Sorgenti autorevoli in ambienti hyperlinkati.

La qualità di un metodo di ricerca richiede la valutazione umana dovuta alla soggettività inerente alla nozione di rilevanza. I motori di ricerca correnti, tipicamente, indicizzano una porzione dimensionabile del WWW e rispondono nell'ordine dei secondi, benché sarebbe più utile prevedere un'algoritmo che ci impieghi più tempo per fornire una risposta più significativa e migliore per l'utente, anche se è molto difficile dire *cosa* un tool di ricerca può riuscire a fare nel tempo extra.

Query e risorse autorevoli.
La ricerca è vista come l'inizio di un'immissione di una query da parte dell'utente. È meglio non dare una vista unica di nozione di query, poiché ci sono più tipi di query e diverse gestioni, per esempio:

1) *query specifiche*
2) *query con ampio argomento (broad-topic queries*)
3) *query per pagine simili (similar page queries)*

Ci concentreremo sui primi 2 tipi di query.
La difficoltà nel gestire le query specifiche è concentrata attorno a ciò che viene chiamato "scarcity problem": ci sono poche pagine che contengono davvero l'info richiesta. Spesso è difficile determinare l'identità di queste pagine. D'altra parte la difficoltà nel broad-topic queries è che ci si aspetta migliaia di pagine rilevanti sul WWW.
Quindi la difficoltà fondamentale è ciò che si chiama "abundance problem".
Il numero di pagine restituite come rilevanti è troppo ampio per essere consultato.
Per rendere effettiva una ricerca di questo tipo si può filtrare da un grande insieme di pagine rilevanti, un piccolo insieme delle più autorevoli o definitive.
Il problema ora è: "come facciamo a decidere cosa è autorevole?".

Analisi della struttura dei link.
Gli hyperlink codificano un considerevole ammontare di giudizi umani nascosti, e noi pretendiamo che questo tipo di giudizio sia precisamente necessario per formulare una nozione di authority.
La creazione di un link sul WWW rappresenta una concreta indicazione sul seguente tipo di giudizio: il creatore della pagina *p* che include una pagina *p* ha conferito in qualche misura authority su *q*. I link ci permettono l'opportunità di trovare autorità potenziali semplicemente attraverso la pagina che punta ad esso.
Ci sono un numero di potenziali "trappole" nell'uso dei link x questo scopo. Prima di tutto i link sono creati per una vastità di motivi, la maggior parte dei quali non hanno niente a che fare con l'autorità. Ad esempio molti sono del tipo "click here to return home" e altri sono dei banner.
Un altro grande problema sta nella difficoltà di bilanciare i criteri di *rilevanza* e *popolarità*, ognuno dei quali introduce ad un'intuitiva nozione di autorità.
Un *hub page* è una pagina considerata autorevole in base alle relazioni che esistono tra le autorità (authorities) per un argomento e quelle pagine che linkano a più autorità tra loro legate. Osserviamo che c'è un certo equilibrio tra gli hub e le autorità nel grafo definito dalla struttura a link e usiamo questo grafo per creare un algoritmo che identifica entrambi i tipi di pagine simultaneamente.
L'algoritmo lavora su *sottografi focalizzati* del WWW che sono costruiti come risultato di una ricerca basata su testo. La tecnica per la costruzione di ogni sottografo è definita x produrre una piccola collezione di pagine simili che contengano il maggior numero di pagine autorevoli x un argomento dato.

Overview.
L'algoritmo vuole identificare le pagine autorità basandosi sul contesto dell'argomento e sull'intero WWW. Tra i problemi coinvolti c'è il filtering dei risultati, poiché una ricerca broad topic può avere milioni di risultati.

Ovviamente questo tipo di ricerca è diverso da una ricerca locale in un sito aziendale o in una intranet. È altresì importante notare come il problema sia diverso dal *clustering*, in cui si vuole dividere una popolazione in tanti insiemi più piccoli.

Costruzione di un focused subgraph del WWW

Possiamo vedere ogni collezione V di pagine iperlinkate come un grafo diretto G=(V,E). I nodi corrispondono alle pagine e gli archi diretti (*p*,*q*) in E indicano i link dalla pagina *p* a *q*.
Il *grado uscente* è il numero del link uscenti, il *grado entrante* è il numero dei link entranti.
Si possono ottenere dei sottografi nel modo seguente: se $W \subseteq V$ è un sottoinsieme di pagine, noi useremo G[W] per denotare il grafo indotto da W (i nodi sono le pagine di W e gli edge sono i link tra le pagine di W).
Supponiamo di utilizzare una query string σ. Vogliamo determinare le pagine autorevoli con l'analisi della struttura dei link. Per prima cosa dobbiamo individuare in quale sottografo del WWW dobbiamo cercare. Il nostro obbiettivo è di focalizzare lo sforzo computazionale sulle pagine rilevanti in modo da poter restringere le analisi ad un insieme Q_σ.
Ma questo ha due controindicazioni:

1) questo insieme può contenere più di 1000000 di pagine e quindi introdurre alti costi di calcolo;
2) è possibile che le migliori authority non appartengano a questo insieme.

Idealmente vorremmo una collezione S_σ con le seguenti proprietà:

1) S_σ è relativamente piccola;
2) S_σ è ricca di pagine rilevanti;
3) S_σ è deve contenere il maggior numero o molte delle authority più "forti".

Se S_σ è piccola allora siamo in gradi di affrontare i costi computazionali non banali. Inoltre, assicurando che sia ricco di pagine importanti, in tal modo rendiamo più facile trovare buone authorities, che cioè sono referenziate in S_σ.

Creazione della collezione di pagine

Dato un parametro *t*, di solito posto a 200, per prima cosa collezioniamo *t* pagine con il miglior ranking ottenute da una ricerca basata su testo su un motore di ricerca. Identifichiamo questo insieme di pagine come *insieme radice* (*root-set*) R_σ.
Questo root-set soddisfa le prime due proprietà, ma generalmente è lontano dal soddisfare la terza proprietà. Questo si può capire perché in genere, le *t* pagine restituite tramite il motore, sono state cercate tramite la query string σ e allora R_σ è chiaramente un sottoinsieme di tutte le pagine che contengono σ (spesso le pagine in Q_σ non rispettano la proprietà 3).
È interessante notare che spesso ci sono pochissimi link tra le pagine in R_σ rendendolo *non strutturato*.
Si può usare il root-set R_σ per produrre un insieme S_σ che soddisferà le condizioni che vogliamo.
Consideriamo una "forte" autorità. Non è detto che essa sia in R_σ ma sicuramente è puntata da una pagina di R_σ. Così si può incrementare il numero di autorità forti nel sottografo espandendo R_σ lungo i link che entrano ed escono da essa.

Algoritmo di costruzione del sottografo

Subgraph(σ, &, t, d)
σ : query string
&: motore di ricerca text_base
t,d:numeri naturali.
R_σ = i primi t risultati di & per la stringa σ
Setta $S_\sigma = R_\sigma$
Per ogni pagina p $\in R_\sigma$
sia $\Gamma^+(p)$ l'insieme di tutte le pagine a cui punta p
sia $\Gamma^-(p)$ l'insieme di tutte le pagine che puntano a p
aggiungi tutte le pagine di $\Gamma^+(p)$ a S_σ.
Se $|\Gamma^-(p)| <= d$, allora
aggiungi tutte le pagine di $\Gamma^-(p)$ a S_σ.
altrimenti
scegli un insieme di pagine da $\Gamma^-(p)$ e inseriscili in S_σ
end if
enf for
return S_σ.

Ora descriveremo un algoritmo per computare hub e authority nel base-set S_σ.
Prima di poter descrivere l'algoritmo si devono considerare la euristiche che devono bilanciare gli effetti dei link che servono solo per navigare (back…).
Denotiamo con $G[S_\sigma]$ come il sottografo indotto dall'insieme S_σ, e in esso distinguiamo due tipi di link: un link è *trasversale* esso è tra pagine con domini diversi, *intrinseco* se è tra pagine dello stesso dominio. Visto che i link intrinsechi servono la maggior parte delle volte solo per la navigazione interna del sito comunicano meno informazioni dei link trasversali sulle authority che sono puntate dalla pagina.
Così è possibile eliminare tutti i link intrinsechi, lasciando solo gli archi corrispondenti ai link trasversali. Questo ci da G_s.
Questa euristica è semplice ma efficace.
Un'altra euristica può essere applicata. Supponiamo che un gran numero di pagine vengano da uno stesso dominio e tutte puntano ad una sola pagina *p*. Molto spesso questa pagina *p* corrisponde ad una pagina che non ci interessa (come banner).
Per eliminare questo problema può essere evitato scegliendo un parametro *m* tipicamente tra 4 e 8 e permettendo solo ad *m* pagine dal singolo dominio di puntare ad ogni pagina *p*. Anche questa è un'euristica efficace.

Calcolo di hub e authorities
Vogliamo estrarre le autorità dalla collezione di pagine G_s semplicemente dell'analisi dalla struttura di G_s.
L'approccio più semplice ordina le pagine in base al loro *grado entrante*. Rigettiamo l'idea se applicata all'inizio della collezione di tutte le pagine contenenti la string query σ.
Ma può essere applicata a quello che ora noi abbiamo costruito esplicitamente come una piccola collezione di pagine rilevanti contenenti la maggior parte della autorità che vogliamo trovare. Le autorità appartengono a G_s e sono linkate dalle pagine dentro G_s.
C'è da dire che questo approccio ha anche dei problemi.
Il problema è che non vengono inserite solo le autorità, ma anche le pagine molto popolari, e per aggirare questo problema c'è bisogno di un uso del contesto delle pagine del base-set, piuttosto che basarsi sulla sola struttura del grafo (Java punta anche alle pagine dei giochino Java).

Ora mostriamo che questo non è il caso del grafo G_σ in quanto è possibile rilevare più info dai link. L'osservazione è che le pagine interessanti per la ricerca iniziale non hanno un grande in-degree, visto che sono tutte autorità su un argomento comune. Poiché sono tutte con argomenti comune è considerabile che ci sia un overlap tra esse. Così in aggiunta alle pagine autorevoli più pesanti ci aspettiamo di trovare quelle che sono dette *hub pages*, che sono le pagine che hanno più link a più autorità rilevanti.
In questo modo considerando le hub possiamo individuare le autorità e scartare i nodi che hanno semplicemente un grande in-degree e così pagine hub e authority esibiscono quello che viene detto *mutual reinforcing relationship*: una buona pagina hub punta a buone authority e una buona authority è puntata da buone pagine hub!
Chiaramente se vogliamo distinguere hub e authority abbiamo bisogno di rompere questo circolo vizioso.

Un algoritmo iterativo
Facciamo uso tra relazioni tra hub e authority tramite un algoritmo iterativo, che mantiene e aggiorna un peso numerico per ogni pagina. Così con ogni pagina *p* associamo un *authority weight* non negativo $x^{<p>}$ e *hub weight* non negativo $y^{<p>}$.
L'algoritmo mantiene l'invariante che il peso di ogni tipo è normalizzato, nel senso che la loro somma quadratica è uguale a 1:

$$\Sigma_{p \in S_\sigma} (x^{\langle p \rangle})^2 = 1 \text{ e } \Sigma_{p \in S_\sigma} (y^{\langle p \rangle})^2 = 1$$

Consideriamo le pagine con i valori *x* e *y* più grandi come le migliori authority e i migliori hub.
È facile far vedere numericamente che vale il *mutual reinforcing relationship* tra hub e authority. Infatti se *p* punta a più pagine con grandi valori di $x^{<p>}$ allora esso dovrebbe ricevere un grande $y^{<p>}$ e se *p* è puntato da molte pagine con un grande valore di *y* allora dovrebbe ricevere un grande valore di *x*.
Dati i pesi $x^{<p>}$ e $y^{<p>}$ l'operazione $\mathcal{I}$ aggiorna i pesi $x^{<p>}$ come segue:

$$x^{\langle p \rangle} \leftarrow \sum_{q:(q,p) \in E} y^{\langle q \rangle}.$$

Dati i pesi $x^{<p>}$ e $y^{<p>}$ l'operazione $\mathcal{O}$ aggiorna i pesi $y^{<p>}$ come segue:

$$y^{\langle p \rangle} \leftarrow \sum_{q:(p,q) \in E} x^{\langle q \rangle}.$$

In tal modo $\mathcal{I}$ e $\mathcal{O}$ sono il motivo principale per cui authority e hub sono mutumente rinforzati.
Ora, per trovare l'equilibrio che vogliamo per i pesi, applichiamo le operazioni $\mathcal{I}$ e $\mathcal{O}$ in modo alternato.
Rappresentiamo l'insieme dei pesi $\{x^{<p>}\}$ come un vettore $\boldsymbol{x}$ a cui corrisponde ogni pagina in G_s. Allo stesso modo useremo il vettore $\boldsymbol{y}$.

Procedura:

```
Itera(G,k)
        G: una collezione di n pagine linkate
        k: numero naturale
        z: vettore di tutti (1,1,…,1) in R^n
        x_0=z;
        y_0=z;
        For i=1 to k do
                Applica l'operazione I ad (x_{i-1}, y_{i-1}), ottenendo nuovi pesi per x, detti x'_i.
                Applica l'operazione O ad (x'_i, y_{i-1}), ottenendo nuovi pesi per y, detti y'_i.
                x_i=normalizza(x'_i)
                y_i=normalizza(y'_i)
        end for
        return (x_k, y_k).
```

Questa procedura appena descritta può essere applicata ad un filtro che caccia le prime *c* autorità e i primi *c* hub al modo seguente:

Filtra(G,k,c)
G: una collezione di *n* pagine linkate
k,c: numero naturale
(x_k, y_k)=Itera(G,k)
Riferisci le *c* pagine con il valore più grande di coordinate in x_k come autorità
Riferisci le *c* pagine con il valore più grande di coordinate in y_k come hub
end.

Applicheremo la procedura di filtro con $G = G_\sigma$ e con *c* tra 5 e 10.
Per scegliere *k* nel modo migliore dovremmo prima mostrare che applicando Itera con un valore arbitrario grande di *k* le sequenze di vettori x_k, y_k convergono a dei punti fissi x^*, y^*.

Similar-Page Queries
L'algoritmo sviluppato in precedenza può essere applicato ad altro tipo di problemi. È possibile utilizzare la struttura dei link per dare una nozione di *somiglianza* tra le pagine.
Se troviamo una pagina *p* di interessa (ed è authority) ci possiamo porre la seguente domanda: cosa hanno considerato gli utenti quando hanno linkato *p* al momento della creazione della pagina e dell'hyperlink?
Se *p* è una pagina molto referenziata allora abbiamo una versione del problema dell'abbonzanza (quello di prima…).
La struttura dei link circostanti (link entranti e/o uscenti) rappresenta un numero enorme di opinioni indipendenti sulla relazione di *p* alle altre pagine.
Usando la nostra nozione di hub e authorities possiamo dare un approccio x il problema del *page similarità* chiedendoci: nella regione locale nella struttura vicino a *p*, quali sono le migliori authorities? Così le authorities possono potenzialmente fornire come un riassunto delle pagine legate a *p* legate ad un argomento ampio.
Infatti i metodi relativi alla costruzione del sottografo focalizzato e il calcolo degli hub e delle authority può essere adattato per il *similar page queries* senza modifiche sostanziali.

Prima noi iniziavamo cercando le *t* pagine contenenti la stringa σ; ora cerchiamo *t* pagine che puntano a *p*. Così possiamo costruire un root-set di *p* detto R_p (*t* pagine che puntano a *p*), poi lo aumentiamo in S_p da cui otteniamo il sottografo G_p, in cui cerchiamo hub ed authority.