Web-Suche

Link-Analyse

Bibliometrik: Zitat-Analyse

- Viele Dokumente enthalten *Bibliographien* (oder *Referenzen*), d.h. eindeutige *Zitierungen* anderer, vorher veröffentlichter Dokumente.
- Bei Verwendung von Zitaten als Links können solche Korpora als gerichteter Graph betrachtet werden.
- Die Struktur dieses Graphen kann unabhängig vom Inhalt interessante Informationen über die Ähnlichkeit von Dokumenten und die Struktur der Korpora liefern.

Einflussfaktor (Impact Factor)

- Von Garfield in 1972 entwickelt, um die Bedeutung (Qualität, Einfluss) von wissenschaftlichen Zeitschriften zu messen.
- Maß dafür, wie oft Artikel einer Zeitschrift von anderen Wissenschaftlern zitiert werden.

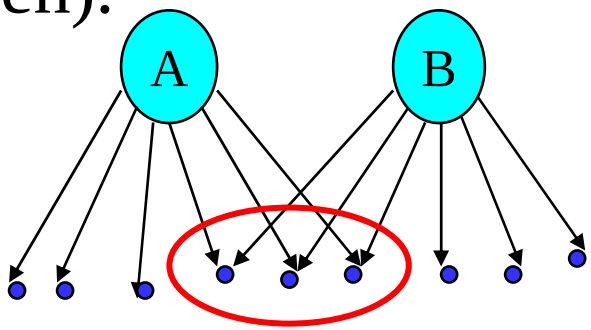
- Wird jährlich vom Thompson Scientific (http://www.isinet.com/) berechnet und veröffentlicht.

- Der *Einflussfaktor* einer Zeitschrift *J* im Jahr *Y* ist die durchschnittliche Anzahl von Zitaten (von allen indizierten Dokumenten, die im Jahr Y veröffentlicht wurden) eines Artikels, der in J im Jahr *Y*–1 oder *Y*–2 veröffentlicht wurde.
- Berücksichtigt nicht die Qualität des zitierenden Artikels.

- Siehe auch http://citeseer.ist.psu.edu/impact.html für einen ähnlichen Index.

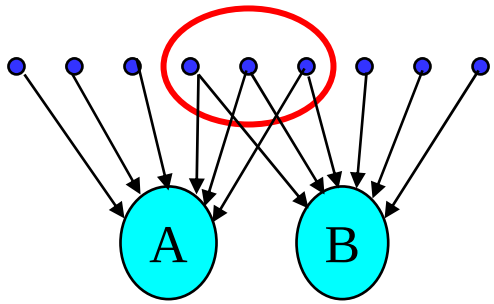
Bibliographische Kopplung

- Maß für die Ähnlichkeit von Dokumenten, das 1963 von Kessler eingeführt wurde.
- Die bibliographische Kopplung von zwei Dokumenten *A* und *B* ist die Anzahl der Dokumente, die *sowohl* von *A* als auch von *B* zitiert werden, d.h. der Umfang des Durchschnitts ihrer Bibliographien (ggf. normiert durch die Größe der Bibliographien).



Ko-Zitatation

- Ein alternatives auf Zitaten basierendes Maß der Ähnlichkeit, das 1973 von Small eingeführt wurde.
- Anzahl der Dokumente, die sowohl *A* als auch *B* zitieren, ggf. normalisiert durch die gesamte Anzahl von Dokumenten die entweder *A* oder *B* zitieren.



Zitate im Vergleich zu Links

- Weblinks sind anders als Zitate:
  - Links sind navigationsfähig.
  - Viele Seiten mit hohem In-Grad sind Portale und keine Inhaltsanbieter.
  - Nicht alle Links (aber auch nicht alle Zitate) sind Bestätigungen.
  - Firmenwebseiten verweisen nicht auf ihre Konkurrenten, Zitate relevanter Literatur werden hingegen durch Peer-Reviewing erzwungen.

Autoritäten

- *Autoritäten* sind Seiten, die anerkannt sind, und die signifikante, vertrauenswürdige und nützliche Information zu einem Thema zu liefern.
- *In-Grad* (Anzahl von Zeigern auf eine Seite) ist ein einfaches Maß der Autorität.
- Jedoch behandelt ein In-Grad alle Links gleich.
- Sollten nicht Links von Seiten, die selbst Autoritäten sind, mehr zählen?

Hubs

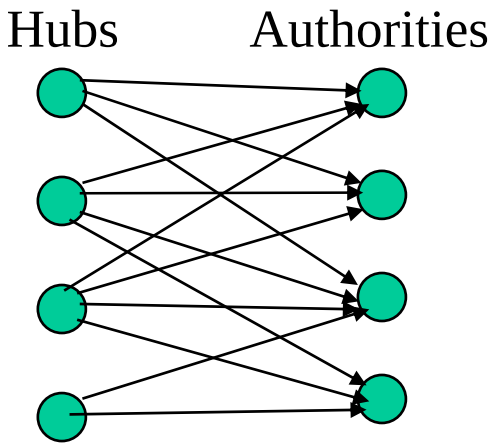
- *Hubs* sind Indexseiten, die viele nützliche Links auf relevante Inhaltsseiten (Themenautoritäten) liefern.
- Hubseiten zum Thema “Information Retrieval” sind z.B unter http://www.cs.utexas.edu/users/mooney/ir-course zu finden.

HITS

- Algorithmus, der 1998 von Kleinberg entwickelt wurde.
- Er versucht, Hubs und Autoritäten zu einem bestimmten Thema rechnerisch durch die Analyse eines relevanten Subgraphen des Webs zu bestimmen.
- HITS basiert auf einer rekursiven Definition:
  - Hubs verweisen auf viele Autoritäten.
  - Auf Autoritäten wird von vielen Hubs verwiesen.

Hubs und Autoritäten

- Zusammen neigen sie dazu, einen bipartiten Graphen zu bilden:

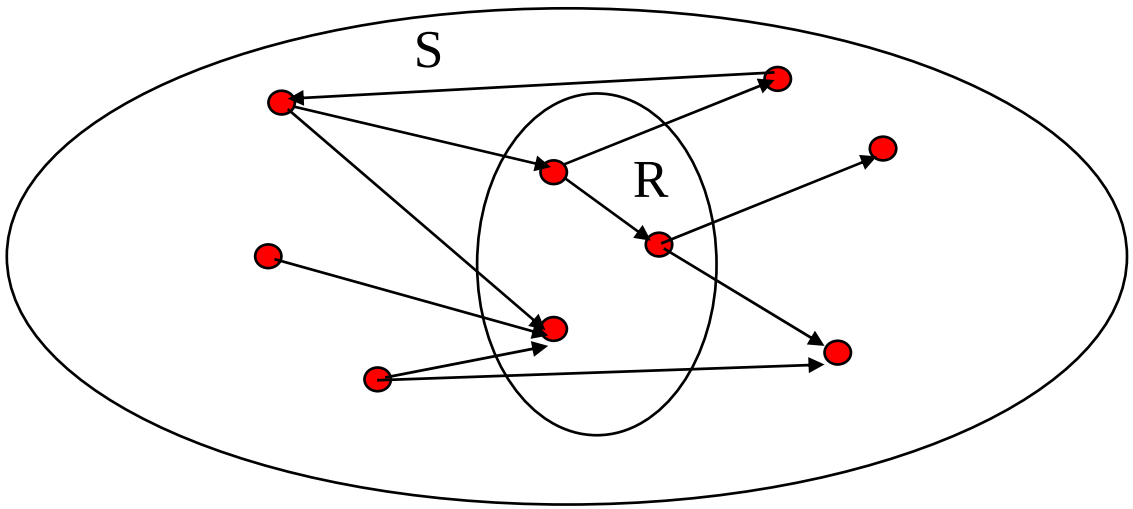


HITS Algorithmus

- Aufgabe: Berechnet Hubs und Autoritäten für ein bestimmtes Thema, das durch eine Anfrage spezifiziert ist.
- Bestimmt zuerst eine Menge relevanter Seiten für die Anfrage, die als *Basis*-Menge *S* bezeichnet wird.
- Analysiert die Linkstruktur des durch *S* induzierten Teilgraphen, um Autoritäts- und Hubseiten in dieser Menge zu finden.

Konstruieren eines Basis-Subgraphen

- Für eine spezifische Anfrage *Q* sei die *Wurzel*-Menge *R* die Menge der von einer Standard-Suchmaschine (z.B. KSM) zurückgegebenen Dokumente.
- *S* := *R*.
- Füge zu *S* alle Seiten hinzu, auf die mindestens eine Seite in *R* verweist.
- Füge zu *S* alle Seiten hinzu, die auf mindestens eine Seite in *R* verweisen.



Aufwandsbegrenzung

- Um den rechnerischen Aufwand zu limitieren:
  - Begrenze die Anzahl der Wurzelseiten auf die besten 200 Seiten, die für die Anfrage gefunden wurden.
  - Begrenze die Anzahl der “Rückwärts-Link”-Seiten auf eine willkürliche Menge von höchstens 50 Seiten, die von einer “Rückwärts-Link”-Anfrage zurückgegeben wurden.
- Um reine Navigationslinks zu eliminieren:
  - Eliminiere Links zwischen zwei Seiten auf dem gleichen Host.
- Um “nicht-autoritätsfördernde” Links zu eliminieren:
  - Erlaube max. m ($m \simeq 4$–8) Seiten von jedem Host als Zeiger auf ein beliebige individuelle Seite.

Autorität und In-Grad

- Selbst in der Basismenge *S* einer gegebenen Anfrage sind die Knoten mit dem höchsten In-Grad nicht notwendigerweise Autoritäten (sondern evtl. nur allgemein bekannte Seiten wie Yahoo oder Amazon).
- Auf ‘wahre’ Autoritätsseiten wird von mehreren Hubs verwiesen (dies sind Seiten, die auf viele Autoritäten verweisen.)

HITS – Iterativer Algorithmus

- Iterativer Algorithmus, der sich langsam einer sich gegenseitig verstärkenden Menge von Hubs und Autoritäten nähert.

- Aufgabe: Bestimme für jede Seite $p \in S$
  - den Autoritätswert a_p (zusammengefasst in einem Vektor $\boldsymbol{a}$)
  - und den Hubwert h_p (Vektor $\boldsymbol{h}$)

HITS-Algorithmus

1. Initialisiere alle $a_p := h_p := 1$
2. Normalisiere die Werte, so dass gilt:

$$\sum_{p \in S} \left(a_p\right)^2 = 1 \qquad \sum_{p \in S} \left(h_p\right)^2 = 1$$

3. Auf Autoritäten wird durch viele gute Hubs verwiesen:

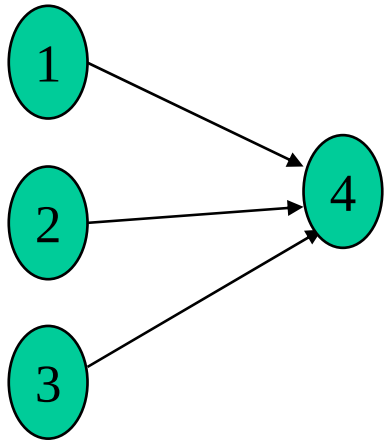
$$a_p = \sum_{q:q \to p} h_q$$

4. Hubs verweisen auf viele gute Autoritäten:

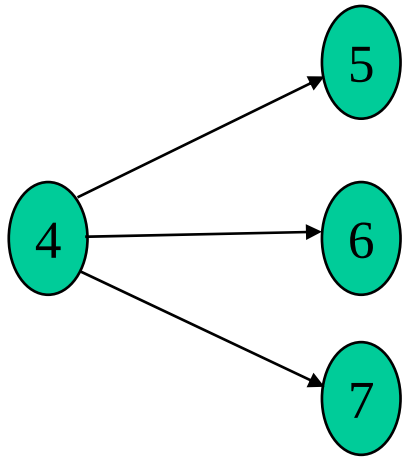
$$h_p = \sum_{q:p \to q} a_q$$

5. Solange die Vektoren sich (signifikant) ändern, gehe zu Schritt 2.

Illustrierte Update-Regeln



$a_4 := h_1 + h_2 + h_3$



$h_4 := a_5 + a_6 + a_7$

HITS im Detail

Initialisiere für alle $p \in S$: $a_p := h_p := 1$

Bis Änderung kleiner als gegebener Schwellwert:

Für alle $p \in S$:

$$a_p := \sum_{q:q \to p} h_q$$

/* aktualisiere Autoritätswerte */

Für alle $p \in S$:

$$h_p := \sum_{q:p \to q} a_q$$

/* aktualisiere Hubwerte */

Für alle $p \in S$: $a_{p:} = a_p / c$ mit $c := \sqrt{\sum_{p \in S} a_p^2}$

/* ***a*** normalisieren */

Für alle $p \in S$: $h_{p:} = h_p / c$ mit $c := \sqrt{\sum_{p \in S} h_p^2}$

/* ***h*** normalisieren */

Darstellung in linearer Algebra

- Definiere A als Adjazenzmatrix für den durch S induzierten Subgraphen.
  - $A_{ij} = 1$ für $i \in S$, $j \in S$ gdw. $i \rightarrow j$ im Graphen.
- Die Autoritätswerte a_p werden in einem Vektor $\boldsymbol{a}$ zusammengefasst, und die Hubwerte h_p in einem Vektor $\boldsymbol{h}$.
- Die Schritte der Iteration ergeben sich zu
  - $\boldsymbol{h} := A\boldsymbol{a}$
  - $\boldsymbol{a} := A^{\mathrm{T}}\boldsymbol{h}$

Konvergenz

- Algorithmus konvergiert zu einem *Fixpunkt*, falls unendlich wiederholt.
- Autoritätsvektor $\boldsymbol{a}$ konvergiert gegen den ersten Eigenvektor von A^TA.
- Hubvektor, $\boldsymbol{h}$, konvergiert gegen den ersten Eigenvektor von AA^T.
- In der Praxis liefern 20 Wiederholungen ziemlich stabile Ergebnisse.

Ergebnisse

- Autoritäten für Anfrage “Java”
  - java.sun.com
  - comp.lang.java FAQ
- Autoritäten für Anfrage “search engine”
  - Yahoo.com
  - Excite.com
  - Lycos.com
  - Altavista.com
- Autoritäten für Anfrage “Gates”
  - Microsoft.com
  - roadahead.com

(Nach [Kleinberg 1998])

Beobachtung

- In den meisten Fällen waren die endgültigen Autoritäten nicht in der anfänglichen Wurzelmenge, die mit Altavista bestimmt wurde.
- Autoritäten wurden durch Vor- und Rückwärtslinks hinzugefügt (und dann durch HITS als Autorität bestimmt).

Finden ähnlicher Seiten durch Verwendung der Linkstruktur

- Aufgabe: Bestimmung ähnlicher Seiten zu einer Seite *P*. (Dieser Ansatz findet Autoritäten in der "Link-Nachbarschaft" von *P*.)

- Sei *t* gegeben (z.B. *t* = 200).
- Sei *R* eine Menge von *t* Seiten, die auf *P* verweisen (die Wurzelmenge).
- Bestimme die Basismenge *S* von *R* wie o.a.
- Lasse HITS auf *S* laufen.
- Gebe die besten Autoritäten in *S* als die "ähnlichsten Seiten von *P*" zurück.

Ergebnisse der Ähnlichkeitssuche

- Gegeben “honda.com”
  - toyota.com
  - ford.com
  - bmwusa.com
  - saturncars.com
  - nissanmotors.com
  - audi.com
  - volvocars.com

PageRank

- Alternative Link-Analyse-Methode, die von Google verwendet wird (Brin & Page, 1998).
- Versucht nicht, die Unterscheidung zwischen Hubs und Autoriäten zu erfassen, sondern klassifiziert Seiten nur nach Autorität.
- Wird eher auf das gesamten Web angewandt als auf eine lokale Nachbarschaft von Seiten, die die Ergebnisse einer Anfrage umgeben.

Grund-Idee PageRank

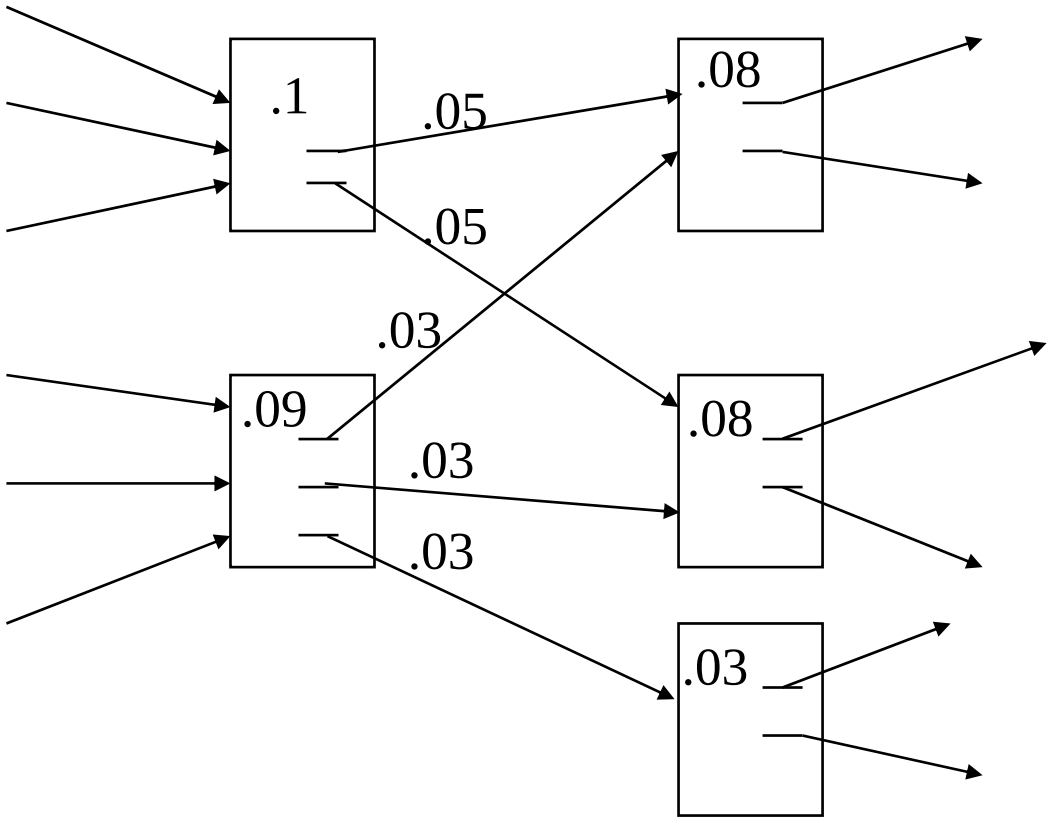
- Die Messung des In-Grades alleine (Zitatzählung) berücksichtigt nicht die Autorität der Quelle eines Links.
- (Vereinfachte) PageRank-Gleichung für Seite *p*:

$$R(p) = \sum_{q:q \to p} \frac{R(q)}{N_q}$$

  - N_q ist die Gesamtzahl der Out-Links von Seite *q*.
  - Eine Seite *q* gibt einen gleichen Anteil ihrer Autorität an alle Seiten weiter, auf die sie verweist (z.B. auf *p*).

Grund-Idee PageRank

- PageRank “fließt” entlang der Kanten:



Grundidee PageRank

- Wiederhole den Fluss-Prozess bis zur Konvergenz:

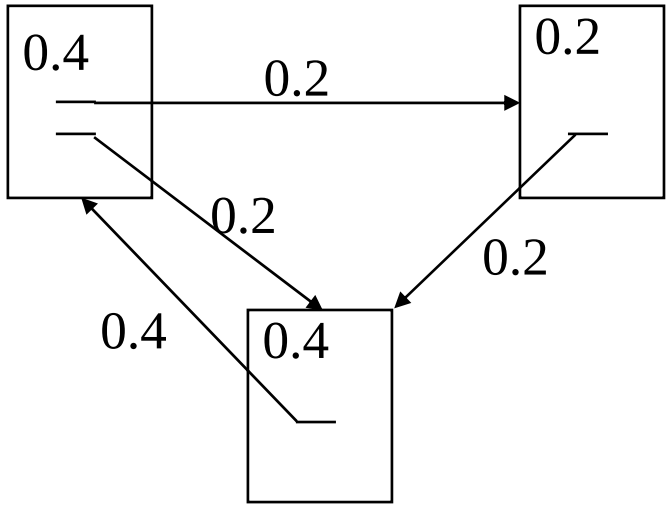
  Sei S die Gesamtmenge der Seiten.

  Initialisiere für alle $p \in S$: $R(p) = 1/|S|$

  Bis sich Werte nicht mehr (viel) ändern (*Konvergenz*)

  Für jedes $p \in S$: $R'(p) = \sum_{q:q \to p} \frac{R(q)}{N_q}$

Beispiel: stabiler Fixpunkt



Lineare-Algebra-Version

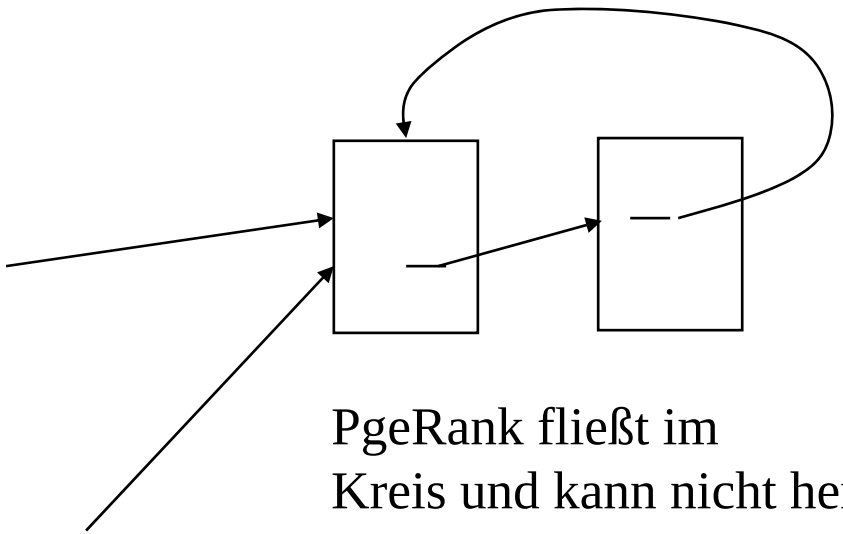
- Betrachte $\boldsymbol{r}\mathbf{:=}(R(p))_{p \in S}$ als einen Vektor in $\mathbf{R}^{|S|}$.
- Sei $\mathbf{A}$ die $|S|\times|S|$-Matrix mit

  $\mathbf{A}_{vu} := 1/N_u$ falls $u \rightarrow v$, und $\mathbf{A}_{vu} := 0$ sonst.

- Dann gilt am Ende des Algorithmus $\boldsymbol{r} = \mathbf{A}\boldsymbol{r}$, d.h. $\boldsymbol{r}$ konvergiert zu dem Eigenvektor von $\mathbf{A}$, der zum Eigenwert 1 gehört.

Problem mit anfänglicher Idee

- Eine Gruppe von Seiten, die nur auf sich selbst verweist, aber auf die durch andere Seiten verwiesen wird, agiert als eine Gewichts-Senke, die das ganze Gewicht absorbiert.
- “Suchmaschinenoptimierer” nutzen diesen Effekt in “Linkfarmen” aus.



PgeRank fließt im
Kreis und kann nicht heraus

Gewichts-Quelle

- Führe eine Gewichts-Quelle E ein, die kontinuierlich den Rank jeder Seite p durch einen festen Betrag $E(p)$ ergänzt.

$$R(p) = \alpha \sum_{q:q \to p} \frac{R(q)}{N_q} + (1-\alpha)E(p)$$

- Mit $\alpha \in [0,1]$ kann der Einfluss von E gesteuert werden, Brin & Page haben mit $\alpha = 0.85$ gute Ergebnisse erzielt.

PageRank-Algorithmus

Sei S die Gesamtmenge der Seiten.

Sei $\alpha \in (0,1)$, z.B. $\alpha = 0.85$.

Für alle $p \in S$: $E(p) := 1/|S|$

Für alle $p \in S$ initialisiere $R(p) := 1/|S|$.

Bis sich die Gewichte nicht mehr (viel) ändern (*Konvergenz*):

$$R(p) = \alpha \sum_{q:q \to p} \frac{R(q)}{N_q} + (1-\alpha)E(p)$$

Lineare-Algebra-Version

- Nach Konvergenz gilt $\boldsymbol{r} = \alpha\mathbf{A}\boldsymbol{r} + (1-\alpha)\mathbf{E}$.
- Wegen $\|\boldsymbol{r}\|_1 = 1$ gilt $\boldsymbol{r} = c(\alpha\mathbf{A} + (1-\alpha)\mathbf{E}\times\mathbf{1})\boldsymbol{r}$

  wobei **1** der Vektor ist, der nur aus 1ern besteht.
- Somit ist $\boldsymbol{r}$ ein Eigenvektor von $\alpha\mathbf{A} + (1-\alpha)\mathbf{E}\times\mathbf{1}$.

Random-Surfer-Modell

- PageRank kann als Modelierung eines “willkürlichen Surfers” betrachtet werden, der auf einer beliebigen Seite startet und dann entweder
  - mit der Wahrscheinlicheit $E(p)$ willkürlich auf die Seite p springt
  - oder willkürlich einem Link auf der aktuellen Seite folgt.

- $R(p)$ modelliert dann die Wahrscheinlichkeit, dass sich dieser willkürliche Surfer zu jeder gegebenen Zeit auf der Seite p befindet.

- Die “Sprünge” in **E** werden benötigt, um zu vermeiden, dass der willkürliche Surfer in Web-Senken “gefangen” wird, aus denen kein Link herausführt.

Konvergenz

- Frühe Experimente in Google verwendeten 322 Millionen Links.
- PageRank-Algorithmus konvergiert (mit einer kleinen Toleranz) in ca. 52 Wiederholungen.
- Die Anzahl der für Konvergenz erforderlichen Wiederholungen ist empirisch O(log *n*) (wobei *n* die Anzahl der Links ist).
- Daher ist die Berechnung ziemlich effizient.

Einfache Titelsuche mit PageRank

- Verwende zunächst die einfache Boolesche Suche, um Titel von Webseiten zu suchen und klassifiziere die gefundenen Seiten dann nach ihrem PageRank.
- Beispiel-Suche nach “university” (aus [Page, Brin 1998]):
  - Altavista gab eine beliebige Menge von Seiten mit “university” im Titel wieder (schien kurze URLs zu bevorzugen).
  - Primitives Google gab die Homepages der amerikanischen Top-Universitäten wieder.

Google-Suche

- Komplette Google-Suche umfasste vor der Kommerzialisierung (basierend auf wissenschaftlichen Veröffentlichungen):
  - Vektorraummodell
  - Abstandsmaß zu Schlüsselwörtern
  - HTML-Tag-Gewichtung (z.B. Titelpräferenz)
  - PageRank
- Details zu aktuellen Google-Komponenten sind Betriebsgeheimnisse.

Personalisierter PageRank

- PageRank kann durch Ändern von **E** beeinflusst (personalisiert) werden: Beschränken des "Random Surfers" auf eine Menge als relevant spezifizierter Seiten.
- Zum Beispiel durch Setzen von $E(p) := 0$, außer auf der eigenen Homepage, wo $E(p) := \alpha$
- Dies führt zu einer Ausrichtung auf Seiten, die im Webgraphen näher zu der eigenen Homepage sind.

PageRank-basiertes Spidering

- Verwende PageRank, um den Spider auf “wichtige” Seiten zu leiten (zu fokussieren).
- Berechne PageRank unter Verwendung der aktuellen Menge der bearbeiteten Seiten.
- Sortiere die Anfrage-Warteschlange des Spiders auf der Basis des aktuell geschätzten PageRanks.

Schlussfolgerungen zur Linkanalyse

- Die Linkanalyse verwendet als Suchhilfe Informationen über die Struktur des Webgraphen.
- Dies ist eine der wesentlichen Innovationen bei der Websuche
- ... und der primäre Grund für den Erfolg von Google.

Ausblick: Suche im Web 2.0

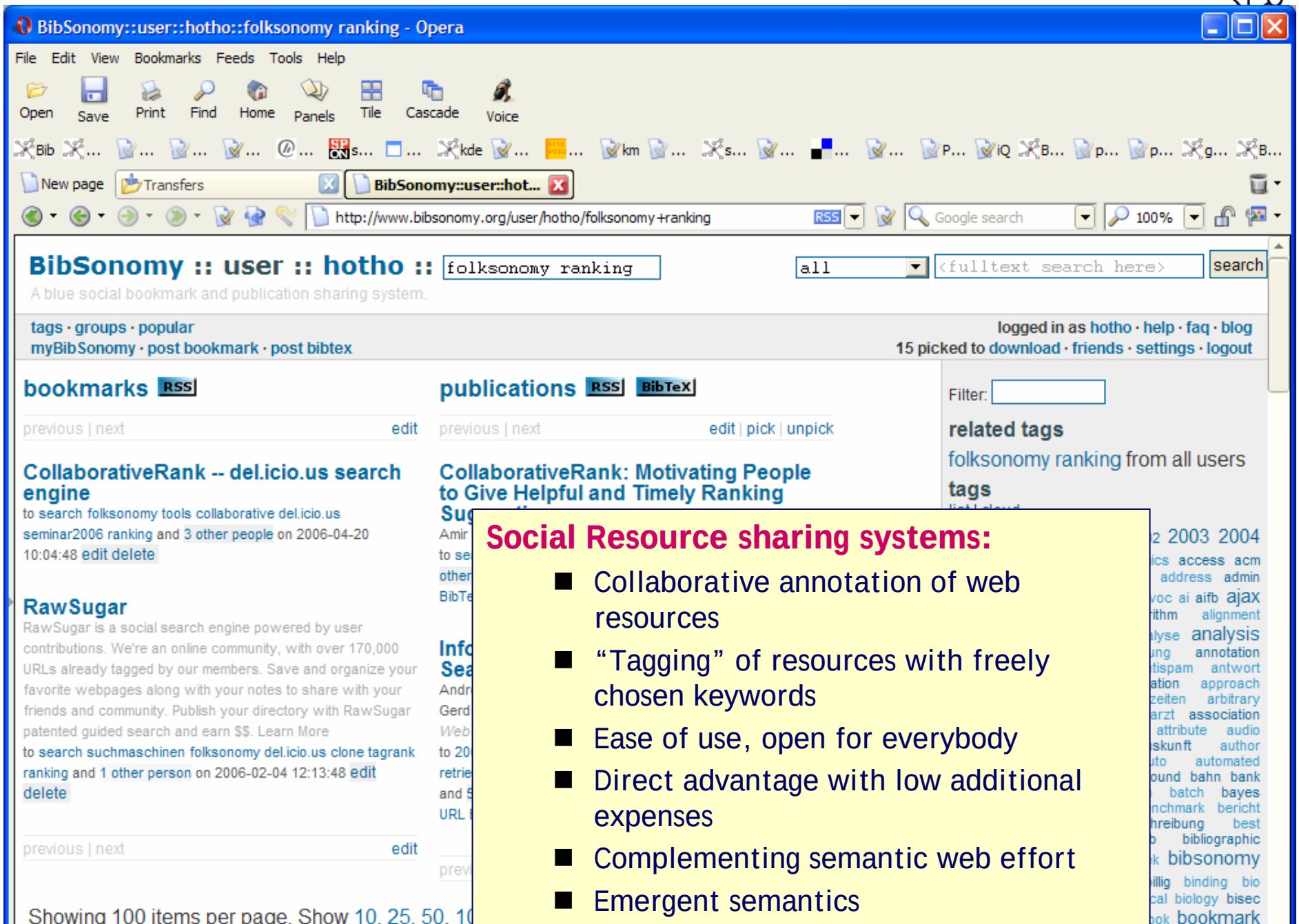
ENDOWED CHAIR OF THE HERTIE FOUNDATION
Knowledge and Data Engineering
DEPARTMENT OF MATHEMATICS & COMPUTER SCIENCE

Information Retrieval in Folksonomies: Search and Ranking

Andreas Hotho, Robert Jäschke,
Christoph Schmitz, Gerd Stumme

Published in York Sure and John Domingue, editor(s), The Semantic Web: Research and Applications, LNAI 4011, pages 411-426, Springer, Heidelberg, 2006.

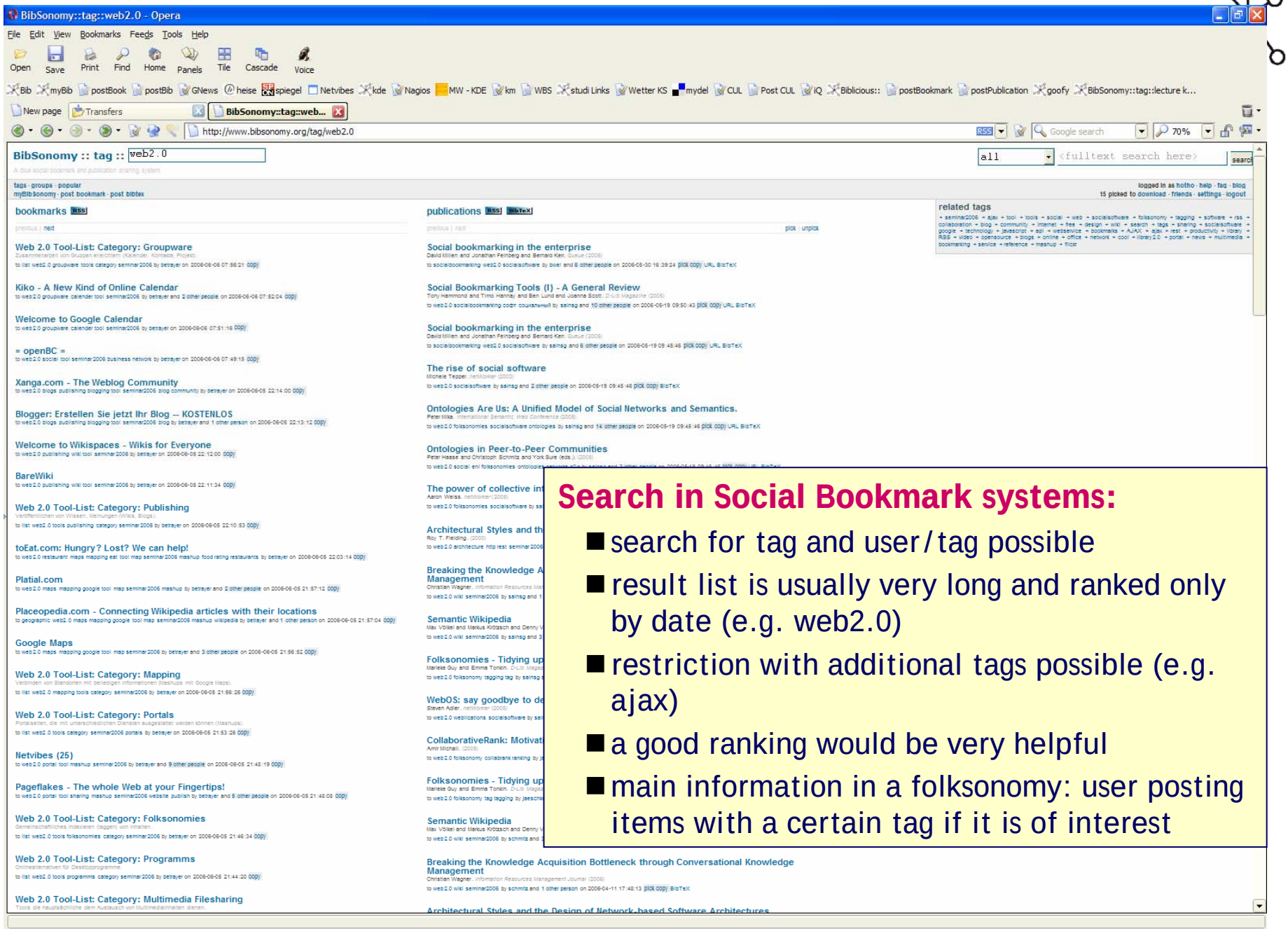
BibSonomy - a Folksonomy/Web 2.0 System



Social Resource sharing systems:

- Collaborative annotation of web resources
- “Tagging” of resources with freely chosen keywords
- Ease of use, open for everybody
- Direct advantage with low additional expenses
- Complementing semantic web effort
- Emergent semantics

BibSonomy - a Folksonomy/Web 2.0 System

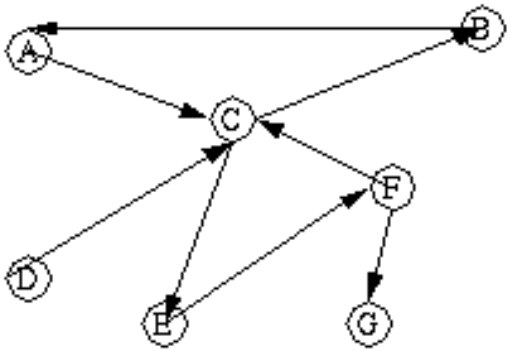


Search in Social Bookmark systems:

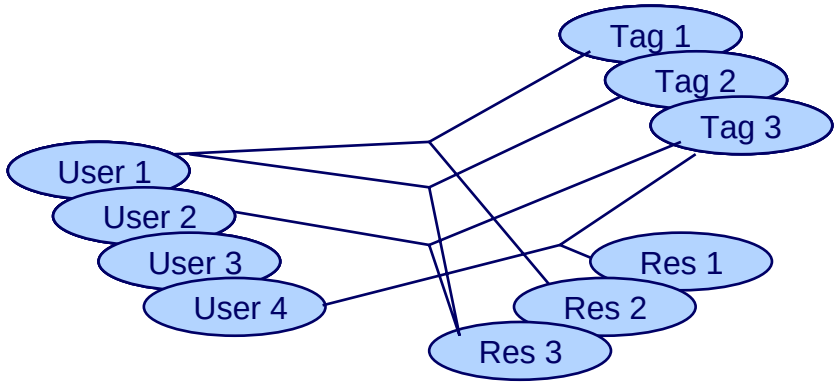
- search for tag and user/tag possible
- result list is usually very long and ranked only by date (e.g. web2.0)
- restriction with additional tags possible (e.g. ajax)
- a good ranking would be very helpful
- main information in a folksonomy: user posting items with a certain tag if it is of interest

Search in Folksonomies

- PageRank in the web: pages are important if a lot of important pages are linking to them
- authority values in a folksonomy are propagated along the hyperlink structure of the folksonomy



Web-Graph



Folksonomies

Formal Model

A *folksonomy* is a tuple $\mathbf{F} := (U, T, R, Y, \prec)$ where

U, T, and R are finite sets, whose elements are called *users*, *tags* and *resources*, resp.

Y is a ternary relation between them, i.e. $Y \subseteq U \times T \times R$, called tag assignments (TAS for short)

and $\prec$ is a user specific subtag/supertag relation, i.e. $\prec \subseteq U \times T \times T$, called subtag/supertag relation.

Converting a Folksonomy into an Undirected Graph

Set V of nodes consists of the disjoint union of the sets of tags, users and resources:

$$V = U \cup T \cup R$$

All co-occurrences of users and tags, tags and resources, users and resources become edges between the respective nodes:

$$\begin{aligned} E = \{\{u,t\} \mid \exists\, r \in R : (u,t,r) \in Y\} \;\cup \\ \{\{t,r\} \mid \exists\, u \in U : (u,t,r) \in Y\} \cup \\ \{\{u,r\} \mid \exists\, t \in T : (u,t,r) \in Y\} \end{aligned}$$

Original PageRank:

Computation of fixed point *r* of the weight spreading function

$$r := \alpha A r + (1-\alpha)e$$

- *A* is the row-normalized adjacency matrix reflecting the graph
- *e* : random surfer vector
- α : weighting factor, eg $\alpha = 0.85$

Adaptation to folksonomy:

- each undirected edge → two directed edges

Problem with the adapted PageRank version:

Graph is undirected → weight flows in one direction and directly "swashes back"

Idea to solve this is to apply a differential approach:

Let R_{AP} be the fixed point with $\alpha = 1$

Let R_{pref} be the fixed point with $\alpha < 1$

$R := R_{pref} - R_{AP}$ is the final weight vector

Additionally: different weights in random surfer vector allow for topic-specific ranking.

Evaluation on del.icio.us dataset

Crawl of del.icio.us from July 27 to 30, 2005 resulted in a folksonomy with

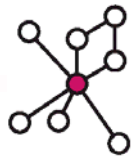
$|U|$ = 75,242 users,
$|T|$ = 533,191 tags and
$|R|$ = 3,158,297 resources, related by in total
$|Y|$ = 17,362,212 tag assignments (TAS).

Results: adapted PageRank

Tag	ad. PageRank
system:unfiled	0,0078404
web	0,0044031
blog	0,0042003
design	0,0041828
software	0,0038904
music	0,0037273
programming	0,0037100
css	0,0030766
reference	0,0026019
linux	0,0024779
tools	0,0024147
news	0,0023611
art	0,0023358
blogs	0,0021035
politics	0,0019371
java	0,0018757
javascript	0,0017610
mac	0,0017252
games	0,0015801
photography	0,0015469
fun	0,0015296

User	ad. PageRank
shankar	0,0007389
notmuch	0,0007379
fritz	0,0006796
ubi.quito.us	0,0006171
weev	0,0005044
kof2002	0,0004885
ukquake	0,0004844
gearhead	0,0004820
angusf	0,0004797
johncollins	0,0004668
mshook	0,0004556
frizzlebiscuit	0,0004543
rafaspol	0,0004535
xiombarg	0,0004520
tidesonar02	0,0004355
cyrusnews	0,0003829
bldurling	0,0003727
onpause_tv_anytime	0,0003600
cataracte	0,0003462
triple_entendre	0,0003419
kayodeok	0,0003407

Results: adapted PageRank



http://slashdot.org/	0,0002613
http://pchere.blogspot.com/2005/02/absolutely-delicious-complete-tool.html	0,0002320
http://script.aculo.us/	0,0001770
http://www.adaptivepath.com/publications/essays/archives/000385.php	0,0001654
http://johnvey.com/features/deliciousdirector/	0,0001593
http://en.wikipedia.org/wiki/Main_Page	0,0001407
http://www.flickr.com/	0,0001376
http://www.goodfonts.org/	0,0001349
http://www.43folders.com/	0,0001160
http://www.csszengarden.com/	0,0001149
http://wellstyled.com/tools/colorscheme2/index-en.html	0,0001108
http://pro.html.it/esempio/nifty/	0,0001070
http://www.alistapart.com/	0,0001059
http://postsecret.blogspot.com/	0,0001058
http://www.beelerspace.com/index.php?p=890	0,0001035
http://www.techsupportalert.com/best_46_free_utilities.htm	0,0001034
http://www.alvit.de/web-dev/	0,0001020
http://www.technorati.com/	0,0001015
http://www.lifehacker.com/	0,0001009
http://www.lucazappa.com/brilliantMaker/buttonImage.php	0,0000992
http://www.engadget.com/	0,0000984

Results: boomerang

Preference for tag: boomerang

PageRank without preference

Tag	ad. PageRank
system:unfiled	0,0078404
web	0,0044031
blog	0,0042003
design	0,0041828
software	0,0038904
music	0,0037273
programming	0,0037100
css	0,0030766
reference	0,0026019
linux	0,0024779
tools	0,0024147
news	0,0023611
art	0,0023358
blogs	0,0021035
politics	0,0019371
java	0,0018757
javascript	0,0017610
mac	0,0017252
games	0,0015801
photography	0,0015469
fun	0,0015296

PageRank with preference

Tag	ad. PRank
boomerang	0,4036883
shop	0,0069058
lang:de	0,0050943
software	0,0016797
java	0,0016389
programming	0,0016296
web	0,0016043
reference	0,0014713
system:unfiled	0,0014199
wood	0,0012378
kassel	0,0011969
linux	0,0011442
construction	0,0011023
plans	0,0010226
network	0,0009460
rdf	0,0008506
css	0,0008266
design	0,0008248
delicious	0,0008097
injuries	0,0008087
pitching	0,0007999

FolkRank with preference

Tag	FolkRank
boomerang	0,4036867
shop	0,0066477
lang:de	0,0050860
wood	0,0012236
kassel	0,0011964
construction	0,0010828
plans	0,0010085
injuries	0,0008078
pitching	0,0007982
rdf	0,0006619
semantic	0,0006533
material	0,0006279
trifly	0,0005691
network	0,0005568
webring	0,0005552
sna	0,0005073
socialnetworkanalysis	0,0004822
cinema	0,0004726
erie	0,0004525
riparian	0,0004467
erosion	0,0004425

Results: Semantic Web

Preference for ressource: http://www.semanticweb.org

PageRank without preference

Tag	ad. PageRank
system:unfiled	0,0078404
web	0,0044031
blog	0,0042003
design	0,0041828
software	0,0038904
music	0,0037273
programming	0,0037100
css	0,0030766
reference	0,0026019
linux	0,0024779
tools	0,0024147
news	0,0023611
art	0,0023358
blogs	0,0021035
politics	0,0019371
java	0,0018757
javascript	0,0017610
mac	0,0017252
games	0,0015801
photography	0,0015469
fun	0,0015296

PageRank with preference

Tag	ad. PRank
semanticweb	0,0208605
web	0,0162033
semantic	0,0122028
system:unfiled	0,0088625
semantic_web	0,0072150
rdf	0,0046348
semweb	0,0039897
resources	0,0037884
community	0,0037256
xml	0,0031494
research	0,0026720
programming	0,0025717
css	0,0025290
portal	0,0024118
.imported	0,0020495
imported-bo...	0,0019610
en	0,0018900
science	0,0018166
.idate2005-04-11	0,0017779
newfurl	0,0017578
internet	0,0016122

FolkRank with preference

Tag	FolkRank
semanticweb	0,0207820
semantic	0,0121305
web	0,0118002
semantic_web	0,0071933
rdf	0,0044461
semweb	0,0039308
resources	0,0034209
community	0,0033208
portal	0,0022745
xml	0,0022074
research	0,0020378
imported-bo...	0,0018920
en	0,0018536
.idate2005-04-11	0,0017555
newfurl	0,0017153
tosort	0,0014486
cs	0,0014002
academe	0,0013822
rfid	0,0013456
sem-web	0,0013316
w3c	0,0012994

Results: Semantic Web

http:// www. semanticweb.org /	0,3761957
http://flink. semanticweb.org /	0,0005566
http:// simile .mit.edu / piggy- bank/	0,0003828
http://www.w3.org/2001/ sw/	0,0003216
http:// infomesh. net /2001/ swintro/	0,0002162
http://del. icio.us / register	0,0001745
http:// mspace.ecs.soton.ac.uk/	0,0001712
http:// www. adaptivepath .com / publications / essays/ archives /000385.php	0,0001637
http:// www. ontoweb.org /	0,0001617
http:// www.aaai.org / AITopics / html / ontol. html	0,0001613
http:// simile .mit.edu /	0,0001395
http:// itip. evcc.jp/ itipwiki/	0,0001256
http:// www. google.be /	0,0001224
http:// www. letterjames.de/ index. html	0,0001224
http:// www.daml.org /	0,0001216
http:// shirky.com / writings / ontology _ overrated . html	0,0001195
http:// jena. sourceforge. net/	0,0001167
http:// www. alistapart.com /	0,0001102
http:// www. federalconcierge.com / WritingBusinessCases. html	0,0001060
http:// pchere. blogspot.com /2005/02/ absolutely- delicious- complete- tool. html	0,0001059
http:// www. shirky.com / writings / semantic_ syllogism. html	0,0001052

Conclusion

Folksonomies might overcome the knowledge aquisition bottleneck through ease of use and growing amount of users.

Our ranking is just based on the structure of the folksonomy – the content of the resources is not used.

Suitable for intranets, where

- resources are typically not hyperlinked,
- community building is important.