb la norma euclidiana.

El sistema, a més, permet realitzar càlculs combinats utilitzant normes p diferents per a cadascun dels operadors que intervenen. Tot això fa que aquest model, tot i no haver-se gairebé utilitzat, sigui un marc amb moltes possibilitats teòriques.

El model de Boole estès fou suggerit per Salton, Fox i Wu [1983]. J. H. Lee, W. Y. Kim, M. H. Kim, i Y. J. Lee avaluaren els operadors de Boole amb el model estès [1993], mentre que les propietats del model foren discutides a [J. H. Lee , 1994] .

4. Models algebraics lineals

Model clàssic d'espai vectorial

L'anomenat model vectorial és segurament el més popular dins de la recerca en RI. Bona part d'aquest coneixement es deu al dilatat treball desenvolupat per Salton i el seus col·laboradors amb el sistema de RI SMART[4], desenvolupat a la Cornell University.

Els models d'espai vectorial es basen en la idea d'assignar a cada document un vector n-dimensional les components del qual són, als models clàssics, els termes del document. Les cerques també es representen amb un vector del mateix espai. Cada component és una variable contínua anomenada pes.

Per comprovar el grau de similitud entre els vectors que identifiquen la cerca i el document es fa servir una correlació quantificada mitjançant el producte escalar d'ambdós vectors. Ordenant els documents recuperats que superin un cert grau de similitud obtenim un ordre jeràrquic de rellevància.

(D6) *Per al model vectorial (correlació del cosinus), els pesos* $w_{i,j} \geq 0$ *associats a la parella* (k_i, d_j) *són variables contínues. A més, als termes de la cerca també se'ls hi assigna pesos. Sigui* $w_{i,q} \geq 0$ *el pes associat a la parella* (k_i, q)*. Llavors, el vector que representa la cerca* $\vec{q}$ *es defineix de manera que* $\vec{q} = (w_{1,q}, w_{2,q}, \ldots, w_{t,q})$ *on t, segons* **(D2)**, *és el nombre total de termes indexats en la col·lecció. Igualment, com abans, el vector per al document* d_j *és representat per* $\vec{d}_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$.

Aleshores la similitud es pot calcular de la següent manera:

$$\textit{(F7)} \quad Sim(d_j, q) = \frac{\vec{d}_j * \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|} = \frac{\sum_{i=1}^{t} w_{i,j} \cdot w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \cdot \sqrt{\sum_{i=1}^{t} w_{i,q}^2}}$$

[4] Veure infra pàgina 49.

*on * representa el producte escalar de dos vectors, · el producte ordinari de nombres reals i* $\left|\vec{\mathbf{d}}_j\right|$ *i* $\left|\vec{\mathbf{q}}\right|$ *els mòduls o normes euclidianes dels vectors del document i la cerca respectivament.*

De fet existeixen altres fórmules a la literatura per calcular la similitud tot i que la correlació del cosinus és la més utilitzada.

El mòdul del vector cerca no afecta a l'ordre de rellevància, ja que donada una consulta, és igual per a tots els documents. La norma del vector document introdueix una normalització a l'espai vectorial dels documents, de manera que sigui la similitud entre els pesos del termes la que faci el producte més gran i no merament un valor més alt degut a una major extensió del document. La introducció dels mòduls equival matemàticament a calcular el cosinus de l'angle que formen els vectors a l'espai de dimensió t.

En la següent taula en detallem quatre alternatives més:

Mesura de similitud	***Expressió de càlcul***
Producte escalar	$Sim(d_j, q) = \sum_{i=1}^{t} w_{i,j} \cdot w_{i,q}$
Coeficient de Dice	$Sim(d_j, q) = \frac{2\sum_{i=1}^{t} w_{i,j} \cdot w_{i,q}}{\sum_{i=1}^{t} w_{i,j}^2 + \sum_{i=1}^{t} w_{i,q}^2}$
Coeficient de Jaccard	$Sim(d_j, q) = \frac{\sum_{i=1}^{t} w_{i,j} \cdot w_{i,q}}{\sum_{i=1}^{t} w_{i,j}^2 + \sum_{i=1}^{t} w_{i,q}^2 - \sum_{i=1}^{t} w_{i,j} \cdot w_{i,q}}$
Coeficient de superposició o de Simpson	$Sim(d_j, q) = \frac{\sum_{i=1}^{t} w_{i,j} \cdot w_{i,q}}{\min\left(\sum_{i=1}^{t} w_{i,j}^2, \sum_{i=1}^{t} w_{i,q}^2\right)}$

Taula 2.2

El sistema és simple i elegant; proporciona una estratègia de jerarquització molt flexible i amb rendiments comparables, almenys, als de sistemes de càlcul més feixucs [R. Baeza-Yates, B. Ribeiro-Neto, 1999]. Per aquesta raó és un model força utilitzat.

Per establir la ponderació dels pesos $w_{i,j}$ i $w_{i,q}$ es fan servir diverses estratègies. Una de les més simples possibles és, de nou, considerar-les variables binàries, – i. e. atendre només a la presència o no del terme dins del document o la cerca– . Però els resultats no són massa satisfactoris i s'han desenvolupat mesures més acurades, com l'esquema tf-idf.

En aquest plantejament cal quantificar, d'una banda, la freqüència d'aparició d'un terme dins d'un document. Aquest paràmetre habitualment es coneix com a factor de freqüència del terme (*tf*) i es considera que dóna una mesura de fins a quin punt descriu aquest terme el contingut del document. És a dir, que com més vegades apareix un terme en un document, més pes semàntic tingui.

Tanmateix, els termes molt corrents gairebé no aporten capacitat de distingir si un document és o no pertinent per a una cerca concreta. Per tal motiu s'introdueix un factor calculat a partir d'una relació inversa respecte la freqüència d'aparició del terme dins dels documents de la col·lecció (freqüència inversa de documents, *idf*) .

El model més habitual seguint aquest esquema tf-idf ve descrit de la manera següent[5]:

(D7) *Sigui N el nombre total de documents en la col·lecció i n_i el nombre de documents en els quals apareix el terme k_i. Sigui $F_{i,j}$ la freqüència absoluta d'aparició del terme k_i en el document d_j. Llavors definim la freqüència normalitzada $f_{i,j}$ del terme k_i (factor tf) en el document d_j com*

$$\textbf{(F8)} \qquad f_{i,j} = \frac{F_{i,j}}{\max_l (F_{l,j})}$$

on el denominador ($\max_l$) correspon al màxim de totes les freqüències $F_{i,j}$. Si un terme k_i no apareix en el document d_j, llavors $f_{i,j} = 0$.

[5] Tanmateix existeixen altres maneres de fer aquest càlcul com la suggerida a [Harman, Candela, 1990] i incorporada al motor de cerques PRISE.

Sigui a més idf_i, un factor de freqüència inversa de presència del terme k_i en documents de la col·lecció definit

(F9) $$idf_i = \log\left(\frac{N}{n_i}\right)$$

Llavors els pesos del model es calculen

(F10) $$w_{i,j} = f_{i,j} \cdot idf_i = \frac{F_{i,j}}{\max_l (F_{l,j})} \cdot \log\left(\frac{N}{n_i}\right)$$

Sigui $F_{i,q}$ la freqüència d'aparició del terme k_i en el text de la cerca q. Podem definir la freqüència normalitzada $f_{i,q}$ del terme k_i de la cerca q com

(F11) $$f_{i,q} = 0{,}5 + \frac{0{,}5 \cdot F_{i,q}}{\max_l (F_{l,q})}$$

de manera que el valor mínim queda fitat a un domini entre 0,5 i 1 (i. e. tots els valors estan formalment representats a la cerca).

Així els pesos s'expressen introduint el factor idf_i

(F12) $$w_{i,q} = f_{i,q} \cdot idf_i = \left(0{,}5 + \frac{0{,}5 \cdot F_{i,q}}{\max_l (F_{l,q})}\right) \cdot \log\left(\frac{N}{n_i}\right)$$

Cal fer notar que l'ús de ***(F9)*** només és possible si n_i mai pot prendre el valor N. Això exclou considerar termes que apareguin en tots els documents. En cas contrari, cal afegir-hi algun terme que elimini la possibilitat que el logaritme deixi de tenir un valor finit.

Un esquema variant és el de les xarxes neurals en que a partir dels pesos inicials es segueix un procés iteratiu similar anàleg a la retroalimentació per rellevància, cosa que queda fora del nostre estudi. Tanmateix assenyalar que no constitueix un altre model ja que els pesos inicials i la similitud es calculen igual, però la iteració amb la xarxa modifica aquests valors de manera que al final poden tenir valors diferents de zero pesos de termes inexistents en un document.

Si repassem la bibliografia, H. P Luhn l'any 1953 i de nou al 1957 fou el primer a introduir teòricament el concepte d'espai vectorial a la RI, que incloïa ja molts dels aspectes considerats actualment. Però hi hauria que esperar fins als anys 60 per tal que es desenvolupés.
Salton i Lesk [1968] inicialment utilitzaren simplement termes ponderats (tf). Sparck Jones [1972] va introduir el factor idf i Salton i Yang [1973] varen demostrar la seva eficàcia millorant la recuperació. Yu i Salton [1976] a més van estudiar l'efecte de la ponderació de termes en la presentació jeràrquica final segons la rellevància dels documents. G. Salton i M. J. McGill documentaren diverses variants per al càlcul de pesos. L'any 1988 Salton i Buckley sintetitzaren 20 anys d'experiència amb el SMART. Per la seva banda, Raghavan i Wong [1986] van elaborar una anàlisi crítica del model vectorial.
La introducció conceptual de les xarxes neuronals es deu a Wilkinson i Hingston [1991]. Les aplicacions pràctiques més conegudes procedeixen de Kwok [1995] i el seus col·legues desenvolupades a través del sistema experimental multilíngüe PIRCS.

Model d'espai vectorial generalitzat

Aquest model proposa un marc elegant i formal per a representar les dependències existents entre termes en lloc de considerar-los independents. Fou introduït per S. K. Wong, Ziarko i C. N. Wong [1985] encara que no ha esta gaire utilitzat.

(D8) *Sigui* $\vec{\mathbf{k}}_i$ *un vector associat amb el terme* k_i*. Considerem els t termes que indexen una col·lecció. Llavors direm que són independents si i només si el conjunt de*

t vectors $\{\vec{\mathbf{k}}_1, \vec{\mathbf{k}}_2, ..., \vec{\mathbf{k}}_t\}$ *són linealment independents i per tant constitueixen una base del subespai de dimensió t.*

Fem notar que de manera general la independència lineal de cada vector no implica la ortogonalitat entre ells, tal i com assumeix el model vectorial més simple. Això es pot veure fàcilment si considerem

(F13) $\vec{\mathrm{d}_j} = \sum_{i=1}^{t} w_{i,j}\vec{\mathbf{k}}_i$

(F14) $\vec{\mathrm{q}} = \sum_{l=1}^{t} w_{l,q}\vec{\mathbf{k}}_l$

llavors podem calcular la similitud entre un document i la cerca

(F 15) $Sim(d_j, q) = \dfrac{\vec{\mathbf{d}_j} * \vec{\mathbf{q}}}{\left|\vec{\mathbf{d}_j}\right| \cdot \left|\vec{\mathbf{q}}\right|} = \dfrac{\sum_{i=1}^{t} w_{i,j}\vec{\mathbf{k}}_i * \sum_{l=1}^{t} w_{l,q}\vec{\mathbf{k}}_l}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \cdot \sqrt{\sum_{l=1}^{t} w_{l,q}^2}}$

que només coincideix amb ***(F7)*** si per a qualsevol parell d'índex k_i i k_j, amb i ≠ j llavors $\vec{\mathbf{k}}_i * \vec{\mathbf{k}}_j = 0$.

(D9) *Donat el conjunt de termes que indexen una col·lecció* $\{k_1, k_2, ..., k_t\}$*, sigui* $w_{i,j}$ *el pes associat a cada parella terme-document* (k_i, d_j) *que tornen a ser binaris (i. e.* $w_{i,j} \in \{0, 1\}$*). Llavors es possible representar tots el casos de coincidència de termes dins dels documents com* 2^t *productes canònics (minterms) donats per* $m_1 = (0,0,...0)$*,* $m_2 = (1,0,...0)$*, ...* $m_{2^t} = (1,1,...,1)$*, id est* m_1 *es refereix a un document sense cap terme;* m_2 *al que només té el terme* k_1 *i així successivament fins que* m_{2^t} *indica el document amb tots el termes. Sigui també* g_i *la funció lògica tal que* $g_i(m_j)$ *retorna el pes binari del terme* k_i *en el minterme* m_j*.*

Siguin $\{\vec{\mathbf{m}}_1, \vec{\mathbf{m}}_2, ..., \vec{\mathbf{m}}_{2^t}\}$ *un conjunt de vectors que formen una base ortonormal de dimensió* 2^t *associats respectivament als mintermes* $\{m_1, m_2, ..., m_{2^t}\}$ *i definits com* $\{\vec{\mathbf{m}}_1 = (1,0,...,0), \vec{\mathbf{m}}_2 = (0,1,...,0), ... \vec{\mathbf{m}}_{2^t} = (0,0,...,1)\}$.

Llavors el vector $\vec{\mathbf{k}}_i$ *associat amb el terme* k_i *s'obté*

$$\textbf{(F16)} \quad \vec{\mathbf{k}}_i = \frac{\sum_{\forall r / g_i(m_r)=1} c_{i,r} \vec{\mathbf{m}}_r}{\sqrt{\sum_{\forall r / g_i(m_r)=1} c_{i,r}^2}}$$

on els factors de correlació $c_{i,r}$ *es calculen*

$$\textbf{(F17)} \quad c_{i,r} = \sum_{\forall d_j / \forall l \quad g_l(\mathbf{d}_j)=g(m_r)} w_{i,j}$$

i la similitud es calcula segons ***(F15)***.

Tot i que aquest mètode amplia l'àmbit conceptual del model vectorial, no ha estat gaire tractat bibliogràficament i l'ús experimental és insignificant. El fet de considerar la dependència entre termes, lluny de millorar l'eficàcia podria ser un desavantatge segons alguns autors. Donada la localitat que es produeix sovint en la dependència entre termes, extrapolar-la a tota la col·lecció podria afectar-ne al rendiment global.

Model d'indexació semàntica latent (LSI)

La tasca de recuperació d'informació a partir de la descripció de documents i consultes mitjançant un conjunt de termes indexats independents pot produir pobres resultats per raons lingüístiques. D'una banda, és fàcil que apareguin documents que siguin semànticament poc rellevants, degut especialment a fenòmens com la polisèmia. Altres situacions, com la sinonímia, poden provocar una recuperació poc exhaustiva. Però això no és tot, ja que com que els termes es consideren independents, no és possible considerar-ne les relacions que s'estableixen entre ells.

El model d'indexació semàntica latent (LSI) afronta aquests problemes a partir de descriure més aviat amb conceptes que no pas amb termes. Això hauria de permetre recuperar documents relacionats, encara que potser parcialment, amb el contingut de la cerca, però que no contenen els termes inclosos en la consulta.

La idea central dels algorismes basats en LSI és descriure el document i la cerca amb un vector d'un espai de dimensió reduïda associat amb els conceptes que contenen i no amn els termes que hi apareixen.

La principal tècnica desenvolupada es coneix com a descomposició en valors singulars (SVD), que és la que presentem tot seguit.

(D10) *Sigui t el nombre de termes de l'índex dins la col·lecció i N el volum total de documents. Identifiquem per* $\vec{q}$ *el vector que defineix la cerca. Definim* $M = (M_{i,j})$ *de dimensió* $t \cdot N$ *on les files es corresponen amb el termes i les columnes amb els documents, tots ells normalitzats amb la norma euclidiana. Sigui* $r = \min(t, N)$ *el rang de la matriu M. A cada element* $M_{i,j}$ *d'aquesta matriu se li assigna un pes* $w_{i,j}$*, associat a la parella terme-document* $[ki, d_j]$*. Aquests pesos es poden calcular mitjançant* ***(F10)*** *o una altra variant del model vectorial. S'aplica el mateix procediment a la cerca de manera que* $\vec{q} = (w_{1,q}, w_{2,q}, ..., w_{t,j})$*.*

Aleshores podem assegurar que existeixen dos matrius ortogonals T_0 $t \cdot t$ *i* D_0 $N \cdot N$ *i una altra diagonal* S_0 $t \cdot (N)$ *amb tots els valors no negatius de manera que (descomposició en valors singulars, SVD)*

(F18) $M = TSD^t$

T_o *és la matriu formada pels vectors propis de la matriu de correlació de termes* $M_o \cdot M_0^t$*. Igualment* D_0^t *és la matriu de vectors propis de la matriu* $M_0^t \cdot M_0$*.*

Dels elements de la diagonal de S_0 *anomenats valor singulars –els únic no nuls –sols n'hi ha r (el rang d'*M_0*) que no s'anul·len. A més s'escull obtenir-los ordenats:*

(F19) $S = diag(s_1, s_2, ..., s_r, 0, ...0),\ s_1 \geq s_2 \geq ... \geq s_r \geq 0$

Sigui S_0' la matriu diagonal formada pels primers k termes més grans de S_o i tots els altres termes nuls.

(F20) $S'=diag(s_1,s_2,...,s_k,0,...0),\ s_1 \geq s_2 \geq ... \geq s_k \geq 0 \qquad k < r$

Llavors la matriu M_o' obtinguda pel producte amb T i D^t de S' és la matriu de rang k més propera calculant mínims quadrats a M.

(F21) $M' = TS'D^t$

Definim un nova matriu S_r k·k diagonal formada pels elements no nuls de S'

(F22) $S_k=diag(s_1,s_2,...,s_k)\ s_1 \geq s_2 \geq ... \geq s_k \geq 0$

Definim les matrius T_k t·k i D_k^t k·(N) eliminant les corresponents columnes i files respectivament associades als termes eliminats de S_o'.
El producte resultant

(F23) $M_k = T_k S_k D_k^t = (M^*_{i,j})$

és el model reduït de rang k que més s'aproxima per mínims quadrats a la matriu M_0 original.
La relació entre els documents es troba mitjançant

(F24) $R_d = (R_{i,j}) = M_k^t M_k = (D_k S_k)(D_k S_k)^t$

Finalment la similitud normalitzada es calcula:

(F25) $Sim(d_j, q) = \sum_{i=1}^{t} w_{i,q} \cdot M^*_{i,j}$

que es correspon al cosinus de l'angle que formen el vectors resultants de les projeccions en el nou espai dels vectors $\vec{q}$ *i* $\overrightarrow{d_j}$.

La importància del mètode recau en la reducció que es realitza en l'espai de termes identificadors que passen a convertir-se hipotèticament en conceptes. Escollir el valor de *k* resulta, per tant, crític i ha d'equilibrar dos efectes contraris. En primer lloc k ha de ser suficientment gran per tal de representar realment l'estructura de dades de cada document. Però alhora ha de ser suficientment petit per permetre filtrar els detalls irrellevants que són presents en les representacions basades en els termes indexats.

A la pràctica se sol agafar un valor comprès entre 50 i 150, típicament al voltant de 100, un valor molt llunyà tant del nombre de documents *N* com del nombre de termes indexats *t*.

El model LSI fou introduït l'any 1988 per Dumais, Furnas, Landauer, Deerwester i Harshman encara que la font principal de referència acostuma a ser [Deerwester, Dumais, Furnas, Landauer, 1990]. Dumais [1991] va observar que, en l'esquema tf-idf, l'ús de logaritmes entròpics donava millor resultats que el simple ús de les freqüències absolutes. Posteriorment Bartell, Cottrell i Belew [1992] indicaren que el model LSI pot ser interpretat com un cas especial d'anàlisis multidimensional escalar.

Diferents experiments de RI mostraren un percentatge d'eficàcia superior del LSI sobre el model vectorial que fou quantificat en un 30 % de mitjana a [Berry, Dumais, O'Brien, 1995].

S'ha observat que l'algorisme de reducció SVD implica una distribució normal o de Gauss de dades, i la matriu original podria no ajustar-se a aquest esquema de manera que podria ser més eficaç una distribució de Poisson [Manning, Schütze, 1999].

Kolda [1997, 1998] proposà un nou model matemàtic igual d'eficaç per al LSI basat en una descomposició matricial semidiscreta (SDD)[6] en lloc de la descomposició SVD per tal de fer front al problema que les tres matrius que genera el mètode poden ocupar molta més memòria que la matriu de termes original. Dowling [2002] ha demostrat la viabilitat del algorisme SDD mitjançant un prototipus experimental.

S'ha suggerit [Dhillon, Modha, 2001] un model alternatiu per reduir temps de càlcul i memòria respecte el LSI-SVD, amb el que guarda certa relació. El sistema intenta determinar també les relacions semàntiques latents entre documents, però tractant

[6] La diferència essencial es troba en les matrius T i D de ***(F18)*** ara només poden prendre valors del conjunt { –1, 0, 1}, d'aquí el nom de semidiscreta. Tanmateix requereix força més temps de càlcul per

d’agrupar-los en classes segons el seu contingut (clustering). De totes les tècniques matemàtiques desenvolupades per aquestes tasques amb caràcter general, «k-means» (k mitjanes) és la més eficaç. Bàsicament, consisteix en establir un conjunt de classes (clusters) disjunts, de manera que cada document, representat per un vector, un punt de l’espai $\mathbb{R}^k$, sigui més proper del centroide que representa la categoria que de qualsevol altre dels vectors-conceptes. Però aquesta tècnica és de classificació i cau fora del nostre abast.

a la descomposició inicial –que només cal fer un cop, en general ja que es poden calcular per separat les dades dels nous documents a afegir–, i un espai de dimensió *k* més gran.

5. Models probabilístics

Els models probabilístics tenen en comú l'intent de donar una estimació de la probabilitat que un document d_j rellevant per a l'usuari sigui recuperat mitjançant una cerca q_i.

Model probabilístic d'independència binària

El model de recuperació independent binari (BIR) intenta solucionar el problema de la RI seguint un esquema com el que segueix. Donada una cerca de l'usuari, hi ha un conjunt de documents *R* que contenen exactament els documents rellevants per a la cerca, i no cap altre. Aquest conjunt de documents són certament una resposta ideal que, si sabéssim descriure, tornaria trivial la tasca de RI. El sistema assumeix que la probabilitat de la rellevància d'un conjunt de documents només depèn de la cerca i de la representació dels documents i que la presència de cada terme és independent dels altres, d'on es justifica el nom actual del model.
Partint d'uns valors probabilístics inicials, basats en els termes que apareixen a la col·lecció, s'estableix una descripció probabilística que es presenta a judici de l'usuari. Aquest decideix quins son rellevants i quins no i, a partir d'aquesta selecció, el sistema redefineix la descripció del conjunt. Repetint el procés i suposant que és convergent l'optimització de la probabilitat de rellevància, s'espera que la descripció arribi suficientment a prop de l'objectiu ideal. La selecció de documents rellevants també pot ser realitzada automàticament.
La magnitud utilitzada per mesurar la similitud entre una cerca *q* i un document d_j és la raó de proporcionalitat entre la probabilitat que el document sigui rellevant i que no ho sigui (odd).

(D11) *Sigui* $w_{i,j} \in \{0, 1\}$ *i* $w_{i,q} \in \{0, 1\}$ *els pesos que corresponen respectivament a l'aparició de termes en el document* d_j *i la cerca q. Sigui R el conjunt de documents rellevants realment o hipotèticament. Sigui* $\overline{R}$ *el conjunt de documents no rellevants (i.e. el complement d'R com indica la notació). Sigui* $P(R|\vec{d}_j)$ *la probabilitat que el*

document d_j sigui rellevant per a la cerca q i $P(\overline{R} \mid \vec{d}_j)$ la probabilitat que no sigui rellevant per a q. Llavors la similitud entre el document d_j i la cerca q es defineix com el logaritme de la raó

$$\textbf{(F26)} \qquad sim(d_j,q) = \log \left(\frac{P(R \mid \vec{\mathbf{q}}, \vec{\mathbf{d}}_j)}{P(\overline{R} \mid \vec{\mathbf{q}}, \vec{\mathbf{d}}_j)}\right) = \log \left(\frac{P(R \mid \vec{\mathbf{d}}_j)}{P(\overline{R} \mid \vec{\mathbf{d}}_j)}\right)$$

L'última expressió la utilitzarem per a simplificar la notació, però cal entendre que totes les probabilitats queden condicionades a una cerca *q* concreta, això és, als termes que hi apareguin.

Evidentment, aquesta expressió directament no es gaire útil per la dificultat de fer-ne una estimació.

Gràcies al teorema de probabilitat total podem escriure:

$$\frac{P(R \mid \vec{\mathbf{d}}_j)}{P(\overline{R} \mid \vec{\mathbf{d}}_j)} = \frac{P(R) \cdot P(\vec{\mathbf{d}}_j \mid R)}{P(\vec{\mathbf{d}}_j)} \cdot \frac{P(\vec{\mathbf{d}}_j)}{P(\overline{R}) \cdot P(\vec{\mathbf{d}}_j \mid \overline{R})} = \frac{P(R)}{P(\overline{R})} \cdot \frac{P(\vec{\mathbf{d}}_j \mid R)}{P(\vec{\mathbf{d}}_j \mid \overline{R})}$$

on $P(\vec{\mathbf{d}}_j \mid R)$ denota la probabilitat de seleccionar el document d_j dins del conjunt *R* de documents rellevants. $P(\vec{\mathbf{d}}_j \mid \overline{R})$ calcula el complementari. *P*(R) correspon a la probabilitat que un document agafat a l'atzar dins de la col·lecció sigui rellevant. És evident que aquest valor és idèntic per a tots els documents a l'igual que el seu complementari $P(\overline{R})$, cosa que permet escriure[7]

$$\textbf{(F27)} \qquad sim(d_j,q) \sim \log \left(\frac{P(\vec{\mathbf{d}}_j \mid R)}{P(\vec{\mathbf{d}}_j \mid \overline{R})}\right)$$

Si, com abans, $\{k_1, k_2, ..., k_t\}$ és el conjunt de termes indexats en la col·lecció que considerem esdeveniments independents i també g_i la funció lògica tal que $g_i(\vec{\mathbf{d}}_j)$ retorna el valor binari de la component i del vector $\vec{\mathbf{d}}_j$, llavors ***(F27)*** es pot escriure

considerant els termes presents en cada document i els que no hi són dels que apareixen a la cerca q:

$$\frac{P(\vec{\mathbf{d}}_j \mid R)}{P(\vec{\mathbf{d}}_j \mid \overline{R})} = \frac{(\prod_{\forall i / g_i(\vec{\mathbf{d}}_j)=1} P(k_i \mid R)) \cdot (\prod_{\forall i / g_i(\vec{\mathbf{d}}_j)=0} P(\overline{k}_i \mid R))}{(\prod_{\forall i / g_i(\vec{\mathbf{d}}_j)=1} P(k_i \mid \overline{R})) \cdot (\prod_{\forall i / g_i(\vec{\mathbf{d}}_j)=0} P(\overline{k}_i \mid \overline{R}))}$$

$P(k_i \mid R)$ representa la probabilitat que el terme k_i estigui present en un document rellevant escollit a l'atzar. $P(\overline{k}_i \mid R)$ calcula la probabilitat que el terme k_i no estigui present. Les probabilitats associades a $\overline{R}$ tenen un significat anàleg. Si considerem que a més, $P(k_i \mid R) + P(\overline{k}_i \mid R) = 1$ i idènticament per a $\overline{R}$

$$\frac{P(\vec{\mathbf{d}}_j \mid R)}{P(\vec{\mathbf{d}}_j \mid \overline{R})} = \frac{(\prod_{\forall i / g_i(\vec{\mathbf{d}}_j)=1} P(k_i \mid R)) \cdot (\prod_{\forall i / g_i(\vec{\mathbf{d}}_j)=0} (1 - P(k_i \mid R)))}{(\prod_{\forall i / g_i(\vec{\mathbf{d}}_j)=1} P(k_i \mid \overline{R})) \cdot (\prod_{\forall i / g_i(\vec{\mathbf{d}}_j)=0} (1 - P(k_i \mid \overline{R})))} =$$

$$= \frac{(\prod_{\forall i / g_i(\vec{\mathbf{d}}_j)=1} P(k_i \mid R)) \cdot (\prod_{\forall i / g_i(\vec{\mathbf{d}}_j)=0} (1 - P(k_i \mid R))) \cdot (\prod_{\forall i / g_i(\vec{\mathbf{d}}_j)=1} (1 - P(k_i \mid R))(1 - P(k_i \mid \overline{R}))}{(\prod_{\forall i / g_i(\vec{\mathbf{d}}_j)=1} P(k_i \mid \overline{R})) \cdot (\prod_{\forall i / g_i(\vec{\mathbf{d}}_j)=0} (1 - P(k_i \mid \overline{R}))) \cdot (\prod_{\forall i / g_i(\vec{\mathbf{d}}_j)=1} (1 - P(k_i \mid R))(1 - P(k_i \mid \overline{R}))} =$$

Si ara recordem que les probabilitats són condicionades als termes que apareixen a la cerca

$$\frac{P(\vec{\mathbf{d}}_j \mid R)}{P(\vec{\mathbf{d}}_j \mid \overline{R})} = \frac{\prod_{\forall i / g_i(\vec{\mathbf{d}}_j)=1} P(k_i \mid R)(1 - P(k_i \mid \overline{R}))}{\prod_{\forall i / g_i(\vec{\mathbf{d}}_j)=1} P(k_i \mid \overline{R}) \cdot (1 - P(k_i \mid R))} \cdot \frac{\prod_{\forall i / g_i(\vec{\mathbf{q}})=1} (1 - P(k_i \mid R))}{\prod_{\forall i / g_i(\vec{\mathbf{q}})=1} (1 - P(k_i \mid \overline{R}))}$$

El quocient de l'esquerra de l'expressió anterior es forma a partir del producte de les probabilitats $P(\overline{k}_i \mid R)$ i $P(\overline{k}_i \mid \overline{R})$ de tots els termes de la consulta. Resulta doncs independent del document d_j i per tant irrellevant per establir-ne la jerarquia de

[7] Notar que en realitat hauríem d'escriure $P(R \mid \vec{\mathbf{q}})$ i $P(\overline{R} \mid \vec{\mathbf{q}})$ respectivament.

similitud entre els document i la cerca. Si a la vegada agrupem tots els termes en un únic producte tindrem de nou

(F28) $$sim(d_j,q) \sim \log \left(\prod_{\forall i / g_i(\vec{d}_j)=1 \ \wedge \ g_i(\vec{q})=1} \frac{P(k_i \mid R)(1-P(k_i \mid \overline{R}))}{(1-P(k_i \mid R)(P(k_i \mid \overline{R})} \right)$$

Si preferim utilitzar els pesos $w_{i,q}$ i $w_{i,j}$ que valen zero quan els termes no estan presents a la cerca q o en el document dj, el sumatori induït per la propietat bàsica dels logaritmes es pot estendre a tots els termes de la col·lecció, de manera que queda definitivament

(F29) $$sim(d_j,q) \sim \sum_{i=1}^{t} w_{i,q} \cdot w_{i,j} \cdot (\log \frac{P(k_i \mid R)}{1-P(k_i \mid R)} + \log \frac{1-P(k_i \mid \overline{R})}{P(k_i \mid \overline{R})})$$

Com que inicialment no tenim cap document recuperat hem de fer una estimació dels valors de les probabilitats $P(k_i \mid R)$ i $P(k_i \mid \overline{R})$. Hi ha diverses variants per fer això. Una de les més habitual és considerar aquest valor constant per a tots els termes k_i i igual a 0.5 i estimar que la distribució de l'índex dins del conjunt de documents no rellevants segueix aproximadament la de tota la col·lecció[8]. Així si tal i com hem definit a **(D4)** N és el nombre total de documents i n_i el que conté el terme k_i

(F30) $$P(k_i \mid R) = 0{,}5$$

(F31) $$P(k_i \mid \overline{R}) = \frac{n_i}{N}$$

Amb aquests valors el sistema recupera documents que contenen termes de la cerca i ofereix el resultats jerarquitzats per similitud. Ara el sistema pot realitzar un nou procés per millorar-ne els resultats – retroalimentació per rellevància – procedint tal i com segueix.

[8] Expressat mitjançant el càlcul de Laplace per a situacions equiprobables.

(D12) *Sigui V el subconjunt de documents recuperats inicialment i jerarquitzats segons* ***(F29)****. De tots aquests documents es pot escollir un subconjunt format pels r amb més similitud amb q, on aquest valor llindar està prèviament definit. Sigui a més* V_i *el subconjunt d'elements de V tal que llurs documents contenen l'índex* k_i*.*

Llavors, per tal de millorar el valor de $P(k_i \mid R)$ assumim que es pot aproximar molt millor amb la distribució del terme k_i entre els documents recuperats – que es suposa que són en conjunt més rellevants que no pas un conjunt qualsevol agafat a l'atzar– . De manera equivalent, podem calcular un valor aproximat de $P(k_i \mid \overline{R})$ considerant que els documents no recuperats són no rellevants. Així, les coses queden:

(F32) $$P(k_i \mid R) = \frac{V_i}{V}$$

(F33) $$P(k_i \mid \overline{R}) = \frac{n_i - V_i}{N - V}$$

El procés es pot repetir reiteradament de manera automàtica. Només cal establir un límit d'iteracions i un llindar mínim de diferència entre els resultats d'un pas i el següent per tal que quan no es superi els sistema interpreti que s'ha arribat al límit de la convergència (i. e. la millor distribució de document rellevants possible).

També és possible fer intervenir l'element humà seguint la concepció original per escollir els documents rellevants que formen en cada iteració el conjunt *V*.

(F32) i ***F(33)*** presenten problemes per a valors petits de *V* i V_i. Per evitar-ho es pot optar per sumar un valor constant 0.5 en els numeradors i 1 en els denominadors, encara que molts cops és més convenient substituir el factor d'ajustament constant 0,5 pel quocient n_i/N, cosa que fa que quedi

(F34) $$P(k_i \mid R) = \frac{V_i + \frac{n_i}{N}}{V + 1}$$

(F35) $$P(k_i \mid \overline{R}) = \frac{n_i - V_i + \frac{n_i}{N}}{N - V + 1}$$

Cal fer notar que com sigui que el sistema calcula la funció de classificació a partir de $P(K_i|R)$, $P(k_i|\overline{R})$ i els complementaris respectius, a més de establir una classificació per als documents també calcula de facto una jerarquia de termes. Això significa a la pràctica que es poden suggerir documents relacionats – retroalimentació per rellevància – i enquadrar-los dins de categories.

Per tant, sintetitzant, el procés reiteratiu el que fa és anar seleccionant els termes amb més probabilitat de ser rellevants i, a partir d'ells, els documents que són presentats segons l'ordre de rellevància.

Però no tot són avantatges, al menys des de la perspectiva teòrica. La necessitat de conjecturar la separació inicial entre conjunts de documents rellevants i no rellevants sense tenir-ne ni un espai de mostra. En segon lloc el fet que el mètode no consideri la freqüència d'aparició d'un terme en els documents, ja que tots el pesos són binaris. Això de vegades es soluciona introduint $F_{i,j}$ i algun factor per a normalitzar-ne la longitud com en el cas del sistema XAPIAN.

Finalment el fet de considerar que la presència dels termes és independent. Tanmateix, aquest últim aspecte, com ja hem esmentat en el model vectorial generalitzat, podria resultar una bona decisió a la pràctica.

La discussió bibliogràfica del model probabilístic és molt amplia. Indiquem-ne algunes de les referències essencials. Fou introduït per Robertson i Spark Jones [1976] i és discutit amb gran amplitud a Rijsbergen [1979]. Spark Jones [1979a, 1979b] va experimentar amb el model usant valors procedents de retroalimentació amb l'usuari per determinar les probabilitats inicials. Croft i Harper [1979] proposaren les estimacions que aquí hem exposat.

Més tard, Croft [1983] hi afegí un factor de ponderació basat en freqüències. Fuhr [1989] discutí les possibilitats d'ús de funcions de recuperació polinòmiques per a la indexació probabilística. Fuhr [1992] revisà les diverses variants clàssiques del model. Finalment, Cooper [1994] va posar en evidència diversos problemes relacionats amb l'ús dels resultats jerarquitzats mitjançant probabilitat en RI.

Regressió logística per etapes (SLR)

Aquesta variant probabilística construeix un model de predicció de la rellevància del documents a partir d'una regressió basada en un conjunt de dades obtingudes en un procés previ d'entrenament per etapes.

(D12) *Sigui $\{X_1, X_{2,...,}X_s\}$ un conjunt finit de paràmetres estadístics associats a les característiques de la cerca i del conjunt de documents de la col·lecció. Llavors el logaritme de la raó donat per* ***(F26)*** *es pot calcular com*

$$\textbf{(F36)} \qquad sim(d_j,q) = \log \left(\frac{P(R \mid \vec{\mathbf{q}}, \vec{\mathbf{d}}_j)}{P(\overline{R} \mid \vec{\mathbf{q}}, \vec{\mathbf{d}}_j)}\right) = c_0 + \sum_{i=1}^{S} c_i X_i$$

on $\{c_0, c_{1,...,}c_s\}$ són un conjunt de coeficients reals que depenen exclusivament de la col·lecció.

A la pràctica no cal afegir-hi el terme c_o donat que es tracta d'una constant – l'ordenada a l'origen de la funció de regressió multivariable – i, per tant, afecta per igual a tots els valors de $sim(d_j, q)$.

L'entrenament del sistema serveix per determinar-ne el coeficients c_i i, un cop establerts, el sistema estima per a cada cerca q el valor dels paràmetres estadístics concrets. Els paràmetres estadístics tenen que veure amb la quantitat i l'aparició de termes en la col·lecció, la longitud de la cerca, la mida dels documents, entre d'altres aspectes que han resultat rellevants en el procés de recerca.

Des d'un punt de vista matemàtic, el model implica una hipòtesi més atenuada que la independència binària. En aquest cas s'assumeix l'anomenada suposició de dependència vinculada, que estableix que hi ha alguna mena de relació entre el grau de dependència de les parelles rellevants (q_i, d_j) i les no rellevants.

Aquesta hipòtesi fou introduïda per Cooper [1991] en un treball crític sobre les suposicions matemàtiques assumides en models probabilístics com el BIR. Cooper mateix, junt a Gey i Dabhey [1992], van introduir el model de regressió logística que és

la base del sistema de recuperació CHESHIRE II[9] de la Universitat de Califòrnia a Berkeley.

Model de xarxa d'inferències

Es basa en una visió epistemològica del problema de la RI[10]. S'associen variables aleatòries als termes, als documents i a les cerques de l'usuari. Una variable aleatòria associada amb el document d_j representa l'esdeveniment d'observar aquest document en la cerca dels que són rellevants. Aquest fet comporta uns valors de confiança o creença sobre les variables associades amb els termes d'indexació. Així, l'observació d'un document es considera la causa per augmentar la creença en les variables associades amb els seus termes d'indexació.

Els termes i els documents es representen amb vèrtexs d'un graf dirigit. Els arcs estan dirigits des d'un document als termes per a indicar que l'observació d'un document produeix una millora en el valor de creença sobre els seus termes d'indexació.

Per la seva banda, la variable aleatòria associada a la cerca representa l'esdeveniment que s'ha trobat la informació requerida a través de la consulta. El nivell de confiança o creença en tal cerca depèn dels valors que tenen els termes que hi apareixen, cosa que significa que en el graf els arcs van des dels termes a la cerca.

El model complet de xarxa d'inferència no té necessàriament que basar-se només en documents, termes i consultes, sinó que pot incloure expressions més complexes, encara que la forma de procedir és equivalent. Tanmateix, aquí presentem el model més simple comparable amb la resta.

(D13) *Sigui* $\vec{\mathbf{k}}$ *un vector de dimensió t definit per* $\vec{\mathbf{k}} = (k_1, k_2, ...k_t)$*, on les components són variables aleatòries binàries associades als termes d'indexació de la col·lecció – defineixen per tant* 2^t *estats possibles per a* $\vec{\mathbf{k}}$ *–. Sigui, a més,* d_j *una variable*

[9] Veure infra pàgina 66.

[10] Aquesta perspectiva interpreta la probabilitat com un cert grau de creença sense que necessàriament existeixi vinculat un clar referent estadístic tal i com pressuposa comprendre la probabilitat com un cas límit de l'estadística.

aleatòria associada al document homònim i, igualment, q una variable aleatòria binària associada a la cerca de l'usuari.
Llavors, la similitud entre el document d_j i la cerca q es calcula com

(F37) $$sim(d_j,q) = P(q \wedge d_j) = \sum_{\forall \vec{\mathbf{k}}} P(q \wedge d_j \mid \vec{\mathbf{k}}) \cdot P(\vec{\mathbf{k}})$$

on d_j i q són una representació abreujada per a d_j=1 i q =1, respectivament.

El fet que els documents siguin les causes dels termes i no a la inversa fa que d_j sigui independent de tot $\vec{\mathbf{k}}$. Per tant

$$\sum_{\forall \vec{\mathbf{k}}} P(q \wedge d_j \mid \vec{\mathbf{k}}) \cdot P(\vec{\mathbf{k}}) = \sum_{\forall \vec{\mathbf{k}}} P(q \wedge d_j \wedge \vec{\mathbf{k}}) = \sum_{\forall \vec{\mathbf{k}}} P(q \mid d_j \cdot \vec{\mathbf{k}}) \cdot P(d_j \cdot \vec{\mathbf{k}})$$

Però com que els vèrtex k_i separen el corresponent al document d_j i la cerca q llavors

$$\sum_{\forall \vec{\mathbf{k}}} P(q \mid d_j \cdot \vec{\mathbf{k}}) \cdot P(d_j \cdot \vec{\mathbf{k}}) = \sum_{\forall \vec{\mathbf{k}}} P(q \mid \vec{\mathbf{k}}) \cdot P(d_j \cdot \vec{\mathbf{k}}) = \sum_{\forall \vec{\mathbf{k}}} P(q \mid \vec{\mathbf{k}}) \cdot P(\vec{\mathbf{k}} \mid d_j) \cdot P(d_j)$$

Però la dependència causal dels valors de la creença dels k_i respecte de d_j també comporta que siguin mútuament independents, així $P(\vec{\mathbf{k}} \mid d_j)$ es pot calcular com un producte, de manera que finalment ***(F37)*** queda

(F38) $$sim(d_j,q) = \sum_{\forall \vec{\mathbf{k}}} P(q \mid \vec{\mathbf{k}}) \cdot \left(\prod_{\forall i / g_i(\vec{\mathbf{k}})=1} P(k_i \mid d_j) \cdot \prod_{\forall i / g_i(\vec{\mathbf{k}})=0} P(\bar{k}_i \mid d_j) \right) \cdot P(d_j)$$

on les funcions g_i són de nou les definides a **(D2)**.
Com que els vèrtex arrel de la xarxa d'inferències són els documents, se'ls hi ha d'assignar una probabilitat a priori que reflecteixi l'esdeveniment «ser observat»[11].
La primera opció que es va proposar [Turtle, Croft, 1990] es correspon amb assignar a tots el mateix valor, que només depèn del volum de documents. Així, la probabilitat

[11] O de manera equivalent, «que es doni».

d'observar un document d_j val $1/N$, on N és el nombre de documents de la col·lecció – supra **(D4)** –. Tenim doncs

(F39) $P(d_j) = 1/N$

Aquesta condició equival a la del model de Boole, que no fa cap diferència a priori entre els documents. De fet, és possible ajustar aquest model per tal que la cerca es comporti en termes de recuperació exactament com aquest model simple. Per fer això és necessiten especificar les següents probabilitats condicionades

$$\textbf{(F40)}\quad P(k_i|d_j) = \begin{cases} 1 \text{ si } g_i(\vec{d}_j) = 1 \\ 0 \text{ en cas contrari} \end{cases}$$

$$\textbf{(F41)}\quad P(q|\vec{\mathbf{k}}) = \begin{cases} 1 \text{ si } \exists\, \vec{\mathbf{q}}_{CC_a} / (\vec{\mathbf{q}}_{CC_a} \in \vec{\mathbf{q}}_{DNF}) \wedge (\forall\, k_i, g_i(\vec{\mathbf{k}}) = g_i(\vec{\mathbf{q}}_{CC_a})) \\ 0 \text{ en cas contrari} \end{cases}$$

on $\vec{\mathbf{q}}_{CC_a}$ i $\vec{\mathbf{q}}_{DNF}$ són els definits per al model clàssic de Boole a **(D3)**.

L'expressió ***(F40)*** bàsicament indica que quan s'observa el document d_j només s'activen els vèrtex associats als termes d'indexació presents en l'esmentat document. Per la seva banda ***(F41)*** imposa que al menys un dels vectors que formen $\vec{\mathbf{q}}_{DNF}$ ha de coincidir amb els termes actius a $\vec{\mathbf{k}}$.

Una altra opció és adoptar estratègies pròpies del model vectorial com és la jerarquització per rellevància tf-idf. En aquest, la probabilitat a priori reflecteix el coneixement que tenim de la importància de la normalització del document per tal de

no privilegiar els documents de mida més gran[12]. D'aquesta manera $P(d_j)$ com la inversa de la norma euclidiana de $\vec{\mathbf{d}}_j$ –supra **(D2)**–.

(F42) $$P(d_j) = \frac{1}{\left|\vec{\mathbf{d}}_j\right|}$$

Per capturar l'impacte dels factors *tf* assignem a $P(k_i|d_j)$ el valor $f_{i,j}$ segons ***(F8)***

(F43) $$P(k_i|d_j) = f_{i,j}$$

Ara especifiquem la influència del factor *itf*. Per això necessitem calcular la contribució individual de cada k_i.

(D14) *Sigui* $\vec{\mathbf{k}}_i$ *un vector que fa referència a un estat del vector* $\vec{\mathbf{k}}$ *en què el vèrtex* k_i *és actiu i tots els altres inactius, id est* $\vec{\mathbf{k}}_i = \vec{\mathbf{k}}$ *quan* $(g_i(\vec{\mathbf{k}}) = 1 \wedge \forall_{j \neq i}\, g_j(\vec{\mathbf{k}}) = 0)$.

Ara recuperant ***(F9)*** podem definir la influència de cada vèrtex en la cerca q de la següent manera

(F44) $$P(q|\vec{\mathbf{k}}) = \begin{cases} idf_i, & si \;\; \vec{\mathbf{k}} = \vec{\mathbf{k}}_i \;\wedge\; g_j(\vec{\mathbf{q}}) = 1 \\ 0 & si \;\; \vec{\mathbf{k}} \neq \vec{\mathbf{k}}_i \;\vee\; g_j(\vec{\mathbf{q}}) = 0 \end{cases}$$

Aplicant a ***(F38)*** les expressions ***(F42)***, ***(F43)*** i ***(F44)*** podem escriure

(F45) $$sim(d_j,q) = \sum_{\forall \vec{\mathbf{k}}_i} P(g \mid \vec{\mathbf{k}}_i) \cdot P(k_i \mid d_j) \cdot \left(\prod_{\forall\, l \neq i} P(\bar{k}_l \mid d_j) \right) \cdot P(d_j) =$$

$$= \left(\prod_{\forall i} P(\bar{k}_i \mid d_j) \right) \cdot P(d_j) \cdot \left(\sum_{\forall \vec{k}_i} P(k_i \mid d_j) \cdot P(q \mid \vec{\mathbf{k}}_i) \cdot \frac{1}{P(\bar{k}_i \mid d_j)} \right) =$$

[12] En el marc de la teoria de Bayes cal tenir un coneixement previ del domini on s'aplica per especificar-ne les probabilitats a priori o causals.

$$= \left(\prod_{\forall i} P(\bar{k}_i \mid d_j)\right) \cdot \frac{1}{\left|\vec{\mathbf{d}}_j\right|} \cdot \left(\sum_{\forall i \ / \ g_i(\vec{\mathbf{d}}_j)=1 \ \wedge \ g_i(\vec{\mathbf{q}})=1} f_{i,j} \cdot idf_i \cdot \frac{1}{1-f_{i,j}} \right)$$

expressió que dóna una jerarquització per rellevància tf-idf però diferent que l'equivalent del model vectorial degut al terme inicial del producte de probabilitats $P(\bar{k}_i \mid d_j)$.

Una de les particularitats que fa més útil aquest model és poder combinar diferents procediments de cerca per tal de millorar el resultat final de jerarquització per rellevància. Així, seguint el que hem exposat podem tenir un vèrtex que es correspon amb una cerca estàndard basada em un conjunt de termes i una segona cerca formulada amb termes de Boole.

(D15) *Siguin q_i^v les cerques formulades com un conjunt de termes que formen les components d'un vector i q_i^b les formulades segons l'àlgebra de Boole. Sigui I la formulació complexa de la necessitat d'informació que inclou totes les cerques q_i^v i q_i^b, i. e. I és una reunió d'interseccions de cerques. Llavors*

(F46) $$sim(d_j,q) = P(I \wedge d_j) = \sum_{\forall \vec{\mathbf{k}}} P(I \wedge d_j \mid \vec{\mathbf{k}}) \cdot P(\vec{\mathbf{k}})$$

Per a calcular les probabilitats d'*I* només cal considerar que les cerques q_i^v i q_i^b són totes independents. Per tant, la probabilitat global de cada intersecció (operador lògic $\wedge$) es calcula mitjançant el producte de totes les $P(q_i^v)$ *i* $P(q_i^b)$ que intervenen en ella, –i . e. $\prod_{\forall i,j} P(q_i^v) \cdot P(q_j^b)$ –.

Les reunions (operador lògic $\vee$) s'obtenen calculant la probabilitat que no es doni alhora el producte de totes $P(\bar{q}_i^v)$ i $P(\bar{q}_i^b)$, –i. e. $\prod_{\forall i,j} 1 - P(\bar{q}_i^v) \cdot P(\bar{q}_j^b)$ –.

A nivell bibliogràfic l'origen del model es deu a la tesi doctoral de H. Turtle [1990]. Amb col·laboració amb el seu director de tesi W. B. Croft, publicà diversos treballs. Destaquem [Turtle, Croft, 1991] on avaluen el rendiment del model i comproven com el rendiment en la recuperació de documents millora quan s'utilitzen alhora els dos tipus de cerca en lloc de cadascuna per separat.

Haines i el mateix Croft [1993] van estudiar la possibilitat d'utilitzar aquesta tècnica per a refinar les cerques utilitzant retroalimentació per rellevància. Callan, Lu i Croft de nou [1995] usen aquest model per a la cerca de col·leccions de documents distribuïdes. L'any següent, per la seva banda, Callan [1996] va dissertar sobre el seu ús en filtrat d'informació, alhora[13] que Ribeiro-Neto i Muntz [1996] generalitzaven el concepte de xarxa d'inferència mitjançant les xarxes de creences a les quals ens referirem en el proper epígraf.

Model de xarxa de creences

El model de xarxa de creences també es basa, com l'anterior, en una interpretació epistemològica del concepte de probabilitat, però a diferència de l'anterior es fonamenta en la definició d'un precís espai mostral. Així la xarxa resultant té una topologia que separa totalment els vèrtex corresponents als documents dels de la cerca. És a dir, els arcs del graf estan dirigits des dels termes als documents i a les cerques.

(D16) *Sigui* $K = \{k_1, k_2, ...k_t\}$ *l'univers del discurs que defineix l'espai mostral per al model de xarxa de creences. Sigui també* $u \subset K$ *un subconjunt de K. A cada subconjunt u se li assigna un vector* $\vec{\mathbf{k}}$ *tal que* $g_i(\vec{\mathbf{k}}) = 1$ *si i només si* $k_i \in u$ *–amb* g_i *supra* **(D2)–**. *A més, sigui* k_i *una variable aleatòria binària associada al terme homònim* k_i. *Aquesta variable pren el valor* 1 *quan l'índex* k_i *és un membre del conjunt de conceptes de* $\vec{\mathbf{k}}$ *i en cas contrari val* 0.

[13] *Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieva*l, a Zurich.

D'aquesta manera considerem cada terme d'indexació com un concepte bàsic i tot K com l'espai de conceptes. Associem també els conceptes a subconjunts car és força útil per expressar els operadors de la lògica de Boole o la teoria de conjunts. Els documents i cerques s'incorporen a l'espai mostral tal i com segueix:

(D17) *Un document d_j de la col·lecció queda definit com un conjunt – i. e. un concepte – format pels termes que l'indexen. Anàlogament, una cerca q de l'usuari la representem com un concepte descrit a partir dels termes utilitzats per descriure-la.*

Finalment, cal definir la distribució de probabilitat dins de l'univers del discurs.

(D18) *Sigui c un concepte genèric dins de l'espai mostral K que representa un document o una cerca indistintament. Llavors definim la probabilitat de c com el grau de cobertura del concepte de l'espai K calculat*

(F47) $$P(c) = \sum_{\forall u} P(c \mid u) \cdot P(u)$$

on P(u) és la probabilitat que u esdevingui dins K.
I la probabilitat que dóna d'ordre jeràrquic per rellevància

(F48) $$sim(d_j,q) = P(d_j|q)$$

(F47) expressa el grau de cobertura contrastant cada concepte de *K* amb *c* i sumant-ne totes les contribucions ponderades per la probabilitat que cada concepte es doni a *K*. Però com que no és possible inicialment conèixer la probabilitat que un concepte esdevingui en l'espai *K* , cal acceptar que tots ells són equiprobables, de tal manera que

(F49) $$P(u) = \left(\frac{1}{2}\right)^t$$

(F48) es justifica a partir del teorema de Bayes. Si definim la funció jerarquitzadora a partir de $P(d_j \wedge q)$ com a ***(F37)***, considerant que $P(d_j|q) = P(d_j \wedge q)/P(q)$ i que $P(q)$ és

constant per a tots els documents de la col·lecció llavors tenim que $P(d_j|q)$ i $P(d_j \wedge q)$ són proporcionals, cosa que justifica escollir la funció més simple possible.
Combinant ***(F47)*** i ***(F48)*** tenim

$$\textbf{(F50)} \qquad sim(d_j,q) = P(d_j|q) = \sum_{\forall u} P(d_j \wedge q \mid u) \cdot P(u)$$

Ara bé, com que tant els vèrtex dels documents com els de les cerques deriven dels termes, *q* i *dj* són mútuament independents aleshores recordant la relació entre *u* i $\vec{\mathbf{k}}$

$$\sum_{\forall u} P(d_j \wedge q \mid u) \cdot P(u) = \sum_{\forall u} P(d_j \mid u) \cdot P(q \mid u) \cdot P(u) = \sum_{\forall \vec{\mathbf{k}}} P(d_j \mid \vec{\mathbf{k}}) \cdot P(q \mid \vec{\mathbf{k}}) \cdot P(\vec{\mathbf{k}})$$

i per tant

$$\textbf{(F51)} \qquad sim(d_j,q) = \sum_{\forall \vec{\mathbf{k}}} P(d_j \mid \vec{\mathbf{k}}) \cdot P(q \mid \vec{\mathbf{k}}) \cdot P(\vec{\mathbf{k}})$$

La manera de calcular les probabilitats condicionades $P(d_j \mid \vec{\mathbf{k}})$ i $P(q \mid \vec{\mathbf{k}})$ permet modelar diferents estratègies corresponents a models d'IR diferents.
Per al model vectorial ho especifiquem de la següent manera.

$$\textbf{(F52)} \quad P(q|\vec{\mathbf{k}}) = \begin{cases} \dfrac{w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,q}^2}} & si \ \vec{\mathbf{k}} = \vec{\mathbf{k}}_i \wedge g_j(\vec{\mathbf{q}}) = 1 \\ 0 & en\ cas\ contrari \end{cases}$$

$$\textbf{(F53)} \quad P(d_j|\vec{\mathbf{k}}) = \begin{cases} \dfrac{w_{i,j}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2}} & si \ \vec{\mathbf{k}} = \vec{\mathbf{k}}_i \wedge g_j(\vec{\mathbf{d}_j}) = 1 \\ 0 & en\ cas\ contrari \end{cases}$$

segons les definicions i notació ja assenyalats a **(D6)** i **(D14)**.

Amb aquestes probabilitats s'obtenen exactament els mateixos resultats que amb el model clàssic vectorial, cosa que no es podia fer amb el model de xarxa d'inferències.

Podem comparar amb més profunditat els dos models. Els càlculs en les xarxes d'inferències depenen del terme $P(\vec{\mathbf{k}}|d_j)$; mentre que en els de xarxes de creences la dependència és amb $P(d_j|\ \vec{\mathbf{k}})$. El primer cas sempre pot ser calculat com un producte de les probabilitats individuals $P(k_i|d_j)$, donada la independència mútua dels k_i. En canvi, a $P(d_j|\ \vec{\mathbf{k}})$ no serà possible, en general, la seva descomposició en un producte de probabilitats basades en termes. Com a conseqüència, $P(d_j|\vec{\mathbf{k}})$, pot expressar qualsevol funció de probabilitat definida per $P(\vec{\mathbf{k}}|d_j)$, però el cas a l'inrevés no és cert. Per tant, les xarxes d'inferència són un cas particular de les xarxes de creences. Això significa que aquestes últimes permeten reproduir els resultats de les primeres, inclòs les cerques basades alhora en conjunts de termes i de Boole, a més d'oferir el model vectorial i el model de BIR entre d'altres possibilitats. Tot i la potència conceptual del model, encara no ha estat gaire explotat.

6. Motors de RI de text complet d'ús genèric

Introducció

Aquest capítol està dedicat a l'estudi de les opcions de models de RI per les quals han optat diversos sistemes, comercials o no, totalment desenvolupats i a punt per a ser utilitzats[14], que siguin de propòsit general, és a dir, no vinculats a un gestor de bases de dades o proveïdor d'informació textual concret, ni incorporat a qualsevol altre programari.

Hi ha nombrosos sistemes experimentals en diversos laboratoris acadèmics arreu del planeta. Tanmateix són això, sistemes que serveixen per fer proves cercant millors rendiments i no preparats per a ser instal·lats i funcionar.

Respecte els proveïdors d'informació, és força difícil, sinó impossible, aconseguir informació mínimament detallada sobre el motors de cerca de proveïdors com Dialog[15] o LexisNexis[16]. A més, aquests sistemes de recuperació d'informació acostumen a incloure opcions de cerca a partir de serveis de valor afegit preparats per llurs àrees de documentació, tal com tesaurus, directoris o mapes conceptuals que s'interrelacionen amb les prestacions dels models de RI. Aquestes dificultats ens portaren des d'un bon principi a negligir-los en el nostre estudi.

Altres productes[17] són gestors de bases de dades, que integren tant informació estructurada –bases de dades relacionals–, com no estructurada, organitzant-se la RI en text complet en camps estructurats que es creen, amb més o menys intervenció humana, en el procés d'indexació. Només hem inclòs un cas que podríem considerar d'aquesta categoria, VERITY ULTRASEEK, ja que és resultat de la convergència d'una empresa amb un passat en desenvolupament de sistemes de RI i d'un producte amb tal finalitats.

---

[14] Per a una perspectiva espanyola recent d'alguns d'aquests projectes: [Sanchis, Moreno, Gil, 2002]. Un cas a part és Karpanta, sistema de RI desenvolupat a la Universitat de Salamanca, basat en el model vectorial sobre una base de dades relacional interrogada amb SQL i codificació en Visual Basic, dissenyada per a usos bàsicament docents. Una implementació en línia està disponible a: <http://milano.usal.es/dtt.htm>. Documentació: <http://www.ub.es/biblio/bid/04figue1.htm>. [Consulta: 23 de juny de 2003].

[15] <http://www.dialog.com>. [Consulta: 8 de maig de 2003].

[16] <http://www.lexisnexis.com>. [Consulta: 8 de maig de 2003].

[17] És el cas, per exemple, de RetrievalWare, (http://www.convera.com/Products/products_rw.asp>). [Consulta: 8 de maig de 2003].

Tampoc ens ocuparem aquí d'aquells enginys de cerca dissenyats expressament per indexar i recuperar documents d'Internet, com és el cas de l'AltaVista Search Search[18] o del Google Search Appliance[19]. Ni de projectes emparentats amb els motors de cerca, com l'URSA[20], un sistema basat en un model algebraic i capaç de processar texts i recuperar informació transparent als llenguatges. O el PIRCS[21], que barreja càlculs propis del model probabilístic, basat en xarxes d'inferències de Bayes i models lingüístics, que va més enllà de l'àmbit conceptual que aquí hem tractat[22].

Queden fora també els kits de desenvolupament de RI (SDK, Software Development Kit), eines que utilitzen en aquest àmbit els programadors per afegir un motor de cerca a aplicacions[23].

Finalment, no hem inclòs en aquestes pàgines enginys de RI pensats per a diverses morfologies de la informació, encara que també puguin incloure el text.

Respecte als motors de RI d'ús genèric, encara que segurament no hi seran presents tots, considerem que és suficientment exhaustiu l'inventari com per a representar quina ha estat la implementació real dels models matemàtics en sistemes de RI concrets, més enllà dels nombrosos experiments de laboratori.

Hem intentat situar cada sistema estudiat dins del seu context cronològic i històric. Així mateix, es descriuen les característiques generals de tipus informàtic, com ara el llenguatge de codificació, el sistema operatiu que suporten, l'arquitectura i d'altres aspectes relacionats. També es donen dades per a la seva adquisició i s'esmenten els detalls coneguts sobre el model d'IR. Tanmateix, en general, no es citen d'altres característiques habituals, com els lematitzadors en una o més llengües (stemmers), o els diccionaris de paraules buides (stopwords). Tampoc s'aborden aspectes relacionats amb la manera de construir i administrar els fitxers invertits i altres diccionaris complementaris que duguin, a no ser que sigui absolutament rellevant.

---

[18] Versions Desktop o Interprise: <http://solutions.altavista.com/en/products/index.shtml>. [Consulta: 8 de maig de 2003].

[19] <http://www.google.com/appliance/product_info.html>. [Consulta: 8 de maig de 2003]

[20] Unicode Retrieval System Architecture. Web del projecte: <http://crl.nmsu.edu/Research/Projects/tipster/ursa> [Consulta: 8 de maig de 2003].

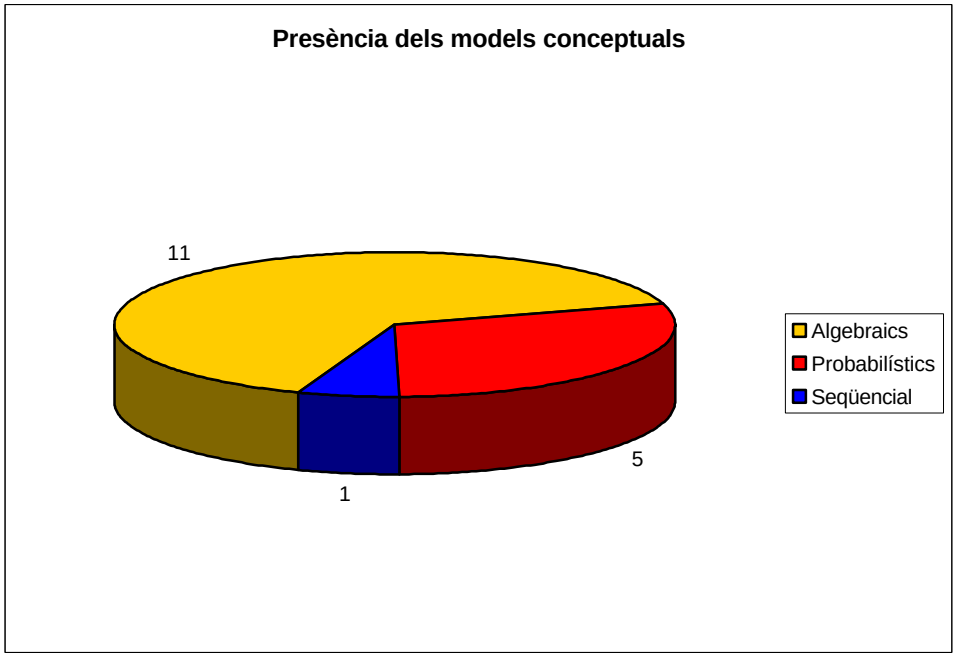
[21] <http://ir.cs.qc.edu/pircs.html>; [Consulta: 8 de maig de 2003].

[22] Segons aquests models el pes d'un terme depen de la seva probabilitat linguïstica i no de la distribució de la col·lecció.

[23] Un bon exemple és Onix Text Retrieval Toolkit (<http://www.lextek.com/onix>), que es basa en el model vectorial i la correlació dels cosinus, però que permet escollir en indexar la manera en que es determinen els pesos atenent a aspectes com els factors tf, idf, el nombre de termes d'un document o el nombre de documents. [Consulta: 8 de marc de 2003].

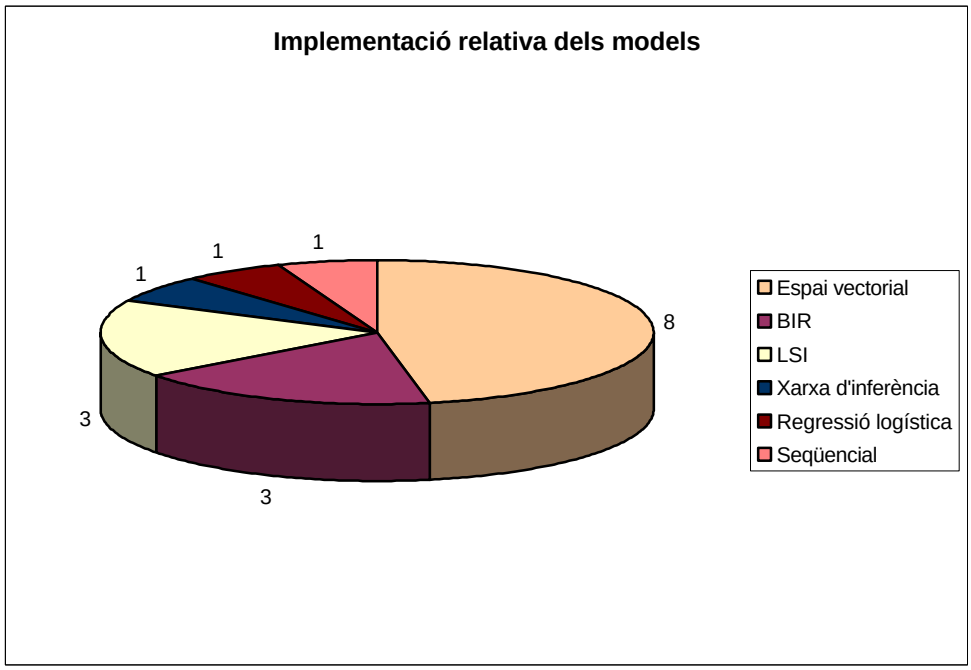
Una visió de conjunt

A partir de l'estudi del sistemes de RI elaborat s'observa un predomini clar dels models conceptuals algebraics sobre els probabilístics, mentre que els de Boole o altres basats en la teoria de conjunts són simplement inexistents (Gràfic 3.1). Cal considerar aquest fet com a fruit de la major facilitat algorísmica que ofereixen, sobretot els models amb pesos que segueixen l'esquema tf-idf.



Gràfic 3.1

Si comparem els tipus d'algorismes concrets, el model basat en l'espai vectorial és el que més s'ha utilitzat, a causa d'aquesta facilitat que esmentàvem fa un moment. Cal considerar que en aquest apartat s'agrupen sistemes com l'SMART, que permeten calcular jerarquies de rellevància i pesos de maneres diverses i altres que ho fan seguint només alguna variant concreta com ISEARCH, PRISE o MANAGING GIGABYTES. El model probabilístic d'independència binària (BIR) i el d'indexació semàntica latent mitjançant SVD són els altres models que també han assolit certa importància (gràfic 3.2).



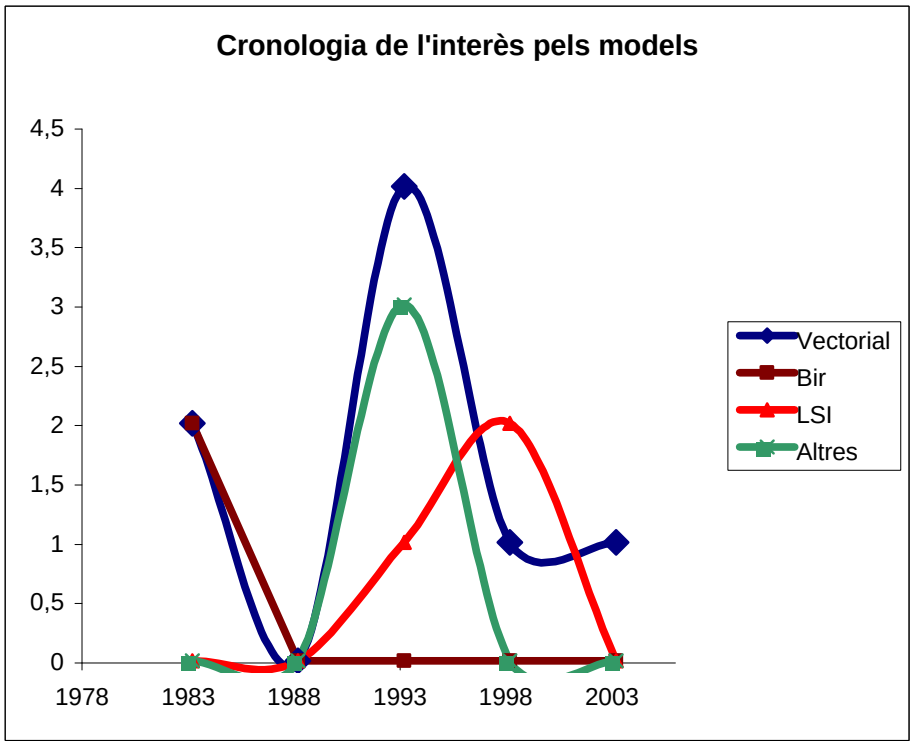
Gràfic 3.2

En centrar-nos en l'evolució cronològica de l'interès per uns o altres models des dels anys 80 fins a l'actualitat, observem que es desenvolupen enginys basats en el model vectorial en la primera meitat de la dècada dels 80. Es recupera l'interès en força d'ells a l'època de màxim desenvolupament de sistemes, vers la meitat del 90 i després decau sense ser abandonat del tot. El gràfic és però, una mica enganyós, ja que el model SMART realment es va mantenir viu durant 30 anys i nosaltres hem establert com a data d'inici del projecte principis dels 80. Tanmateix, el que més ens interessa és l'efecte comparatiu.

El model probabilístic d'independència binària té la seva època de glòria a principis dels 80 mentre que, per la seva banda, el LSI destaca a partir de la segona meitat de la década dels 90 (gràfic 3.3).

Assenyalem també que no hem trobat cap giny que es comencés a desenvolupar a la segona meitat de la dècada dels 80, encara que sí es van endegar alguns dels projectes ja en funcionament.

L'interès per la confecció d'aquests enginys sembla haver decaigut notablement. Les raons lligades a la maduresa dels motors actuals es tracten en el capítol següent. Actualment, aquests sistemes s'incorporen com a motors de cerca en aplicacions concretes que necessiten eines de RI.



Gràfic 3.3

Les taules 3.1 i 3.2 de les pàgines següents mostren algunes de les característiques i particularitats del sistemes estudiats.

Sistema	Model matemàtic de recuperació	Any 1a versió	Projecte Actiu	Aplicació comercial	Versió gratuïta	Desenvolupadors
Smart	- Vectorial (producte escalar, cos, tf-idf, ...)	1981	No	No	Sí	Gerard Salton, Chris Buckley a la Cornell University
Personal Librarian (SIRE)	-Vectorial (cos, tf-idf) - Lògica de Boole	1983-1986	No	Sí	No	Matt Koll
OKAPI	- Independència binària	1984	No	No	No	Polytechnic of Central London, City University.
Xapian (Omsee)	- Independència binària	1984	Sí	No	Sí	Martin Porter, Cambridge University Xapian project.
Smartlogic Discovery	- Independència binària	1984	Sí	Sí	No	Martin Porter, Discovery. Applied Psychology Research Ltd
Glimpse	- Cerca seqüencial	1993	Sí	Sí (Shareware)	Sí	Udi Manber, Sun Wu, i Burra Gopal a la University of Arizona.
Telcordia LSI	- Index semàntic latent	1994	Sí	Sí	No	Telcordia Technologies
Inquery	- Xarxa d'inferència	1994	Sí	Si	No	Inquery: CIIR U. of Massachusetts.
Managing Gigabytes (MG)	- Vectorial	1994	No	No	Sí	Tim C. Bell, University of Canterbury; Alistair Moffat, University of Melbourne; Ian Witten, University of Waikato; Justin Zobel, RMIT. New Zealand Digital Library Project
Isearch	- Vectorial (cos, tf-idf)	1994	No	No	Sí	Nassib Nassar.
IB	- Vectorial (cos, tf-idf)	1995	No	Sí	No	Nassib Nassar. Basis Systeme netzwerk
Cheshire II	- Lògica de Boole - Regressió logística - Regressió logística & lògica de Boole	1995	Sí	No	Sí	Ray Larson de la UC Berkeley.
Prise	- Vectorial	1995	No	No	Sí	Retrieval Group del NIST (National Institute of Standards and Technology).
LSI ++	- Index semàntic latent	1996	No	No	Sí	Todd A. Letsche. University of Tennesse
Verity Ultraseek	- Vectorial	1997	Sí	Sí	Sí (30 dies)	Ultraseek, Inktomi, Verity.
GTP	- Index semàntic latent	1998	Sí	No	Sí	S. Howard, H. Tang, M. Berry, i D. Martin de la University of Tennessee.
Amberfish	- Vectorial (cos, tf-idf)	2002	Sí	Sí	Sí	Nassib Nassar, per a Etymon..

Taula 3.1

Sistema	Demo en línia	Arxius accessibles FTP		Arquitectura desenvolupada	Plataformes per a binaris servidors/local
		Font	Binaris		
Smart	No	Sí (C)	Sí	- Mòdul Local	Solaris Linux
Personal Librarian	No	No (C, ?)	Sí	- Mòdul local - Xarxa local (d'igual a igual)	Windows 16 bits MA OS
OKAPI	No	No (C, Perl)	No	. Mòdul local - Servidor/Client CGI (?)	Solaris Linux
Xapian (Omsee)	No	Sí (C++)	No	- Mòdul local - Servidor/Client Web	Solaris Linux
Smartlogic Discovery	No	No	No	- Servidor/Client - Mòdul local	Solaris Windows NT/200x Altres UNIX
Glimpse	Sí	Sí (C)	Sí	- Mòdul Local	Solaris Linux Mac OS Altres UNIX
Telcordia LSI	Sí	No (C++ & Java)	No	-Local -Servidor/Client CGI	Solaris Linux Windows Plataforma JAVA
Inquery	Sí	No (C)	No	- Mòdul local - Servidor/Client Web	IBM RS/6000 AIX Compaq Alpha UNIX HP-UX Solaris Windows NT/2000
Managing Gigabytes	No	Sí (C)	No	- Mòdul Local	Solaris (UNIX) Linux (UNIX) Windows 32 bits
Isearch	No	Sí (C++)	No	- Local - Servidor/Client Web - Servidor/Client Z39.50 (ISITE)	Solaris Linux Altres UNIX Windows NT
IB	No	No (C++)	No	- Local - Servidor/Client Web - Servidor/Client Z39.50	Solaris Linux Altres UNIX Windows NT
Cheshire	Sí	Sí (C)	No	- Servidor/Client Z39.50 - Servidor/Client Web - Mòdul Local	Solaris Aplha HPUX Linux Windows NT
Prise	No	No (C)	Sí Password	- Servidor/Client Z39.50	Solaris (UNIX)
LSI++	No	No (C++)	No	-Servidor/Client CGI	Solaris Altres GCC
VERITY ULTRASEEK	No	No(C++, ?)	Sí	- Servidor/Client	Solaris Linus WindowsNT/200x
GTP	No	Sí Password (C++, Java)	Sí Password	- Local - Servidor/Client (?)	Solaris (UNIX) Linux (UNIX) Plataforma JAVA
Amberfish	No	No (C++)	No	- Servidor/Client (kit no totalment desenvolupat)	Solaris (UNIX) Linux (UNIX) Mac OS Free CÇBSD

Taula 3.2

SMART

L'SMART és sense dubtes, el sistema de recuperació d'informació més conegut dels que s'han realitzat. El seu origen es basa en les recerques de Gerard Salton sobre el model vectorial. Els primers treballs es remunten a l'any 1961 a Harvard, de manera que al 1964 ja disposava d'una versió experimental en funcionament. A partir de 1965 continuà la seva tasca a la Universitat de Cornell Altrament el modern SMART el podem situar sobre l'any 1981, quan s'implementa sobre el sistema operatiu UNIX i esdevé transportable.

El disseny del programari ha tingut diverses aportacions, al marge de Salton, però cal destacar la de Chris Buckley. El sistema és accessible[24] amb llicència GPL (General Public License)[25], però sembla que ha estat bàsicament abandonat des de la mort de Salton l'any 1995.

Avui en dia encara està considerat el sistema més interessant per a l'ensenyament i molt útil per a la recerca. La raó principal és que mai ha abandonat l'àmbit de l'experimentació, cosa que li dóna molta flexibilitat, car permet la modificació de diferents paràmetres per veure'n els resultats, aspecte que rarament resulta d'interès fora de l'àmbit docent i acadèmic. Així s'explica que es puguin escollir diverses maneres de fer les ponderacions de termes corresponents al model vectorial.

Els aspectes associats negatius són la falta d'optimització per a una tasca concreta, la manca de documentació –només algun tutorial en línia no massa explícit[26]– i, sobretot, una interfície d'usuari basada en caràcters que exigeix un cert grau de coneixement de l'entorn UNIX. A més, cal poder programar scripts realitzats amb cshell si es vol treure un rendiment personalitzat del sistema, encara que totes les funcions habituals que pot necessitar un usuari estàndard ja estan implementades. Tot això fa que aquells que l'utilitzen realment només com a usuaris acostumin a ser membres de la comunitat acadèmica.

---

[24] Disponible el binari per a Solaris (Sun OS) a: <ftp://ftp.cs.cornell.edu/pub/smart/>. [Consulta: 4 de juny de 2003].
Disponibles per a LINUX a: ftp://pi0959.kub.nl/pub/Smart/smart. [Consulta: 2 de gener de 2003]

[25] Més informació a : <http://www.gnu.org/copyleft/gpl.html>. [Consulta: 8 de juny de 2003].

[26] Disponibles a:
<http://pi0959.kub.nl:2080/Paai/Onderw/Smart/hands.html.>
<http://www.csse.monash.edu.au/courseware/cse4500/subjects/tutorials/SMART%20Tutorial.pdf.>
[Consulta: 4 de juny de 2003].

La versió actual lliurada és la 11[27] i es basa en uns 350 fitxers amb un total de 45.000 línies de codi en C.

Un càlcul aproximat estableix que es necessiten 0,4 vegades la mida original de la col·lecció en format textual per indexar-la. El sistema inclou un diccionari, informació sobre la localització del text i un fitxer invertit amb les referències que apunten als termes indexats.

Com ja hem indicat, tot i que el sistema està a punt per ser utilitzat, conserva entre el seus objectius el fet que es pugui utilitzar com a banc de proves del model vectorial.

Podem descompondre el pes $w_{i,j}$ **(D6)** en tres parts, dues corresponents a l'esquema tf-idf i un tercer bloc a la normalització:

(F54) $$w_{i,j} = \frac{f_{i,j} \cdot idf_i}{normal}$$

Per a calcular el factor d'aparició d'un terme (*tf*) dins d'un document d_j es permeten les següents opcions seguint la notació de **(D7)**:

Factor freqüència del terme	***Expressió de càlcul*** $(f_{i,j})$
Binari	$f_{i,j} \in \{0, 1\}$
Freqüència absoluta	$F_{i,j} = F_{i,j}$
Freqüència normalitzada (al màxim)	$f_{i,j} = \frac{F_{i,j}}{\max_l (F_{l,j})}$
Amplificat	$f_{i,j} = \left\{ \begin{array}{cc} 0{,}5 + \frac{0{,}5 F_{i,j}}{\max_l (F_{l,j})} & si \quad F_{i,j} \neq 0 \\ 0 & si \quad F_{i,j} = 0 \end{array} \right\}$
Logarítmic	$f_{i,j} = \left\{ \begin{array}{cc} 1 + \ln(F_{i,j}) & si \quad F_{i,j} \neq 0 \\ 0 & si \quad F_{i,j} = 0 \end{array} \right\}$

Taula 3.3

[27] Tot i que existeix com a mínim una versió 12 utilitzada en el TREC 4 l'any 1995.

El factor binari es correspon amb els primers dissenys de l'SMART i només considera la presència o absència del terme, sense atendre a factor ponderadors. La freqüència absoluta només considera el nombre de vegades que un terme apareix, sense cap normalització, cosa que pot facilitar la recuperació dels documents de mida més gran en detriment dels més curts. La freqüència normalitzada pel màxim de les freqüències aparegudes en el document *dj* és el típic esquema utilitzat en el model tf-idf – o sigui, es correspon amb ***(F8)*** –. El factor amplificat s'utilitza quan es desitja augmentar el valor dels termes presents respecte els no presents. S'utilitza especialment en el cas de la representació dels pesos de la cerca ***(F11),*** ja que valors molt baixos disminueixen dràsticament l'eficàcia en la recuperació. Finalment, el factor logarítmic s'utilitza si es vol atenuar la importància dels termes amb freqüències absolutes molt grans respecte els que les tenen més petites.

El càlculs del factor de freqüència inversa (*idf*) de presència del terme k_i en documents de la col·lecció admeten les expressions detalladas a continuació segons **(D7)**:

Factor de freqüència inversa de terme	***Expressió de càlcul*** *(idf_i)*
Inversa o tf-idf	$idf_i = \log(\frac{N}{n_i})$
Quadrat	$idf_i = \left[\log(\frac{N}{n_i})\right]^2$
Probabilístic	$idf_i = \log(\frac{N - n_i}{n_i})$
Nul	1

Taula 3.4

El factor invers és l'utilitzat més freqüentment **(F9)** i atenua la importància de termes molt poc presents a la col·lecció que donen quocients molt grans. Quan es desitja que aquest efecte suavitzador no sigui tan important es pot optar pel càlcul del *idf* quadrat. L'anomenat factor probabilístic respon al càlcul de la raó (odd) entre el nombre de

documents on no és present un terme i aquells on si que hi és. El factor nul consisteix en no considerar la influència de la freqüència d'aparició del terme dins de la col·lecció. Finalment, considerem els termes corresponents al normalització segons **(D6)**:

Normalització	***Expressió de càlcul*** *(normal)*
Nul·la o correlació de producte escalar	$normal = 1$
Cosinus	$normal = \sqrt{\sum_{i=1}^{t} f_{i,j}^2}$
Suma de termes	$normal = \sum_{i=1}^{t} f_{i,j}$
Suma de termes a la quarta potència	$normal = \sum_{i=1}^{t} f_{i,j}^4$
Al màxim	$normal = \max_l(f_{l,j})$

Taula 3.5

L'element neutre del producte ordinari es pren quan no es desitja realitzar cap tipus de normalització – similitud del producte escalar, supra taula 2.2 –. La normalització del cosinus (norma p=2) correspon a la situació més estàndard i és la recollida a ***(F7)***. La suma de termes (norma p=1) dóna més importància als pesos grans a l'hora de calcular el factor de normalització que el cas anterior –i. e. pesos molt grans en front de la resta fan davallar fortament la similitud de *q* i *d*. Aquest efecte encara s'amplifica més si la suma es fa sobre els termes a la quarta potència. Finalment, el factor corresponent al màxim s'utilitza si es vol que el valor de normalització sigui al més petit possible.

PERSONAL LIBRARIAN

L'empresa Personal Library Software fou creada l'any 1983 per Matt Koll. El capital intel·lectual es fonamentava en l'experiència acumulada els anys anteriors en el projecte SIRE (Syracuse Information Retrieval Experiment), a la universitat de

Syracuse. La principal innovació que va realitzar junt amb els seus col·legues fou integrar la cerca basada en la lògica de Boole i la jerarquització de resultats segons el model tf-idf i la similitud del cosinus.

L'any 1986 va aparèixer PERSONAL LIBRARIAN, el primer producte comercial que oferia consultes en llenguatge natural, classificació jeràrquica de resultats, i cerca basada en un exemple[28].

L'empresa va anar creixent des de les 3 persones inicials fins a unes 60 l'any 1996. Oferí amb èxit serveis en línia, funcions de cerca en bases de dades en CD-ROM, i aplicacions a mida de les necessitats de les empreses. També va saber adaptar el seus productes a l'entorn del web[29]. Però les negociacions i posterior adquisició per part d'Amèrica Online (AOL) significà a la pràctica el final del desenvolupament d'aquests tipus de programari[30].

A l'actualitat PERSONAL LIBRARIAN està disponible en forma de shareware, però no s´ha actualitzat des de 1995 – versions 4.15 per a Windows 16 bits i 4.1 per a Mac OS[31] –.

La interfície gràfica permet l'administració i la cerca, que tant pot fer-se en llenguatge natural com utilitzant expressions amb operadors de Boole.

El sistema està pensat per funcionar sobre una xarxa local i indexa els següents formats: text pla, Word 2.0, WordPerfect 5.x i un format propietari variant de l'ASCII anomenat "PL Standard", que permet estructurar els registres. L'usuari pot afegir gràcies a això, notes en els registres i marques per a retornar-hi directament (bookmarks). A més, l'administrador hi pot afegir funcions d'enllaç hipertextual, però codificant directament. Es poden crear diccionaris i tesaurus, fins i tot a partir de les llistes de termes automàticament suggerides.

Les bases de dades es poden estructurar en capítols i subcapítols i el sistema pot accedir a més d'una alhora.

El giny es basa en el model vectorial i en la correlació del cosinus, però el pesos tenen en consideració, a més dels aspectes del model tf-idf, la proximitat d'un document al

[28] Aquesta tècnica extreu del text termes relacionats amb els de l'equació de la cerca realitzada per estendre-la. És un tipus bàsic de retroalimentació per rellevància.

[29] PLWeb Turbo, una eina de cerca a la WWW. Més informació a: <http://www.pls.com/plweb.htm>. [Consulta: 15 de juny de 2003].

[30] Tot i que a mitjans de l'any 2001 AOL negava que s'hagués abandonat aquesta línia de recerca.

[31] Disponibles a: <http://www.pls.com/downinst.htm>. [Consulta: 15 de juny de 2003]. Anteriorment existien versions per a UNIX i, molt antigues, per a Vax.

seu inici i la proximitat del termes de la cerca, tot amb uns factors de ponderació que no podem precisar. En el cas d'utilitzar operadors de Boole – per a precisar molt la cerca –, es procedeix com en el cas citat supra del SIRE.

A més també, existeixen altres productes relacionats. Citem a Callable Personal Librarian (CLP), un paquet de programari que serveix per a desenvolupar enginys als distribuïdors d'informació molt utilitzat comercialment.. Disposa del nucli indexador i del motor de cerca i d'un conjunt d'eines complementàries útils per a programadors en C. L'enginy (versió 6.5)[32] és més avançat, té més capacitat d'indexació (Acrobat, HTML, e-mail) i només funciona sobre plataformes UNIX (Solaris 2.6, Digital Unix 4.0D, HP-UX 11, AIX 4.3.2 , SGI IRIX 6.2 (64 bit) i Red Hat Linux 5.2.

OKAPI

El projecte OKAPI és original del Politècnic de Londres – actualment la Universitat de Westminster – on es va començar a desenvolupar l'any 1982. Al juliol de 1989 les recerques es van traslladar a la City University, també de Londres. Avui en dia sembla que ja no està actiu quant a actualitzacions.

L'objectiu principal era crear un catàleg en línia que recuperés registres mitjançant eines probabilístiques i que oferís un bon disseny d'interfície d'usuari. Posteriorment es va desenvolupar un segon projecte per a investigar sobre lematització, correcció de lletrejat, taules de referència creuades per a sinònims i ajuda a la cerca. A partir de 1988 es van avaluar expansions de la cerca i retroalimentació per rellevància. Vers l'any 1993 s'hi va afegir un tesaurus i, finalment, una interfície gràfica configurable per l'usuari per millorar les prestacions de cerca interactiva.

El model de RI incorporat es probabilístic d'independència binària, ajustant-se a ***(F29)***, ***(F30)*** i ***(F31)***, segons els treballs de Robertson i Spark Jones.

El programa està codificat en C, amb material addicional en Perl. La interfície ha estat desenvolupada amb C, C++ i Tcl/Tk. Només és directament compilable sobre UNIX.

[32] <http://www.pls.com/cpl.htm>. [Consulta: 15 de juny de 2003].

Hi ha fitxers binaris disponibles per a Solaris 2.x i per a Red Hat Linux 6.x[33].

XAPIAN

El projecte Xapian, actiu a l'actualitat, està basat en el codi d'un projecte anterior, l'Omsee[34], conegut també per Open Muscat (Muscat de codi obert), desenvolupat i lliurat en llicència GPL per BrightStation PLC. Quan aquesta empresa va tancar l'abril de 2001, alguns dels desenvolupadors de Omsee van continuar treballant-hi.

L'origen del Muscat es remunta als treballs de recerca realitzats pel Dr. Martin Porter a la Universitat de Cambridge. L'any 1984, en companyia de John Snyder, fundà la Cambridge CD Publishing, amb l'objectiu d'explotar-ne la tecnologia. Quan la focalització de l'interès de l'IR canvià del CD al web, va passar a anomenar-se Muscat Ltd, on es desenvolupà la biblioteca de programes Muscat 3.6, reemplaçada actualment pel nou nucli del Xapian, totalment recodificat. Muscat Ltd va ser adquirida posteriorment per Maid PLC, que canvià de nom, anomenant-se primer Dialog Corporation, i posteriorment BrightStation PLC fins a la seva desaparició.

XAPIAN és escrit amb C++ i la biblioteca d'enllaços permet utilitzar llenguatges suportats per SWIG[35], tal com PHP, Perl o Java. S'ha desenvolupat per a plataformes UNIX i s'ha provat almenys sobre x86, Linux, FreeBSD, OpenBSD i Solaris –.

La versió més actual de XAPIAN 0.5.4 es distribueix en tres parts: la biblioteca de programes pròpiament dita, que constitueix el nucli del motor de cerca; una col·lecció de petits programes de mostra i Omega, una aplicació complementària que s'encarrega de la indexació de termes i d'interfície CGI de l'usuari final[36]. No es subministren arxius binaris, sinó únicament el codi font.

El límit de la mida del fitxer ve fixat pel sistema operatiu i Omega permet indexar directament HTML, PDF, PostScript i text pla. El ventall pot ser ampliat si hi ha

[33] És possible adquirir els fitxers binaris a canvi del preu simbòlic de 100 lliures esterlines per a usos no comercials a: <http://www.soi.city.ac.uk/~andym/OKAPI-PACK/registration_details.php>. També es pot adquirir per 1000 lliures el codi font. [Consulta: 8 de juny de 2003].

[34] Durant un breu període de temps, el projecte es denominà Omseek, abans de prendre el nom actual de Xapian.

[35] Més informació a: <http://www.swig.org>. [Consulta: 8 de juny de 2003].

[36] Disponible a: <http://xapian.sourceforge.net/download.php>. [Consulta: 8 de juny de 2003]. A més s'hi pot trobar un tutorial força complert.

filtres automàtics, com per exemple Microsoft Word, o bé pot ser substituït per un robot indexador de webs.

El model utilitzat per jerarquitzar els documents és una variant del d'independència binària ***(F29)***

$$\textbf{(F55)}\quad Sim\,(d_j,q) = \sum_{i=1}^{t} \frac{(C+1)F_{i,j}}{C\cdot L_j + F_{i,j}} w_{i,q}\cdot w_{i,j}\cdot(\log\frac{P(k_i \mid R)}{1-P(k_i \mid R)} + \log\frac{1-P(k_i \mid \overline{R})}{P(k_i \mid \overline{R})})$$

$F_{i,j}$ representa la freqüència absoluta d'aparició del terme k_i en el document d_j **(D7)**. Si N és el nombre de termes i $long_d$ la longitud absoluta de d_j, llavors L_j és la longitud normalitzada del document definida com el quocient entre $long_d$ i la mitjana de les longituds de tots els documents:

$$\textbf{(F56)}\qquad L_j = \frac{long_d}{\frac{1}{N}\sum_{l=1}^{N} long_l}$$

Com que aquesta expressió origina problemes quan les longitud són molt petites respecte la mitjana, el sistema fixa un valor mínim.

C és una constant que s'escull mentre es crea el conjunt de documents V rellevants per a la consulta, és a dir, que tenen al menys un terme dels de la cerca. Aquest valor és normalment més petit o igual que 1000 i fita també el nombre de documents presentats.

Notar que per a valor petits de C, ***(F55)*** es converteix en ***(F29)***, per tant no es veu afectat ni per $F_{i,j}$ ni per L_j. Altrament, quan C pren valors prou grans, convergeix cap a $F_{i,j}/L_j$, que tendeix a afavorir documents no gaire extensos on apareix molts cops un terme invocat des de l'expressió de la cerca.

El sistema expandeix la cerca bàsicament segons el criteri general exposat del model BIR a partir de **(D12).**

XAPIAN admet cerques amb operadors de Boole. En aquest cas, un cop ha seleccionat els documents que compleixen l'equació lògica, aplica el procediment esmentat anteriorment per tal d'escollir els C millors documents presentats segons el valor decreixent de ***(F55)***.

Finalment, assenyalem que el sistema permet fer cerques amb termes que formen frases exactes, dins d'un nombre especificat de paraules o en un ordre específic. Això és possible gràcies a un vector $wdp_{i,j}$ (within-document positions) que guarda totes les posicions del terme k_i dins del document d_j.

SMARTLOGIC DISCOVERY

Aquest sistema es pot considerar una versió comercial paral·lela al XAPIAN. Quan BrightStation PLC va desaparèixer l'any 2001, Smartlogic continuà amb la comercialització del producte Muscat 3.6. L'abril de 2002 aquesta fou adquirida per Applied Psychology Research Ltd (APR), qui llicència l'ús del producte sota el nom SMARTLOGIC DISCOVERY. Disposen de fitxers binaris per a servidors Windows NT/2000/2003, Solaris i Linux, encara que sota comanda poden compilar el programari per a qualsevol sistema operatiu habitual[37].

Encara que APR no facilita gairebé informació sobre el cor del sistema, la història recent i les prestacions que ofereix permeten col·legir que no hi ha modificacions substancials respecte a Muscat 3.6 des del punt de vista del model d'IR que es correspon amb el probabilístic d'independència binària amb modificacions iguals o similars a les exposades pel XAPIAN. També hi són presents funcionalitats, tal com la cerca amb operadors de Boole o funcions de proximitat (near), basades en vectors $wdp_{i,j}$.

Permet indexar text pla (txt, rtf), formats de Microsoft Office (doc, xls, ppt), PDF, HTML, XML, email via passarel·les IMAP.

El que sí s'han desenvolupat notablement, donat el seu caràcter estrictament comercial, són les funcions d'enllaç amb bases de dades relacionals mitjançant drivers JDBC[38] i un robot indexador de webs.

El sistema té l'habilitat de classificar per rellevància els termes de metadades de documents estructurats amb marques, cosa que permet millorar les capacitats de suggerir pàgines. El programari també disposa d'una funció correctora dels errors de

[37] Es pot establir els contactes per adquirir les llicències a través de: <http://www.aprsmartlogik.com/products/discover>. [Consulta: 8 de juny de 2003].

lletrejat basada en trigrames, és a dir, a partir del càlcul de la probabilitat que una paraula s'escrigui a continuació d'altres dues. Finalment, no podria faltar una potent i integrada interfície gràfica que incorpora moltes utilitats per a l'usuari final, com planificadors o eines de comunicació. Però totes aquestes interessants funcionalitats no afecten el model matemàtic d'IR amb el que realitzar la classificació per rellevància.

GLIMPSE

Aquest sistema va aparèixer l'any 1993, però el projecte segueix totalment viu de manera que les actualitzacions són recents. Sense cap dubte és un cas a part dins dels sistemes de RI pel que fa al model car no es basa en l'ús d'un fitxer invertit. Les raons que els creadors aporten és la mida de l'índex generat, problema especialment significatiu en maquinaris domèstics, i la necessitat d'un lletrejat acurat.

La tècnica desenvolupada es basa en la utilització d'un esquema híbrid entre un fitxer invertit complet i la pura cerca seqüencial. L'índex generat, que ve a ocupar menys d'un 5% de la mida original, no disposa dels llocs exactes on és cada terme, sinó que apunta a una àrea on es pot trobar la resposta. Llavors entra en funcionament un procés flexible de cerca seqüencial per determinar exactament el text buscat i presentar-lo l'usuari.

Per a crear l'índex, el sistema explora tota la col·lecció i el construeix de manera similar a un fitxer invertit, però no es guarda cada ocurrència d'un terme, sinó que cadascun només es guarda un cop i se li assignen els números de bloc entre els 256 (1 byte) en que s'ha de dividir tota la col·lecció. El fet que no apareguin totes les ocurrències que corresponen a un mateix bloc disminueix dràsticament la mida del fitxer.

La cerca es realitza en dues fases. Primer es localitzen els blocs que contenen els termes de la cerca. Llavors dins de cada bloc es realitza una cerca seqüencial. Això és possible gràcies a la mida relativament petita de cadascun dels blocs. Encara que seria factible realitzar aquesta tasca amb altres tècniques més ràpides, com el hashing o estructures d'arbre, s'escull la cerca seqüencial per la flexibilitat que permet. D'aquesta

[38] Més informació a: <http://java.sun.com/products/jdbc/features.html>. [Consulta: 8 de juny de

manera es poden especificar una quantitat d'errors com insercions, supressions, substitucions i les seves combinacions; aspectes formals definits per l'usuari com paràgrafs o missatges de correu; o la utilització de símbols de reemplaçament (wildcards) i conjunts de caràcters ([A-E]).

Curiosament, les cerques amb operadors de Boole resulten especialment lentes. Quan s'utilitza l'operador i (and) cal determinar primer els blocs en que es troben els termes relacionats i després comprovar dins de tots els blocs seleccionats, document per document, si els dos termes hi són presents.

Els principals inconvenients del sistema són que no s'optimitza la rapidesa i que no presenta els resultats classificats segons la seva rellevància. En aquest sentit, el comportament és assimilable a una resposta segons la lògica de Boole (model pseudo-Boole), però amb les possibilitat gens menyspreables ja esmentades.

El programari està totalment escrit amb C i indexa només text pla. La versió actual que es llicència[39] tant per a usos docents, com de recerca i comercials és la 4.17.3 – de 16 de maig de 2003 –. Hi ha demostracions en línia del funcionament del sistema[40].

El projecte segueix totalment viu, encara que està dedicat sobretot a una aplicació desenvolupada utilitzant GLIMPSE com a generador de l'índex i motor de cerca. Es tracta de Webglimpse, que inclou una interfície d'administració de web i un robot que explora Internet. Però cal indicar que GLIMPSE està totalment preparat per funcionar independentment de manera local.

La versió comercial de webglimpse permet presentar els resultats jerarquitzats amb funcions definibles per l'administrador del sistema, però sempre sobre els resultats prèviament recuperats, tal i com ja hem indicat.

TELCORDIA LSI

El disseny d'aquest enginy es remunta a les idees exposades a [Deerwester, Dumais, Furnas, Landauer, 1990] a Bellcore sobre els models d'indexació semàntica latent.

2003].

[39] Es pot obtenir la llicència escaient a: <http://glimpse.cs.arizona.edu>. [Consulta: 8 de juny de 2003].

[40] Accessible a: « http://glimpse.cs.arizona.edu/index.php?dir=subdemo&page=usersearch.html». [Consulta: 8 de juny de 2003].

L'any 1992 ja s'estava desenvolupant un prototipus de recerca que es va cedir sota llicència universitària a diversos grups acadèmics. Un cop enllestides les primers versions del programari necessari per a la seva comercialització– per exemple, la interfície per consultes a través de la xarxa desenvolupada en Perl l'any 1994 – es va treure al mercat com Bellcore LSI. L'any 1997 l'empresa fou absorbida, però conservant identitat pròpia- pel gegant Science Applications International Corporation (SAIC), de manera que es va veure abocada a canviar a Telcordia, i d'aquí com s'identifica comercialment en l'actualitat el sistema[41].

Originalment el giny es va codificar en C amb alguns scripts de UNIX per a plataformes Solaris. Posteriorment, algunes parts s'han reescrit en C++ i en Java i el sistema s'ha compilat en Windows, Linux i diversos UNIX[42]. SAIC comercialitza una versió en la plataforma multiportable «Java 2 Enterprise Edition» (J2EE).

No es poden aconseguir directament fitxers en línia, però existeix una «demo»[43] amb comentaris per mostrar al possible usuari la potència del model que recupera documents que no contenen el terme citat, precisament per la relació semàntica latent amb altres termes introduïts a la cerca.

TELCORDIA LSI incorpora retroalimentació per rellevància i està preparat per a indexar documents en llengües diverses. La seva capacitat de relacionar conceptes el fa especialment hàbil en casos de recuperació multilíngüe.

La funció de similitud per a jerarquitzar els documents recuperats es correspon amb ***(F25)***.

INQUERY

L'origen del sistema INQUERY cal buscar-lo en la tesi doctoral de Howard R. Turtle l'any 1990. L'any 1992 es creà el The Center for Intelligent Information Retrieval [44](CIIR) per al desenvolupament i la transferència científica. El seu projecte inicial fora

[41] Es pot demanar la llicència d'ús a: <http://lsi.argreenhouse.com/lsi/request.html>. [Consulta: 8 de juny de 2003].

[42] De fet, és possible fer-ho en qualsevol plataforma que admeti un compilador de C GNU. Més informació sobre els compiladors a: <http://gcc.gnu.org>. [Consulta: 11 de maig de 2003].

[43] <http://lsi.argreenhouse.com/lsi-bin/lsiQuery>. [Consulta: 8 de juny de 2003].

[44] A la Universitat de Massachusetts a Amhers.

el desenvolupament d'un motor de cerca basat en una xarxa de creences de Bayes. El projecte va durar tres anys. L'any 1996, un cop que el sistema INQUERY disposà de totes les seves funcionalitats (v 4.x) es varen transferir els drets de comercialització a Sovereign Hill Software, participada pel CIIR i creada per ser receptora de les tecnologies ja consolidades. Al gener de 1999, quan el producte es trobava en la seva versió 5, l'empresa fou adquirida per Dataware Technologies, la qual fora absorbida finalment per Open Text Corporation[45] al gener del 2001. Tanmateix, no figura a l'actualitat a la seva família de productes comercialitzats[46]. Però constitueix el motor de cerca de BRS/Search, un avançat gestor de base de dades documental pensat especialment per a tecnologies hipermèdia[47].

INQUERY és un dels sistemes que ha tingut més èxit si es considera el nombre d'empreses i institucions que l'han implementat. No és possible a l'actualitat aconseguir en línia una còpia gratuïta, encara que diverses universitats disposen de llicències experimentals. En canvi, es pot accedir a CIIRDEMO, una col·lecció de «demos» en la qual s'utilitza INQUERY per cercar dins de bases de dades[48].

El programari està escrit en Ansi C i s'ha compilat sobre plataformes Windows NT/2000, Solaris, Compaq/Digital UNIX, IBM RS/6000 AIX, i SGI Irix. Altrament, les versions inicials no comercials foren provades fins i tot sobre MSDOS, amb i sense interfície gràfica windows de 16 bits.

El sistema permet donar més pes a uns termes que a d'altres amb diverses opcions; indicar distàncies entre termes; termes que han d'estar en blocs de text de dimensió definida per l'usuari; expressions exactes; agrupar termes de la cerca com a sinònims per a una consulta determinada, entre d'altres funcions[49].

Les cerques poden fer-se mitjançant operadors de Boole i llavors els resultats s'ajusten a les equacions ***(F38)***, ***(F39)***, ***(F40)*** i ***(F41)***; utilitzant expressions del llenguatge natural ***(F42)***, ***(F43)***, ***(F44)*** i ***(F45)***; o bé amb una barreja d'ambdues ***(F46)***.

[45] <http://www.opentext.com>. [Consulta: 16 de febrer de 2003].
[46] <http://www.opentext.com/products>. [Consulta: 16 de juny de 2003]. Aquesta firma comercial no ha contestat les nostres demandes d'informació sobre aquest punt.
[47] <http://www.opentext.com/brs/brs_search.html>. [Consulta: 13 de febrer de 2003].
[48] <http://ciir.cs.umass.edu> . [Consulta: 15 de juny de 2003].
[49] Informació detallada amb exemples a: <http://ciir.cs.umass.edu/irdemo/inqinfo/inqueryhelp.html#queryform>. [Consulta: 15 de juny de 2003].

MANAGING GIGABYTES (MG)

MG es un motor de cerca i recuperació d'informació en codi lliure (llicència pública general)[50] desenvolupat a partir de les idees de Tim C. Bell (Universitat de Canterbury), Alistair Moffat (Universitat de Melbourne), Ian Witten (Universitat de Waikato) i Justin Zobel (RMIT) i amb finalitats pedagògiques i experimentals. La primera versió pública del programa aparegué vers l'any 1994 lligada a la preparació de la monografia «*Managing Gigabytes: Compressing and Indexing Documents and Images*» [Witten, Moffat, Bell, 1994][51] que incloïa a l'annex documentació sobre el funcionament del sistema i resultats experimentals.

El projecte es va mantenir viu fins a finals de l'any 1999 gràcies a diverses col·laboracions altruistes. Tanmateix, l'última versió disponible, la MG-1 3.0, fou desenvolupada dins dels projectes The New Zealand Digital Library i Greenstone software[52], on ocupa el paper de nucli de RI.

La codificació està totalment realitzada en Ansi C i corre sobre sistemes UNIX com Solaris i Linux i, en el cas més recent, es pot compilar en plataformes Windows de 32 bits. MG disposa d'una senzilla aplicació que fa d'interfície gràfica (xmg, dissenyada per a X-Windows) que evita a l'usuari final realitzar la recuperació en una interfície basada en caràcters, però no cobreix cap altre funcionalitat.

Una de les característiques especials d'aquest enginy, i que no es dóna en altres, és que els documents que formen la col·lecció estan comprimits (mètode de Huffman) per reduir el volum de dades a emmagatzemar pel maquinari. Això obliga a un procés de descompressió si es vol accedir a ell.

El sistema indexa text pla, però no interpreta correctament la codificació amb 8 bits. Admet cerques amb operadors de Boole. Si s'opta per una resposta jerarquitzada, la

[50] Dades sobre la llicència d'ús a: <http://mds.rmit.edu.au/mg/intro/copying.html>. [Consulta: 8 de juny de 2003].

[51] Existeix una segona edició renovada [Witten, Moffat, Bell, 1999].

[52] A: <http://www.nzdl.org/html/mg.html>. [Consulta: 8 de juny de 2003].
L'última versió lligada als autors originals (v 2.1) es localitza a <http://www.cs.mu.oz.au/mg/mg-1.2.1.tar.gz>. Versions antigues estan disponibles també a: <http://www.cs.mu.oz.au/mg/oldversions>. [Consulta: 8 de juny de 2003].

funció de similitud es calcula mitjançant un esquema que respon a ***(F54)***, encara que existeix un model experimental MG-2, del qual no s'ha lliurat cap versió ni documentació, que permet utilitzar altres tècniques pròpies d'un espai vectorial. Per a calcular el terme idf_i, tenim

$$\textbf{(F57)} \qquad idf_i = \log\left(\frac{N}{n_i} + 1\right)$$

Aquesta correcció és necessària si es vol evitar que l'argument del logaritme agafi el valor unitat. Tal possibilitat existeix, ja que el sistema no exclou aquells termes que apareguin en tots els documents– ni en la immensa majoria d'ells –.

Els factors $f_{i,j}$ es corresponen amb les freqüències absolutes, mentre que el factor de normalització inclou els casos de la següent taula:

Normalització	***Expressió de càlcul*** *(normal)*
Nul·la o correlació de producte escalar	1
Cosinus	$Normal = \sqrt{\sum_{i=1}^{t} F_{i,j}^2}$
Suma de termes	$Normal = \sum_{i=1}^{t} F_{i,j}$
Arrel quadrada de la suma de termes	$Normal = \sqrt{\sum_{i=1}^{t} F_{i,j}}$
Logaritme del factor de longitud	*Sigui Oj el nombre de termes totals emmagatzemats en el document* d_j $Normal = \log_2 O_j$
Arrel quadrada del factor de longitud	$Normal = \sqrt{O_j}$

Taula 3.6

La documentació existent és prou amplia i disponible a: <http://www.mds.rmit.edu.au/mg>. [Consulta: 8 de juny de 2003]

La versió 3.0 de MG permet utilitzar diversos mètodes per cerques de termes inexactes, amb patrons que permeten recuperar texts amb errors de lletrejat, subcadenes, prefixes i sufixos o variants lingüístiques amb idèntics lexemes. Es poden atorgar pesos addicionals als termes de la cerca i en el procés es pot accedir a la freqüència d'aparició del termes de l'equació de consulta dins la col·lecció.

ISEARCH

Aquest sistema és un altre cas de programari de codi obert per a indexació i recuperació de documents de text. Fou desenvolupat originalment l'any 1994 per Nassib Nassar per a la National Science Foundation, per tal de ser el component encarregat de fer de motor de cerca de text dins del sistema Isite, que havia de substituir al ja antiquat freeWAIS. El treball va continuar després coordinat per Archibald Warnock (ISEARCH 2)

ISEARCH pot funcionar de manera totalment independent de l'altre mòdul d'Isite –un desenvolupament del protocol Z39.50[53] – i de fet ha tingut força èxit. També es va desenvolupar l'Isearch-cgi, una senzilla, però útil, interfície basada en scripts per tal de recuperar cerques de pàgines web.

El programari fou codificat amb C++ i a més de cerca de text complet admet una estructuració en camps gràcies a la definició de tipus de documents (SGMLTAG doctype). Pot indexar documents de text pla, carpetes de correu, compilacions de correu (digest), HTML i altres codificacions basades en llenguatges de marques. La concepció i la documentació que existeix permet afegir-hi nous tipus de documents desenvolupant el mòdul en C++ escaient. No disposa de lematitzador, encara que existia la intenció de instal·lar-lo i no suporta text amb 8 bits, cosa que redueix molt les possibilitats d'indexar documents en altres llengües que no siguin l'anglès.

Ha estat compilat almenys per a les següents plataformes: Linux, Solaris, FreeBSD, Digital Unix, Ultrix, HP/UX, AIX, DG/UX, IRIX i Windows. La darrera versió[54] que

[53] Norma ISO 23950.

[54] A <ftp://ftp.cnidr.org/pub/software/Isearch> i <ftp://ftp.cnidr.org/pub/software/Isearch-cgi> es poden trobar versions més antigues del sistema.

coneixem és la 1.47 de 1999, i no tenim cap indicació que el projecte es mantingui actualitzat.

El sistema suporta cerques de termes, cerques basades en la lògica de Boole[55] i truncament de termes per la dreta (prefixes).

El model matemàtic utilitzat es correspon amb el d'espai vectorial. La funció de classificació segons rellevància s'ajusta totalment a la correlació del cosinus ***(F7)*** mentre que els pesos es calculen segons el model tf-idf ***(F10)***. Tanmateix el sistema permet assignar pesos extraordinaris als termes de la cerca, tant superiors com inferiors, de manera que es pot adaptar a la importància dels termes. Això pot ser útil especialment en les cerques que utilitzin operadors de Boole com la negació, que pot ser atenuada.

IB

IB, d'Inter Basis, és una edició comercial de ISEARCH desenvolupada per Basis Systeme netzwerk (BSn)[56] entre 1995 i 1996, però sense desenvolupaments posteriors.

IB 2.x té les mateixes característiques que el seu antecessor, però amb noves possibilitats. Per exemple, disposa de 50 tipus de documents SGML/XML predefinits. Suporta conjunts de caràcters ampliats i missatges en diverses llengües. Encara que no necessita llistes de paraules buides, poden ser especificades durant la indexació o en la cerca, fins i tot en llistes i idiomes diferents.

L'evolució del producte lligada al creixement d'Internet gràcies al www comportà a la mateixa època l'aparició de IB 2.0 WebCat, que incorpora una interfície en format HTML amb algunes característiques molt útils en el nou entorn: detecció automàtica i

---

A <http://www.freesoft.org/software/Isearch> es poden trobar els fitxers font de la v 1.47, així com els binaris de Linux i de Windows, a més d'un complet tutorial i altres informacions de referència. [Consulta: 8 de juny de 2003].

[55] Mitjançant l'addició de la clau -*infix*, una raresa heretada del Cobol i el Fortran i de les calculadores TI (Texas Instruments), però que permet escriure l'expressió amb lògica directa.. Però el que és realment extraordinari és que per a aquest tipus de consultes també es pot utilitzar la notació polonesa inversa (RPN, Reverse Polish Notation) dels llenguatges basats en pila, com el Forth i les calculadores HP (Hewlett Packard). Així, per exemple, l'expressió «pàgina document *or* groc verd *or and*» equival en lògica directe a «(pàgina *or* document) *and* (groc *or* verd)».

[56] L'última versió coneguda és la 2.5. Informació a: <http://www.bsn.com/main.html>. Des d'aquesta pàgina es facilita accés a versions de prova de l'enginy, però els enllaços ja no són operatius. [Consulta: 15 de juny de 2003].

transparent de enllaços, conversió en temps real de qualsevol document a format HTML, vinculació de formats a aplicacions externes concretes (MIME).
IB 2.0 WebCat ha estat un producte amb cert èxit comercial.

CHESIRE

El projecte CHESIRE II, impulsat des de la Universitat de California, Berkeley, per Ray L. Larson, consisteix en un sistema de catàleg en línia amb un motor de cerca que utilitza tècniques probabilístiques de RI. La primera versió del sistema apareix l'any 1995 amb la voluntat de fer de pont entre els vells catàlegs de referències i els que donen accés a col·leccions en text complet encara que els posteriors desenvolupaments de les interfície d'usuari han donat prioritat a aquest rol.
CHESIRE incorpora una arquitectura client/servidor amb dos tipus de clients diferents, un Z39.50 i un web mitjançant un intèrpret CGI que utilitza els índexs i la base de dades directament. La codificació és en llenguatge C. La darrera versió disponible és la 2.38 per a plataformes UNIX i Windows NT[57].
Els documents es guarden en format SGML, cosa que permet definir diferents tipus de documents (DTD) i barrejar, per exemple, registres bibliogràfics en format MARC (format SGML) amb documents HTML a l'hora de presentar una classificació jeràrquica de documents rellevants. Qualsevol de les tipologies pot ser seleccionada en el procés de recuperació.
Per millorar el procés de recuperació s'ha incorporat una eina de retroalimentació per rellevància que permet recuperar nous documents a partir de la selecció realitzada per l'usuari.
El sistema permet fer cerques, tant en expressions amb operadors de Boole com amb llenguatge natural. En el primer cas es presenta un simple llistat sense cap classificació de rellevància. Per a determinar els documents que compleixen la sentència es recorre a comprovar els vectors de cada document en les posicions *i* que contenen les

[57] <ftp://cheshire.berkeley.edu/pub/cheshire>. [Consulta: 8 de juny de 2003].
Documentació i tutorials al web del projecte: <http://cheshire.lib.berkeley.edu/index.html>. També està disponible una demostració en línia de consulta a un catàleg que utilitza CHESIRE [Consulta: 8 de juny de 2003].

freqüències absolutes de cada terme k_i, de manera que es presenten els valors que donen valor 1 segons l'expressió ***(F1)***.

Els resultats de les consultes probabilístiques sí que es presenten jerarquitzades per rellevància. El model probabilístic utilitzat és la regressió logística, de manera que la funció de similitud ve donada per ***(F36)***.

En el cas del CHESIRE la suma de coeficients s'escriu com

$$\textbf{(F58)} \qquad sim(d_j,q) = \sum_{i=1}^{6} c_i X_i$$

Les variables $\{X_1, X_2,...,X_6\}$ s'expressen segons s'indica a la següent taula:

Variable	***Significat***	***Expressió de càlcul***
X_1	Mitjana per a tots els termes comuns a d_j i q del logaritme decimal de la freqüència absoluta d'aparició del termes k_i en la cerca q.	*Sigui $O_{j,q}$ el nombre de termes per als quals $F_{i,q} \cdot F_{i,j} \neq 0$ i $F_{i,q}$ la freqüència del terme k_i a la cerca q. Llavors* $X_1 = \frac{\sum_{\forall\ F_{i,q}\cdot F_{i,j}\neq 0} \log(F_{i,q})}{O_{j,q}}$
X_2	Arrel quadrada del nombre de termes de la cerca, exclosos els termes buits (longitud de la cerca).	*Sigui L_q el nombre de termes no buits de la cerca q. Llavors* $X_2 = \sqrt{L_q}$
X_3	Mitjana per a tots els termes comuns a d_j i q del logaritme decimal de la freqüència absoluta d'aparició del termes k_i en el document d_j.	$X_3 = \frac{\sum_{\forall\ F_{i,q}\cdot F_{i,j}\neq 0} \log(F_{i,j})}{O_{j,q}}$
X_4	Arrel quadrada de la mida del document.	*Sigui L_{d_j} la mida en bytes del document d_j* $X_4 = \sqrt{L_{d_j}}$
X_5	És la mitjana dels valors idf_i per a tots els termes comuns a d_j i q	$X_5 = \frac{\sum_{\forall\ F_{i,q}\cdot F_{i,j}\neq 0} \log(\frac{N}{n_i})}{O_{j,q}}$
X_6	Logaritme decimal de tots el termes comuns de d_j i q	$X_6 = \log(O_{j,q})$

Taula 3.7

Els valors dels coeficients $\{c_1, c_2,...,c_6\}$ foren obtinguts utilitzant la col·lecció TIPSTER[58] amb els següents resultats:

Coeficient	c_1	c_2	c_3	c_4	c_5	c_6
Valor	1,269	-0,310	0,679	-0.0674	0,223	2,01

Taula 3.8

Una de les possibilitats que presenta el sistema és combinar una cerca de Boole – q_{boole} – i probabilística alhora – q –. En aquest cas, el sistema estableix el conjunt de documents D_{boole} que compleixen l'expressió de Boole q_{boole}.
Llavors la funció de similitud val

$$\textbf{(F59)}\ Sim(d_j, q) = \begin{cases} \sum_{i=1}^{6} c_i X_i(q_{boole}) & \forall\ d_j \in D_{boole} \\ 0 & \forall\ d_j \notin D_{boole} \end{cases}$$

on $X_i(q_{boole})$ es calcula només sobre el conjunt solució D_{boole}.

PRISE

Aquest sistema fou presentat l'any 1995 pel Written Natural Language Processing Group del National Institute of Standards and Technology (NIST), una agència del govern dels EUA, i per tant no subjecta a drets de copyright dins del seus país, amb

[58] TIPSTER fou un projecte finalitzat la tardor de 1998 i finançat per la Defense Advanced Research Projects Agency (DARPA) per al desenvolupament del tractament i recuperació automàtica d'informació a partir de documents. Per tal de fer experiments normalitzats es va compilar una col·lecció de documents de centenars de MB. Informació sobre el projecte a: <http://www.nist.gov/itl/div894/894.02/related_projects/tipster>. [Consulta: 8 de juny de 2003].

l'objectiu de facilitar el desenvolupament d'aquestes eines tant en els entorns acadèmics com comercials.

El programari està codificat en llenguatge Ansi C i es composa d'un senzill mòdul client/servidor basat en el protocol Z39.50 i un motor d'indexació i de cerca que és pròpiament PRISE[59]. De tota manera, el disseny del motor està isolat del mòdul Z39.50 per tal de facilitar la substitució d'un o l'altre.

El fitxers fonts es poden compilar sobre plataformes Solaris (2.6) i Sun OS (4.1x) encara que la portabilitat d'aquest sistema operatiu el fa apte amb canvis menors per córrer sota altres versions de UNIX, inclòs Linux. La versió actual del paquet és la 2.0[60] i data de l'any 1998 i el giny ja no està dins dels projectes actius del NIST[61].

Entre les característiques del sistema citem que permet la retroalimentació per rellevància (només la versió 2.0), el truncament de termes, funció *or*, expressions exactes amb omissió d'espai en blanc i etiquetes SGML i operadors de proximitat. Indexa text pla, però no admet la codificació de 8 bits.

PRISE utilitza un model d'espai vectorial amb la forma tf-idf de calcular els pesos proposada a [Harman, Candela, 1990]. En aquest cas la funció de similitud la podem escriure com

(F60) $$sim(d_j,q) = \frac{\sum_{\forall\ F_{i,q}\cdot F_{i,j}\neq 0} \log_2 F_{i,j}\cdot idf_i}{\log_2 O_j}$$

Aquesta expressió calcula la suma per a tots els termes no buits de l'expressió de la cerca. *Oj* – factor de longitud – és, com a la taula 3.6c el nombre de termes totals emmagatzemats –i. e. excloses per tant les paraules buides– en els índexs del document *dj*, computant les repeticions. El factor idf_i es calcula segons una variant de ***(F57)***

[59] El conjunt s'acostuma anomenar Z39.50/PRISE o simplement ZPRISE.

[60] Accessible a: <http://www-nlpir.nist.gov/projects/zprise/index.html>. Cal obtenir prèviament el «password» lliurant un missatge a <dimmick@nist.gov> amb còpia a <paul.over@nist.gov> [Consulta: 15 de juny].
Hi ha disponible una petita guia d'instal·lació i funcionament a: <http://www-nlpir.nist.gov/works/papers/zp2/zp2.html> [Consulta: 15 de juny].

[61] Tanmateix des de l'any 2000 disposen d'un kit experimental de RI dissenyat per a diferents morfologies de la informació desenvolupat per funcionar sobre una màquina virtual Java 2. La funció de classificació es basa en la combinació de pesos de diversos camps.
Informació a: <http://www.itl.nist.gov/iad/894.02/projects/irf>. [Consulta: 15 de juny de 2003].

(F61) $$idf_i = \log_2(\frac{N}{n_i} + 1)$$

LSI ++

Aquest motor de cerca és fruit de la tesina de postgrau de Todd A. Letsche[62] a la Universitat de Tennesse (1996). L'objectiu era desenvolupar una nova versió de LSI més eficient per a grans col·leccions que les dissenyades fins a aquell moment. El programari està disponible amb llicència GPL[63]. Ara bé, LSI ++ és estrictament un motor de cerca i no duu cap eina d'indexació.

Desenvolupat en C++ sobre UNIX i compilat per a Solaris, la intenció de ser totalment portable fa que no presenti problemes amb compiladors GCC.

Todd A. Letsche va dissenyar també un CGI per tal que serveixi d'interfície d'usuari per al model client-servidor, que permet escollir el nombre de factors *k* **(D10)** per controlar-ne la precisió. L'enginy possibilita la retroalimentació per rellevància.

VERITY ULTRASEEK

Verity és una empresa constituïda al maig de 1988 que ha desenvolupat els seus propis motors de cerca durant anys[64], però que en l'actualitat comercialitza un sistema incorporat en adquirir Inktomi al novembre de 2002. Aquest producte, que s'anomenava Inktomi Enterprise Search, fou incorporat a la seva vegada en comprar Ultraseek, que havia comercialitzat la primera versió d'aquest enginy l'any 1997.

---

[62] Disponible a: <http://www.cs.utk.edu/~library/TechReports/Thesis/Letsche.ps.Z>. [Consulta: 8 de juny de 2003].

[63] Es pot demanar la llicència a: < http://www.cs.utk.edu/~lsi/request2.html>. [Consulta: 8 de juny de 2003]. Hi ha disponible documentació interna, però no sobre el funcionament a: <http://www.cs.utk.edu/ berry/lsi++/index.html>.

[64] Verity Topic (1989) basat en el model vectorial. Permetia, a més de les cerques de Boole, utilitzar tòpics, diccionaris construïts a base de relacionar termes, cosa que modificava la seva ponderació.
Verity Search97 (1997) també estava basat en el model vectorial i integrava totes les prestacions típiques del moment com capacitat d'indexar diversos formats, integració amb robots per recórrer la www, formats a mida dissenyats per l'usuari per a cerques per camps, ... La cerca podia fer-se en llenguatge natural o amb operadors de Boole. Admetia operadors de relació (<,>,...), proximitat, truncaments, paràmetres de reemplaçament, expansió de conceptes basats en recompte estadístic i cerca basada en un exemple.

La versió actual és la 5.0.4 i és possible obtenir en línia una demostració de prova vàlida durant 30 dies[65].

El sistema es basa en un model vectorial que treballa sobre una estructura de camps, de manera que poden ser ponderats per l'administrador. Sense dubtes la seva orientació cap a l'empresa fa que una de les característiques principals sigui poder integrar dades procedents de bases de dades relacionals, amb possibilitats de filtrar termes, camps, taules, consultes o informes. Indexa HTML, XML, text, RTF, MS Office, Adobe Acrobat PDF, PostScript, FrameMaker, Lotus SmartSuite, WordPerfect, objectes binaris de mida gran (binary large objects BLOBs) i més de 100 formats de dades més. Ofereix mòduls per a idiomes diferents.

Hi ha fitxers binaris disponibles per a Solaris 2.6, Solaris 7 and Solaris 8; Windows NT 4.0; Microsoft Windows 2000; i Red Hat Linux 6.0 o posterior.

Permet realitzar cerques de Boole, amb frases exactes, per camps, amb truncaments, utilitzant metadades i amb paràmetres de reemplaçament.

GTP

General Text Parser, GTP[66] és un enginy orientat a objectes (C++ i Java) per a crear estructures de dades per a RI. El seu desenvolupament arranca de la tasca realitzada per a l'elaboració de la tesina de postgrau de Stefen O. Howard. A més hi han col·laborat H. Tang, M. Berry i D. Martin, tots a la Universitat de Tennessee el sistema permet analitzar fitxers de text pla en directoris de manera recurrent, proveu diferents opcions de pes per als termes i realitza descomposicions LSI, tant SVD com SDD. La funció de similitud s'ajusta a una correlació del cosinus.

El software ha estat compilat sobre Solaris, RedHat Linux usant compilador GCC i també sobre Javac, JVM 1.3, 1.4.

[65] Disponible a: <http://downloadcenter.verity.com/dlc/index.jsp>. [Consulta: 15 de juny de 2003].

[66] Es pot demanar la llicència a través de: <http://www.cs.utk.edu/~lsi/gtp-request.html>. Un cop es disposa del <password > el codi font és accessible a: <http://www.cs.utk.edu/~lsi/cgi-bin/gtp-admin/startup.cgi>. [Consulta: 15 de juny de 2003].

També s'ha desenvolupat una millora del sistema – Parallel General Text Parser, PGTP[67] – que permet fer els càlculs de LSI de manera distribuïda (computació paral·lela), però no té l'opció de càlcul amb SDD.

AMBERFISH

Aquest recent enginy és un altre producte desenvolupat per Nassib Nassar i distribuït[68], sota els termes de la versió 2 de GNU General Public License (GPL), per Etymon, qui proveeix de manteniment, llicència comercial i suport.

AMBERFISH està dissenyat per córrer sobre plataformes UNIX i ha estat compilat amb èxit sobre Linux, Solaris, Free CÇBSD i Mac OS. Està codificat en C++ i el principal problema és que no disposa d'una interfície gràfica web/CGI, encara que el programari està dissenyat per interactuar-hi. També pot suportar una interfície Z39.50.

Accepta text pla i XML. Aquests documents els tracta com a una jerarquia niada la qual cosa permet estructurar la cerca. És capaç de realitzar consultes de múltiples bases de dades i permet la indexació modular. Admet els operadors lògics *and* i *or* – però no la negació –. Això és motivat perquè realitza la cerca de cada terme per separat i finalment combina l'expressió de Boole. La cerca de tots els documents que no contenen un terme és onerosa quant a temps de càlcul.

El sistema no extreu prèviament els lexemes dels termes, sinó que permet el truncament per la dreta en l'equació de cerca. També accepta les frases exactes.

La jerarquització per rellevància dels documents recuperats es basa en el model vectorial, que sembla ajustar-se plenament a la correlació del cosinus ***(F7)*** i al model tf-idf per determinar el valor dels pesos ***(F10)***.

[67] Accessible, amb el «password» corresponent, a: <http://www.cs.utk.edu/~lsi/cgi-bin/pgtp-admin/startup.cgi>. [Consulta: 15 de juny de 2003].

[68] La llicència es demana a través d'un formulari accessible a: <http://www.etymon.com/amberfish/amberfish_download.html>. [Consulta: 15 de juny de 2003].

7. Perspectiva sobre passat i futur

Progressos en models matemàtics

Els sistemes de recuperació de la informació (RI) van emergir als any 50 i 60 estretament vinculats al desenvolupament de la informàtica i per tant basats en estàtics processaments de lots (batch). Els anys 70 van iniciar la simbiosi entre els ordinadors i les telecomunicacions i, conseqüentment, l'accés al SRI es convertí en quelcom més dinàmic i interactiu. A la pràctica, això significà que la interacció humà-màquina s'havia de convertir en el centre de la RI. Tanmateix, la cosa no ha estat totalment així, i la major part de l'esforç ha anat a parar en millorar l'eficàcia de la representació automàtica i la cerca, tractant els SRI com a processos estàtics i no interactius.

Cal considerar que hi ha diversos aspectes que expliquen aquest comportament. D'una banda, la pròpia tecnologia condiciona el desenvolupament dels sistemes. Podem tenir un bon exemple comparatiu en l'evolució de les interfícies del sistemes operatius: al principi inexistents; després rudimentàries i basades en caràcters, però cada cop anaven integrant tasques més complexes amb un únic comandament; finalment l'aparició posterior dels entorns gràfics. Però és evident que no tot s'explica per aquestes limitacions ja que, seguint amb el mateix exemple, no podem justificar per qüestions merament de desenvolupament tecnològic la relativament important diferència cronològica d'implementació d'entorns gràfics en els sistemes operatius dels ordinadors de la casa Apple (Mackintosh) i els anomenats compatibles PC, liderats per IBM i Microsoft.

Hi ha raons de senzillesa. Per molt cost que sigui generar, provar i millorar models matemàtics, resulta molt més complexe estudiar les variants del comportament humà. Això també explica la inversió de temps i esforços en una tasca que aporta resultats tangibles en menys temps. L'èxit d'una visió física seguint [Ellis, 1998], quantificable, enfront d'un model més cognitiu i intangible.

Però hi ha altres motius lligats a l'empenta del mercat. Si hi ha poc usuaris i poden adaptar-se a la màquina, per què caldria invertir-hi recursos en alguna cosa que ningú necessita? Tornarem a insistir en aquests aspecte.

Sembla clar que, independentment de la forma que puguin agafar les interfícies d'interacció humà-máquina, darrere ha d'existir un motor que funciona amb algorismes matemàtics, per la senzilla raó que els estats numèrics són l'única cosa que els ordinadors poden manipular. El problema plantejat és el següent: donada una cerca cal recuperar tots els documents que són suficientment rellevants. En termes matemàtics, cal definir el significat de «suficientment», cosa que és relativament fàcil a la pràctica amb algun tipus de fita –. Emperò, resulta més complicat establir què entén el model per «rellevant», i d'aquí totes les variants que hem estudiat, que no fan sinó oferir definicions diferents.

Aquest plantejament evident obvia el fet que el que la cerca expressa pot no ajustar-se al que l'usuari considera rellevant. I és en aquests punt, en el fet d'intentar facilitar que sigui així, i de la manera més intuïtiva possible, on queda més per fer. Però, en qualsevol cas, la inversió de mitjans esmerçada en la millora de l'eficàcia i eficiència en la recuperació de documents cal considerar-la imprescindible, ja que tampoc serviria de res, hipotèticament, que l'usuari pugui expressar de manera unívoca el contingut de l'objecte del seu interès si el sistema és incapaç de recuperar-lo.

El paradigma actual

No cal esperar una revolució, almenys en els propers anys, en els models, com va significar el pas de sistemes basats en cerques de Boole als que usen llenguatges naturals. Per entendre la raó cal fer una mica d'història.

Els sistemes ponderats ja havien demostrat la seva eficàcia durant els 60 i els 70 (SMART, SIRE), i diversos especialistes havien assenyalat la seva importància i utilitat[69]. Tanmateix, els usuaris dels sistemes RI eren bibliotecaris, documentalistes, especialistes en el camp sobre el que realitzaven la cerca o, en el pitjor dels casos, persones força formades intel·lectualment, capaces d'aprendre a gestionar l'estricta lògica de Boole i dels llenguatges controlats. Si a això li sumem les típiques sinergies contraries als canvis, bé per por, comoditat o per interessos corporativistes, entendrem

[69] Una recopilació de tot aquest rerafons de feina teòrica i pràctica, així com una descripció perfectament ambientada de la eclosió dels sistemes de RI no centrats en la lògica de Boole es pot seguir a [Brenner, 1996].

que fins a l'expansió dels 90, en què augmentà espectacularment el nombre d'usuaris potencials, sobretot gràcies a internet i després la www, no apareguin les necessitats comercials de sistemes que introdueixen eines més flexibles i intuïtives, encara que havien començat a covar-se a finals dels 80 amb la proliferació del PC i els CD-Roms.

La situació actual és força diferent. Encara que la recerca continua, no sembla que els models matemàtics hagin de millorar de manera qualitativa la relació matemàtica recuperació/precisió. Les millores que puguin produir-se en aquest sentit semblen més aviat destinades a fer més eficient la tasca informàtica, reduint els temps d'execució o la quantitat de memòria necessària. Per tant, es pot parlar de maduresa[70], tot i que la suma del percentatge de rellevància i precisió que aconsegueixen es mou al voltant del 100, prou lluny de l'ideal 200[71].

Un sistema basat en freqüències té limitacions evidents sobre el seu poder de predicció, més si considerem que els conceptes són compostos impossibles de ponderar de manera unívoca a partir dels seus components. Per això, es podria considerar la situació com a transitòria per insuficient. Però no cal oblidar que el prototipus de documentalista perfecte, que aplica les categories pertinents a un document i només aquestes, tan sols existeix a la ficció, com han demostrat experiència i recerca. Tampoc les tècniques d'intel·ligència artificial semblen poder oferir-nos, almenys a mig termini, noves perspectives, tret d'àmbits potser molt especialitzats.

Així, no ens pot estranyar que els esforços principals ja s'hagin desplaçat cap a d'altres aspectes d'interès que no els simples models de RI, com és el cas del filtrat, la recuperació multilingüe, la recuperació de web o vídeos, i la recuperació interactiva (interacció humà-màquina). Així, els tallers del TREC[72], el més important fòrum sobre RI, ja no incorporen des de l'edició del 2000, els experiments «ad hoc» que eren des del seu inici la principal tasca desenvolupada. Aquests tenien per objectiu investigar el

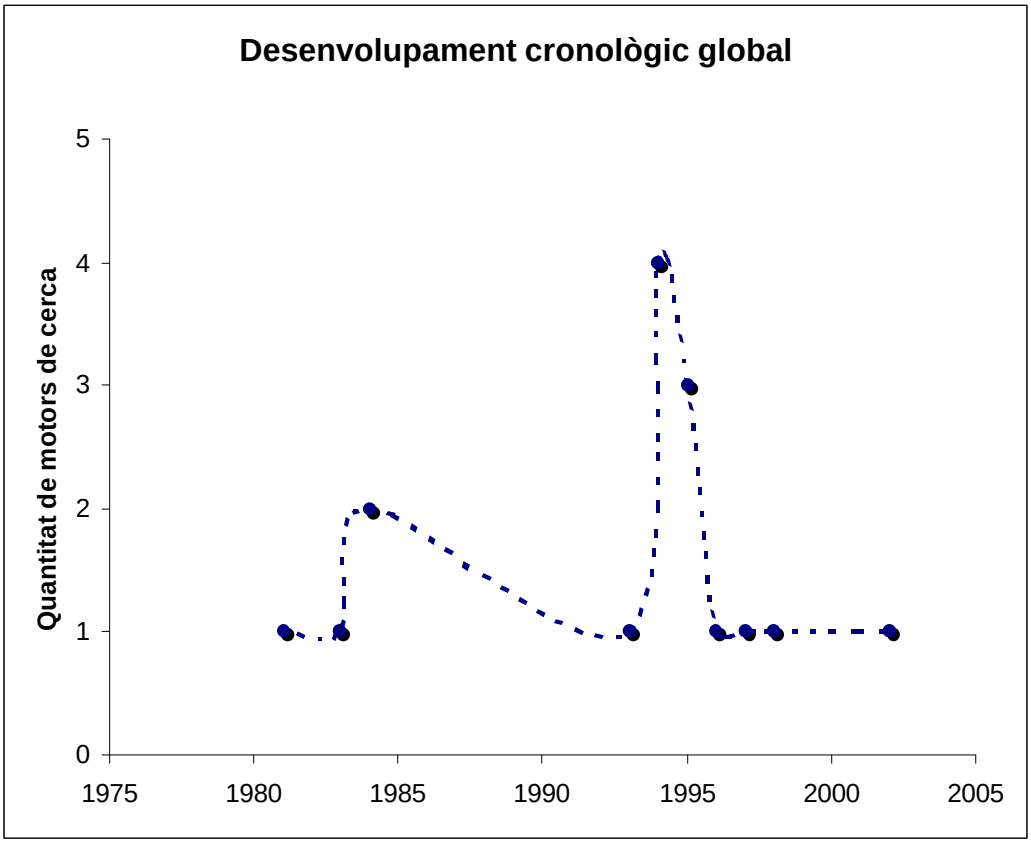
[70] [Harman, 1992b], [Saracevic, 1996].

[71] No podem evitar la comparació, tot i les brutals diferències, amb el principi d'incertesa de Heisenberg per al món quàntic, on és impossible determinar amb tota precisió la posició i la velocitat d'una partícula. Aquí, el principi d'incertesa s'aplica en el sentit que recuperació exhaustiva i gran precisió són incompatibles. Una altra analogia amb el paradigma de la física...

[72] El National Institute of Standards and Technology (NIST) organitza des de 1992 les Text Retrieval Conferences (TREC), un fòrum amb tallers experimentals creats per a fomentar la recerca en recuperació de text amb els següents objectius: estimular la recerca de RI en grans col·leccions; incrementar el bescanvi d'idees entre els entorns acadèmics, industrials i governamentals; augmentar la velocitat de transferència de tecnologia des dels laboratoris de recerca als productes comercials, mostrant les substancials millores en les metodologies de RI en casos reals; i millorar les tècniques d'avaluació dels sistemes.

rendiment del sistemes quan cerquen sobre un important conjunt estàtic de documents de referència, utilitzant unes consultes o tòpics especialment dissenyats. Des de la 9a edició es considera que no calen més experiències d'aquest tipus[73].

Si estudiem la distribució de temps dels projectes de RI que hem analitzat supra, la qual hauria de ser suficientment representativa, s'observa clarament com es dispara el desenvolupament d'aquest enginy amb un pic vers l'any 1995 (gràfic 4.1) , per després tornar a decaure. Evidentment, aquest interès està connectat amb la popularització de la RI a través d'Internet i la www.



Gràfic 4.1

El pas del model de Boole a un llenguatge natural i un sistema jerarquitzat és un canvi conceptualment brusc, per molt que realment a la pràctica hagi estat prou suau, sense oblidar que la lògica de Boole amagada sota interfícies més o menys amigables estigui

[73] [Voorhees, Harman, 2000]

encara avui en dia darrere dels cercadors d'Internet com Google[74] o Altavista[75], tot i que recorrin a tècniques de jerarquia per rellevància basades en visibilitat a la xarxa, solvència dins d'ella, sufragis universals o restringits o qualsevol altra tècnica.

Competència entre models

Si bé hem deixat clar que els sistemes actuals compleixen acceptablement bé la seva missió, la pregunta gens ingènua que cal formular és quin dels models provats funciona millor. Però la veritat és que no estem en condicions de respondre-la.

Conforme la RI ha anat madurant, s'ha anat fent més evident que les diferents tècniques tenen punts fort i dèbils, de manera que alguns documents es recuperen més fàcilment utilitzant un tipus de model i per a documents diferents és preferible emprar altres mètodes. A més cap dels esquemes s'ha mostrat realment superior als altres, Això ha quedat prou evidenciat en les diverses edicions de TREC[76] i altres experiments similars. L'excepció podria ser el model de regressió. Això podria deure's a que els criteris d'optimització per a la regressió – i. e. mínims quadrats o algun altre semblant – no han d'estar necessàriament ben relacionats amb les mesures del rendiment de recuperació, com la mitjana de la precisió amb la qual han estat avaluats.

Bartell, Cottrell, Belew [1994] van comprovar amb èxit l'eficàcia de barrejar diversos algorismes de RI diferents per millorar la recuperació d'informació. La dificultat radica especialment en ponderar els efectes jerarquitzadors de rellevància que subministren cadascun dels algorismes. Belkin, Cool, Croft i Callan [1993] observaren com la progressiva combinació de cerques representant una mateixa necessitat informativa, millorava el rendiment de la recuperació rellevant de documents. Per la seva banda, Lee [1997] treballà en l'expansió automàtica de cerques a partir d'una donada per millorar-ne el rendiment.

[74] <http://www.google.com>. [Consulta: 23 de juny de 2003].
[75] <http://www.altavista.com>. [Consulta: 23 de juny de 2003].
[76] Els resultats en brut poden obtenir-se aconseguint prèviament un «password» a: <http://trec.nist.gov/results>. Però resulten més clars els breus, però interessants resums (Overviews) de D. Harman de les primeres edicions de TREC disponibles a: <http://trec.nist.gov/pubs.html>. [Consultes 23 de juny de 2003].

Tanmateix, s'han apuntat alguns aspectes que caldria assenyalar. Losee [1997], en un estudi comparatiu entre els models que segueixen l'àlgebra de Boole i els probabilístics, demostrà empíricament les limitacions dels primers i, recomanà, a més, la utilització de models que assumeixen la dependència entre termes, tot i els costos de còmput. Els experiments de Rijsbergen i els seus col·legues a finals del 70 semblaven haver demostrat que no es perdia rendiment negligint aquestes relacions. Però en aquella època es treballava amb col·leccions petites. A l'actualitat, sense anar més lluny, les tècniques d'anàlisi de context local amb l'INQUERY, i els treballs dins del TREC amb frases i paràgrafs, poden ser interpretats com un intent de capturar dependències.

Alguns treballs [Berry, Dumais, O'Brien, 1995] suggereixen un rendiment dels algorismes LSI fins a un 30 % superior que els models vectorials. Però aquestes dades no han estat assumides en general per la comunitat científica.

De tot això es pot col·legir que el millor sistema de RI, des de la perspectiva purament matemàtica, és aquell que pugui incorporar diferents tècniques de càlcul amb una cerca expandida i optimitzada per a cadascuna. Caldria també considerar aquí un altre aspecte que no hem tractat, la manera que cada model incorpora la retroalimentació per rellevància.

Però, sigui quin sigui el resultat, els canvis no seran brutals, ja que només influiran en la millora dels resultats a presentar a l'usuari a través d'una interfície. El canvi que es va produir quan el model de Boole deixà de ser el gran protagonista fou brutal, no per afectar els resultats, que ho feia, sinó perquè alliberava l'usuari d'una sintaxi rígida permetent una aproximació successiva als documents realment buscats mitjançant alguna de les tècniques de retroalimentació.

8. Conclusions i treball futur

Per a concloure

En aquest estudi hem presentat de manera sistemàtica els principals models matemàtics de RI formulats fins a aquests moments i agrupats en tres grans paradigmes: teoria de conjunts, algebraics i probabilístics. Posteriorment hem estudiat la seva implementació en motors de cerca de caràcter general, observant que no tots els algorismes han acabat, fins ara, realment incorporats a un sistema de RI.

Els esquemes de funcionament basats en el model clàssic vectorial són els més incorporats a causa de la seva senzillesa, mentre que els probabilístics d'independència binària foren utilitzats només al principi dels anys 80, relativament poc després de la seva formulació i només han subsistit acompanyant els projectes que des de llavors els han mantingut en funcionament. Durant els anys 90 s'han desenvolupat algorismes probabilístics més sofisticats basats en xarxes d'inferència o regressió logística. També han aparegut models algebraics que intenten determinar les relacions semàntiques latents (LSI).

Dels experiments sistemàtics realitzats per a millorar el còmput general de documents rellevants recuperats sense perdre precisió, no s'ha pogut concloure la superioritat evident d'un sistema enfront d'un altre. Més aviat es pot concloure que són parcialment complementaris, i que un enginy capaç de formular i resoldre cerques utilitzant-los paral·lelament obtindria un rendiment força més elevat.

En els darrers anys no s'han generat gairebé nous sistemes de RI, si bé és veritat que es segueix treballant per millorar l'eficàcia i eficiència dels que existeixen. D'altra banda, diverses aplicacions comercials han incorporat aquests enginys com a motors de cerca. Tot això ens permet parlar d'una certa maduresa en aquests tipus de tecnologies. Per tant, no cal esperar en els propers anys canvis substancials en aquesta àrea dels sistemes d'informació.

El que queda per fer

Des de la perspectiva teòrica queda per formalitzar els algorismes de realimentació per rellevància de cadascun dels models tractats. Posteriorment, caldria comparar-los amb la casuística que hem abordat.

Un altre punt a considerar és l'estudi dels models estructurats, que aquí hem deixat totalment de banda, i veure'n la seva implementació, força més recent.

També resten pendents d'abordar els mètodes de classificació i els enginys que els incorporen.

Finalment, l'estudi centrat en el sistemes de RI de text complet es podria estendre a altres morfologies de la informació, com la imatge o el so, on el grau de desenvolupament és força més escàs

9. Bibliografia

1. Gianni Amati, Claudio Carpineto and Gianni Romano (2001). «FUB at TREC-10 Web Track: A Probabilistic Framework for Topic Relevance Term Weighting» [en línia]. **En:** NIST Special Publication 500-250: The Tenth Text REtrieval Conference (TREC 2001). NIST, National Institute of Standards and Technology, 2001. < http://trec.nist.gov/pubs/trec10/papers/fub01.pdf». [Consulta: 26 d'abril 2003]
2. Gianni Amati, Cornelis Joost Van Rijsbergen (2002). «Probabilistic models of information retrieval based on measuring the divergence from randomness». *ACM Transactions on Information Systems*, vol. 20, no 4 (October 2002), p. 357-389.
3. P. Anick, J. Brennan, R. Flynn, D. Hanssen, B. Aley, J. Robbins (1990). «A direct manipulation interface for Boolean information retrieval via natural language query.». **En:** *Proceedings of the thirteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieva*l. New York: ACM Press, p. 135-150.
4. R. Baeza-Yates, B. Ribeiro-Neto (1999). *Modern Information Retrieval*. New York: ACM Press.
5. B. T. Bartell, G. W. Cottrell, R. K. Belew (1992). «Latent Semantic Indexing is an Optimal Special Case of Multidimensional Scaling». **En:** *Proceedings of the fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieva*l. New York: ACM Press, p. 161-167.
6. B. T. Bartell, G. W. Cottrell, R. K. Belew (1994) «Automatic combination of multiple ranked retrieval systems». **En:** *Proceedings of the seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieva*l. New York: ACM Press, p.173 - 181. Disponible en línia <http://citeseer.nj.nec.com/bartell94automatic.html> [Consulta: 1 de maig de 2003].
7. N.J. Belkin, C. Cool, W.B. Croft J.P. Callan (1993). «The effect of multiple query representations on information retrieval performance». *Proceedings of the sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM Press, p. 339-346. Disponible en línia <http://citeseer.nj.nec.com/ belkin93effect.html> [Consulta: 1 de maig de 2003].
8. M. Berry, S. T. Dumais, G. W. O'Brien (1995). «Using linear algebra for intelligent information retrieval». *SIAM Review*, vol 37, no 4 (1995), p. 573-595. Disponible en línia <http://citeseer.nj.nec.com/berry95using.html> [Consulta: 1 de maig de 2003].
9. A. Bookstein (1978). «On the perils of merging Boolean and weighted retrievals systems». *Journal of the American Society for Information Science*, v. 29, n 3 (March 1978), p. 156-178.
10. A. Bookstein (1980). «Fuzzy requests: an approach to weighted Boolean searches». *Journal of the American Society for Information Science*, vol. 31, no. 4 (July 1980), p. 240-247.
11. A. Bookstein (1985). «Implications of Boolean structure for probabilistic retrieval». **En:** *Proceedings of the Eighth Annual International ACM SIGIR Conference on Research and Development in Information Retrieva*l. New York: ACM Press, p. 11-17.
12. E. H. Brenner (1996). *Beyond Boolean: New approaches to information retrieval*. Philadelphia: NFAIS.
13. J. P. Callan (1996). «Document filtering with inference networks». **En:** *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieva*l. New York: ACM Press, p. 262-269.
14. J. P. Callan, Z. Lu, W. B. Croft (1995). «Searching Distributed Collections With Inference Networks». **En:** *Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieva*l. New York: ACM Press, p. 21-28.
15. W. S. Cooper (1991). «Some inconsistencies and misnomers in probabilistic information retrieval». **En:** *Proceedings of the fourteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York: ACM Press, p. 57-61.
16. W. S. Cooper (1994). «The formalism of probability theory in IR: A foundation or an encumbrance?». **En:** *Proceedings of the seventeenth Annual International ACM SIGIR*

Conference on Research and Development in Information Retrieval, New York: ACM Press, p. 242-247.

17. W. S. Cooper, F.C. Gey, D.P. Dabhey. (1992). «Probabilistic Retrieval Based on Staged Logistic Regression». **En:** *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York: ACM Press, p. 198-210.
18. W. B. Croft (1983). «Experiments with representations in a document retrieval system». *Information Technology: Research and Development,* vol. 2, no 1 (January 1983), p. 1-21.
19. W. B., Croft, D. J. Harper (1979). «Using probabilistic models of retrieval without relevance information». *Journal of Documentation*, vol. 35, no 4 (1979) p. 285--295.
20. J. Dowling. (2002). *Information Retrieval using Latent Semantic Indexing and a Semi-Discrete Matrix Decomposition* [en línia]. [Melbourne]: Monash University, October, 2002. <http://www.pcug.org.au/~jdowling/BCompHons.PDF>. [Consulta: 1 maig 2003].
21. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer (1990). «Indexing by latent semantic analysis», *Journal of the American Society for Information Science*, vol 41, no 6 (September 1990) p. 391-407.
22. I. S. Dhillon, D. S. Modha (2001). «Concept Decompositions for Large Sparse Text Data Using Clustering». *Machine Learning*, vol 42, no 1-2 (January-February 2001), p. 143-175.
23. S. T. Dumais (1991). «Improving the retrieval of information from external sources». *Behavior Research Methods, Instruments, & Computers*, vol 23, no 2 (1991), p. 229—236.
24. S. T. Dumais, G. W. Furnas, T. K. Landauer, S Deerwester (1988). «Using latent semantic analysis to improve information retrieval». **En:** *Proceedings of the SIGCHI conference on Human factors in computing systems*, New York: ACM Press, p. 281-285.
25. D. Ellis. «Paradigms and research traditions in information retrieval research». *Information Services & Use*, vol 18, no 4 (1998), p. 225-241.
26. W. B. Frakes, Ricardo Baeza-Yates Eds. (1992). *Information Retrieval: Data Structures & Algorithms*. Englewood Cliffs, NJ: Prentice Hall.
27. N. Fuhr (1989). Models for retrieval with probabilistic indexing. *Information Processing & Retrieval*, vol. 25, no 1 (1989) p. 55-72.
28. N. Fuhr (1992). «Probabilistic models of information retrieval». *Computer Journal*, vol. 35, no 3 (1992) p. 244-255.
29. G. W. Furnas, S. Deerwester, S. T. Dumais, T. K. Landauer, R. A. Harshman, L. A. Streeter, K. E. Lochbaum (1988). *Proceedings of the eleventh Annual International ACM SIGIR Conference on Research and Development in Information Retrieva*l. New York: ACM Press, p. 465-480.
30. F. C. Gey (1994). «Inferring probability of relevance using the method of logistic regression». **En:** *Proceedings of the seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieva*l. New York: ACM Press, p. 222-231.
31. D. Haines, W. B. Croft (1993). «Relevance feedback and inference networks». **En:** *Proceedings of the sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieva*l. New York: ACM Press, p. 2-11.
32. D. Harman (1992a). «Relevance feedback and other query modification techniques». **En:** W. B. Frakes, Ricardo Baeza-Yates Eds. *Information Retrieval: Data Structures & Algorithms*. Englewood Cliffs, NJ: Prentice Hall.
33. D. Harman (1992b). «Overview of the Second Text REtrieval Conference (TREC-2)». [en línia]. **En:** *The Second Text REtrieval Conference (TREC 2).* National Institute of Standards and Technology, august 2000. <http://trec.nist.gov/pubs/trec2/t2_proceedings.html>. [Consulta: 8 d'abril de 2003].
34. D. Harman, G. Candela (1990). «Retrieving records from a gigabyte of text on a minicomputer using statistical ranking». *Journal of the American Society for Information Science*, vol 41, no 8 (December 1990), p. 581-589.
35. Djoerd Hiemstra, Arjen de Vries (2000). «Relating the new language models of information retrieval to the traditional retrieval models». [en línia]. **En:** *CTIT Technical Report TR-CTIT-00-*

09. [Enschede]: University of Twente, may 2000. <http://www.ub.utwente.nl/webdocs/ctit/1/00000022.pdf> [Consulta: 26 d'abril de 2003].

36. K. L. Kwok (1995). «A Network Approach to Probabilistic Information Retrieval». *ACM Transactions on Information Systems*, vol. 13, no 3 (July 1995), p. 325-354.
37. D. H. Kraft and D. Buel (1983). «Fuzzy sets and generalised boolean retrieval systems». *International Journal of Man-Machine Studies*, vol. 19, no. 1 (January 1983), p. 45-56.
38. T. G. Kolda (1997). *Limited-Memory Matrix Methods with Applications* [en línia] [Maryland]: University of Maryland, College Park [1997]. PhD thesis, The Applied Mathematics Program. <http://citeseer.nj.nec.com/115586.html> . [Consulta: 1 de maig de 2003].
39. T. G. Kolda, D. P. O'Leary (1998). «A semidiscrete matrix decomposition for latent semantic indexing information retrieval». *ACM Transactions on Information Systems*, vol. 16, no 4 (octubre 1998), p. 322-346. Disponible en línia <http://citeseer.nj.nec.com/kolda97 semidiscrete.html> [Consulta: 1 de maig de 2003].
40. Ray R. Larson, Jerome McDonough, Paul O' Leary, Lucy Kuntz (1996). «Cheshire II: Designing a Next-Generation Online Catalog». *Journal of the American Society for Information Science*, vol. 47, no. 37 (1996) p. 555-567.
41. J. H. Lee (1994). «Properties of extended Boolean models in information retrieval». **En:** *Proceedings of the seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieva*l. New York: ACM Press, p. 182-190.
42. J. H. Lee (1997). «Combining Multiple Evidence from Different Relevant Feedback Networks». **En:** Rodney W. Topor, Katsumi Tanaka (Eds.): *Database Systems for Advanced Applications '97, Proceedings of the Fifth International Conference on Database Systems for Advanced Applications (DASFAA), Melbourne, Australia.* Melbourne: World Scientific, p. 421-430. Disponible en línia <http://citeseer.nj.nec.com/9989.html> [Consulta: 1 de maig de 2003].
43. J. H. Lee, W. Y. Kim, Y. H. Lee, (1993). «Ranking documents in thesaurus-based Boolean retrieval systems». *Information Processing and Management*, vol. 30, n 1 (1993), p. 79-91.
44. J. H. Lee, W. Y. Kim, M. H. Kim, Y. J. Lee (1993). «On the Evaluation of Boolean Operators in the Extended Boolean Retrieval Framework». **En:** *Proceedings of the sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieva*l. New York: ACM Press, p. 291-297.
45. J. J. Lee, P. B. Kantor (1991). «A study of probabilistic information retrieval systems in the case of inconsistent expert judgments». *Journal of the American Society for Information Science*, vol. 42, no 3 (April 1991), p. 166-172.
46. R. M. Losee (1997). «Comparing Boolean and Probabilistic Information Retrieval Systems across Queries and Disciplines». *Journal of the American Society for Information Science*, vol 48, no 2 (February 1997), p. 143-156.
47. R. M. Loose, A. Bookstein (1988). «Integrating Boolean queries in conjunctive normal form with probabilistic retrieval models». *Information Processing and Management*, vol. 24, no. 3 (1988), p. 315-321.
48. H. Luhn (1953). «A new method of recording and searching information». *American Documentatio*n, vol 4, no 1 (1953), p. 14-16.
49. H. Luhn (1957). «A statistical approach to mechanized encoding and searching of literary information». *IBM Journal of Research and Development*. Vol 1, no 4, p. 309-317.
50. U. Manber, S. Wu (1993). «GLIMPSE: A Tool to Search Through Entire File Systems». [en línia]. Tuckson: The University of Arizona, October 1993. «http://glimpse.cs.arizona.edu/pubs/glimpse.pdf». [Consulta 26 d'abril de 2003].
51. C. D. Manning, H Schütze (1999). Foundations of statistical natural language processing. Massachusetts: MIT Press.
52. M. E. Maron, J. L. Kuhns (1960). «On relevance, probabilistic indexing and information retrieval», *Journal of the Associations of Computing Machinery*, vol. 7, no.. 3 (July 1960), p. 216-244.

53. S. Miyamoto, T. Miyake (1986). «Fuzzy information retieval based on a fuzzy pseudothesaurus». EEE Transactions on Systems and Man Cybernetics, vol. 16, no 2 (1986), p. 278-282.

54. S. Miyamoto, T. Miyake, K. Nakayama (1983). «Generation of a pseudothesaurus for information retrieval based on cooccurrences and fuzzy set operations. IEEE Transactions on Systems and Man Cybernetics, vol. 13, no 1 (1983), p. 62-70.

55. Y. Ogawa, T. Morita, and K. Kobayashi (1991). «A fuzzy document retrieval system using the keyword connection matrix and its learning method». *Fuzzy Sets and Systems*, vol. 38 (1991), pp. 17-41.

56. Vijay V. Raghavan, S. K. M. Wong (1986). «A critical analysis of vector space model for information retrieval». *Journal of the American Society for Information Science*, vol. 37, no 5 (September 1986), p. 279—287.

57. T. Radecki (1976) « Mathematical model of information retrieval system based on the concept of Fuzzy thesaurus». *Information Processing & Management*, vol. 12, no. 5 1976, p. 313-318..

58. T. Radecki (1979), «Fuzzy Set Theoretical Approach to Document Retrieval». *Information Processing & Management*, vol. 15, no.5 (1979) p. 247-259.

59. B. A. Ribeiro-Neto, R. Muntz (1996). «A belief network model for IR». **En:** *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retriev*al. New York: ACM Press, p. 253-260.

60. C. J. Rijsbergen (1975). *Information Retrieval*. London: Butterworths.

61. C. J. Rijsbergen (1975/79). *Information Retrieval* [en línia]. [Glasgow: University of Glasgow]. < http://www.dcs.gla.ac.uk/Keith/Preface.html> [Consulta: 14 abril 2003].

62. C. J. Rijsbergen (1979). *Information Retrieval*. 2a ed. London: Butterworths.

63. S.E. Robertson, K. Sparck Jones (1960). «Relevance weighting of search terms». *Journal of the American Society for Information Science*, vol. 27, no.3 (May 1976), p. 129-146.

64. W. M. Sachs, W. (1976). «An Approach to Associative Retrieval Through the Theory of Fuzzy Sets». *Journal of the American Society for Information Science*, vol. 27, no.2 (March 1976), p. 85-87.

65. G. Salton, C. Buckley (1988). «Term-weighting approaches in automatic text retrieval». *Information Processing and Management*, vol. 24, no.5 (1988) p. 513-523.

66. G. Salton, E. Fox, H. Wu (1983). «Extended Boolean information retrieval». *Communications of the ACM*, vol. 26, n11 (November 1983), p. 1022-1036.

67. G. Salton, M. E Lesk (1968). «Computer evaluation of indexing and text processing». *Journal of the Associations of Computing Machinery*, vol. 15, no.1 (January 1968), p. 8-36.

68. G. Salton, M. J. McGill (1983). *Introduction to Modern Information Retieval*. New York: McGraw-Hill.

69. G. Salton, E. A. Fox, H.Wu (1983). «Extended boolean information retrieval». *Communications of the ACM*, vol. 26, no.11 (November 1993), p. 1022-1036.

70. G. Salton, C.S. Yang (1973). «On the specification of term values in automatic indexing». *Journal of Documentation*, vol. 29, no.4 (April 1973), p- 351-372.

71. E. Sanchis, L. Moreno, I. Gil Eds. (2002). *I Jornadas de Tratamiento i Recuperación de la Información (JOTRI).* Valencia: Editorial de la UPV.

72. T. Saracevic (1996). «Modeling interaction in information retrieval (IR): a review and proposal». **En:** *Proceedings of the American Society for Information Science*, vol 33. Maryland: ASIS, p. 3-9.

73. K. Sparck Jones (1972). « A Statistical interpretation of term specificity and its application in retrieval». J*ournal of Documentation*, vol. 28, no.1 (March 1972), p. 11-20.

74. K. Sparck Jones (1979a) «Experiments in relevance weighting of search terms». *Information Processing and Management*, vol. 15 (1979), p. 133-144.

75. K. Sparck Jones (1979b) «Search Term Relevance Weighting Given Little Relevance Information». Journal of Documentation, vol. 35, no.1, (March 1979), p. 30-48.

76. K. Spark Jones, S. Walker and S.E. Robertson [1998]. *A probabilistic model of information retrieval: Development and status*. [En línia]. Technical Report 446. Cambridge: University of Cambridge. <http://citeseer.nj.nec.com/jones98probabilistic.html> [Consulta: 1 de maig de 2003].

77. V. Tahani (1976). «A Fuzzy Model of Document Retrieval Systems». Information Processing & Management, vol. 12, no.3 (1976), p. 177-187.

78. H. Turtle (1991). *Inference Networks for Document Retrieval.* Ph. D. dissertation. [Amherst]: University of Massachusetts. Disponible en línia <http://citeseer.nj.nec. com/turtle91inference. html> [Consulta: 1 de maig de 2003].

79. H. Turtle, W. C. Croft. (1990). «Inference networks for document retrieval». **En:** *Proceedings of the thirteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieva*l. New York: ACM Press, p. 1-24.

80. H. Turtle, W. C. Croft. (1991). «Evaluation of an inference network-based retrieval model». *ACM Transactions on Information Systems*, vol. 9, no 3 (July 1991), p. 187-222.

81. J. Verhoeff, William Goffman, Jack Belzer (1961): «Inefficiency of the use of Boolean functions for information retrieval systems». *Communications of the ACM*, vol. 4 , no.12 (December 1961), p. 557-558.

82. E. Voorhees, D. Harman (2000). <Overview of the Ninth Text REtrieval Conference (TREC-9) [en línia]. **En:**. *The Ninth Text REtrieval Conference (TREC 9).* National Institute of Standards and Technology, February 2002. <http://trec.nist.gov/pubs/trec9/t9_proceedings.html>. [Consulta: 8 d'abril de 2003].

83. R. Wilkinson, P. Hingston (1991). «Using the cosine measure in a neural network for document retrieval». **En:** *Proceedings of the fourteenth annual international ACM SIGIR Conference on Research and development in Information Retrieva*l. New York: ACM Press, p. 202–210.

84. I. H. Witten, A. Moffat, C. Bell (1994). *Managing Gigabytes: Compressing and Indexing Documents and Images.* New York: Van Nostrand Reinhold.

85. I. H. Witten, A. Moffat, C. Bell (1999). *Managing Gigabytes: Compressing and Indexing Documents and Images.* 2a ed. San Francisco: Morgan Kaufmann Publishing.

86. S. K. Wong, W. Ziarko, C. N. Wong (1985). **«**Generalized vector spaces model in information retrieval ». **En:** *Proceedings of the eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieva*l. New York: ACM Press, p. 18-25.

87. C.T. Yu, G. Salton (1976). «Precision weighting. an effective automatic indexing method». *Journal of the Associations of Computing Machinery*, vol. 23, no 1 (June 1976), p. 76-88.