CAPITOLO 5
RICERCARE SU INTERNET

Paolo FERRAGINA e Fabrizio LUCCIO

Ove si spiega come sia possibile frugare in un lampo tra gli innumerevoli dati disseminati nella rete.

Nella storia recentissima del genere umano, cioè nel corso degli ultimi diecimila anni, la conservazione del sapere è passata in modo cruciale attraverso l'invenzione della scrittura. A nulla valse l'accorata opposizione di Socrate secondo cui questa invenzione avrebbe avuto per effetto il:

> produrre la dimenticanza nelle anime di coloro che l'impareranno, perché fidandosi della scrittura si abitueranno a ricordare dal di fuori mediante segni estranei e non dal di dentro e da sé medesimi.[1]

L'affermazione è molto profonda ma dovette apparire estrema anche ai suoi tempi, al punto che Platone per esporla la scrisse. Il sapere era già così vasto da non poter essere raccolto e conservato stabilmente nella mente di ogni singolo uomo, e attraverso una sconfinata messe di scritti, immagini, e recentemente di registrazioni sonore e video, è approdata nei supporti elettronici su cui operano i «browser» e i «motori di ricerca» di cui trattiamo in questo capitolo. Nella stessa occasione Socrate osservò anche qualcosa di molto più preoccupante: la scrittura non avrebbe procurato ai discepoli la vera sapienza perché essi, meri uditori senza insegnamento:

> crederanno di essere conoscitori di molte cose mentre, come accade per lo più, in realtà non le sapranno; e sarà ben difficile discorrere con essi perché sono divenuti portatori di opinioni invece che sapienti.

Questo pericolo per chi usa Internet è più serio oggi di allora.

---

[1] Platone, *Fedro 275*.

Parliamo dunque della ricerca su Internet. Essa viene effettuata tramite sistemi efficientissimi che rendono possibile reperire in un attimo un'enorme messe di informazioni memorizzate nella rete in punti potenzialmente molto lontani dal richiedente e sostanzialmente imprevedibili anche per un utente esperto. Esamineremo qui la recentissima storia di questi sistemi per capire come funzionano, accennando alle reti elettroniche su cui operano ed entrando nel merito di alcuni loro algoritmi di funzionamento; e poiché questi algoritmi sono molto complessi potremo solo cercare di darne conto a grandi linee, sufficienti per comprenderne i principi di base e le difficoltà che sono state incontrate nella loro realizzazione, rimandando ogni approfondimento alla letteratura specializzata. Compresi i principi su cui funzionano i sistemi di ricerca il lettore potrà sviluppare una riflessione personale sul loro impatto sulla società di oggi e di domani: argomento di dominio delle scienze sociali ma alla cui discussione dovrebbe legittimamente partecipare solo chi abbia qualche conoscenza scientifica di base.

La prima riflessione scientifica è che il sapere di cui possiamo fruire, costituito da tutto ciò che appare nei libri, nei film, nelle registrazioni sonore e così via, può essere completamente descritto con una sequenza di enorme lunghezza costruita con un alfabeto arbitrario che, nell'informatica, comprende solo i due caratteri 0 e 1. Anche le emozioni che riguardano ogni uomo al suo interno contribuiscono a quella sequenza quando si prova a comunicarle, perché i sensi dell'uomo hanno una «soglia» di percezione sotto la quale non si distinguono segnali diversi. Così in un CD si può registrare la musica in modo praticamente indistinguibile dai suoni emessi dagli strumenti originali, benché la registrazione sia di fatto realizzata in una sequenza di areole sulla superficie del disco che si trovano in uno tra due stati fisici possibili, a loro volta descritti con simboli binari. O un'immagine si può rappresentare su uno schermo video ad alta definizione come insieme di punti (detti «pixel») così piccoli da essere percepiti dall'occhio umano come parte di un'immagine continua, e ciascun punto può assumere un colore rappresentato, con lo standard di oggi, da una sequenza di ventiquattro cifre binarie (tre gruppi di otto cifre che rappresentano le gradazioni del rosso-verde-blu: per esempio la sequenza 01100101 00000000 00000000 rappresenta un bel rosso lacca scuro). Quindi sia i punti che i loro colori appartengono a un universo discreto, ma così ricco di possibilità da essere percepito come continuo.

È dunque su sequenze di caratteri che si svolge qualsiasi ricerca di dati. Conoscere l'alfabeto di riferimento, binario in ogni applicazione informatica, è ovviamente essenziale per realizzare le diverse operazioni richieste ma non ha alcuna

importanza come base teorica del discorso, perché si può passare da un alfabeto a un altro senza cambiare la natura dei fenomeni studiati. Così per esempio possiamo riferirci a numeri rappresentati nella consueta notazione decimale o più in genere a testi scritti con la tastiera di un computer lasciando a questo, o agli stessi circuiti della tastiera, il compito di trasformare il testo nella sequenza binaria che sarà successivamente elaborata. In queste trasformazioni di alfabeto non c'è proprio nulla d'interessante. Molto d'interessante c'è invece nelle proprietà delle sequenze, nel modo di catalogarle o confrontarle tra di loro, di estrarne parti predeterminate, o ripetute, o comuni tra diverse di esse. Su queste basi avviene la ricerca di dati nella rete: oggi in maniera «sintattica» utilizzando proprietà morfologiche delle sequenze; in un prossimo futuro anche in maniera «semantica», cioè guidata dal significato di una sequenza o di alcune sue parti. Ma per avventurarci in questo studio dobbiamo prima stabilire alcuni concetti di base su due entità che vivono in simbiosi ma sono in ogni altro senso distinte, che si chiamano Internet e Web.

5.1 I GRAFI DI INTERNET E DEL WEB

Come abbiamo visto nel Capitolo 2 un «grafo» è un'entità costituita da un insieme di «nodi» e da un insieme di «archi» che rappresentano relazioni tra nodi. Nata alla fine del XVIII secolo, la teoria dei grafi ha assunto grande importanza nel secolo scorso come mezzo per rappresentare entità in relazione tra loro ed è essenziale per lo studio della rete Internet e delle sue applicazioni.

Una rete di computer (nel seguito diremo semplicemente 'rete') è un insieme di apparati che si scambiano messaggi in forma elettronica attraverso connessioni che fanno uso di cavi, fibre ottiche, collegamenti radio o a raggi infrarossi: per quanto ci riguarda le connessioni sono rappresentate con gli archi di un grafo di cui i computer costituiscono i nodi. Internet è in realtà una rete di reti in quanto diverse istituzioni che possiedono molti computer collegati tra loro entrano in Internet come una singola entità. Queste unità sono chiamate in gergo «sistemi autonomi» (in sigla AS) e costituiscono i nodi del grafo di Internet, ciascuno dei quali può rappresentare un utente dotato di un singolo computer, o un insieme di computer connessi tra loro all'interno di un edificio, fino a un'intera rete, anche molto complessa, che può contenere computer geograficamente lontanissimi tra loro. Sistemi autonomi di quest'ultimo tipo sono per esempio gli «Internet service provider» (ISP), compagnie che regolano il traffico dei messaggi su Internet vendendo i loro servizi ad altri utenti.

La Figura 5.1 mostra un possibile frammento del grafo di Internet: i nodi sono rappresentati da cerchi collegati con archi a tratto continuo che possono essere percorsi in entrambe le direzioni. Un messaggio diretto da G a B, per esempio una e-mail, potrà seguire il percorso G – E – F – D – B (l'utente G paga il servizio al proprio provider E che regola i costi con i successivi), ma la cosa è molto più complicata di così. La rete Internet è immensa: anche se non si sa quanti computer vi siano collegati (domanda d'altronde mal posta perché la rete varia continuamente) si può stimare che tale numero supererà il miliardo entro il 2010 ove si considerino tutti i computer all'interno degli AS. A causa di queste enormi dimensioni, della struttura 'anarchica' di Internet che cresce e evolve senza alcun controllo centralizzato, e dell'inevitabile variazione continua delle connessioni dovute a problemi tecnici o di manutenzione, il percorso seguito da un messaggio può essere diverso da quello previsto prima del suo invio o addirittura variare durante la sua trasmissione. Ciò ha caratterizzato la rete sin dalla sua nascita alla fine degli anni Sessanta, distinguendola per esempio dalle reti telefoniche o di distribuzione di energia elettrica. È opportuno esporre brevemente di che si tratta.

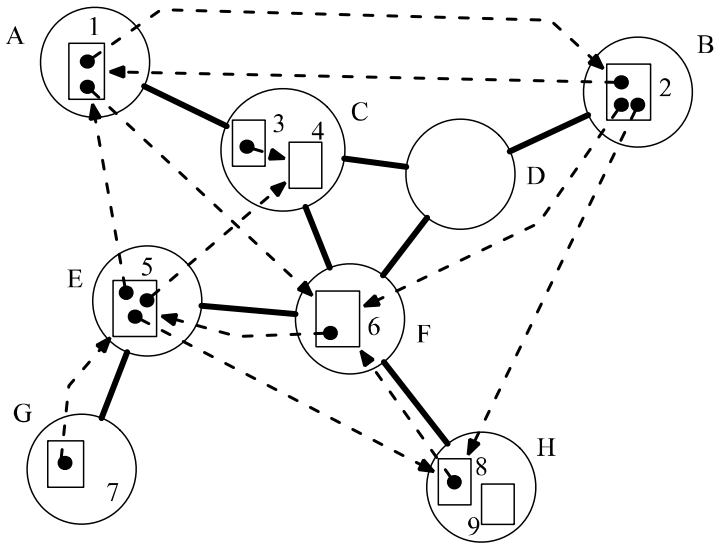

Figura 5.1: La struttura dei grafi di Internet e del Web. I cerchi rappresentano gli AS e gli archi segnati con tratto continuo rappresentano i collegamenti tra essi. I rettangoli rappresentano le pagine Web e gli archi segnati con frecce tratteggiate rappresentano i link tra esse.

Le reti telefoniche tradizionali (ma Internet sta mischiando le carte anche qui) funzionano a «commutazione di circuito»: due utenti che parlano al telefono

utilizzano un «canale» loro riservato che si realizza su una catena di collegamenti predeterminata dalla centrale telefonica e su certe frequenze assegnate ai due: se qualcosa va storto si interrompe la trasmissione. L'idea iniziale, per Internet, fu di introdurre una «commutazione di messaggio» nella quale l'istradamento nella rete era stabilito, nodo per nodo, a seconda della posizione del destinatario e del traffico momentaneo: un messaggio destinato a un utente 'vicino' poteva quindi essere spedito attraverso un percorso molto più lungo del necessario se il canale breve era pesantemente impegnato, e la scelta del nodo successivo era demandata al nodo in cui il messaggio si trovava al momento. Questo metodo subì presto una variazione che è quella adottata oggi ed è nota come «commutazione di pacchetto». Il messaggio, che come abbiamo visto è una sequenza di segnali binari, è diviso in «pacchetti» la cui lunghezza è specificata all'inizio di ogni pacchetto assieme all'indirizzo di destinazione. Può quindi avvenire che diversi pacchetti dello stesso messaggio siano istradati su percorsi diversi per essere rimessi in fila dal destinatario. In tal modo un messaggio è sempre accettato dalla rete, anche se può trascorrere del tempo prima che tutti i pacchetti abbiano raggiunto la destinazione finale e il messaggio possa essere ricostruito. In una rete telefonica, invece, un numero può risultare 'occupato' prima di averlo completato anche se è libero il telefono del destinatario, perché nella catena di connessioni prevista tra le due locazioni un tratto risulta saturato da altre chiamate in corso.[2]

Vediamo ora cosa si intende per Web (o 'www' per World Wide Web, letteralmente una «ragnatela vasta come il mondo»). Nato nel 1989 presso il CERN di Ginevra e basato sul già noto concetto di «ipertesto», il Web è un insieme di documenti detti «pagine» che si richiamano tra loro per raffinare o approfondire uno stesso soggetto, o per richiamare un nuovo soggetto in qualche relazione col primo. La fortuna del Web è legata indissolubilmente a quella di Internet perché le sue pagine sono registrate nelle memorie dei computer della rete, permettendo a un utente di passare da una pagina all'altra seguendo un «link» contenuto nella prima,

[2] Un tempo ciò accadeva di frequente. Oggi la cosa è piuttosto rara per il miglioramento delle tecniche di trasmissione, ma può ancora accadere se si chiama un paese lontano o se si prova a fare una chiamata nazionale la notte dell'ultimo dell'anno. Succede difficilmente se si utilizzano sistemi «VOIP» (voice over Internet protocol) in cui, lungo quasi tutto il percorso, il messaggio è trasmesso su Internet: in questo caso però i pacchetti possono essere ritardati dal traffico generando una certa frammentazione del messaggio vocale.

in tempo brevissimo e senza lasciare il proprio posto di lavoro, ovunque risieda la pagina richiesta.[3]

Visto in questo modo anche il Web si rappresenta con un grafo ove i nodi sono le pagine e gli archi sono i link tra queste, ora dotati di un senso di percorrenza dalla pagina che contiene il link a quella richiamata da esso. La Figura 5.1 mostra anche una porzione del grafo del Web in cui gli archi sono «orientati» mediante frecce. Poiché i motori di ricerca raccolgono, ispezionano e rendono disponibili le pagine Web dovremo riferirci a questo grafo di cui è bene stabilire subito alcune proprietà.

Anzitutto il grafo è letteralmente immenso e ininterrottamente variabile a causa della continua creazione di nuove pagine e alla trasformazione o alla cancellazione di altre. Sulle dimensioni del grafo si leggono notizie a volte infondate e fantasiose: quel che è certo è che i motori di ricerca mettono a disposizione degli utenti decine di miliardi di pagine ma quelle esistenti sono molte di più, anche se di reperibilità meno diretta.[4]

Altra osservazione importante è che, sebbene le pagine Web siano memorizzate nei computer di Internet, i due grafi non hanno alcun'altra relazione tra loro. Riferendosi all'esempio di Figura 5.1, la pagina 1 punta alla pagina 2 ma non vi è alcun collegamento diretto tra i due nodi A e B di Internet, cioè tra i computer che contengono tali pagine; d'altra parte i due nodi C e F sono connessi ma non vi sono link tra le pagine ivi contenute. Inoltre gli archi del Web sono orientati mentre quelli di Internet non lo sono. Nei fatti il grafo di Internet è «fortemente connesso», cioè, a meno di temporanee interruzioni di qualche collegamento, esiste sempre un percorso che connette qualunque coppia di nodi; questo invece non si verifica nel Web ove esistono anche nodi che non possono essere raggiunti da alcun altro perché hanno solo link uscenti (i nodi 3 e 7 nell'esempio), nodi da cui non se ne raggiungono altri perché hanno solo link entranti (4 nell'esempio), o nodi che non hanno alcun collegamento (9 nell'esempio). Un approfondimento di carattere algoritmico di queste importanti caratteristiche è presente nel riquadro seguente.

---

[3] Come sa chiunque abbia avuto accesso a Internet, le pagine Web appaiono sotto forma di schermate sui monitor e i link sono indicati con parole o frasi messe graficamente in evidenza. Selezionando un link con un'operazione detta «click» si accede alla pagina indicata dal link.

[4] Si parla di «Web indicizzabile» come insieme delle pagine che potrebbero essere raggiunte dai motori di ricerca, anche se questi non le raccolgono tutte (vedi oltre). Un altro Web, detto «profondo» (deep), include una quantità ancora più vasta di informazioni organizzate in basi di dati locali e/o reperibili utilizzando apposito software. Un importante settore di studio riguarda oggi la possibilità di estendere le funzioni dei motori di ricerca anche all'informazione del Web profondo.

Matrici di adiacenza e percorsi in un grafo

Un grafo G di *n* nodi può essere rappresentato all'interno di un computer mediante una «matrice di adiacenza» M di *n* righe e *n* colonne, le une e le altre messe in corrispondenza ai nodi di G. La cella di M nella riga *i* e colonna *j*, indicata con M[*i,j*], corrisponde alla coppia di nodi *i*, *j* e si pone M[*i,j*]=1 se G contiene un arco dal nodo *i* al nodo *j*, M[*i,j*]= 0 altrimenti. I grafi di Internet e del Web di Figura 5.1 hanno le matrici di adiacenza I di dimensioni 8×8, e W di dimensioni 9×9, indicate nella Figura 5.2. Si noti che per un grafo non orientato come quello di Internet la matrice I è simmetrica rispetto alla diagonale: nell'esempio l'arco A-C può essere percorso nei due sensi, quindi si ha I[A,C]=1 e I[C,A]=1; questa simmetria può non esistere invece per un grafo orientato come quello del Web: nell'esempio si ha W[1,2]=1 e W[2,1]=1 perché nel grafo vi sono i due archi distinti nelle due direzioni, ma si ha anche che W[1,6]=1 e W[6,1]=0.

	A	B	C	D	E	F	G	H
A	0	0	1	0	0	0	0	0
B	0	0	0	1	0	0	0	0
C	1	0	0	1	0	1	0	0
D	0	1	1	0	0	1	0	0
E	0	0	0	0	0	1	1	0
F	0	0	1	1	1	0	0	1
G	0	0	0	0	1	0	0	0
H	0	0	0	0	0	1	0	0

I

	1	2	3	4	5	6	7	8	9
1	0	1	0	0	0	1	0	0	0
2	1	0	0	0	0	1	0	1	0
3	0	0	0	1	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	1	0	0	1	0	0	0	1	0
6	0	0	0	0	1	0	0	0	0
7	0	0	0	0	1	0	0	0	0
8	0	0	0	0	0	1	0	0	0
9	0	0	0	0	0	0	0	0	0

W

	1	2	3	4	5	6	7	8	9
1	1	0	0	0	1	1	0	1	0
2	0	1	0	0	1	2	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	1	0	0	0	2	0	0	0
6	1	0	0	1	0	0	0	1	0
7	1	0	0	1	0	0	0	1	0
8	0	0	0	0	1	0	0	0	0
9	0	0	0	0	0	0	0	0	0

W^2

Figura 5.2: Le matrici I e W dei grafi di Internet e del Web di Figura 5.1, e la matrice W^2.

In matematica il quadrato di una matrice M di dimensioni *n*×*n* è una nuova matrice M^2 delle stesse dimensioni, i cui elementi $M^2[i,j]$ sono calcolati in modo un po' bizzarro come «prodotto» della riga *i* per la colonna *j* di M, secondo la formula:

$$M^2[i,j] = M[i,1]\times M[1,j] + M[i,2]\times M[2,j] + \ldots + M[i,n]\times M[n,j] \qquad (1)$$

Questa formula ha per noi una grande importanza che illustreremo attraverso la matrice quadrato di W, anch'essa indicata in Figura 5.2. Per esempio abbiamo posto $W^2[6,4] = 1$: infatti la riga 6 e la colonna 4 di W sono rispettivamente: 0 0 0 0 1 0 0 0 0 e 0 0 1 0 1 0 0 0 0 , e applicando la formula (1) abbiamo:

$W^2[6,4]$ =
= M[6,1]×M[1,4]+M[6,2]×M[2,4]+ … +M[6,5]×M[5,4]+ … +M[6,9]×M[9,4]
= 0×0 + 0×0 + 0×1 + 0×0 + 1×1 + 0×0 + 0×0 + 0×0 + 0×0 = 1

ove il valore finale 1 è dato dall'incontro di due 1 nelle caselle W[6,5] (in riga 6) e W[5,4] (in colonna 4). Ma poiché queste due caselle indicano che nel grafo del Web vi sono un arco dal nodo 6 al 5, e uno dal 5 al 4, concludiamo che vi è un percorso dal 6 al 4 che attraversa esattamente due archi. Dunque la matrice quadrato di un grafo indica i percorsi di due archi ivi contenuti, e un ulteriore esame della W^2 ci aiuterà a capire meglio il meccanismo. Abbiamo W[1,2] = 1 ma $W^2[1,2] = 0$ perché vi è un arco (percorso di lunghezza uno) da 1 a 2 ma non vi è alcun cammino di lunghezza due tra gli stessi nodi (vedi Figura 5.1). Abbiamo poi $W^2[2,6] = 2$ derivante dal prodotto tra la riga 2 e la colonna 6 di W, rispettivamente 1 0 0 0 0 1 0 1 0 e 1 1 0 0 0 0 0 1 0, secondo il calcolo:

$$W^2[2,6] = 1\times1 + 0\times1 + 0\times0 + 0\times0 + 0\times0 + 1\times0 + 0\times0 + 1\times1 + 0\times0 = 2$$

Qui si incontrano due coppie di 1: W[2,1]=W[1,6]=1 e W[2,8]=W[8,6]=1 a indicare che nel grafo del Web vi sono i due cammini di due archi 2–1–6 e 2–8–6 che collegano il nodo 2 al nodo 6.

Si procede con la stessa regola nel calcolo delle successive potenze W^3, W^4, … della matrice che indicano il numero di percorsi di lunghezza 3, 4, … tra ogni coppia di nodi, ossia il numero di link che bisogna seguire sul Web per spostarsi da una pagina a un'altra. Per esempio gli elementi di W^3 si ottengono, sempre utilizzando l'espressione (1), come prodotto di una riga di W^2 per una colonna di W (o viceversa); si ha così:

$$W^3[7,6] = 1\times1 + 0\times1 + 0\times0 + 1\times0 + 0\times0 + 0\times0 + 0\times0 + 1\times1 + 0\times0 = 2$$

che corrisponde ai due cammini di tre link 7–5–1–6 e 7–5–8–6.

Gli algoritmi classici calcolano la potenza di una matrice a partire da due matrici con esponente più basso, costruendo tutte le coppie riga-colonna e applicando a ciascuna di esse l'espressione (1). Poiché tali coppie sono n^2 in numero, e per calcolare l'espressione (1) è necessario un tempo sostanzialmente proporzionale a n (infatti l'espressione contiene n moltiplicazioni e n-1 addizioni), il tempo di calcolo complessivo richiesto dall'algoritmo è proporzionale a (ovvero 'è di ordine') n^3. Naturalmente il tempo effettivamente richiesto da un algoritmo dipende, oltre che dalla sua formulazione, dal calcolatore e dal linguaggio di programmazione impiegati; ma un comportamento 'cubico' come quello presente significa per

esempio che se il numero n di nodi del grafo raddoppia passando da n a $2n$, il tempo cresce come $(2n)^3 = 8n^3$, cioè diviene otto volte maggiore[5]. E poiché il grafo del Web contiene miliardi di nodi tale tempo diverrebbe incredibilmente grande se il calcolo dovesse essere sostenuto da un solo computer. Poiché, come vedremo, questo calcolo è uno degli ingredienti principali per stabilire un ordinamento d'importanza tra le pagine del Web, la soluzione è quella di dividere il lavoro tra moltissimi computer con tecniche di «calcolo distribuito» troppo complesse per poter essere accennate qui.

5.2 I BROWSER E UN PROBLEMA «DIFFICILE»

Cinque anni dopo la sua nascita il Web conteneva un numero di pagine stimabile in una decina di milioni e il loro reperimento da parte degli utenti costituiva un problema molto serio. Ciò dette origine alla nascita di complessi sistemi software con le funzioni di «browser»[6], e successivamente alla creazione di «motori di ricerca». Vediamo di che si tratta.

Una pagina Web è descritta, nel computer che la contiene, mediante uno speciale linguaggio che permette di specificarne i componenti (in particolare scritte ed elementi grafici) e come essi debbano essere mostrati o manipolati sullo schermo di un computer. Ciascuna pagina è caratterizzata da un indirizzo che è ovviamente una sequenza binaria, ma a cui si associa per comodità un nome più facile da ricordare: per esempio «www.unipi.it/index.htm» è il nome (URL in gergo) della pagina Web principale dell'Università di Pisa. Per evitare confusione è bene tener presente che le pagine Web si presentano di norma in gruppi correlati tra loro perché pertinenti a uno stesso ente, e sono contenute in uno stesso computer detto «Web server». Ci si riferisce al gruppo di pagine come al «sito Web» dell'ente. Il sito è caratterizzato dal suo «nome di dominio», per esempio «unipi.it»; la sua pagina principale ha un indirizzo che include quello del Web server («www.unipi.it» nell'esempio), e punta ad altre pagine del gruppo attraverso link in essa contenuti: è il caso delle pagine 3 e 4 in Figura 5.1.

---

[5] Per questo tipo di analisi cfr. § 2.3.

[6] Browser si potrebbe tradurre in italiano «brucatore» o per esteso «frugatore», ma entrambe le voci sono poco eleganti. In effetti si usa solo la voce inglese.

L'indirizzo delle pagine secondarie è un raffinamento del nome precedente tramite una struttura gerarchica e lessicale di barre, ad esempio: www.unipi.it/ricerca/dottorati è l'indirizzo della pagina che contiene la lista dei dottorati di ricerca di Pisa, le cui pagine specifiche sono raggiungibili da questa con link successivi. Il browser consente dunque di visualizzare una pagina Web attraverso il suo indirizzo. Se non si conosce l'indirizzo di una pagina secondaria si può accedere ad essa raggiungendo la pagina principale del sito e seguendo i successivi link; ma si deve conoscere o immaginare almeno il nome del sito per impiegare il browser in questa cosiddetta «ricerca per navigazione».

Dopo il primo browser sviluppato al CERN per il Web originale, sono seguiti prodotti commerciali famosi come Netscape Navigator, Internet Explorer, Firefox e tanti altri. Una osservazione fondamentale è che copie delle pagine Web vengono riunite in grandi gruppi e concentrate in nodi della rete a ciò dedicati per permettere di reperirle più in fretta (vedi oltre). Un punto di concentrazione, cioè un computer o una rete locale di computer, prende il nome intraducibile di «cache», e «caching» ne è la tecnica di impiego. Facciamo ora riferimento alla Figura 5.3 che mostra una tipica organizzazione di connessioni e cache nel grafo di Internet, in maggior dettaglio rispetto alla Figura 5.1.

Un ruolo preminente è svolto dai cosiddetti «proxy», cioè computer con funzione di cache locale all'AS. Anzitutto essi mantengono una copia delle pagine Web richieste più frequentemente dai browser dell'AS, che spesso sono le stesse per diversi browser: tali pagine possono essere state richieste all'esterno dell'AS attraverso Internet, o al suo interno perché contenute in altri computer dell'AS stesso. Nell'esempio di Figura 5.3 se gli utenti A e B hanno richiesto la stessa pagina questa è memorizzata nel proxy 1 per uso di entrambi. Inoltre i proxy memorizzano e forniscono alla rete Internet le pagine Web dell'AS richieste frequentemente dall'esterno: in tal modo le richieste vengono servite immediatamente senza traversare e affollare la rete interna all'AS. Nell'esempio se alcune pagine contenute nei computer A e B sono spesso richieste dagli altri utenti della rete, copie di tali pagine sono mantenute nel proxy 1 e di lì smistate direttamente all'esterno. Un livello di caching ancora superiore è fornito dalle CDN (per «Content Delivery Networks») introdotte nel 2000: si tratta di sottoreti di computer con funzione di cache di enormi dimensioni che servono vaste aree geografiche. È bene notare che le stesse pagine Web possono essere duplicate e memorizzate in molte cache di tutti i livelli.

Consideriamo ora il problema di distribuire le pagine nelle cache della rete in modo da minimizzare il tempo complessivo di accesso a quelle prevedibilmente richieste dai browser in un certo periodo di tempo. Vedremo che il problema è sostanzialmente insolubile dal punto di vista algoritmico perché è di «complessità esponenziale»: esso rientra infatti in una classe di problemi «difficili» di cui si è dato conto ampiamente nel Capitolo 3 di questo volume, per i quali dobbiamo accontentarci di soluzioni approssimate. Per mostrare tutto questo riprendiamo il «problema della bisaccia» (cfr. § 2.2) apparentemente diversissimo ma, come vedremo, correlato al nostro per quanto riguarda il tempo necessario a risolverlo.

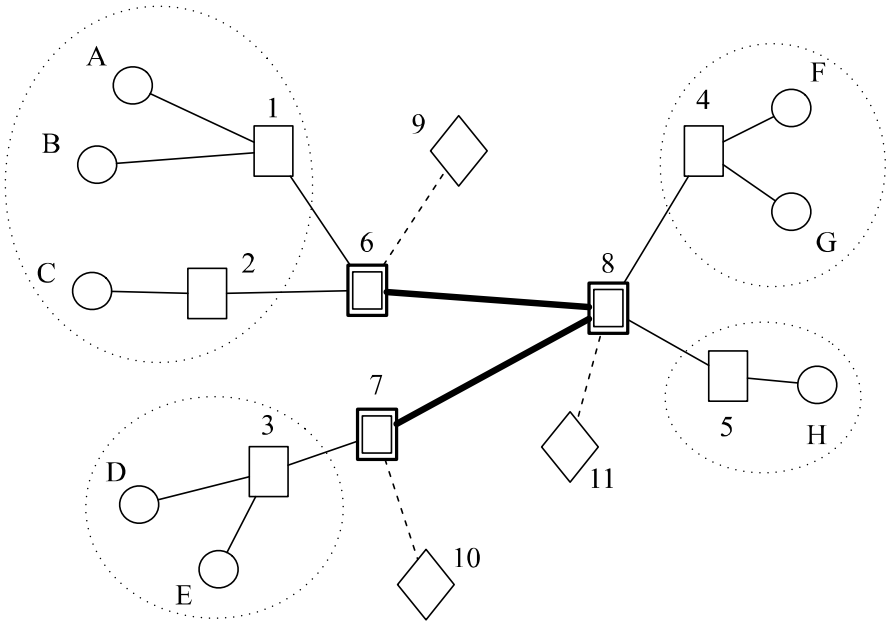

Figura 5.3: Diversi livelli di caching per le pagine Web. I cerchi A, B, …, H sono postazioni di browser, per esempio singoli computer degli utenti, e mantengono al loro interno le pagine Web richieste più di recente. Le zone racchiuse in linee punteggiate sono semplicissimi sistemi autonomi e i rettangoli ivi contenuti sono «proxy». I rettangoli doppi sono «router», cioè apparati che instradano i messaggi nella rete Internet indicata a tratto spesso. Infine i rombi sono grandi sottoreti CDN.

Consideriamo l'esempio di Figura 5.4. Nella tabella si considerano 7 oggetti, per comodità numerati da 1 a 7, e per ogni oggetto si riporta il peso e il valore corrispondente (l'oggetto 1 ha peso 15 e valore 13, l'oggetto 2 ha peso 45 e valore 25 e così via). Nell'ultima riga della tabella i valori 1 e 0 indicano rispettivamente se un oggetto è stato selezionato o no (nell'esempio gli oggetti selezionati sono l'1,

il 3, il 4 e il 7). Supponiamo che la capacità C della bisaccia sia 70. Ricordiamo che il problema consiste nel decidere quali oggetti selezionare in modo da trovare una soluzione che massimizzi la somma dei valori, mentre la somma dei pesi rimane limitata dalla capacità della bisaccia. Come già sottolineato nel Capitolo 2 questo problema, a meno di banali casi particolari, non ammette ad oggi altra strategia di soluzione che quella di provare tutte le possibili combinazioni degli elementi dati, scartando quelle la cui somma dei pesi superi la capacità C e scegliendo fra tutte le altre una di quelle che massimizza la somma dei valori. Poiché tutte le possibili combinazioni sono in numero di 2^n, la soluzione richiede tempo esponenziale e quindi enorme anche per un limitato numero di oggetti.[7]

C = 70	Oggetti	1	2	3	4	5	6	7
	Peso	15	45	11	21	8	33	16
	Valore	13	25	14	15	6	20	13
		1	0	1	1	0	0	1

Figura 5.4: Un esempio di problema della bisaccia: scegliere tra sette oggetti un gruppo di essi che abbia massimo valore e il cui peso complessivo non superi 70. Pesi e valori degli oggetti sono indicati nelle righe Pesi e Valori, rispettivamente. L'ultima riga indica la scelta ottima {1, 3, 4, 7} attraverso la posizione degli 1. Il peso complessivo è 15+11+21+16 = 63 < 70; il valore complessivo è 13+14+15+13 = 55.

Nel Riquadro *Il problema della Cache* mostriamo la relazione che c'è fra il problema della bisaccia e quello della distribuzione delle pagine Web nella cache. Qui sottolineiamo che una formulazione generale del secondo problema è molto complicata perché deve tener conto della probabilità con cui viene richiesta ogni pagina Web, e della frequenza complessiva delle richieste provenienti da ogni AS. Basterà riferirsi a un modello semplificato per vedere che il secondo problema «è difficile almeno quanto» il primo: cioè se esistesse un algoritmo non esponenziale per risolvere il problema della cache, questo sarebbe applicabile con semplici trasformazioni al problema della bisaccia che dunque a sua volta non sarebbe esponenziale. Il che, come sappiamo, non è ritenuto possibile. In tal modo possiamo dimostrare che anche il problema della cache è altrettanto difficile.[8]

---

[7] Per esempio per $n = 20$ si ha $2^{20} > 1$ milione.

[8] Questo meccanismo di «riduzione» tra problemi è lo standard per assegnare a un nuovo problema un livello di complessità già noto per altri.

Il problema della Cache

Ammettiamo dunque, per semplificare al massimo il problema, che la rete contenga k AS indicati con AS_1, …, AS_k e n pagine indicate con p_1, …, p_n; che ciascuna pagina sia memorizzata nel proprio AS di origine; che tutte le pagine siano richieste con la stessa probabilità e che ogni AS richieda lo stesso numero di pagine; infine, che esista in tutta la rete una sola cache di dimensione B, presso un solo AS, in cui possano essere duplicate le pagine. Come nel problema della bisaccia possiamo definire un vettore P di pesi, ove P[j] è la dimensione del file[9] che rappresenta p_j: la somma dei «pesi» delle pagine poste nella cache non potrà superare la sua dimensione B. Più difficile è definire l'equivalente del vettore dei valori per il nuovo problema.

A tale scopo indichiamo con A^j l'AS proprietario della pagina p_j; indichiamo con $d(i,j)$ la distanza (cioè il numero di passi nella rete Internet) tra un generico AS_i e A^j; e indichiamo con $c(i,j)$ la distanza minima tra AS_i e la pagina p_j tenendo presente che questa è contenuta in A^j ma potrebbe trovarsi anche nella (unica) cache presente nella rete. Così se p_j è nella cache e la distanza tra questa e AS_i è minore di $d(i,j)$ abbiamo $c(i,j) < d(i,j)$, altrimenti $c(i,j) = d(i,j)$. Il valore $u(i,j) = d(i,j)-c(i,j) \geq 0$ indica l'eventuale vantaggio per AS_i che la pagina p_j sia contenuta nella cache: vantaggio effettivo se $u(i,j) >0$. A questo punto possiamo definire il valore V[j] di p_j come il vantaggio, calcolato su tutti gli AS, che p_j sia contenuta nella cache. Abbiamo:

$$V[j] = \sum_{i=1..k} u(i,j).$$

Analogamente a quanto fatto nel problema della bisaccia possiamo allora considerare un vettore ausiliario A per indicare una scelta di pagine da memorizzare nella cache.

Il problema del caching, pur in questa formulazione estremamente semplificata, è quindi almeno difficile quanto quello della bisaccia. Quindi, a maggior ragione, è esponenziale il problema del caching nella sua versione generale. Che fare dunque per questo problema e per tutti quelli parimente difficili? Se la soluzione ottima si trova esponenzialmente lontana da noi nel tempo dovremo accontentarci di una soluzione approssimata: e qui il metodo di soluzione cambia da problema a problema.

[9] Si definisce file un contenitore di informazioni digitalizzate organizzate sequenzialmente.

Per il problema della bisaccia, per esempio, è naturale dare precedenza agli oggetti che abbiano un rapporto alto tra valore e peso, scegliendoli in ordine decrescente di tale rapporto finché la bisaccia può contenerli: questa euristica è illustrata in Figura 5.5. Per il problema del caching, in particolare nelle CDN, sono state studiate varie strategie: la più semplice è quella basata sulla «popolarità» delle pagine richieste dai browser degli AS prossimi alla CDN, che si memorizzano in questa, fino a sua saturazione, in ordine decrescente rispetto al numero delle richieste. Migliori risultati si ottengono con algoritmi più sofisticati che fanno uso della conoscenza delle connessioni della rete, dando per esempio la precedenza in una CDN C_i alle pagine p_j con più alto valore del prodotto $\pi_j \times d(i,j)$, ove π_j è la frequenza di richieste della pagina p_j. Notiamo che questi algoritmi si basano su ragionevoli ipotesi teoriche ma la loro qualità si valuta in modo completamente sperimentale.

oggetto	3	1	7	5	4	6	2
P	11	15	16	8	21	33	45
V	14	13	13	6	15	20	25
V/P	1.27	0.87	0.81	0.75	0.71	0.61	0.55

Figura 5.5: Gli oggetti della Figura 5.4 ordinati per valore decrescente del rapporto V[*i*]/P[*i*]. La scelta euristica di elementi è {3,1,7,5} con peso totale 50 e valore totale 46: confrontando questa soluzione con quella ottima di Figura 5.4 (peso 63, valore 55), si nota che la bisaccia ha capacità libera di 70 - 50 = 20 ma non può ospitare alcuno degli oggetti non scelti.

5.3 I MOTORI DI RICERCA

I browser sono strumenti utilissimi se l'utente conosce l'indirizzo della pagina desiderata, ma inutili se si desidera reperire le pagine attraverso il loro contenuto senza conoscerne l'indirizzo. È a quest'ultima necessità che rispondono i motori di ricerca, indispensabili oggi per reperire informazioni nello sconfinato grafo del Web. I tre motori più famosi si chiamano Google, Yahoo! e Bing; sono basati su principi simili ma sufficientemente diversi perché, rispondendo alla stessa interrogazione, indichino al richiedente pagine Web in parte differenti e presentate con stile d'accesso proprio di ciascun motore. Non faremo qui un confronto dei diversi motori, limitandoci a indicarne la struttura comune e a presentare gli algoritmi che realizzano alcune loro funzioni fondamentali.

La necessità di raccogliere informazioni in tutto il Web, classificarle opportunamente e presentarle agli utenti nel brevissimo tempo che tutti conosciamo, richiede per ogni motore di ricerca l'impiego di un numero enorme di computer raggruppati in diversi centri operativi distribuiti nel mondo. Anche se le diverse compagnie sono riluttanti a fornire dati precisi, si stima che per ciascun motore operino centinaia di migliaia di computer raccolti in sottoreti dedicate a funzioni diverse e non sempre completamente distinte tra loro. Possiamo distinguere le funzioni di un motore di ricerca in due categorie: quelle destinate alla costruzione di un archivio di pagine, e quelle destinate alla risoluzione di una interrogazione posta dall'utente. Tra le prime ricordiamo il «crawling» e l'analisi del grafo del Web, l'analisi delle pagine raccolte, e la costruzione infine di un «indice» che contenga tutte le informazioni utili alla risoluzione veloce delle interrogazioni. Queste funzioni vengono ripetute a intervalli di tempo regolari per garantire che l'indice (e quindi le risposte fornite agli utenti) siano al passo con le continue variazioni del Web.

Per rispondere all'interrogazione di un utente il motore di ricerca esegue una serie di altre funzioni che utilizzano l'indice corrente al fine di recuperare le pagine «rilevanti» per l'interrogazione stessa. Questa solitamente si presenta sotto forma di una sequenza di «parole chiave» relative al bisogno di informazione che si cela dietro quella interrogazione[10]. Il processo di astrazione che porta l'utente a scegliere le parole chiave è critico in quanto da esso dipende la qualità dei risultati restituiti dal motore di ricerca. È infatti evidente che una interrogazione può essere risolta mediante diversi gruppi di risposte «rilevanti», che dipendono dall'utente che l'ha formulata e dalle sue intenzioni del momento. Ad esempio l'interrogazione "Lufthansa" potrebbe avere per l'utente una connotazione «navigazionale» perché questi vuole trovare il sito della compagnia aerea; «commerciale» perché vuole acquistare un biglietto; «informativa» perché l'utente è interessato alle vicende societarie o ai numeri del suo «call center».

Chiaramente si potrebbe estendere un'interrogazione mediante l'aggiunta di alcune parole che precisino meglio l'intenzione dell'utente, ma ciò raramente accade: statistiche recenti mostrano che più dell'80% delle interrogazioni sono costituite al

[10] Questo approccio è detto «bag of words» a sottolineare che ciò che conta è l'insieme delle parole costituenti una interrogazione e non il loro ordine. Tuttavia il lettore potrà verificare sul suo motore preferito che l'ordine delle parole ha un impatto sui risultati restituiti. I motori offrono anche delle opzioni che consentono di rendere più precisa l'interrogazione; queste però sono poco utilizzate dagli utenti che preferiscono comunque la (limitante) semplicità del «bag of words».

più da due termini e la media è circa 2.5. Non si tratta solo di pigrizia a comporre interrogazioni selettive, ma anche di difficoltà da parte degli utenti a trovare le parole chiave adeguate a esprimere le loro necessità di informazione. In una ricerca pubblicata dal Corriere della Sera nell'aprile del 2001, su un campione di 856 navigatori italiani tra i venticinque e i cinquantacinque anni che utilizzavano Internet regolarmente, il 33% di questi trovava sempre serie difficoltà nell'uso dei motori di ricerca dell'epoca, e il 28% trovava difficoltà solo alcune volte. Tutto ciò induceva il giornalista a concludere che il metodo d'impiego dei motori, non mutato da allora, «genera stress, frustrazione e senso di smarrimento nel mare del Web. […] E quasi un italiano su tre sogna un motore di ricerca automatico e intelligente che non agisca solo per parole chiave». Su questo aspetto torneremo alla fine del capitolo: per ora notiamo che gli algoritmi di ricerca sul Web devono superare, a ogni interrogazione, numerose difficoltà: non ultima quella che gli utenti esaminano in media non più di dieci risposte tra quelle ottenute, quindi una sola pagina di risultati.

Nonostante tutti questi problemi i motori di ricerca svolgono oggi egregiamente il loro compito, con prestazioni in efficienza e rilevanza delle risposte che migliorano di giorno in giorno grazie ai progressi della ricerca industriale e accademica in campo algoritmico. Descriviamo ora le caratteristiche salienti delle funzioni di un motore di ricerca notando però che, come è facile immaginare, non tutti gli algoritmi impiegati sono pubblicamente noti.

5.3.1 Il crawling

Nel linguaggio di Internet il termine «crawling» indica la raccolta delle pagine dal grafo del Web, possibilmente di 'tutte' le pagine, senza altro scopo che quello di renderle disponibili ai successivi algoritmi di analisi, catalogazione e «indicizzazione». Un programma di crawling è detto in gergo «crawler» (nuotatore), o anche «spider» (ragno) o «robot»: il termine più appropriato è forse l'ultimo, ed è anche il meno usato.[11]

---

[11] «Crawler» indica letteralmente colui che si trascina sul terreno, ma il significato sportivo di chi nuota in 'stile libero' è in genere più evocativo tra i cultori di Internet, che amano anche 'navigare' sulla rete. Questi termini sono completamente fuorvianti: il crawler è un programma che risiede in un computer del motore di ricerca e richiede che gli siano inviate le pagine Web tramite il protocollo di trasmissione impiegato dalla rete, senza spostarsi da nessuna parte (in caso contrario potrebbe essere visto come virus informatico); così come l'utente che naviga in Internet rimane sulla propria sedia in attesa che gli siano mostrate le informazioni richieste: più che un navigatore è un turista potenziale che consulta cataloghi di paesi lontani sul divano di un'agenzia di viaggi.

È importante sottolineare la fondamentale differenza tra browser e crawler: mentre il primo richiede il recupero di specifiche pagine indicate da un operatore, il crawler richiede una serie di pagine con un metodo completamente automatico. Gli algoritmi di funzionamento di browser e crawler sono quindi totalmente distinti e determinano la differenza di ruolo tra un sistema di browsing (per esempio Internet Explorer) e un motore di ricerca (per esempio Google).

Nel riquadro seguente esporremo più in dettaglio come funziona un algoritmo di crawling.

Un algoritmo di crawling

Un algoritmo di crawling impiega due strutture di dati (cfr. Riquadro *Strutture dati* del Capitolo 2) dette «coda» e «dizionario» con lo stesso significato che hanno questi termini nel linguaggio dei trasporti (una coda di auto) e della linguistica (un dizionario di termini). Una coda Q è un accumulo di elementi allineati in attesa di un servizio. L'elemento che si trova in testa a Q è il primo a ricevere tale servizio, e perché ciò avvenga l'elemento viene tolto dalla coda facendo emergere in testa a essa l'elemento successivo: questa operazione si indica con Q→E, ove E indica l'elemento ormai liberato all'esterno. Per inserire in Q un nuovo elemento E, questo si pone in fondo alla coda e sarà servito dopo tutti quelli ivi contenuti: l'operazione si indica con E→Q.

Anche un dizionario D è un accumulo di elementi (non necessariamente termini di un linguaggio umano) in attesa di esame, ma questi possono essere gestiti in modo più flessibile rispetto alla coda: ciò che qui interessa è che vi sia un metodo rapido per stabilire se un elemento è già presente nel dizionario e possibilmente estrarlo (D→E), e un metodo di inserzione di un nuovo elemento (E→D): si noti che i due meccanismi non possono essere scelti indipendentemente l'uno dall'altro perché la possibilità di eseguire una ricerca rapida nel dizionario dipende dal modo in cui gli elementi vi sono inseriti e organizzati. Un elemento chiave per l'efficienza di queste operazioni è che esista un ordinamento tra gli elementi del dizionario, ma in informatica tale ordinamento può essere sempre artificialmente indotto. Abbiamo infatti osservato nell'introduzione di questo capitolo che gli elementi di D, qualunque sia la loro natura, devono essere rappresentati come sequenze binarie nei computer. Tali sequenze possono allora essere ordinate per esempio in ordine «alfabetico» (nell'alfabeto binario il carattere 0 precede 1, dunque un ordinamento tra quattro sequenze è per esempio: 010 0110 10 1011 110).

Vediamo ora in modo necessariamente molto semplificato come funziona un crawler secondo l'algoritmo indicato in Figura 5.6. Notiamo anzitutto che i proprietari di un sito Web possono vietare di estrarre informazioni dal sito stesso, in tutto o in parte, ponendo nel sito un file «robots.txt» che specifica al suo interno le pagine a cui è negato il diritto di accesso. L'algoritmo impiega una coda Q per contenere gli indirizzi di pagine Web da esaminare, e due dizionari D_{url} e D_{pagine} per contenere rispettivamente gli indirizzi delle pagine Web già esaminate e i «file» che contengono le relative informazioni. All'inizio delle operazioni D_{url} e D_{pagine} sono vuoti, mentre Q contiene un insieme di indirizzi a partire dai quali il crawler inizierà la sua esplorazione del Web. Come è facile immaginare la scelta di questi indirizzi è cruciale per raggiungere pagine di rilevanza generale in tempi ragionevoli: tipicamente si utilizzano gli indirizzi di «portali», ossia pagine Web che contengono una lista nutrita di risorse presenti su Internet (ad esempio, DMOZ, Yahoo!, ecc.) e di siti governativi, educativi (ad esempio Wikipedia) o ricreativi famosi e quindi ricchi di link verso altre pagine importanti del Web.

Input: U_1, …, U_k gli indirizzi scelti inizialmente

Output: D_{url} e D_{pagine}

Passo 1: Si pongano U_1, …, U_k nella coda Q

Passo 2: Si ripete

Q → U si estrae l'indirizzo in testa alla coda

se U non è presente in D_{url} allora

richiedi R(U), il file robots.txt presente nel sito U

se la pagina U non è vietata da R(U) allora

richiedi T(U), il testo della pagina U

U → D_{url}

T(U) → D_{pagine}

scandisci T(U) e, per ogni link U' non in D_{url}, poni U' → Q

fintanto che Q non è vuota

Figura 5.6: Struttura essenziale di un algoritmo di crawling.

L'algoritmo di Figura 5.6 non è elementare. Per comprenderlo si deve notare che quando un nuovo link U' è reperito all'interno di una pagina l'indirizzo è inserito in Q per un futuro esame se non è contenuto in D_{url}, cioè se la sua pagina T(U') non è ancora stata scaricata. Si noti però che lo stesso link U' può essere contenuto in due o più pagine e queste possono essere incontrate prima che T(U') sia scaricato. Per evitare che U' sia caricato più volte in D_{url} bisogna quindi controllare che gli elementi estratti da Q non siano già stati elaborati. A tale proposito si deve eseguire una ricerca rapida nel dizionario D_{url} del tipo *Ricerca Dicotomica* (cfr. Figura 2.23).

Ovviamente un algoritmo di crawling realmente utilizzabile include molte altre funzionalità e ottimizzazioni rispetto al codice precedente, tutte ugualmente importanti. Ne citeremo alcune per dar conto della complicazione della sua progettazione e realizzazione. Anzitutto il grafo del Web varia con una velocità altissima tanto che si stima che si rinnovi del 30% ogni anno, e i crawler sono in genere addestrati a seguirne le variazioni più rapide durante una stessa sessione di funzionamento, e devono essere sufficientemente veloci da consentire frequenti ripetizioni della scansione del Web per mantenere aggiornato l'indice del motore. In particolare un crawler deve ottimizzare tre parametri e cioè: il numero N di pagine Web gestibili prima che i suoi algoritmi e le sue strutture dati vengano 'sopraffatte' dal numero di pagine presenti in D_{url}; la velocità S con cui è in grado di scaricare pagine dal Web, che oggi raggiunge picchi di migliaia di pagine al secondo; infine la quantità di «risorse computazionali» (CPU[12], spazio di memoria e disco) utilizzate per portare a termine le operazioni. Chiaramente più grandi sono N e S, maggiore sarà il costo di mantenimento della coda Q, e delle strutture di dati D_{url} e D_{pagine}; di contro, più efficiente è la gestione di D_{url} e D_{pagine}, minore sarà la quantità di risorse computazionali utilizzate e quindi il consumo di energia, problema estremamente rilevante visto l'altissimo numero di computer utilizzati per il funzionamento di questi motori.

Vi sono anche altri aspetti che un progettista di crawler deve tenere in considerazione: per esempio la riduzione dell'interferenza con il sito esaminato per evitare che questo risulti intasato dalle continue richieste di pagine dei crawler a scapito del servizio offerto agli utenti; la scelta delle pagine da ri-esaminare più

[12] Si definisce CPU (Central Processing Unit) l'unità centrale di elaborazione di un qualunque calcolatore.

frequentemente come news, blog, ecc.; l'uso di tecniche di «calcolo distribuito» e di «resistenza ai guasti» per garantire che il crawler non si interrompa mai nel suo funzionamento e raggiunga elevati valori di S e di N. Per approfondire questi argomenti si veda la nota bibliografica in fondo al capitolo.

5.3.2 Il grafo del Web in maggior dettaglio

È naturale a questo punto chiedersi quanto sia ampio il grafo del Web e quale sia la sua struttura di interconnessione, poiché da queste caratteristiche dipende l'efficacia del processo di crawling chiamato a un compito colossale.

Abbiamo già notato come il Web sia enorme e cambi velocemente: non c'é quindi alcuna speranza che un crawler raccolga in D_{pagine} tutte le pagine presenti sul Web e ci si deve accontentare di un sottoinsieme ampio e più rilevante possibile. I parametri N e S definiti sopra influenzano direttamente la dimensione del sottoinsieme raccolto, cioè la cosiddetta «copertura» del grafo, ma per quanto riguarda la qualità delle pagine estratte il crawler deve adottare una tecnica di visita del grafo più complessa di quella dell'algoritmo di Figura 5.6. A tale scopo al posto della coda Q si impiega solitamente una struttura di dati più sofisticata, detta «coda con priorità» (cfr. Capitolo 4), in cui l'ordine di estrazione degli elementi dipende da una priorità a essi assegnata. Nel nostro caso gli elementi sono indirizzi di pagine Web e la priorità è quindi legata alla rilevanza di quelle pagine. Tanto maggiore è questa priorità, quanto prima la pagina sarà visitata dal crawler. L'obiettivo è quindi quello di assegnare una bassa priorità alle pagine poco rilevanti o a quelle già viste, anche se duplicate in un altro sito (questo riguarda almeno il 20% del Web), o alle pagine di un sito troppo grande da essere raccolto nella sua interezza. In quest'ultimo caso si tiene conto della 'profondità' di una pagina nel suo sito, indicata dal numero di barre presenti nell'indirizzo, per esempio la pagina http://www.unipi.it/ricerca/index.htm è meno profonda della pagina http://www.unipi.it/ricerca/dottorati/index.htm. Si segue la ratio che le pagine che appaiono 'in fondo' a un sito sono presumibilmente le meno rilevanti, e comunque sono sempre raggiungibili dalle loro pagine antenate che a loro volta saranno state indicizzate e probabilmente contengono link a esse. Studi recenti hanno dimostrato che il numero e la qualità dei link di ingresso e uscita di un sito sono indicatori efficaci per l'assegnazione di queste priorità.

Per quanto riguarda l'influenza della struttura del grafo del Web sul funzionamento dei crawler, notiamo anzitutto che se questo grafo fosse formato da numerosissime componenti disgiunte, la coda del crawler dovrebbe contenere un indirizzo di

partenza in ciascuna di esse per poterle raggiungere; oppure se il grafo fosse costituito da una catena di pagine, la scelta casuale di un indirizzo da mettere nella coda porterebbe in media a visitare solo la metà del Web.

Nel novembre del 1999 uno studio, oggi storico, ha analizzato la struttura di un sottografo del Web di quel tempo formato da circa duecento milioni di pagine. Risultò così che «quella» porzione di Web era formata da quattro componenti principali indicate nella Figura 5.7, aventi tutte circa la stessa dimensione: un sottografo SCC fortemente connesso, denominato «core» o nucleo[13]; un sottografo IN con cammini che finiscono in pagine di SCC; un sottografo OUT con cammini che partono da SCC; e una serie di «tentacoli» e «tubi», cioè pagine collegate in catene non passanti per SCC o del tutto isolate. Questi risultati sono stati confermati successivamente da studi eseguiti su campioni di Web più recenti e più ampi: non solo il grafo rilevato ha sempre la forma indicata nella Figura 5.7 e le dimensioni delle sue componenti hanno circa lo stesso rapporto tra loro, ma la struttura del grafo è «scalabile», cioè qualsiasi suo sottografo di dimensioni non troppo piccole ha la stessa struttura del grafo da cui proviene. Questi risultati inizialmente sorprendenti sono oggi giustificati da studi matematici sulle leggi di crescita delle reti.

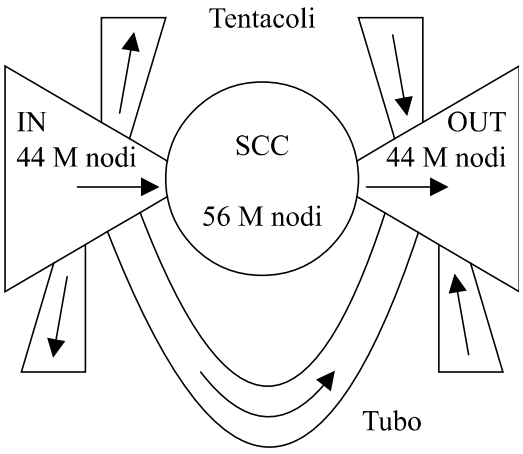

Figura 5.7: La caratteristica forma a «papillon» del grafo del Web (1999). I sottografi SCC, IN, OUT, e Tentacoli e Tubi, occupano ciascuno circa un quarto del grafo totale.

[13] Un grafo orientato si dice fortemente connesso se esiste un cammino tra una qualunque coppia di nodi in qualunque ordine essi siano presi.

La struttura del grafo del Web indica che le pagine da inserire nella coda del crawler devono essere scelte in IN o al più in SCC. Si è visto che IN e SCC contengono gran parte delle tipologie di siti che abbiamo indicato in precedenza, per cui la scelta di far partire il crawling da questi siti risulta oculata. Se gli indirizzi nella coda fossero scelti in modo casuale tra tutti quelli del Web avremmo complessivamente una probabilità su due di finire in IN o in SCC, quindi basterebbero pochi indirizzi iniziali scelti a caso per raggiungere, e quindi raccogliere, una parte significativa di tutte le pagine esistenti sul Web. Sfortunatamente questo processo è di difficile realizzazione perché non si conosce la lista di tutti gli indirizzi delle pagine esistenti né è chiaro come effettuarne il campionamento in modo uniforme.

5.3.3 Indicizzazione delle pagine e ricerca dei risultati

Le pagine raccolte dal crawler vengono esaminate da un imponente insieme di algoritmi che estraggono da esse alcune informazioni utilizzate successivamente per rispondere alle interrogazioni poste dagli utenti. Ciò comporta che intere sottoreti di computer siano dedicate a costruire grandi e sofisticate strutture di dati dette «indici».

Ogni «documento» cui si riferisce l'indice contiene una pagina *p* del Web corredata di altre informazioni raccolte nella rete e correlate a *p*. Tra queste informazioni vi sono in particolare i cosiddetti «testi àncora», cioè porzioni di testo estratte da altre pagine e che circondano i link che puntano a *p*, di cui costituiscono quindi una sorta di descrizione terza presumibilmente affidabile. I motori di ricerca danno molta importanza ai testi àncora perché consentono di ampliare i risultati di una interrogazione recuperando pagine che non contengono le parole richieste ma sono correlate a essa. Poniamo per esempio che una pagina *p* contenga le immagini di varie specie di insetti ma non contenga la parola 'insetti' nel suo corpo testuale (o addirittura non contenga alcun testo). È però possibile che un appassionato entomologo realizzi una propria pagina Web con un link a *p* del tipo belle immagini di insetti. Tale pezzo di testo è un'àncora per *p*: le parole «belle, immagini, insetti» in esso contenute vengono aggiunte a quelle reperite in *p* e sono considerate fortemente caratterizzanti per tale pagina. Sfortunatamente, come spesso accade nel Web, a un uso pregevole di queste informazioni se ne accompagna un uso malizioso. Nel 1999 effettuando l'interrogazione «more evil than Satan» (più diabolico di Satana) Google restituiva come primo risultato la pagina di Microsoft, probabilmente a seguito della creazione di molte pagine con link a quella di Microsoft e aventi testi àncora contenenti le parole «evil» e

«Satan». Questa situazione imbarazzante fu risolta da Google in poche settimane, ma un incidente simile accadde poi nel novembre del 2003 con l'interrogazione «miserable failure» (fallimento miserevole) e la restituzione nella prima posizione di Google della pagina del presidente George W. Bush. Questo tipo di attacchi ha preso il nome di «google bombing», ed è stato ripetuto in diversi luoghi e in diverse lingue sfruttando l'importanza che i motori di ricerca attribuiscono ai testi àncora.

Costruiti i documenti a partire dalle pagine raccolte dal crawler, il motore di ricerca li analizza per estrarre i «termini» in essi contenuti e le informazioni correlate. I termini di tutte le pagine vengono inseriti in un dizionario e la verifica se un termine dell'interrogazione appare in questo dizionario costituisce il primo passo del processo di ricerca. Si noti però che un termine non è solo una parola o, in genere, una sequenza di caratteri alfabetici, ma può essere qualunque componente di un'interrogazione come un numero (730 o 800-156156), un'abbreviazione («e-ticket»), un modello dei nostri oggetti preferiti (N95, B52, Z4), un pezzo di ricambio (BH0241140500), il codice di un corso universitario (AA006), ecc. Cioè in genere un termine è più che una sequenza di lettere e numeri e può includere anche caratteri di interpunzione. Il dizionario dei termini è dunque gigantesco e deve essere organizzato in modo che si possa rispondere velocemente alle interrogazioni future sul suo contenuto. Non è infatti immaginabile che a ogni ricerca di un utente corrisponda una scansione lineare del dizionario, ed è cruciale quindi disporre di algoritmi e strutture di dati adeguate a gestire efficientemente in tempo e spazio questa grande mole di sequenze di caratteri.[14]

Il dizionario costituisce solo una parte della messe di informazioni estratte dal motore di ricerca e legate ai vari termini che appaiono nei documenti indicizzati. Queste informazioni sono numerose e di tipologia diversa, e devono essere opportunamente organizzate per renderne rapida l'identificazione nella risposta a un'interrogazione. Questa fase prende il nome di indicizzazione e consiste nella costruzione di una struttura di dati, denominata «liste invertite», che consta di due parti: il dizionario dei termini e una serie di liste, una per termine, contenenti le

[14] Numerosi risultati sperimentali hanno dimostrato che il numero n di termini distinti di un testo T segue una legge matematica che ha la forma $n = k\,|T|^{\alpha}$, con k pari a qualche decina, $|T|$ numero di parole del testo, e α uguale circa a 1/2. La dimensione attuale del Web indicizzato dai motori di ricerca è di decine di miliardi di pagine, ciascuna con almeno qualche centinaio di termini da cui $n > 10 \times 10^6 = 10^7$. Quindi il dizionario può contenere decine di milioni di termini distinti, ciascuno di lunghezza arbitraria.

occorrenze di quel termine e altre informazioni che indicano la «rilevanza» di ogni sua occorrenza. L'aggettivo «invertite» si riferisce al fatto che nei documenti l'ordine delle occorrenze dei termini è quello della sequenza testuale, mentre in queste liste l'ordine è quello che hanno i termini stessi nel dizionario.

Impiegando questo metodo si memorizzano per ogni termine i documenti che lo contengono e le posizioni in questi ove il termine appare. Questa lista è detta «posting list» del termine, e viene memorizzata in un vettore a partire da una posizione indicata in una tabella del «lessico». I documenti sono posti in un'altra tabella e sono identificati da numeri interi (in gergo «docID») assegnati, per esempio, durante la visita del crawler. La memorizzazione delle posizioni di un termine nei documenti che lo contengono incide significativamente sullo spazio totale richiesto dall'indice, pertanto la scelta di mantenere questa informazione dipende dalla tipologia di interrogazioni che il motore di ricerca è in grado di soddisfare. Nel caso di interrogazioni sulla semplice esistenza di un termine, sarebbero sufficienti i soli docID dei documenti che lo contengono, mentre interrogazioni sull'esistenza di frasi (cioè di più termini consecutivi in un ordine dato) richiedono di conoscere le posizioni dei termini in quei documenti. Alcuni motori usano queste posizioni anche per stimare la rilevanza di un documento rispetto a un'interrogazione, sulla base della distanza tra i termini dell'interrogazione in quel documento.

Illustriamo ora su un esempio la struttura delle liste invertite semplificandole per quanto è possibile, e sottolineando che nessun motore di ricerca rivela completamente le tecniche impiegate per la loro rappresentazione «compressa» e per la ricerca su di esse.

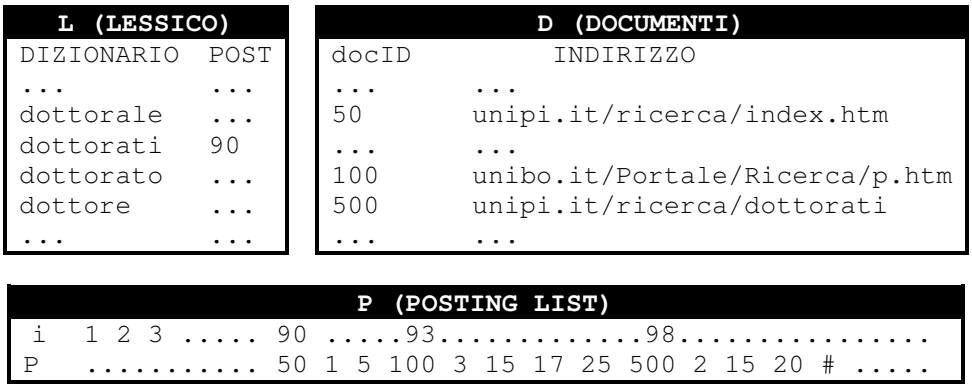

Figura 5.8: Indicizzazione a liste invertite. L'elenco dei documenti che contengono il termine «dottorati» inizia in posizione 90 della posting list P, indicata dal riferimento presente in POST per quel termine. Il termine «dottorati» è contenuto nei documenti 50, 100 e 500, il cui indirizzo è indicato nella tabella D dei documenti.

Vediamo dunque come è costruita la *posting list* P, riferendoci alla Figura 5.8. A ogni termine *t* («dottorati», nell'esempio) è associato un punto di inizio in P ove sono registrati in sequenza, per ogni documento *d* che contiene *t*, il docID del documento stesso (nell'esempio il primo documento è 50), il numero di volte in cui *t* compare in *d* (1 nell'esempio), le posizioni in *d* in cui compare *t* (5 nell'esempio). La *posting list* di *t* si conclude con un terminatore #. Dalla Figura 5.8 rileviamo dunque che il termine *t* = "dottorati" è contenuto nel documento 50 in posizione 5; nel documento 100 nelle tre posizioni 15, 17 e 25; nel documento 500 nelle due posizioni 15 e 20. La presenza di # indica che il termine non è presente altrove, e lì si conclude la sua posting list. Utilizzando questa struttura di dati si individuano quindi i documenti contenenti una parola chiave richiesta da un utente secondo l'algoritmo di ricerca dicotomica di Figura 2.23.

È opportuno costruire la lista P ponendo, per ogni termine, l'elenco dei documenti in ordine di docID crescente (50, 100, 500 nell'esempio), poiché ciò consente di ridurre lo spazio e il tempo necessari a risolvere le interrogazioni future. Infatti se i docID sono in ordine crescente, ciascuno di essi può essere memorizzato per «differenza» rispetto al docID che lo precede, e lo stesso metodo può essere usato per memorizzare le posizioni delle occorrenze del termine in ogni documento che lo contiene. Nella posting list di Figura 5.8 possiamo quindi rappresentare la sequenza di docID «50 100 500» come «50 50 400»: il primo 50 viene rappresentato esattamente non avendo un docID che lo precede, mentre per i

successivi si ha 100-50=50 e 500-100=400. Inserendo anche le occorrenze dei termini, la *posting list* 'ridotta' per il termine «dottorati» diviene:

50 1 5 50 3 15 2 8 400 2 15 5 #

come il lettore potrà facilmente constatare. La *posting list* originale può poi essere ricostruita da quella ridotta mediante una semplice serie di addizioni. Il vantaggio della memorizzazione ridotta consiste nel fatto che i numeri che vi appaiono sono più piccoli di quelli originali i quali, si ricordi, possono essere grandissimi: da ciò deriva una notevole riduzione di spazio di memoria impiegando un'opportuna codifica binaria degli interi.[15]

L'ordine dei docID è importante anche per eseguire con efficienza un'operazione molto richiesta: individuare i documenti che contengono più parole chiave. Immaginiamo che un utente abbia formulato una interrogazione con due termini t_1 e t_2 (l'estensione a un numero maggiore è immediata). Anzitutto si generano due liste L1 e L2 dei docID che contengono t_1 o t_2: per esempio L1 = 10 15 25 35 50 …# e L2 = 15 16 31 35 70 …# (ove abbiamo assunto di aver già risolto le 'differenze', e indicato solo i docID). Il problema è ora quello di individuare i documenti che contengono sia t_1 che t_2, cioè gli elementi comuni delle due liste. Si potranno allora scandire L1 e L2 da sinistra a destra fermandosi sugli elementi comuni. Detti n_1, n_2 i numeri di elementi delle due liste è facile rendersi conto che l'algoritmo richiede un tempo proporzionale a $n_1 + n_2$, poiché ogni confronto tra L1[i] e L2[j] fa avanzare almeno uno dei due indici i o j, e questi non possono esaminare più di $n_1 + n_2$ elementi. Questo costo è molto inferiore al tempo $n_1 \times n_2$ necessario per confrontare ogni elemento di L1 con tutti gli elementi di L2 se le liste non fossero ordinate: ciò causerebbe una grande differenza nei tempi di risposta alle interrogazioni poste dagli utenti poiché, a causa dell'estensione del Web, i valori di n_1 e n_2 sono di qualche centinaio di migliaia di docID.

5.3.4 Valutazione della rilevanza di una pagina

Una completa e corretta caratterizzazione della rilevanza delle pagine Web presenta un forte grado di arbitrarietà. Ciononostante, per soddisfare al meglio le

---

[15] Nella stessa direzione può anche intervenire una rinumerazione dei docID che assegni valori vicini a documenti «simili», e quindi induca differenze più piccole sulle posting list dei loro termini. Questo aspetto è troppo tecnico per essere trattato in questa sede. Ci basti qui osservare che tra i migliori ordinamenti dei docID oggi noti c'é quello naturalmente indotto dagli indirizzi delle pagine, perché pagine di uno stesso dominio contengono spesso un lessico simile.

richieste degli utenti sono stati proposti numerosi e sofisticati algoritmi che permettono di calcolare efficientemente un'approssimazione quantitativa della rilevanza di una pagina. Questa fase prende il nome di «ranking» e costituisce oggi il punto principale di distinzione tra i più importanti motori di ricerca, tanto che su questo si concentrano i maggiori segreti di realizzazione. Infatti, poiché le interrogazioni degli utenti consistono di poche parole chiave, il numero delle pagine che le contengono è solitamente enorme ed è fondamentale che il motore di ricerca restituisca nelle prime posizioni della risposta le pagine probabilmente più rilevanti per il richiedente. Non è azzardato affermare che uno degli ingredienti principali che hanno permesso a Google di raggiungere un'enorme popolarità è proprio l'algoritmo di «page ranking» impiegato, che ai tempi della sua comparsa forniva elenchi di pagine ordinati meglio di quelli degli altri motori di ricerca.[16]

Oggi la situazione è ben più complessa. La misura di rilevanza di una pagina p si ottiene mediante la combinazione di numerosi parametri che dipendono dalla tipologia e distribuzione delle occorrenze dei termini cercati in p, dalla «posizione» di p nel grafo del Web e dalla sua interconnessione alle altre pagine, dalla frequenza di visita di p da parte degli utenti, e da tanti altri fattori non tutti svelati: Google adotta almeno un centinaio di questi parametri! Presenteremo qui di seguito le due principali misure di rilevanza adottate oggi per le pagine Web, premettendo alcune considerazioni che permettano di comprenderne i motivi ispiratori.

È naturale pensare che la rilevanza di un termine t per un documento d dipenda dalla frequenza («Term Frequency») TF[t,d] con cui t occorre in d, e quindi dal «peso» che l'autore di quel documento ha voluto attribuire a quel termine ripetendolo più volte nel testo. D'altra parte considerare la sola frequenza è fuorviante perché per esempio gli articoli e le preposizioni occorrono numerose nei testi senza caratterizzarli in alcun modo. Quindi è necessario introdurre un fattore correttivo che tenga conto della 'capacità di discriminazione' di un termine e sia praticamente nullo nel caso di elementi linguistici secondari. La situazione è però più complicata in quanto un termine apparentemente significativo come per esempio «insetti» può risultare discriminante o meno se la raccolta di documenti indicizzata dal motore è limitata a testi di Informatica (ove l'apparizione del termine «insetti» è inusuale e probabilmente rilevante) o di entomologia (ove tale apparizione è ovvia e quindi irrilevante). È quindi cruciale considerare la rarità di

[16] Al punto che, a tutt'oggi, la pagina di Google presenta il pulsante «Mi sento fortunato» che rimanda l'utente immediatamente al primo risultato senza visualizzare tutti gli altri.

un termine nella raccolta, misurandola come rapporto tra il numero ND di documenti totali presenti in essa e il numero N[*t*] di documenti contenenti il termine *t*. Tanto più il termine *t* è raro quanto più il rapporto ND/N[*t*] è grande, e quindi *t* risulta potenzialmente discriminante per i documenti in cui appare. Solitamente il rapporto non viene utilizzato direttamente per stimare la «capacità di discriminazione» di *t*, ma viene mitigato applicando una funzione logaritmica. Si definisce così il parametro IDF[*t*] = $\log_2$ (ND/N[*t*]), dove IDF indica «Inverse Document Frequency», che risulta poco sensibile a piccole variazioni nel valore di N[*t*]. D'altra parte non è detto che un termine raro sia rilevante perché potrebbe corrispondere a una parole digitata in modo scorretto o desueta. Pertanto le misure di frequenza diretta e inversa sono combinate per formare il cosiddetto «peso testuale» TF-IDF, già proposto alla fine degli anni Sessanta e dato dalla formula:

$$W[t,d] = TF[t,d] \times IDF[t]. \qquad (2)$$

Si noti che se *t* è per esempio un articolo esso appare probabilmente in quasi tutti i documenti della raccolta rendendo il rapporto ND/N[*t*] vicino a uno e quindi il suo logaritmo vicino a zero, ovvero IDF[*t*] e W[*t,d*] praticamente nulli. Allo stesso modo un termine digitato scorrettamente avrà un valore piccolo per TF, e quindi piccolo sarà il valore di W[*t,d*]. Numerosi studi linguistici hanno corroborato la validità empirica della formula (2) che è ora alla base di un qualunque sistema di «information retrieval».

I motori di ricerca di «prima generazione», come Altavista, adottavano il peso TF-IDF come parametro preminente per valutare l'importanza di una pagina Web e ordinare in conseguenza i risultati di un'interrogazione. Questo approccio risultò efficace finché l'accesso al Web era riservato ad agenzie governative e università, e quindi a pagine informative controllate nei contenuti. A partire dalla metà degli anni Novanta il Web si è aperto a tutta la comunità mondiale diventando un enorme 'bazar commerciale', in cui apparire nei risultati di un motore di ricerca voleva dire essere nella 'vetrina del mondo'. Tutto ciò indusse alcune società a costruire pagine Web «truccate» che oltre alle proprie offerte commerciali contenevano, opportunamente occultate, parole chiave tipiche delle interrogazioni più frequenti di tutti gli altri utenti, allo scopo di promuovere arbitrariamente la rilevanza di quelle pagine anche in altri contesti.

Risultò dunque evidente che il peso testuale non poteva essere utilizzato da solo per valutare l'importanza di una pagina, ma occorreva tener conto di altri elementi propri del grafo del Web. A partire dalla metà degli anni Novanta numerose

proposte si susseguirono in ambito accademico e industriale per sfruttare i collegamenti presenti tra le pagine, interpretandoli come un «voto di rilevanza» espresso dall'autore di una pagina p verso quelle puntate dai link in uscita da p. Due tecniche di ranking dettero origine ai cosiddetti motori di ricerca di «seconda generazione»: la prima, denominata PageRank, fu introdotta da Larry Page e Sergey Brin fondatori di Google; la seconda, denominata HITS (Hyperlink Induced Topic Search), fu introdotta da Jon Kleinberg nei laboratori IBM. Nel PageRank si assegna a ciascuna pagina una rilevanza indipendente dal suo contenuto testuale e dall'interrogazione posta dall'utente. In HITS la rilevanza è invece assegnata in funzione di un sottografo del Web definito a partire dall'interrogazione posta dall'utente. Sebbene molto diverse, le tecniche di PageRank e HITS sono entrambe definite ricorsivamente perché la rilevanza di una pagina dipende dalla rilevanza delle pagine che puntano a (o sono puntate da) essa, e comportano calcoli su matrici di grandi dimensioni derivate dalla struttura del Web. Illustriamo ora le due tecniche nello spazio che consente questa trattazione. Il lettore meno versato nell'approfondimento numerico può passare direttamente a leggere § 5.3.6.

5.3.5 Le tecniche di PageRank e HITS

Il PageRank ordina le pagine in funzione della loro 'popolarità' nel grafo del Web misurata in base al numero e alla provenienza degli archi entranti in ogni pagina, cioè dei link che puntano a essa. In termini matematici la popolarità di una pagina p è espressa da un «rango» $R(p)$, calcolato come la probabilità[17] che un utente raggiunga p camminando a caso sulla rete, ove prenderà a ogni passo, con uguale probabilità, uno dei link che incontra nella pagina correntemente visitata. Dette p_1, …, p_k le pagine che hanno almeno un link verso p, e detto $N(p_i)$ il numero di pagine puntate da ogni p_i (cioè il numero di archi uscenti da p_i nel grafo del Web), la formula di base per il calcolo di $R(p)$ è la seguente:

$$R(p) = \sum_{i=1..k} (R(p_i) / N(p_i)). \qquad (3)$$

Cioè solo le pagine che puntano a p contribuiscono al valore di $R(p)$, e il contributo di ciascuna è dato dal proprio rango diviso per il numero di archi uscenti da essa.

La formula (3) è ragionevole in quanto se una pagina p_i con alto rango punta a p ne incrementa proporzionalmente la popolarità, ma tale incremento è diviso per $N(p_i)$ in modo che la popolarità di p_i si divida equamente su tutte le pagine da lei puntate.

---

[17] Chi non avesse familiarità con il concetto di probabilità può trovare una breve introduzione la concetto medesimo nel Riquadro *Eventi, probabilità e valori attesi* del Capitolo 9.

Il calcolo ricorsivo della formula presenta alcuni problemi tecnici perché richiede di specificare il valore iniziale di R(p), per ogni pagina p, e di indicare come debbano essere gestite le pagine che non hanno archi entranti e/o uscenti (si ricordi la discussione sul grafo di Web di Figura 5.7). Per ovviare a ciò, si preferisce considerare una formula leggermente diversa dalla precedente introducendo un fattore correttivo d che tiene conto della possibilità che un utente abbandoni la catena di link per saltare a un'altra pagina scelta a caso nella rete. La formula dunque diviene:

$$R(p) = d \sum_{i=1..k} (R(p_i) / N(p_i)) + (1-d) / n \qquad (4)$$

ove n è il numero di pagine raccolte dal crawler e indicizzate dal motore di ricerca, e d è la probabilità di proseguire nella catena di link. Si pone in genere $d = 0.85$ perché questo valore è sperimentalmente significativo. Al limite se si ponesse $d = 0$ tutte le pagine avrebbero pari rilevanza $R(p)=1/n$; se si ponesse $d = 1$, la rilevanza $R(p)$ dipenderebbe completamente dalla struttura del grafo del Web. La scelta $d = 0.85$ attribuisce, ragionevolmente ma senza eccedere, rilevanza maggiore al rango che deriva dalla struttura del Web.

Il calcolo del rango di tutte le pagine con la formula (4) è eseguito con l'algebra delle matrici. Abbiamo già visto che il grafo del Web è rappresentato da una matrice W le cui successive potenze W^k rappresentano i percorsi di k passi su quel grafo (cfr. Riquadro *Matrici di adiacenza e percorsi in un grafo*). Introduciamo allora una matrice Z di dimensione $n \times n$ i cui elementi hanno valore $Z[i,j] = d \times W[i,j] + (1-d)/n$. Per quanto detto in relazione alla formula (4) il valore $Z[i,j]$ rappresenta la probabilità che un utente percorra il link da p_i a p_j, e le successive potenze Z^k della matrice indicano le probabilità che si eseguano percorsi di k passi. Notiamo inoltre che i ranghi delle pagine possono essere rappresentati in un vettore R di n elementi ($R[i]$ è il rango della pagina p_i) e indichiamo con R_k il contenuto del vettore dopo aver eseguito la k-esima iterazione dell'algoritmo di calcolo (R_0 contiene i valori di rango assegnati alle pagine all'inizio del calcolo). Possiamo allora calcolare le successive configurazioni di R come:

$$R_1 = R_0 \times Z,\ R_2 = R_1 \times Z = R_0 \times Z^2,\ \ldots,\ R_k = R_0 \times Z^k. \qquad (5)$$

Tale formula rappresenta un valore teorico di probabilità che deriva da un modello di calcolo detto «catena di Markov»: il PageRank di Google è dunque una diretta applicazione di una precedente teoria matematica. Questa teoria è troppo difficile per essere esposta qui: diciamo solo, per comprendere come possa essere eseguito

il calcolo, che in questo modello il valore limite R_k, per $k \to \infty$, non dipende dal valore iniziale R_0 che può quindi essere assegnato arbitrariamente.

Ovviamente il calcolo indicato in (5) non è banale per le dimensioni delle matrici in gioco. Si tratta di matrici gigantesche, forse tra le più grandi generate dall'uomo e con cui gli studiosi si siano trovati a operare nella pratica. Basti osservare che il Web indicizzato da Google consiste oggi di almeno cinquanta miliardi di pagine, per cui Z consta di almeno 25×10^{20} elementi! Il calcolo è però semplificato dal fatto che a noi non interessa il valore finale di R_k ma solo l'ordinamento tra i valori assunti dalle sue componenti: se $R(p_1) > R(p_2)$ allora p_1 è più importante di p_2. Pertanto quando il valore delle componenti di R_k risulta sufficientemente stabile il calcolo suddetto viene interrotto. Prove sperimentali hanno dimostrato che un centinaio di iterazioni sono solitamente sufficienti.

Concludiamo la discussione sul PageRank ricordando che esso induce un ordinamento «assoluto» delle pagine che è funzione solo della struttura del grafo e non dipende quindi in alcun modo dalla interrogazione posta dall'utente (si dice che la rilevanza delle pagine è di tipo «query independent»). Pertanto il PageRank può essere calcolato durante la fase di indicizzazione delle pagine e memorizzato con esse, per poi essere recuperato a tempo di interrogazione per ordinare le pagine che contengono i termini della ricerca. I dettagli d'uso del PageRank sono sconosciuti, ma numerosi aneddoti suggeriscono che Google combini questo metodo con il peso TF-IDF e con un centinaio di altri parametri minori estratti automaticamente o manualmente dal Web.[18]

Studiamo ora la struttura algoritmica di HITS, potenzialmente più interessante del PageRank in quanto «query dependent». Per una data interrogazione *q* si recuperano dal Web l'insieme P di tutte le pagine che contengono tutti i termini di *q* e a queste si aggiungono le pagine che puntano a P o sono puntate da pagine di P. L'insieme risultante prende il nome di «base» e contiene dunque pagine che sono correlate con *q* direttamente (in quanto contenenti le parole dell'interrogazione) o indirettamente (in quanto connesse a P). La situazione è indicata nella Figura 5.9a.

---

[18] In una recente intervista, Udi Manber (VP Engineering di Google) ha svelato alcuni di questi parametri che dipendono dal linguaggio (capacità di gestire sinonimi, diacritici, errori di battitura, ecc.), dal tempo (alcune interrogazioni hanno risposte da ricercarsi nei trenta minuti precedenti, altre hanno bisogno di risultati più maturi), da modelli personalizzati (estratti dalla «storia» delle interrogazioni poste dall'utente o dalla sua navigazione).

Costruito il sottografo avente come nodi le pagine della base e come archi i link tra esse, si calcolano per questi nodi due misure di rilevanza dette «authority» (autorevolezza) e «hubness» (buona rassegna). La prima, indicata con A(p), misura l'autorevolezza della pagina p relativamente all'interrogazione q; la seconda, indicata con H(p), misura quanto la pagina p sia una buona rassegna per q. Il metodo di calcolo segue il senso comune secondo il quale una pagina p è tanto più una buona rassegna (e quindi il valore H(p) è grande) quanto più autorevoli sono le pagine citate da p; di contro, una pagina p è da considerarsi una buona autorità per l'interrogazione q (e quindi il valore A(p) è grande) quanto più sono prestigiose le rassegne che citano p. Indicando con z_1, …, z_k le pagine che puntano a p, e con y_1, …, y_h le pagine puntate da p (Figura 5.9b), possiamo formalizzare le intuizioni precedenti con le seguenti due formule:

$$A(p) = \sum_{i=1..k} H(z_i) \quad \text{e} \quad H(p) = \sum_{i=1..h} A(y_i). \tag{6}$$

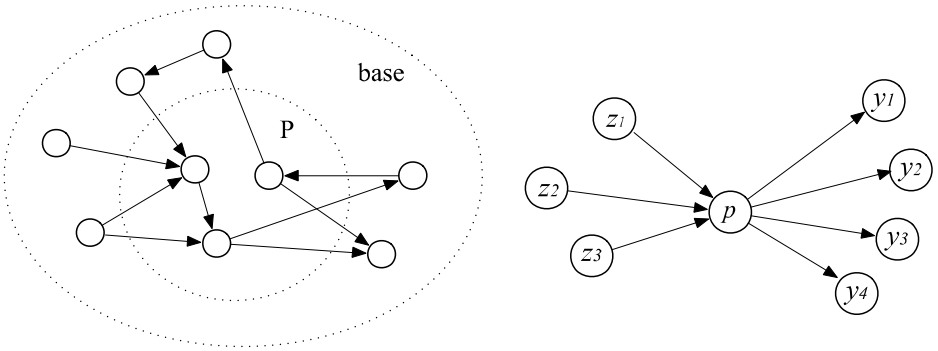

Figura 5.9. a) L'insieme P di pagine che contengono i termini di un'interrogazione q, e la sua «base»; b) Le pagine z_i e y_i che contribuiscono a determinare rispettivamente la «authority» e la «hubness» di una pagina p.

Similmente a quanto fatto per il PageRank, si definisce la matrice B di adiacenza del sottografo indotto dalla base e si calcolano i vettori A e H con le formule (6), impiegando l'algebra delle matrici a partire da B e da opportuni valori iniziali di A e H. Il calcolo è simile a quello indicato per il PageRank con due differenze sostanziali. La prima riguarda la dimensione delle matrici in gioco che consistono ora solo di qualche migliaio di nodi (numero di pagine della base). La seconda è che la base dipende dall'interrogazione e quindi i valori di A e H non possono essere calcolati in precedenza quando si costruisce l'indice, ma le formule (6)

devono essere risolte 'al volo' appena l'utente pone l'interrogazione. Questo limita fortemente l'applicazione del metodo all'ambito Web, e infatti HITS era stato originariamente proposto per motori di ricerca che operavano su raccolte ridotte di documenti e per un pubblico ristretto di utenti (per esempio su reti intranet aziendali).

Se da un lato i valori di A e H rendono HITS molto adatto a catturare diverse sfaccettature del concetto di rilevanza, il metodo è assai più esposto del PageRank a tecniche di «spam» (cfr. § 5.3.6), anche se oggi si conoscono numerose varianti di HITS «più robuste» della versione originale.

5.3.6 Le altre funzioni dei motori di ricerca

Tra le altre operazioni che un motore di ricerca è chiamato a eseguire ha grande importanza il modo con cui si presentano i risultati all'utente. Abbiamo già visto che il dizionario D_{pagine}, che contiene le pagine raccolte dai crawler, è utilizzato per mostrare agli utenti il risultato della ricerca. In effetti i motori restituiscono brevi frammenti testuali, noti come «snippet», che riportano il contesto di occorrenza dei termini dell'interrogazione in ogni pagina dei risultati, e offrono la possibilità di visualizzare la copia originale della pagina recuperata dal crawler che nel frattempo potrebbe essere stata modificata o potrebbe addirittura essere scomparsa dal Web. Il dizionario D_{pagine} è fondamentale per queste due funzionalità che permettono all'utente di valutare la significatività dei risultati o di trovarne alcuni non più disponibili sul Web ma comunque interessanti per la sua interrogazione.

Dobbiamo però sottolineare che le risposte dei motori di ricerca sono a volte corrotte ad arte con sofisticate tecniche di «spam» che promuovono fraudolentemente la rilevanza di alcune pagine per farle apparire tra le prime nella risposta, o alterano la risposta stessa in modi più subdoli. Così per esempio una tecnica di attacco chiamata «cloaking» impiega copie di pagine presumibilmente rilevanti per gli utenti (per esempio tratte da Wikipedia) per mascherarne altre con contenuti del tutto diversi. Se la pagina artefatta è rilevante per l'interrogazione di un utente il motore di ricerca visualizzerà uno snippet appropriato e interessante per essa, ma l'utente, cliccando sul link contenuto nello snippet, scaricherà una pagina irrilevante se non offensiva nei contenuti.

Una discussione sui metodi di spam non può entrare in questo breve capitolo. È però bene sapere che questi fenomeni hanno un'estensione inaspettatamente grande poiché si stima che più del 20% del Web sia costituito da pagine artefatte che mettono a repentaglio la reputazione e l'utilità dei motori di ricerca. In tutte le loro

fasi di funzionamento questi adottano quindi algoritmi anti-spam molto sofisticati per prevenire il recupero, l'indicizzazione ed eventualmente la restituzione di pagine artefatte. È ovvio che questi algoritmi vengono solo parzialmente svelati!

Osserviamo infine che l'obiettivo dei motori di ricerca si sta spostando verso l'individuazione dell'«intento» che si cela dietro l'interrogazione posta dall'utente, oltre che sulla sua composizione puramente «sintattica». Ciò spiega il moltiplicarsi di metodi diversi per presentare adeguatamente le risposte sullo schermo (iniziati con l'esperienza del motore Vivisimo.com), per integrare diverse sorgenti di informazione (news, Wikipedia, immagini, video, blogs, prodotti, ecc.), o per fornire suggerimenti alla composizione delle interrogazioni (Google Suggest, o Yahoo! Search Suggest). A ciò si aggiunga che gli utenti, da «attori attivi» del processo di ricerca, stanno purtroppo diventando sempre più «spettatori passivi»: pubblicità, suggerimenti, meteo, amici connessi, news particolari, ecc., sono tutte «informazioni» che abbiamo probabilmente indicato come 'interessanti' in qualche scheda personale, o che i motori hanno stabilito che sono per noi 'potenzialmente interessanti': ciò alla luce dei nostri comportamenti di ricerca sul Web che i motori sottopongono a un continuo scrutinio. Tutte queste informazioni appaiono già o appariranno in un futuro oramai prossimo sullo schermo quando si legge l'email, sulle pagine personali di iGoogle o myYahoo!, sulla pagina correntemente visitata sul Web, o anche sul nostro navigatore satellitare, senza che noi se ne abbia fatta esplicita richiesta.

Molte altre caratteristiche dei motori di ricerca meriterebbero di essere studiate attentamente sia dal punto di vista algoritmico che da quello dell'organizzazione generale e dell'impiego. Non potendo farlo qui per ragioni di spazio rimandiamo alla rassegna bibliografica finale.

5.4 VERSO UN WEB SEMANTICO

Nel corso della loro breve vita i motori di ricerca hanno fatto passi da gigante e tuttavia sono nella loro infanzia. Nonostante l'estrema complicazione degli algoritmi che permettono di impiegarli con una certa facilità, essi sono ancora limitati a indicare documenti rilevanti che contengono alcune parole chiave indicate dall'utente. Ma ben altro si potrebbe desiderare! Per esempio ponendo su Google le due parole chiave: 'pioggia' e 'roma' si potrebbero ottenere nelle prime posizioni della risposta una decina di riferimenti a pagine Web relative da una partita di calcio della Roma rinviata per la pioggia: probabilmente non è questo che interessa a tutti. Si noti che non è (solo) una questione di «page ranking»;

vorremmo poter chiedere: «Piove a Roma in questo momento o pioverà nelle prossime ore?», desiderando una risposta del tipo: «Ora è nuvolo ed è prevista pioggia nel pomeriggio; approfondimenti alle pagine…». Ma allo stato attuale tutto questo è impossibile nonostante l'informazione sia completamente disponibile su diversi siti Web. Vediamo perché.

All'indirizzo: http://www.tempoitalia.it/previsioni/meteo/italia/roma.html si incontra una tabella molto ben fatta che contiene le previsioni meteorologiche su Roma per tutta la settimana. Ogni riga corrisponde a un giorno e le caselle indicano ordinatamente: le condizioni del tempo (il consueto ideogramma – per esempio una nuvola – e la descrizione a parole – nuvoloso), la probabilità e l'intensità delle precipitazioni, venti, temperature, ecc. Ma dal sito di tempoitalia.it non potremmo mai ottenere una risposta, non dico alla richiesta di cui sopra, ma nemmeno a quella ovvia: «Posso vedere la tabella delle previsioni meteo per Roma?», perché il motore di ricerca non è capace di interpretare la domanda e il sito 'non sa' di possedere questa tabella. O meglio il sito non sa proprio niente, è solo un insieme di file progettati per mostrare su uno schermo una tabella con determinate scritte: alcune fisse come «tempo», «precipitazioni», «direzione del vento» che fungono da intestazioni, altre inserite continuamente dall'esterno come «poco nuvoloso», «10 mm», «S/SE» che indicano i valori del periodo. Il sito cioè non è progettato per 'sapere' che questa è una tabella di previsioni meteo, ma solo per mostrarla a richiesta.

L'importanza di muoversi verso la costruzione di un motore di ricerca capace di 'interpretare' le richieste degli utenti al di là del semplice esame delle parole chiave è stata da tempo indicata con grande determinazione dallo stesso Berners-Lee, l'inventore del Web. In tal senso si parla di «Web semantico» come prossima forma di impiego della rete. Ingredienti fondamentali saranno l'uso di tecniche algoritmiche per l'elaborazione di testi in linguaggio naturale e per l'apprendimento automatico: campi in cui già esistono importanti esperienze sviluppate in ambiti diversi, che iniziano a essere dirette verso la ricerca sulla rete.

Per quanto riguarda l'organizzazione dei dati per il nuovo Web qualche passo significativo è stato fatto ma è difficile per ora valutarne la portata. Il linguaggio RDF (Resource Description Framework), di cui esiste già una versione standard, consente di aggiungere ai dati contenuti nelle pagine un'informazione che ne descrive il 'significato' in termini molto liberi, che per ora si concentrano nell'indicare l'appartenenza di un dato a una classe predefinita e le sue relazioni con altri dati. Non si pretende che un computer 'comprenda' una richiesta ma

quanto meno possa classificarla e metterla in relazione con dati contenuti in altre pagine che non contengono le parole chiave della richiesta originale.

A fianco del RDF si sta definendo una «ontologia» per il Web (WebOnt), cioè un insieme di definizioni e di regole che caratterizzano le categorie semantiche dei dati e le relazioni logiche tra essi. Una simile operazione è fondamentale anzitutto per risolvere alcuni problemi apparentemente banali ma tuttora aperti come il trattamento di sinonimi e di voci derivate dallo stesso termine. Nell'esempio precedente sul tempo di Roma la richiesta contiene le parole «piove» e «pioverà»: una pagine Web che contenga solo la parola «pioggia» non verrebbe mostrata come risposta. Ma vi sono molti dubbi che la via da seguire sia quella intrapresa con WebOnt perché l'insieme delle sue regole è imposto dall'alto e non può abbracciare l'enorme messe di concetti presenti sul Web, né riflettere compiutamente il libero modo di ragionare degli utenti (e il nome stesso dell'operazione perpetua la barbarie linguistica degli informatici che usano la parola ontologia per indicare un insieme di proprietà anziché la scienza che le studia).

Alcuni sforzi recenti stanno comunque offrendo metodi automatici per aggiungere «semantica» alle pagine Web. Una prima sorgente di queste informazioni è il cosiddetto «Web 2.0» comprendente un processo di «tagging» che porta milioni di utenti ad associare termini o frasi alle proprie immagini, video, pagine, e ogni possibile file che occorre sul Web. Questi termini stanno creando un vero e proprio «linguaggio» parallelo detto «folksonomy», ossia una tassonomia pubblica di concetti emergenti in una comunità. Tra queste ricordiamo: Flickr, Technorati, YouTube, Del.icio.us, Panoramio, CiteULike, Last.fm, ESP Game, ecc. Nati con l'obiettivo di «classificare» i propri oggetti per facilitarne il loro recupero personale, questi sistemi di *tagging* assurgono a un ruolo sempre più importante nei motori di ricerca, dimostrando la loro efficacia nel migliorare le ricerche, individuare lo spam, creare nuove modalità di comunicazione e analisi dei dati, identificare nuovi soggetti che caratterizzano un settore.

Un'altra sorgente di informazione efficace nel migliorare il processo di ricerca e analisi del Web è rappresentata dall'insieme stesso delle ricerche eseguite dagli utenti. Queste informazioni sono catturate dai motori di ricerca, memorizzate in cosiddetti «Query Log», e poi successivamente elaborate al fine di estrarre relazioni di «vicinanza semantica» tra interrogazioni e/o pagine del Web. Immaginiamo che le stesse due interrogazioni q_1 e q_2 siano state formulate da molti utenti, i quali hanno poi raggiunto una stessa pagina p che appariva nei risultati

restituiti dal motore di ricerca per quelle due interrogazioni. Ciò significa che *p* ha probabilmente a che fare con gli argomenti di q_1 e q_2 e potrebbe essere quindi etichettata con le loro parole chiave, arricchendo potenzialmente il vocabolario di termini estratti da *p* in fase di indicizzazione o rendendo quei termini più rilevanti per *p* nelle ricerche future. Allo stesso modo, si può dedurre che q_1 e q_2 sono potenzialmente «correlate» e quindi potrebbero l'una costituire un valido suggerimento per l'altra. Ad esempio le due interrogazioni «iTunes» e «iPod» restituiscono su Google, nel momento in cui scriviamo, la pagina http://www.apple.com/itunes/ come primo risultato. Quindi ci aspettiamo che numerosi utenti raggiungeranno questa pagina determinando un «link semantico» tra i due termini iTunes e iPod. Lo stesso avviene per interrogazioni quali «nds» e «nintendo lite» che restituiscono nelle prime due posizioni il sito http://www.nintendo.com/ds/ come risultato della loro ricerca su Google.

D'altra parte è evidente che gli utenti possono proseguire la loro ricerca su diverse pagine a partire da una stessa interrogazione *q*. In questo caso può accadere che queste pagine siano tutte correlate con *q,* e quindi dovrebbero essere considerate «simili» tra loro; oppure che *q* sia una interrogazione «polisemica» che ammette cioè più interpretazioni. In questo secondo caso le pagine raggiunte successivamente dagli utenti che hanno eseguito l'interrogazione *q* possono avere diversi significati e quindi essere suddivise in gruppi di pagine «simili», ciascun gruppo rilevante per una particolare «interpretazione» di *q*. Si pensi ad esempio all'interrogazione «eclipse»: questa potrebbe indicare il termine inglese per «eclissi» (lunare o solare), oppure un sistema di sviluppo software, o un modello di aereo, o di una macchina, o di un autoradio, ecc. Quindi il «Query Log» presenterà numerose pagine diverse semanticamente, ma tutte potenzialmente rilevanti per l'interrogazione polisemica suddetta. Avremo per esempio utenti che hanno richiesto pagine di Wikipedia per documentarsi sulle eclissi lunari o solari, o su pagine del progetto «eclipse.org», o su pagine che descrivono il jet Eclipse 500, o ancora su pagine della Mitsubishi. In ogni caso l'analisi di queste interrogazioni e di questi click, della struttura del Web, del contenuto di quelle pagine così come di altre informazioni, porterà all'identificazione di gruppi di pagine simili e alla caratterizzazione di *q* come interrogazione polisemica. Pertanto *q* dovrà essere «trattata» dal motore di ricerca con cautela, nel senso che i primi risultati visualizzati per essa dovranno tener conto della sua polisemia e quindi essere allo stesso tempo diversificati e completi, nel senso che dovranno rappresentare nel miglior modo possibile le diverse sfaccettature semantiche racchiuse nei termini che costituiscono *q*.

Dal punto di vista algoritmico tutte le relazioni estratte dai «Query Log» e dalle altre sorgenti di informazione su citate (Web, contenuto, «click» di navigazione,…), vengono modellate mediante un grafo di enormi dimensioni sia in termini di numero di nodi (pagine e interrogazioni) che in termini di archi che li connettono (relazioni «sintattiche» o «semantiche»). L'analisi strutturale di questo grafo ha consentito in questi ultimi anni di estrarre numerose informazioni utili sulla «folksonomy» delle ricerche sul Web, sulle comunità di utenti e sui loro tendenziali interessi, e sulle pagine rilevanti in quanto raggiunte frequentemente dagli utenti del Web.

Va da sé che gli approcci di assegnazione automatica di significato agli oggetti del Web sono proni a errori incontrollati. Ma la «conoscenza collettiva» generata da questo gigantesco processo che coinvolge milioni di utenti, sembra sufficiente a ridurre il manifestarsi di questi errori; ciò limita la necessità di un intervento correttivo e rende il processo di estrazione di conoscenza potenzialmente scalabile alla dimensione corrente del Web. Non è azzardato ipotizzare che queste linee finiranno per dettare le regole ontologiche del gioco (e qui la parola è corretta). Ciò potrà perpetuare lo sviluppo attuale del Web che è guidato dalla popolarità delle informazioni anziché dalla loro qualità: non è una bella cosa, ma per ora non si vedono all'orizzonte uomini saggi e disinteressati cui affidare il controllo della rete.

5.5 NOTE BIBLIOGRAFICHE

La ricerca su Internet è un argomento complesso. Oltre a due libri in italiano a carattere generale e ad alcuni testi in inglese più spiccatamente tecnici, citiamo alcuni articoli che hanno marcato la storia di questo campo di studio, o costituiscono utili rassegne, o contengono risultati particolarmente interessanti.

Il testo (Witten *et al*., 2007) contiene uno studio generale sulle caratteristiche dei motori di ricerca e sulle implicazioni del loro impiego: è un testo chiarissimo scritto da tre autori con preparazioni diverse tra loro ed è un'ottima fonte di approfondimento senza entrare in dettagli algoritmici. (Luccio, Pagli, 2007) contiene una presentazione molto elementare dei concetti matematici su cui si basa il funzionamento della rete e sulla struttura dei grafi di Internet e del Web. Molto più specifici e approfonditi sono i testi (Manning *et al*., 2008) e (Baeza-Yates, Ribeiro-Neto, 2010) sulle basi generali dell'«Information Retrieval», (Chakrabarti, 2002) sul reperimento e l'analisi di dati nel Web, e (Witten *et al*., 1999) che può

essere considerato il riferimento fondamentale sull'organizzazione di grandissimi insiemi di dati.

Gli articoli (Hawking, 2006) e (Lopez-Ortiz, 2005) contengono due ottime e comprensibili rassegne sull'organizzazione generale dei motori di ricerca con particolare riferimento agli aspetti algoritmici. (Zobel, Moffat, 2006) è un articolo di riferimento sulle tecniche di indicizzazione, in particolare sull'impiego delle liste invertite. (Fetterly, 2007) contiene una recente rassegna sullo «spamming» e sulle tecniche fraudolente per ingannare i crawler. (Baeza-Yates *et al*., 2008) e (Microyannidis, 2007) discutono le direzioni verso cui evolve il Web semantico e le ricerche su di esso. (Broder *et al*., 2000) e (Brin, Page, 1998) sono due famosi articoli storici: il primo ha riportato lo studio originale sulla struttura del grafo del Web, il secondo ha proposto l'algoritmo di «page ranking» che ha segnato, dal punto di vista scientifico, l'inizio dell'era di Google.

Segnaliamo infine, come curiosità, il blog (Alpert, Hajaj, 2008) in cui, nel luglio 2008, gli ingegneri di Google hanno affermato di aver accumulato un trilione di indirizzi Web.