max planck institut
informatik

M2CI
CLUSTER OF EXCELLENCE

Fokussierte Informationssuche in sozialen Netzen

Ralf Schenkel

Gemeinsame Arbeit mit Tom Crecelius, Mouna Kacimi, Sebastian Michel, Thomas Neumann, Josiane Parreira, Gerhard Weikum

Social Tagging Networks

Definition: Social Tagging Network

Webseite, wo Menschen

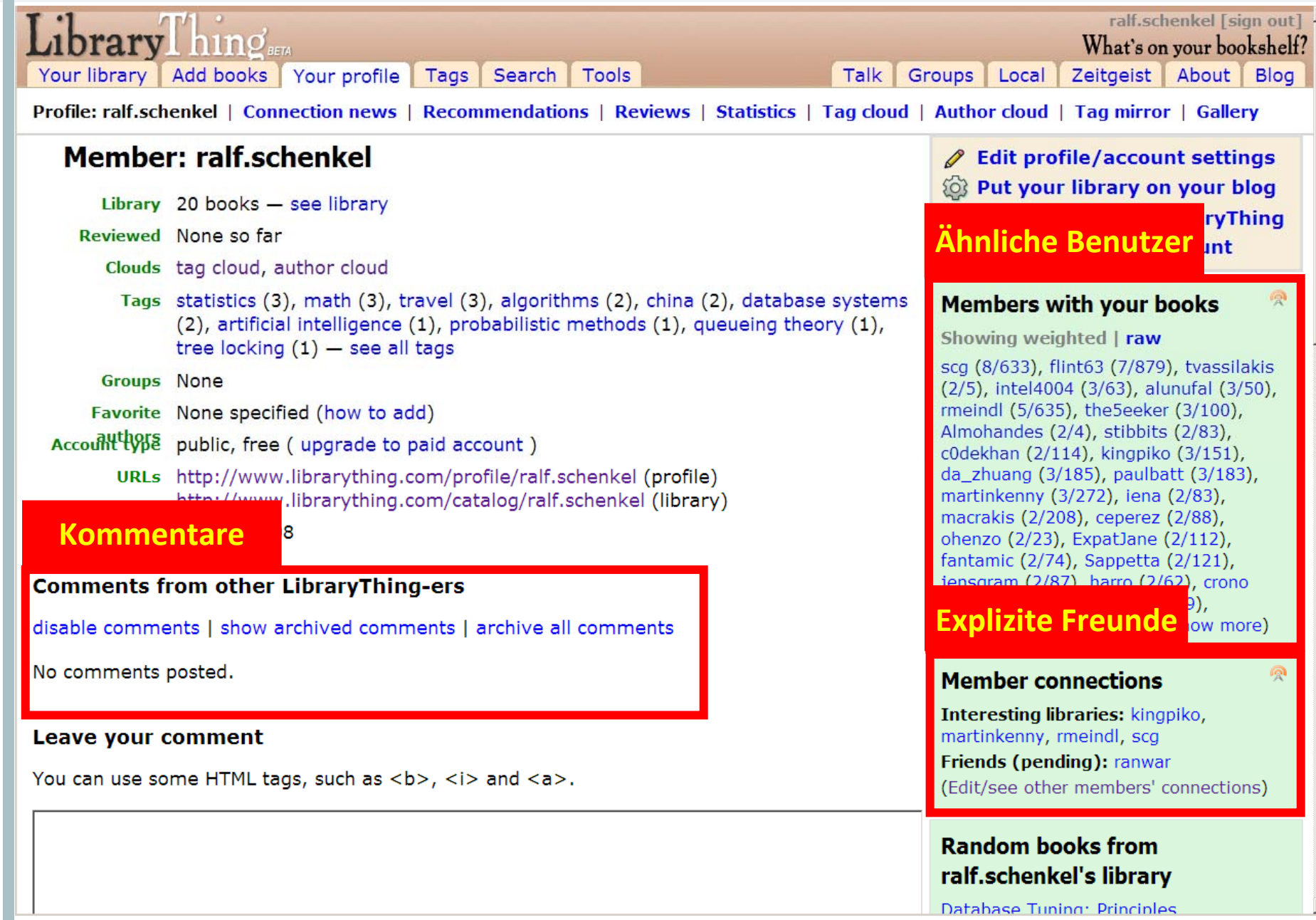
- Information **veröffentlichen** und **annotieren**
- Information **bewerten** und begutachten
- ihre **Interessen** veröffentlichen
- ein Netzwerk von **Freunden** unterhalten
- mit Freunden **interagieren**

Bekannte Beispiele:

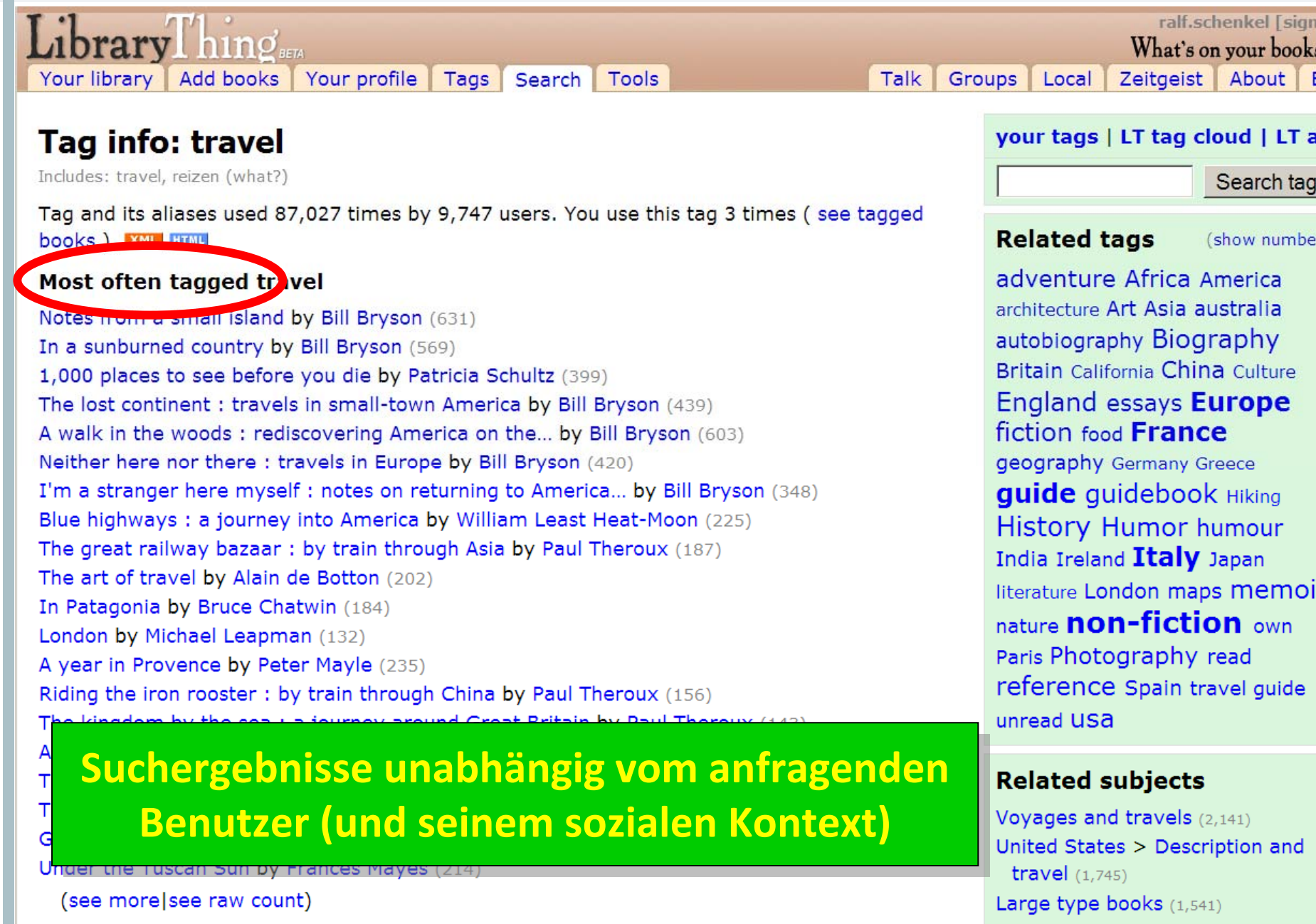
- Flickr (Fotos)
- YouTube (Videos)
- del.icio.us (Bookmarks)
- Librarything (Bücher)
- Discogs (CDs)
- CiteULike (Publikationen)
- Facebook
- Myspace (Medien)

Showcase: librarything.com

librarything.com: Soziale Interaktion



librarything.com: Suche

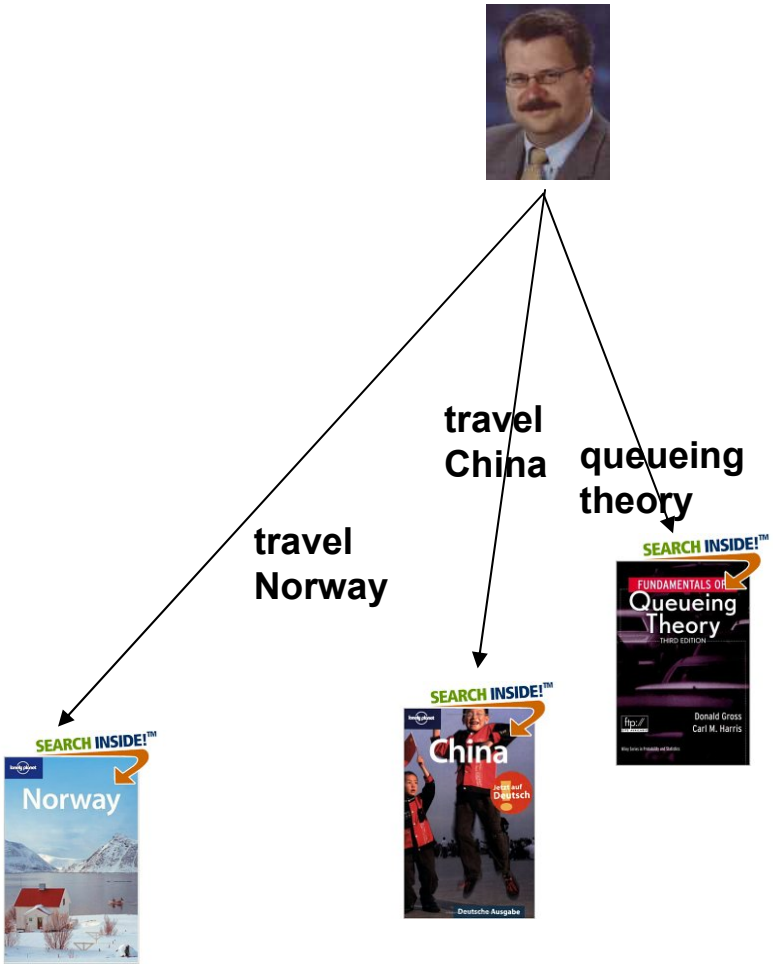


Suchergebnisse unabhängig vom anfragenden Benutzer (und seinem sozialen Kontext)

Outline

- **Modellierung von Social Tagging Networks**
  - Graphmodell
  - Verschiedene Informationsbedürfnisse
- Effektives Scoring für Anfragen
- Effiziente Ausführung von Anfragen
- Zusammenfassung und offene Fragen

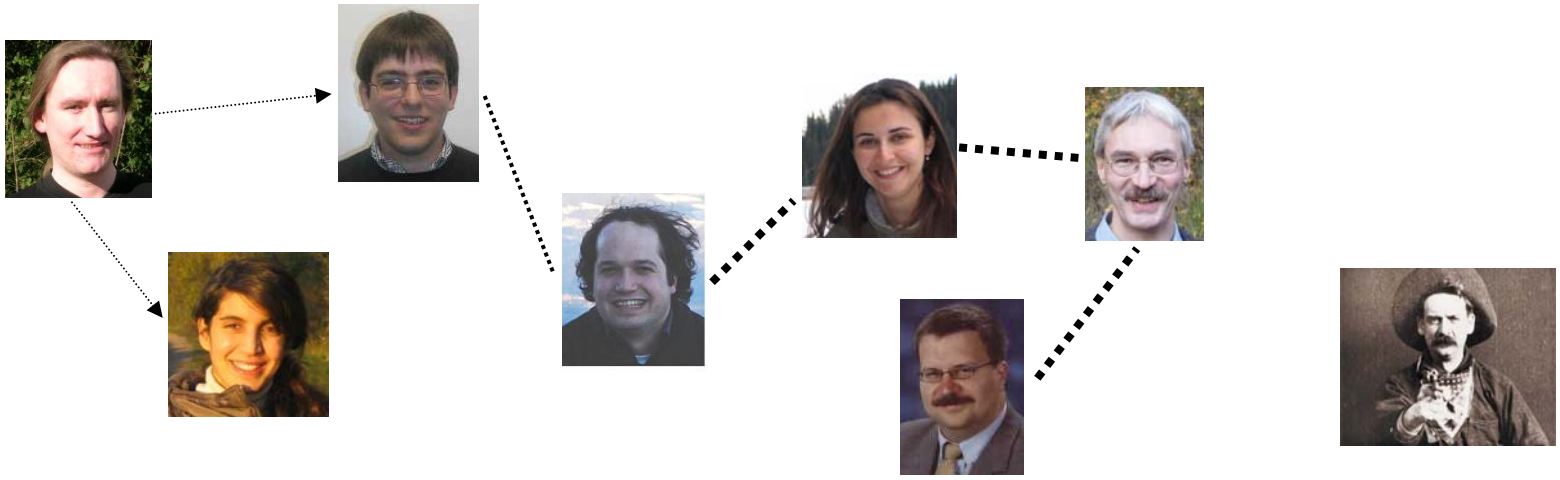
Modellierung von Social Tagging Networks



USERS TAGS ITEMS

CLUSTER OF EXCELLENCE

Modellierung von Social Tagging Networks



USERS

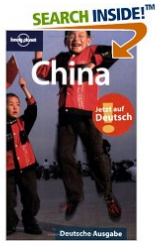
TAGS

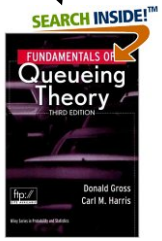
ITEMS

travel
Norway

travel
China

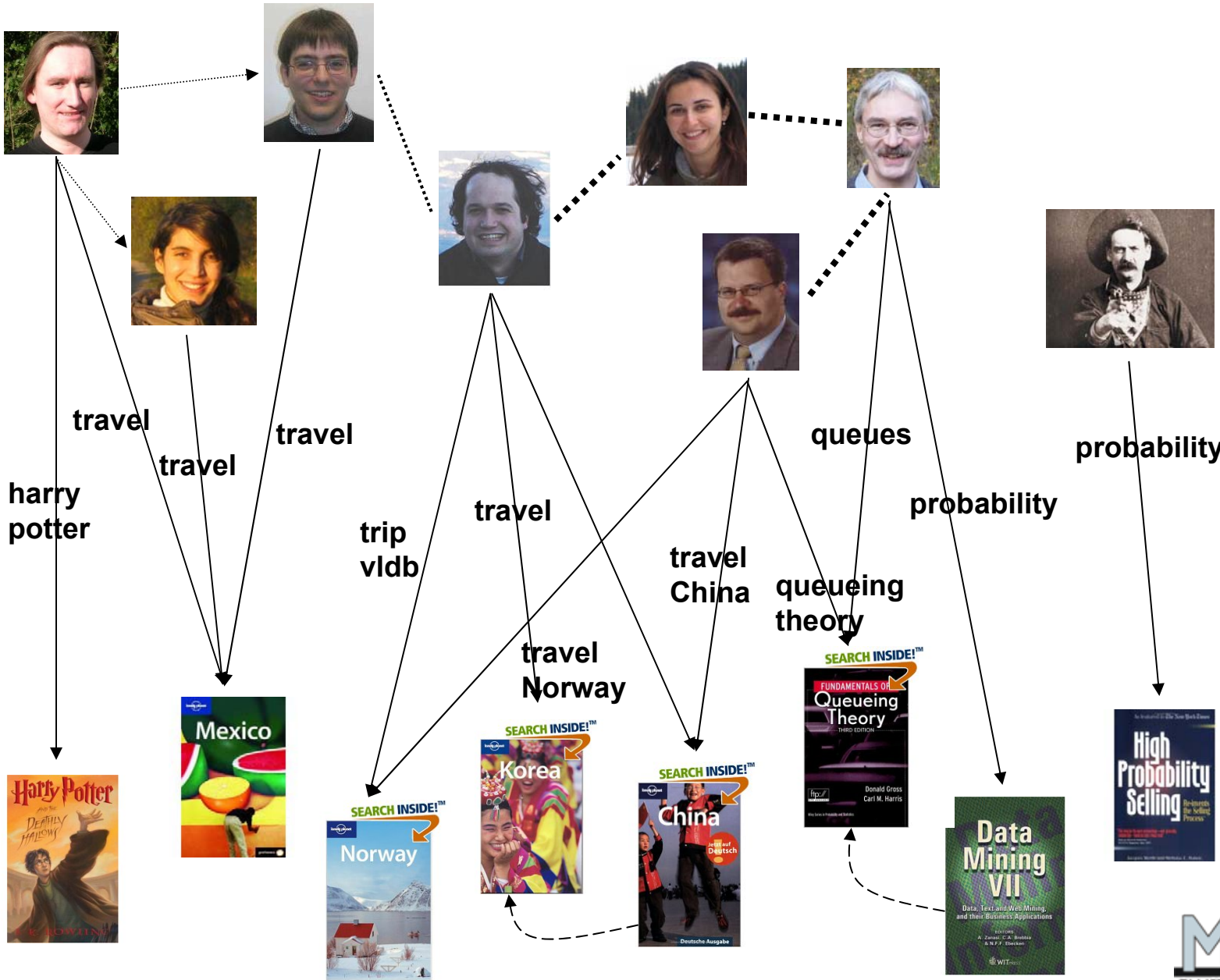
queueing
theory





CLUSTER OF EXCELLENCE

Modellierung von Social Tagging Networks



CLUSTER OF EXCELLENCE

Bausteine eines Social Tagging Network

Graph $G=(U \cup I, E_U \cup E_I \cup E_{UI})$ mit

- 2 Arten von Knoten:
  - Users U (optional gewichtet)
  - Items I (optional gewichtet)
- 3 Arten von Kanten:
  - E_U: User-User (optional gewichtet)
  - E_I: Item-Item (optional gewichtet)
  - E_{UI}: User-Item (beschriftet mit Tags T, opt. gewichtet)

Informationsbedarf 1: Global







harry potter

travel

travel

travel

harry potter

trip vldb

travel

travel Norway

travel China

queues

queueing theory

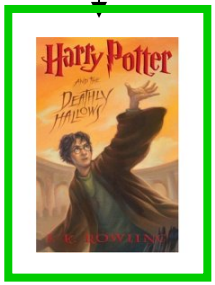
probability

probability

USERS

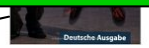
TAGS

ITEMS










Tags aller Benutzer sind gleich wichtig

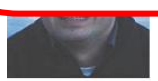
CLUSTER OF EXCELLENCE

Informationsbedarf 2: Ähnliche Benutzer

Informationsbedarf 3: Freunde



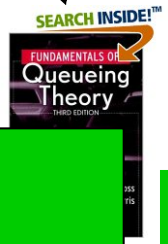


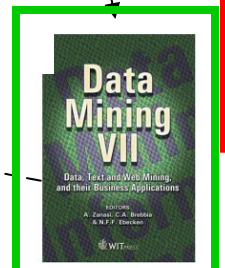



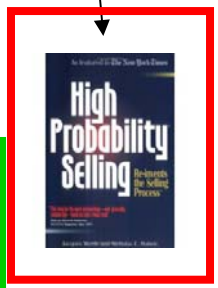
probability

travel
travel
travel
harry potter
trip vldb
travel
travel Norway
travel China
queues
queueing theory
probability
probability

USERS TAGS ITEMS

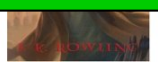


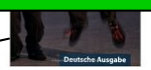




?

Tags von bekannten Freunden sind wichtiger





CLUSTER OF EXCELLENCE

Wunschliste: Social-Aware Social Search

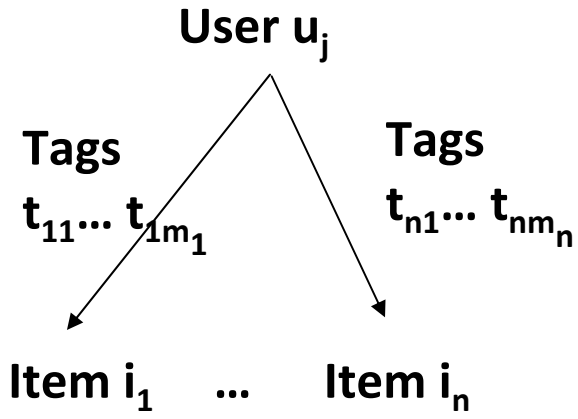
- Ergebnisse einer Suche hängen ab von
  - **Globaler** Popularität von Items
  - **Inhaltlichem Kontext** des anfragenden Benutzers (Items, Tags)
  - **Sozialem Kontext** des anfragenden Benutzers (Freunde)
- Automatische **Anfrageexpansion** (nicht nur Synonyme)
- **Skalierbare** Anfrageauswertung
- **Erklärung** der Ergebnisse

(Ähnliche Wunschliste für soziale Empfehlungssysteme)

Outline

- Modellierung von Social Tagging Networks
- **Effektives Scoring von Anfragen**
  - **Quantizierung von Freundschaften**
  - **Benutzerspezifische Scores**
  - **Experimentelle Evaluierung**
- Effiziente Ausführung von Anfragen
- Zusammenfassung und offene Fragen

Notation



tags(u): Tags, die von User u vergeben wurden

items(u): Items, die von User u getaggt wurden

items(t): Items, die mit Tag t getaggt wurden

df(t): Anzahl von Items, die mit Tag t getaggt wurden

tf_u(i,t): Häufigkeit, mit der Benutzer u Item i mit Tag t getaggt hat

tf(i,t): Häufigkeit, mit der Item i mit Tag t getaggt wurde

Quantifzierung von Freundschaftsgraden

- Globaler „Freundschafts“grad:

$$P_{global}(u,u') = \frac{1}{|U|}$$

- Inhaltsbasierter Freundschaftsgrad
- Graphbasierter Freundschaftsgrad
- Integrierter Freundschaftsgrad

Inhaltsbasierter Freundschaftsgrad

Verschiedene Möglichkeiten:

- **Gemeinsamkeiten der Tagverwendung:**

$$P_{content}(u,u') = \frac{2\,|\,tags(u) \cap tags(u')\,|}{|\,tags(u)\,| + |\,tags(u')\,|}$$

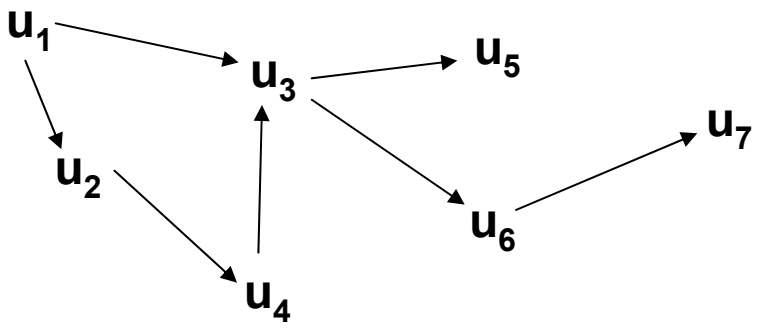
- **Gemeinsamkeiten der annotierten Items:**

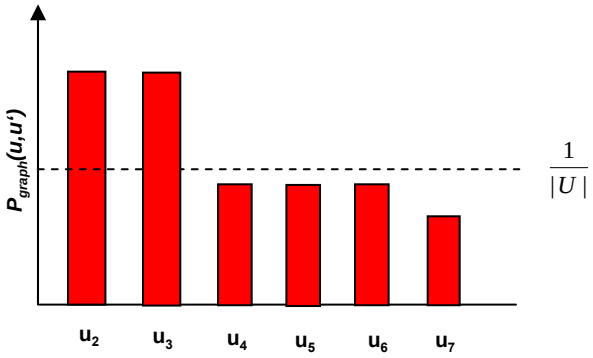
$$P_{content}(u,u') = \frac{2\,|\,items(u) \cap items(u')\,|}{|\,items(u)\,| + |\,items(u')\,|}$$

Für beide:

- $P_{content}(u,u) := 0$
- **Normalisierung so dass** $\sum_{u' \neq u} P_{content}(u,u') = 1$

Graphbasierter Freundschaftsgrad





Ungewichtete Kanten:

$$w(u_i, u_{i+1}) = 1$$

$$w(u_{i1}, \ldots, u_{i_j}) = \sum_{k=1}^{j-1} w(u_{i_k}, u_{i_{k+1}}) = j - 1$$

$$P_{graph}(u,u') = \frac{1}{\min\limits_{\text{path } p=u\ldots u'} w(p)}$$

Kanten gewichtet mit $P_{content}$:

$$w(u_i, u_{i+1}) = P_{content}(u_i, u_{i+1})$$

$$w(u_{i1}, \ldots, u_{i_j}) = \prod_{k=1}^{j-1} w(u_{i_k}, u_{i_{k+1}})$$

$$P_{graph}(u,u') = \max_{\text{path } p=u\ldots u'} w(p)$$

Für beide:

- $P_{graph}(u,u) := 0$
- **Normalisierung so dass** $\sum_{u' \neq u} P_{graph}(u,u') = 1$

Integrierter Freundschaftsgrad

Kombination von

- Inhaltsbasiertem Freundschaftsgrad
- Graphbasiertem Freundschaftsgrad
- Hintergrundmodell (globalem Freundschaftsgrad)

$$F(u,u') = \alpha \cdot P_{graph}(u,u') + \beta \cdot P_{content}(u,u') + (1-\alpha-\beta)\frac{1}{|U|}$$

$(0 \leq \alpha, \beta \leq 1;\ \alpha+\beta \leq 1)$

Exkurs: Scores im Text Retrieval

Allgemeines Framework:

$$score(i,t) \propto tf(i,t) \cdot idf(t)$$

Wichtigkeit von t für Item i (je *häufiger*, desto besser)

Wichtigkeit von t im Corpus (je *seltener*, desto besser)

Handoptimierte Instanz: Okapi BM25

$$score(i,t) = \frac{(k_1+1)tf(i,t)}{k_1 + tf(i,t)} \cdot \log \frac{|I| - df(t) + 0.5}{df(t) + 0.5}$$

Linearkombination zum Scoring von Anfragen

$$score(i,t_1 \ldots t_n) = \sum_{j=1}^{n} score(i,t_j)$$

Benutzerspezifisches Scoring

$$tf(i,t) = \sum_{u \in U} tf_u(i,t)$$

$$= |U| \cdot \sum_{u \in U} \underbrace{\frac{1}{|U|}}_{\text{Globaler Freundschaftsgrad}} tf_u(i,t)$$

Definition einer benutzerspezifischen *social* frequency:

$$sf_u(i,t) = |U| \cdot \sum_{u' \in U} F(u,u') tf_u(i,t)$$

Definition eines benutzerspezifischen *social score*:

$$score_u(i,t) = \frac{(k_1+1) sf_u(i,t)}{k_1 + sf_u(i,t)} \cdot \log \frac{|I| - df(t) + 0.5}{df(t) + 0.5}$$

Automatische Tagexpansion

Problem: Benutzer verwenden verschiedene Tags für ähnliche Dinge

$\Rightarrow$ schlechte Ausbeute (fehlende relevante Ergebnisse)

Beispiel:
MPI, MPII, MPI-INF, MPI-CS, Max-Planck-Institut, D5, AG5, DB&IS, UdS, …

Lösung:

1. Definition von **ähnlichen Tags**
2. **Expansion** von Anfragen mit ähnlichen Tags
3. **Modifikation** der Scoringfunktion für expandierte Anfragen

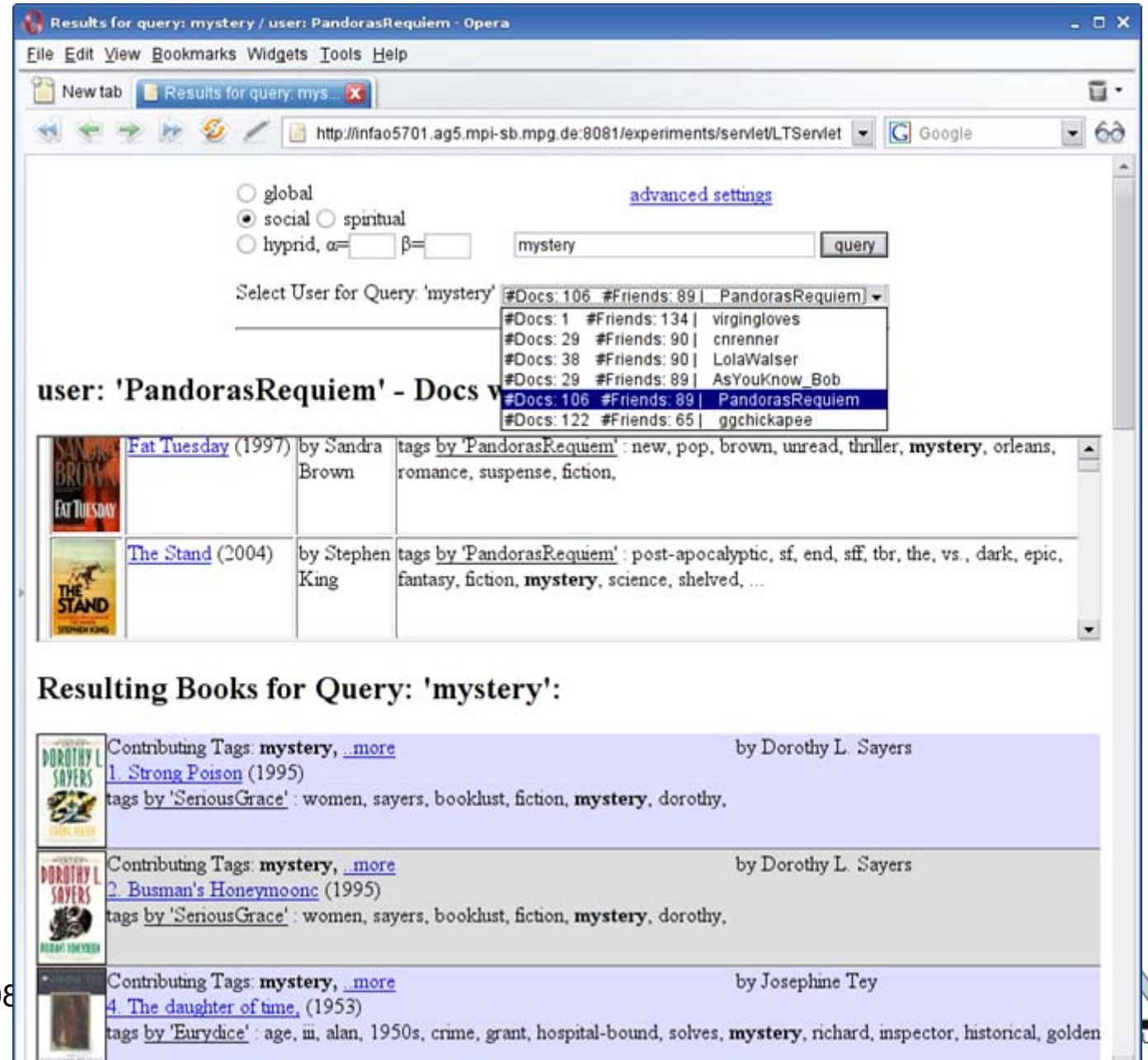
Experimentelle Evaluierung: Güte

Systematische Evaluierung der Ergebnisgüte schwierig

Drei mögliche Ansätze:

- **Benutzerstudien** (Anfragen und Ergebnisbewertungen durch Menschen)
- Anfragen und Bewertungen aus **externer Information** abgeleitet (Beispiel: DMOZ Kategorien)
- **Automatisierte Bewertungen** aus dem Kontext des Benutzers
  - Items, die von Freunden getaggt wurden
  - Items, die in der Zukunft getaggt werden

Prototyp [SIGIR 2008 Demo]



Benutzerstudie: LibraryThing

- Registrieren der Autoren bei Librarything
- Hinzufügen (eines Teils der) eigenen Bücher
- Auswählen von Freunden mit ähnlichem Profil
- Definieren von 33 Anfragen
- Crawlen (eines Teils) von LibraryThing
- Poolen und Bewerten der Ergebnisse
- Bestimmen von NDCG[10] für die unbekannten Ergebnisse (~gewichtete Präzision nach 10 Ergebnissen)

Benutzerstudie: LibraryThing

β (Inhaltsähnlichkeit)

α (Freundschaftsgraph)

	0,0	0,1	0,2	0,5	0,8	1,0
0,0	0,58	0,61	0,62	0,61	0,61	0,60
0,1	0,63	0,63	0,64	0,63	0,63	
0,2	0,62	0,62	0,63	0,64	0,64	
0,5	0,58	0,59	0,60	0,60		

- **Güte insgesamt sehr hoch (1.0 ist optimal)**
- **Kombination von sozialer und inhaltlicher Ähnlichkeit gewinnt**

Outline

- Modellierung von Social Tagging Networks
- Effektives Scoring von Anfragen
- **Effiziente Auswertung von Anfragen**
  - **Threshold-Algorithmen**
  - **ContextMerge**
  - **Experimentelle Evaluierung**
- Zusammenfassung & offene Fragen

Algorithmischer Überblick

- Input: Anfrage $q=\{t_1 \ldots t_n\}$ von Benutzer u
- Output: k Items mit höchsten Scores
- Ziele:
  - Unnötige Berechnungen vermeiden
  - Minimierung von Disk-I/O und CPU-Last
  - Verwenden von vorberechneten Daten auf Platte

Exkurs: Threshold-Algorithmen für Text IR

Input:

- Anfrage $q=\{t_1 \ldots t_n\}$
- Listen $L(t_p)$ mit Paaren $\langle i, score(i,t_p) \rangle$, sortiert nach $score(i,t_p)\downarrow$

Output: *k* Dokumente mit höchsten aggregiertem Score

Algorithmus:

Vorberechnung von $score_u(i,t)$ unmöglich

(hohe Dynamik, zu viele und zu lange Listen)

$\Rightarrow$ Threshold-Algorithmen nicht anwendbar

Zerlegung der Social Frequency (β=0)

$$sf_u(i,t) = |U| \cdot \sum_{u' \in U} F(u,u') tf_u(i,t)$$

$$= |U| \cdot \sum_{u' \in U} \left[\alpha P_{graph}(u,u') + (1-\alpha) \frac{1}{|U|} \right] tf_u(i,t)$$

$$= |U| \left[\sum_{u' \in U} \alpha P_{graph}(u,u') tf_u(i,t) + (1-\alpha) \sum_{u \in U} \frac{tf_u(i,t)}{|U|} \right]$$

$$= \underbrace{\alpha |U| \sum_{u' \in U} P_{graph}(u,u') tf_u(i,t)}_{\text{Abhängig von User u}} + \underbrace{(1-\alpha) tf(i,t)}_{\text{Unabhängig von User u}}$$

Berechne $sf_u(i,t)$ „on the fly“ aus $tf(i,t)$, Freuden von u und ihren getaggten Items

ContextMerge

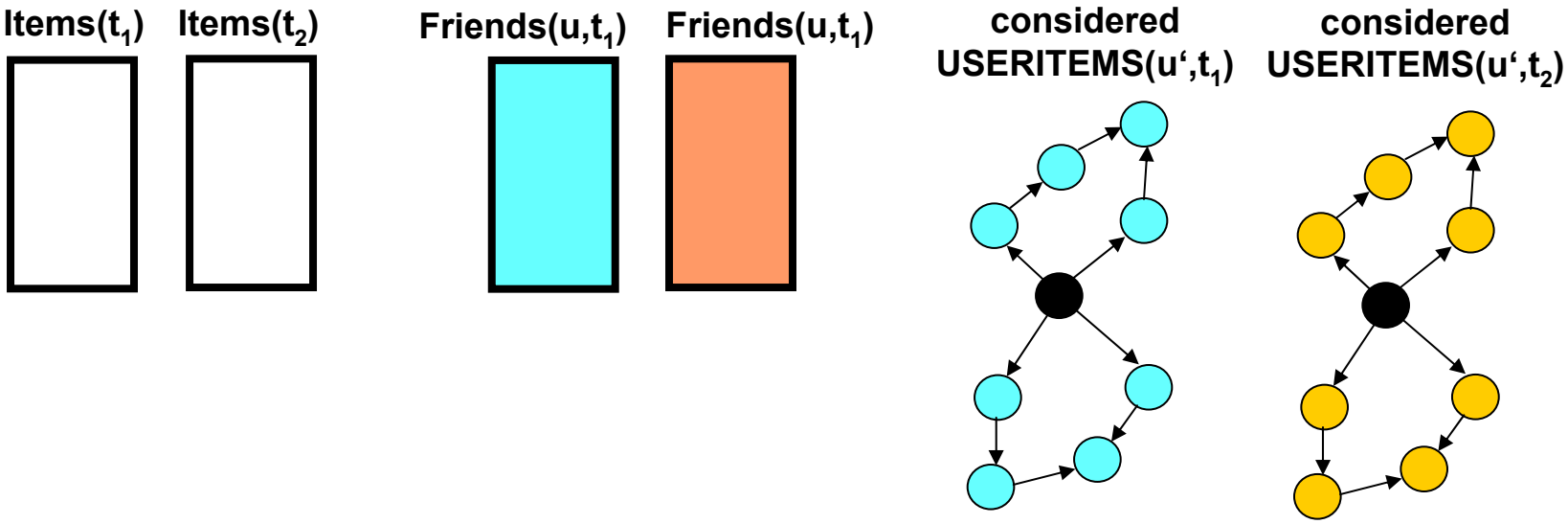
Vorberechnete Listen:

- ITEMS(t): Paare $<i, tf(i,t)>$, sortiert nach $tf(i,t)\downarrow$
- FRIENDS(u): Paare $<u', P_{graph}(u,u')>$, sort. nach $P_{graph}(u,u')\downarrow$
- USERITEMS(u',t): Paare $<i, tf_{u'}(i,t)>$, unsortiert

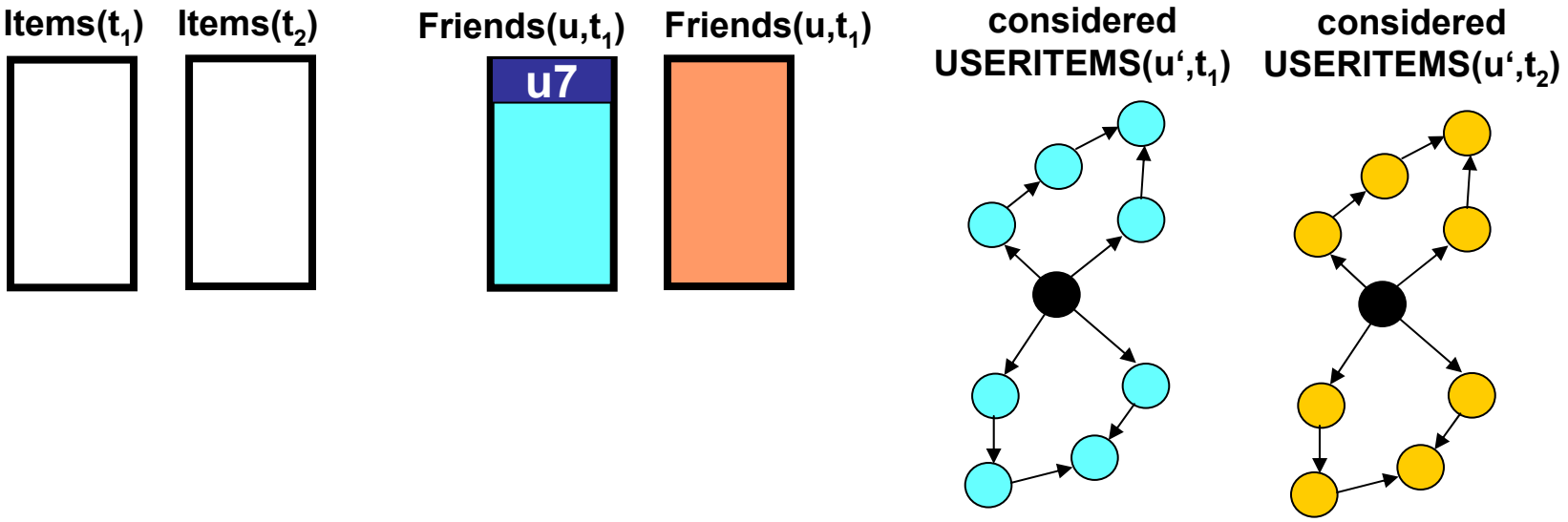
Anpassung von Threshold-Algorithmen für Query $u, t_1 \ldots t_n$:

- Scanne ITEMS(t_p) und *n* Kopien von FRIENDS(u), wähle „beste“ Liste
  - Falls ITEMS(t_p): lies nächsten Eintrag
  - Falls FRIENDS(u,p): Lies USERITEMS(u',t_p) für nächsten Freund u'
  - Aktualisiere Kandidaten und top-k
  - Prüfe Abbruchbedingung

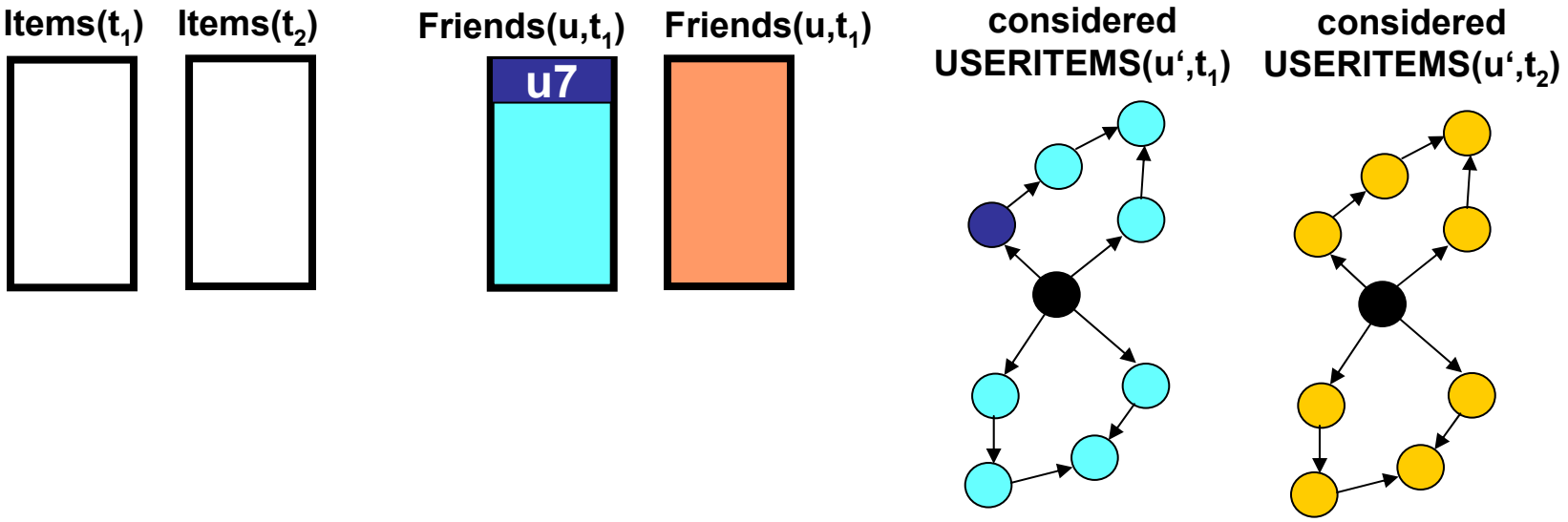
ContextMerge: Schematische Ausführung



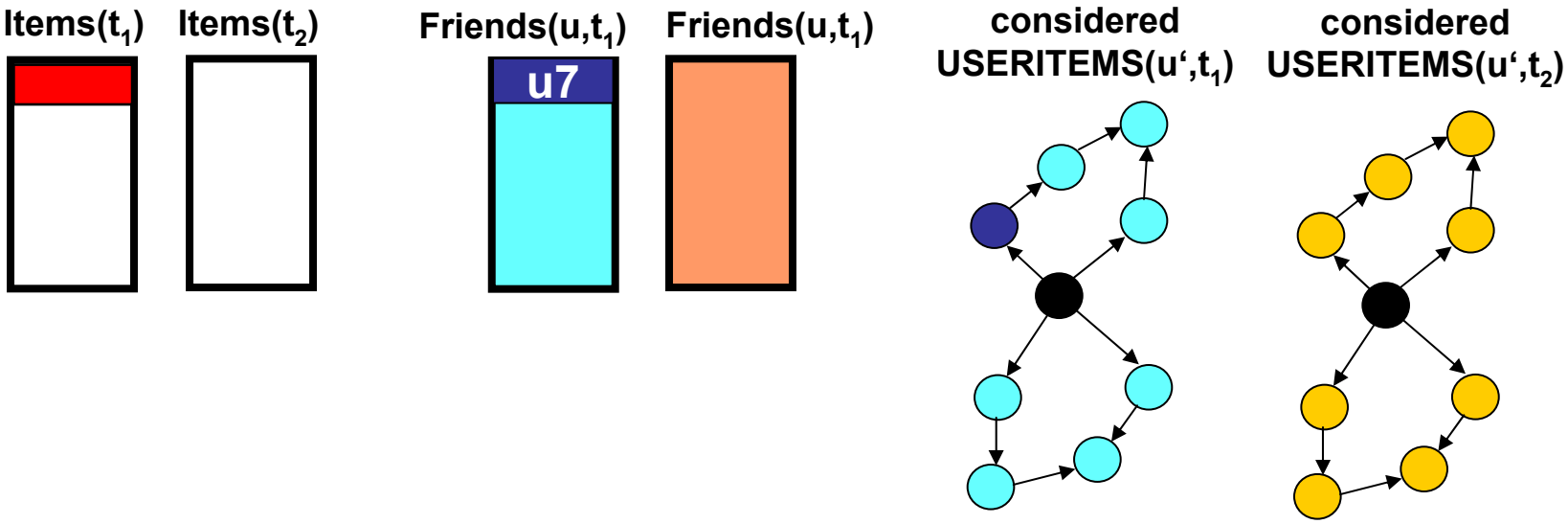
ContextMerge: Schematische Ausführung



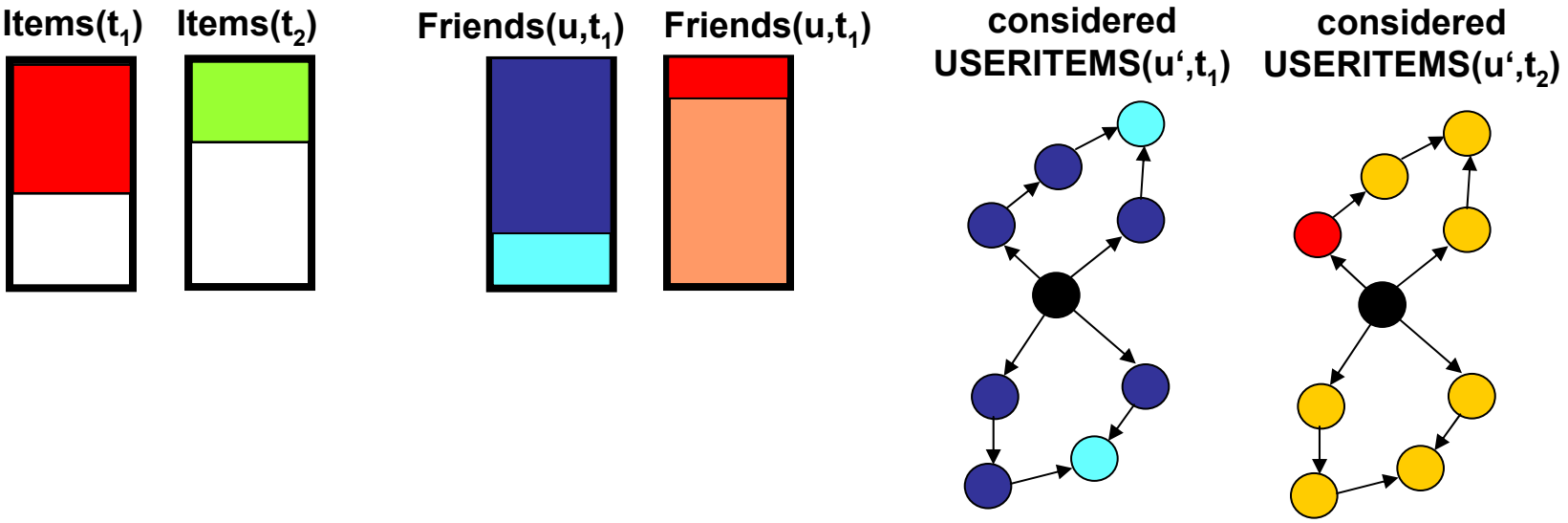
ContextMerge: Schematische Ausführung



ContextMerge: Schematische Ausführung



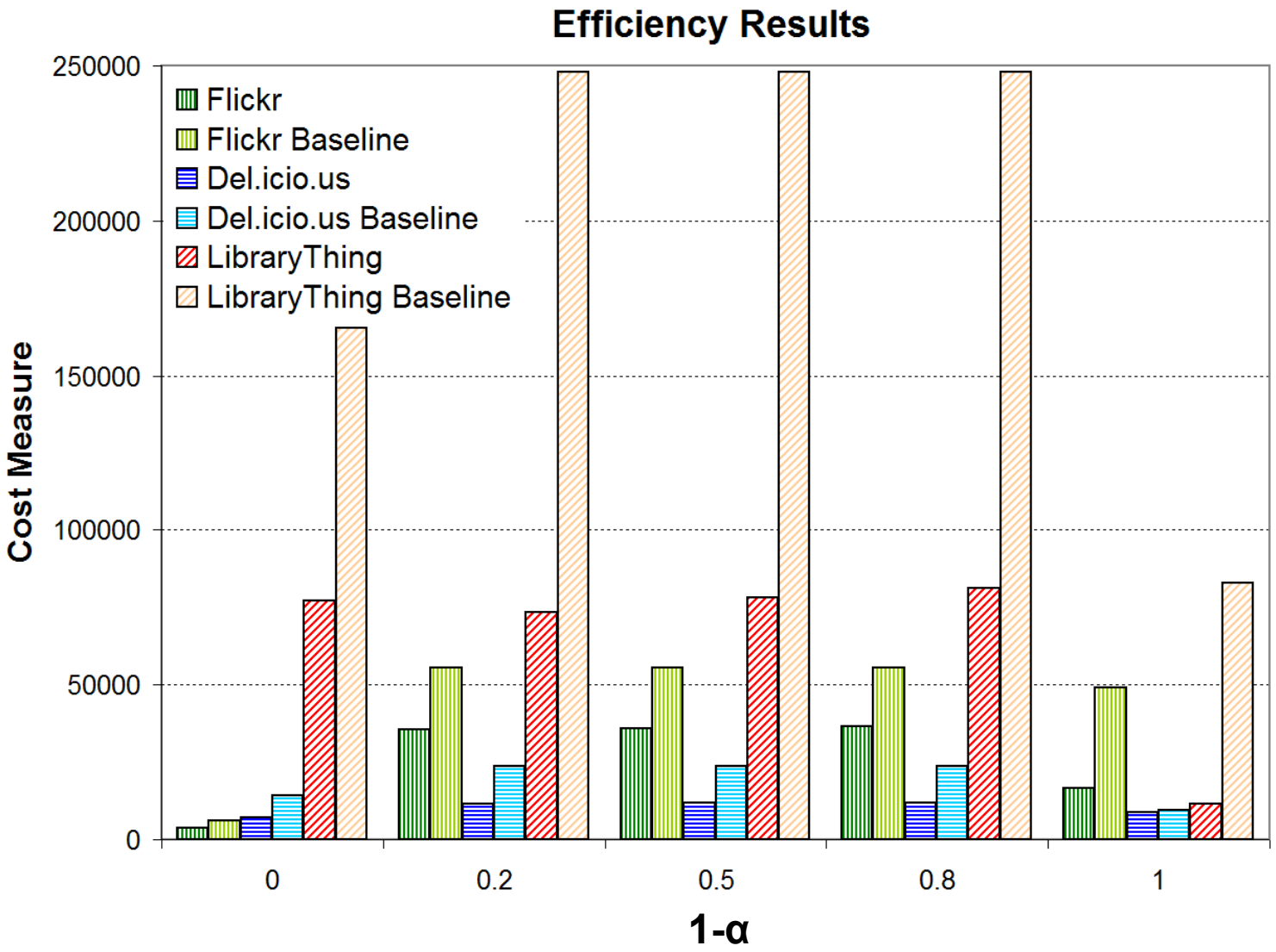
ContextMerge: Schematische Ausführung



Experimentelle Evaluierung: Effizienz

- Testbed: 3 große Crawls realer sozialer Netze
  - Flickr: 10 mio Fotos, ~50,000 Benutzer
  - Del.icio.us: ~175,000 Bookmarks, ~12,000 Benutzer
  - Librarything: ~6.5 mio Bücher, ~10,000 Benutzer
- Anfragen:
  - ~150 häufige Tagpaare in jedem Crawl
  - Für jede Anfrage: wähle Benutzer mit „genügend“ Ergebnissen & Freunden
- Gütemaß: #seq. Zugriffe + 100#wahlfr. Zugriffe
- Baseline: full join + sort

Experimentelle Evaluierung: Effizienz



Outline

- Modellierung von Social Tagging Networks
- Effektives Scoring von Anfragen
- Effiziente Ausführung von Anfragen
- **Zusammenfassung und offene Fragen**

Zusammenfassung

- **Social-Aware Social Search** ist notwendig
- **Kontextabhängiger Score**
  - integriert globale Popularität mit inhaltlichem und sozialem Kontext des Benutzers
  - Unterstützt dynamische Tagexpansion
- **ContextMerge**: skalierbare Implementierung

Offene Fragen

- Aussagekräftiger und verfügbarer **Benchmark**
- **Queryklassifikation** und **Autoparametrisierung**
- Unterstützung inkrementeller **Änderungen**
- Erweiterung auf **Ratings**, Gewichte, …
- Erweiterung auf **Nicht-Tags** (Bildfeatures, …)
- Nachvollziehbare **Erklärung** der Ergebnisse
- Ausnutzen der **Dynamik** (aktuelle Themen, wachsende Gruppen, …)

Social-Aware Search & Recommendations at planet scale