Enterprise and Desktop Search

Lecture 2: Searching the Enterprise Web

Pavel Dmitriev
Yahoo! Labs
Sunnyvale, CA
USA

Pavel Serdyukov
University of Twente
Netherlands

Sergey Chernov
L3S Research Center
Hannover
Germany

Outline

- Searching the Enterprise Web
  - What works and what doesn't (Fagin 03, Hawking 04)
- User Feedback in Enterprise Web Search
  - Explicit vs Implicit feedback (Joachims 02, Radlinski 05)
  - User Annotations (Dmitriev 06, Poblete 08, Chirita 07)
  - Social Annotations (Millen 06, Bao 07, Xu 07, Xu 08)
  - User Activity (Bilenko 08, Xue 03)
  - Short-term User Context (Shen 05, Buscher 07)

Searching the Enterprise Web

Searching the Workplace Web

Ronald Fagin   Ravi Kumar   Kevin S. McCurley

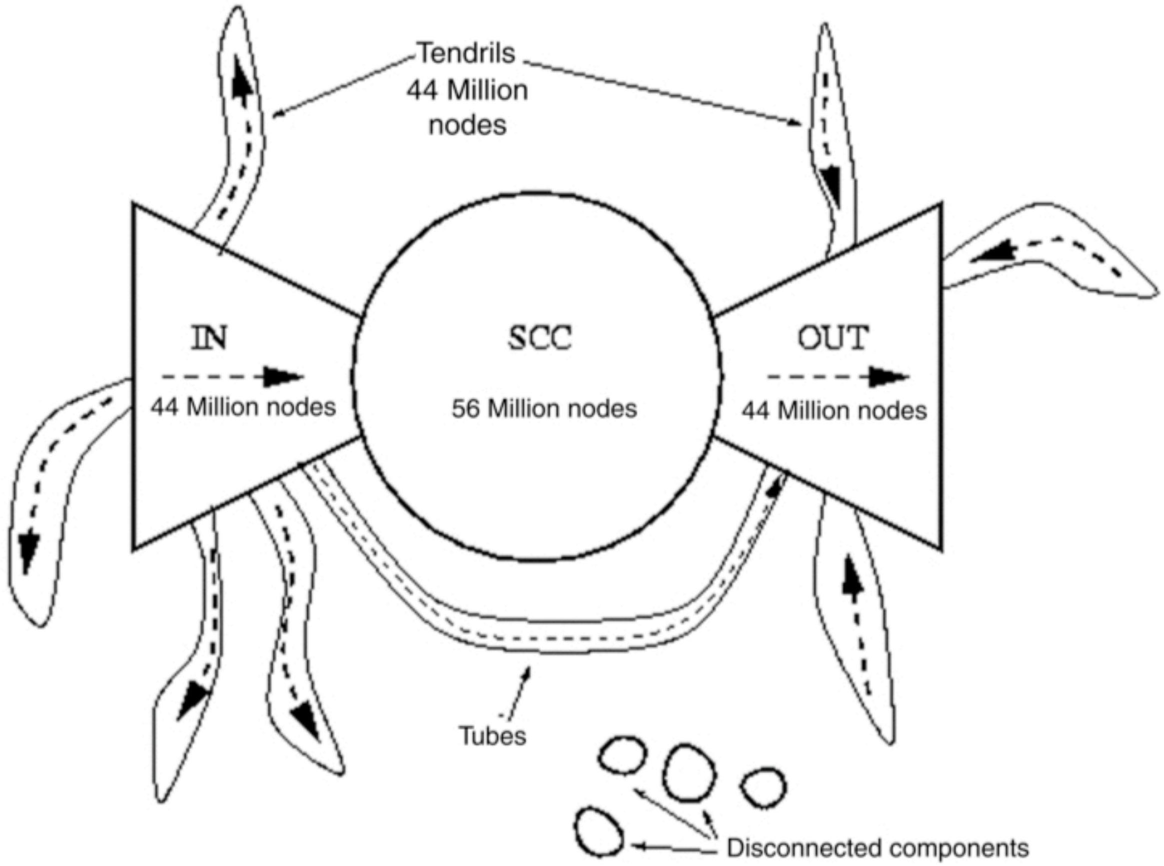
Jasmine Novak   D. Sivakumar

John A. Tomlin   David P. Williamson

IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120

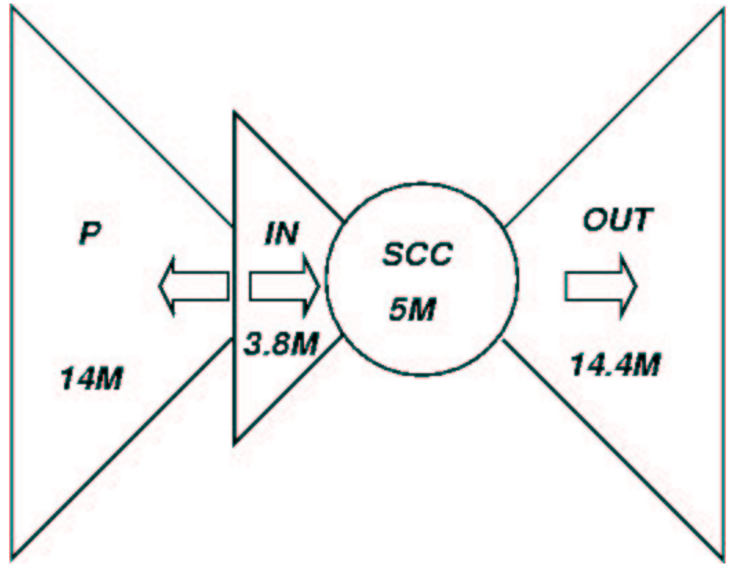
- How is Enterprise Web different from the Public Web?
  - Structural differences
- What are the most important features for search?
  - Use Rank Aggregation to experiment with different ranking methods and features

Enterprise Web vs Public Web: Structural Differences



Structure of the Public Web [Broder 00]

Enterprise Web vs Public Web: Structural Differences



Structure of Enterprise Web [Fagin 03]

- Implications:
  - More difficult to crawl
  - Distribution of PageRank values is such that larger fraction of pages has high PR values, thus PR may be less effective in discriminating among regular pages

Rank Aggregation

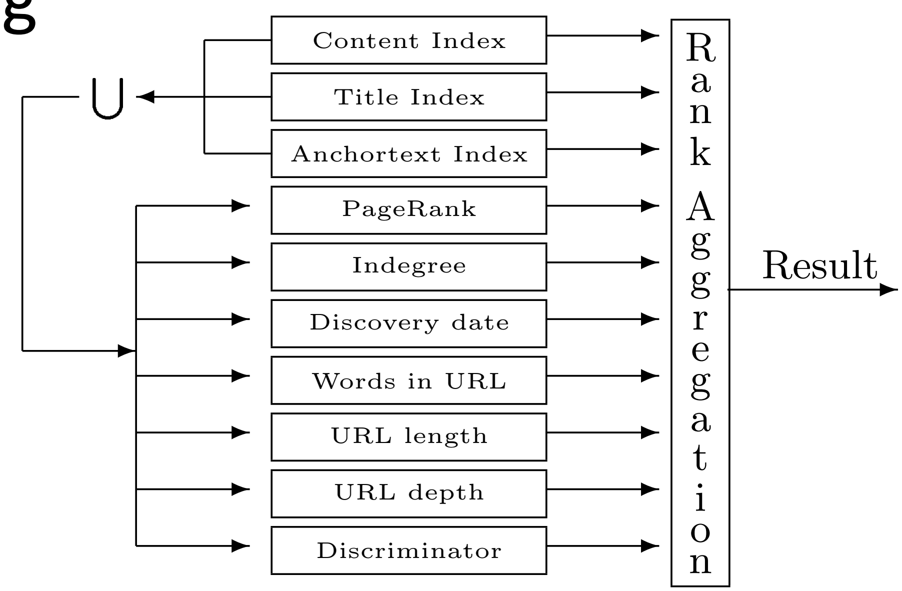
- Input: several ranked lists of objects
- Output: a single ranked list of the union of all the objects which minimizes the number of "inversions" wrt initial lists

- NP-hard to compute for 4 or more lists
- Variety of heuristic approximations exist for computing either the whole ordering or top k [Dwork 01, Fagin 03-1]

Rank Aggregation can also be useful in Enterprise Search for combining rankings from different data source

What are the most important features?

- Create 3 indices: Content, Title, Anchortext (aggregated text from the <a> tags pointing to the page)
- Get the results, rank them by tf-idf, and feed to the ranking heuristics
- Combine the results using Rank Aggregation
- Evaluate all possible subsets of indices and heuristics on very frequent (*Q1*) and medium frequency (*Q2*) queries with manually determined correct answers



Results

α	$I_{R1}(\alpha)$	$I_{R3}(\alpha)$	$I_{R5}(\alpha)$	$I_{R10}(\alpha)$	$I_{R20}(\alpha)$
Ti	29.2	13.6	5.6	6.2	5.6
An	24.0	47.1	58.3	74.4	87.5
Co	3.3	−6.0	−7.0	−4.4	−2.7
Le	3.3	4.2	1.8	0	0
De	−9.7	−4.0	−3.5	−2.9	−4.0
Wo	3.3	0	−1.8	0	1.4
Di	0	−2.0	−1.8	0	0
PR	0	13.6	11.8	7.9	2.7
In	0	−2.0	−1.8	1.5	0
Da	0	4.2	5.6	4.6	0

α	$I_{R1}(\alpha)$	$I_{R3}(\alpha)$	$I_{R5}(\alpha)$	$I_{R10}(\alpha)$	$I_{R20}(\alpha)$
Ti	6.7	8.7	3.4	3.0	0
An	23.1	31.6	30.4	21.4	15.2
Co	−6.2	−4.0	3.4	0	5.6
Le	6.7	−4.0	0	0	−5.3
De	−18.8	−8.0	−10	−8.8	−7.9
Wo	6.7	−4.0	0	0	0
Di	−6.2	−4.0	0	0	0
PR	6.7	4.2	11.1	6.2	2.7
In	−6.2	−4.0	0	0	0
Da	14.3	4.2	3.4	0	2.7

$I_{Ri}(a)$ is “influence” of the ranking metnod *a*

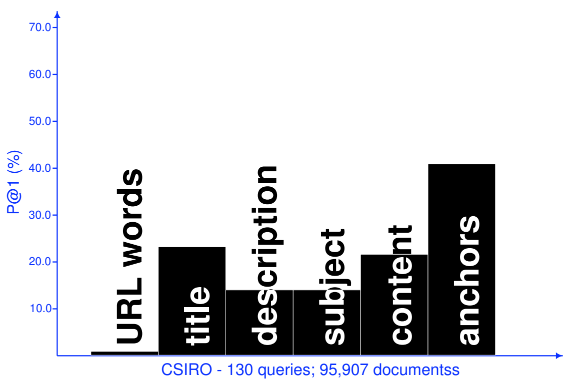
<u>Observations</u>:

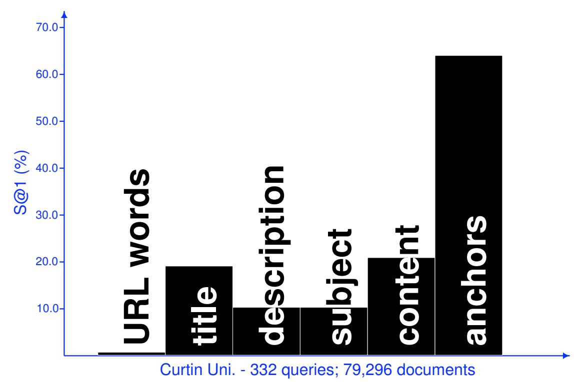
- Anchortext is by far the most influential feature
- Title is very useful, too
- Content is ineffective for *Q1*, but is useful for *Q2*
- PR is useful, but does not have a huge impact

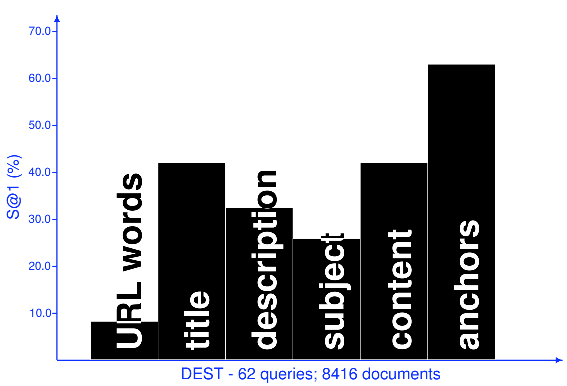
Challenges in Enterprise Search

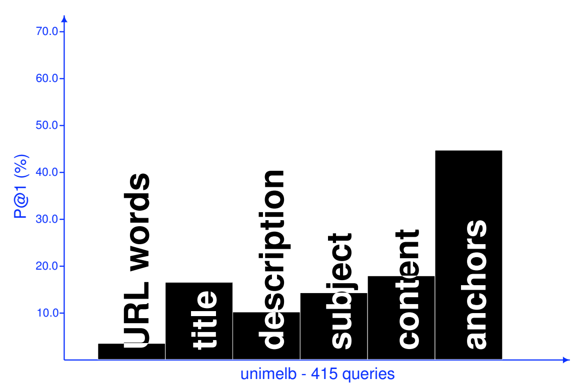
David Hawking

CSIRO ICT Centre,
GPO Box 664,
Canberra, Australia 2601
David.Hawking@csiro.au









This study confirms most of the findings if [Fagin 03] on 6 different Enterprise Webs (results for 4 datasets are shown)

- Anchortext and title are still the best
- Content is also useful

Summary

- Enterprise Web and Public Web exhibit significant structural differences
- These differences result in some features very effective for web search not being so effective for Enterprise Web Search
  - Anchortext is very useful (but there is much less of it)
  - Title is good
  - Content is questionable
  - PageRank is not as useful

Using User Feedback in Enterprise Web Search

Using User Feedback

- One of the most promising directions in Enterprise Search
  - Can trust the feedback (no spam)
  - Can provide incentives
  - Can design a system to facilitate feedback
  - Can actually implement it
- We will look at several different sources of feedback
  - Clicks (very briefly)
  - Explicit Annotations
  - Queries
  - Social Annotations
  - Browsing Traces

Sources of Feedback in Web Search

- ~~Explicit Feedback~~
  - ~~Overhead for user~~
  - ~~Only few users give feedback~~
    ~~=> not representative~~
- Implicit Feedback
  - Queries, clicks, time, mousing, scrolling, etc.
  - No Overhead
  - More difficult to interpret

[Joachims 02, Radlinski 05]

Using Click Data to Improve Search

- Very active area of research in both academia and industry, mostly in the context of Public Web search, but can be applied to Enterprise Web search as well

- The idea is treat clicks as relevance votes ("clicked"="relevant"), or as preference votes ("clicked page" > "non-clicked page"), and then use this information to modify the search engine's ranking function

See RuSSIR'07, "Machine Learning for Web-Related Problems", lecture 3.

Explicit and Implicit Annotations

Using Annotations in Enterprise Search

Pavel A. Dmitriev
Department of Computer Science
Cornell University
Ithaca, NY 14850
dmitriev@cs.cornell.edu*

Nadav Eiron
Google Inc.
1600 Amphitheatre Pkwy.
Mountain View, CA 94043*

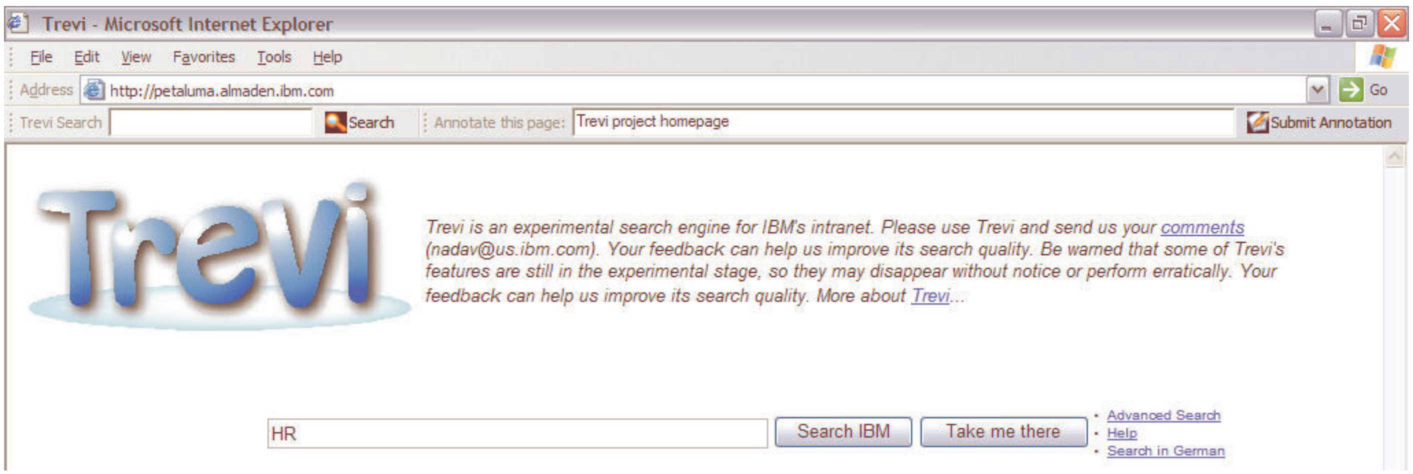
Marcus Fontoura
Yahoo! Inc.
701 First Avenue
Sunnyvale, CA, 94089
marcusf@yahoo-inc.com*

Eugene Shekita
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120
shekita@almaden.ibm.com

- Anchortext is the most important ranking feature for Enterprise Web Search
- But the quantity of the anchortext is very limited in the Enterprise
- Can we use user annotations as a substitute for anchortext?

Explicit Annotations

- Create a Toolbar to allow users annotate pages they visit



- Provide incentives to annotate:
  - Personal annotation appears in the toolbar every time user visits the page
  - Aggregated annotations from all users appear in search engine results

Examples of Explicit Annotations

Annotation	**Annotated Page**
change IBM passwords	Page about changing various passwords in IBM intranet
stockholder account access	Login page for IBM stock holders
download page for Cloudscape and Derby	Page with a link to Derby download
ESPP home	Details on Employee Stock Purchase Plan
EAMT home	Enterprise Asset Management homepage
PMR site	Problem Management Record homepage
coolest page ever	Homepage of an IBM employee
most hard-working intern	an intern’s personal information page
good mentor	an employee’s personal information page

Implicit Annotations

- Mine annotations from query logs
  - Treat queries as annotations for relevant pages
  - While such annotations are of lower quality, a large number of them can be collected easily

```
LogRecord ::= <Query> | <Click>
Query ::= <Time>\t<QueryString>\t<UserID>
Click ::= <Time>\t<QueryString>\t<URL>\t<UserID>
```

- How to determine "relevant" pages? [Joachims 02, Radlinski 05]

Strategy 1

- Assume every clicked page is relevant
  - Simple to implement
  - Produces a large number of annotations
  - But may attach an annotation to an irrelevant page

Strategy 2

- *Session* = time ordered sequence of clicks a user makes for a given query

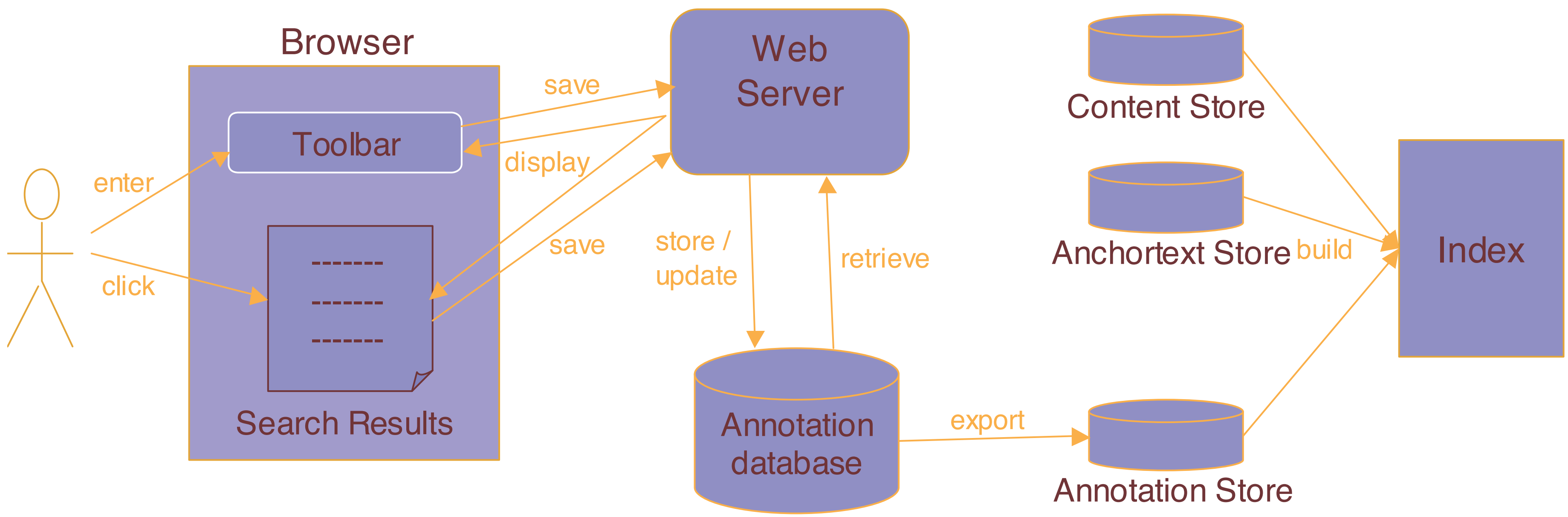
- Assume only the last click in the session is relevant
  - Produces less annotations
  - Avoids assigning annotations to irrelevant pages

Strategies 3 & 4

- *Query Chain* = time ordered sequence of queries executed over a short period of time

- Strategy 3: Assume every click in the query chain is relevant
- Strategy 4: Assume only the last click in the last session of the query chain is relevant

Using Annotations in Enterprise Web Search



Flow of Annotations through the system

Experimental Results

- Dataset: 5.5M index of IBM intranet
- Queries: 158 test queries with manually identified correct answers
- Evaluation was conducted after 2 weeks since starting collecting the annotations

Baseline	EA	IA 1	IA 2	IA 3	IA 4
8.9%	13.9%	8.9%	8.9%	9.5%	9.5%

Table 2: Summary of the results measured by the percentage of queries for which the correct answer was returned in the top 10. EA = Explicit Annotations, IA = Implicit Annotations.

P-TAG: Large Scale Automatic Generation of Personalized Annotation TAGs for the Web

Paul - Alexandru Chirita[1,*], Stefania Costache[1], Siegfried Handschuh[2], Wolfgang Nejdl[1]

[1]L3S Research Center / University of Hannover, Appelstr. 9a, 30167 Hannover, Germany
{chirita,costache,nejdl}@l3s.de

[2]National University of Ireland / DERI, IDA Business Park, Lower Dangan, Galway, Ireland
Siegfried.Handschuh@deri.org

- Want to generate personalized web page annotations based on documents on the user's Desktop
- Suppose we have an index of Desktop documents on the user's computer (files, email, browser cache, etc.)

Extracting tags from Desktop documents

- Given a web page to annotate, the algorithm proceeds as follows:
  - Step 1: Extract important keywords from the page
  - Step 2: Retrieve relevant documents using the Desktop search
  - Step 3: Extract important keywords from the retrieved documents as annotations
- Users judged 70%-80% of annotations created using this algorithm as relevant

Query-Sets: Using Implicit Feedback and Query Patterns to Organize Web Documents

Barbara Poblete
Web Research Group
University Pompeu Fabra
Barcelona, Spain
barbara.poblete@upf.edu

Ricardo Baeza-Yates
Yahoo! Research &
Barcelona Media Innovation Center
Barcelona, Spain
ricardo@baeza.cl

- When have lots of annotations for a given page, which ones should we use?
- This paper proposes to perform frequent itemset mining to extract recurring groups of terms from annotations
  - Show that this type of processing is useful for web page classification
  - May also be useful for improving search quality by eliminating noisy terms

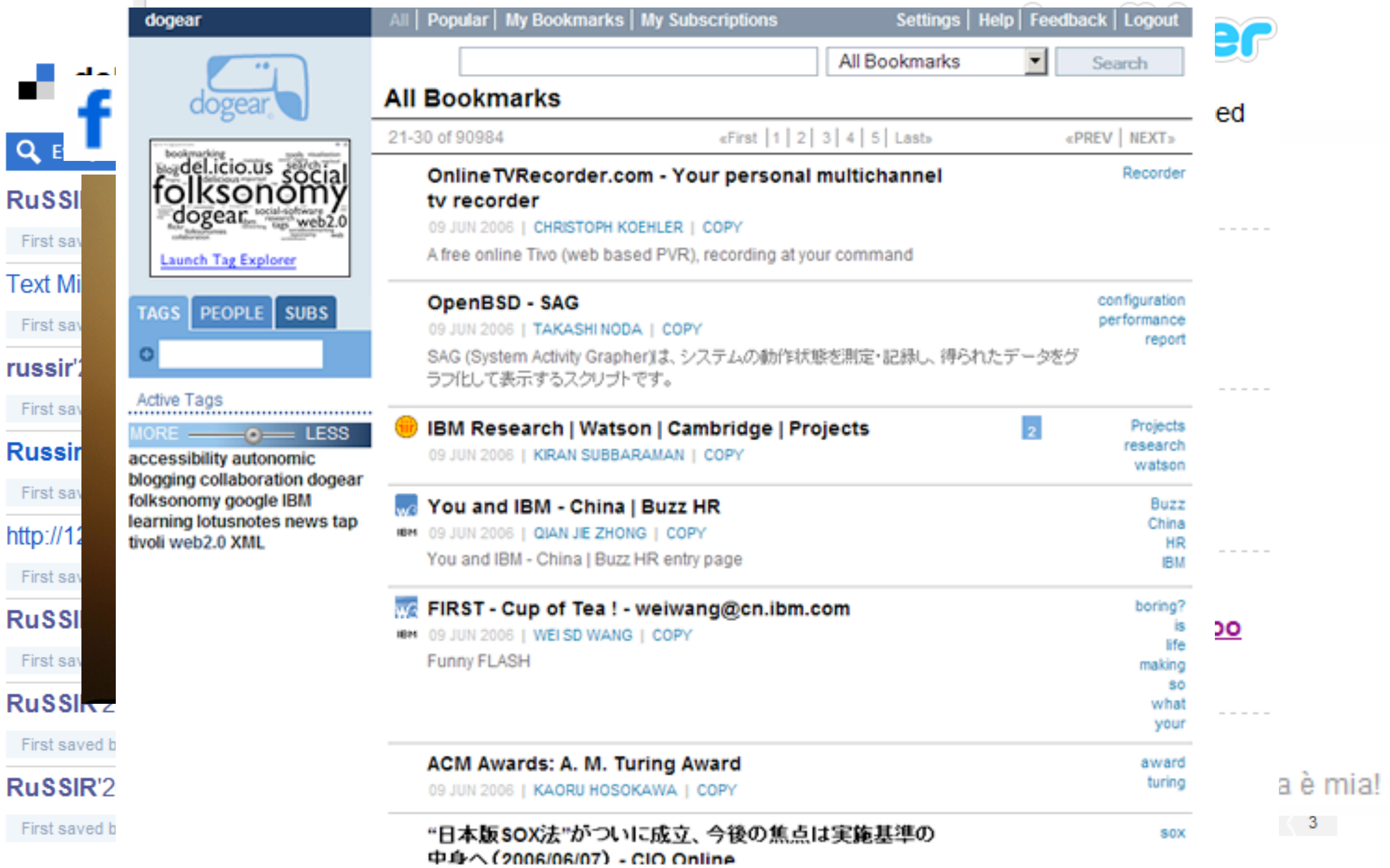
Summary

- User Annotations can help improve search quality in the Enterprise
- Annotations can be collected by explicitly asking users to provide them, or by mining query logs and users' Desktop contents
- Post-processing the resulting annotations may help to improve the search quality

Social Annotations

Tagging

- Easy way for the users to annotate web objects
- People do it (no one really knows why)



Tagging

- Very popular on the Web, becoming more and more popular in the Enterprise
  - Users add tags to objects (pages, pictures, messages, etc.)
  - Tagging System keeps track of <*user*, *obj*, *tag*> triples and mines/organizes this information for presenting it to the user (more in Lecture 3)
- In this lecture we will see how tags can be used to improve search in enterprise web

Using Tagging to Improve Search

- Approach 1: Merge tags with content or anchortext
- Approach 2: Keep tags separate and rank query results by

$$\alpha \times content\_match + (1-\alpha) \times tag\_match$$

- Other approaches: explore the social/collaborative properties of tags
  - Give more weight to some users and tags vs others
  - Compute similarities between tags and documents and incorporate it into ranking

Optimizing Web Search Using Social Annotations

Shenghua Bao[1*], Xiaoyuan Wu[1*], Ben Fei[2], Guirong Xue[1], Zhong Su[2], and Yong Yu[1]

[1]Shanghai JiaoTong University
Shanghai, 200240, China
{shhbao, wuxy, grxue, yyu}@apex.sjtu.edu.cn

[2]IBM China Research Lab
Beijing, 100094, China
{feiben, suzhong}@cn.ibm.com

- Observation: similar (semantically related) annotations are usually assigned to similar (semantically related) web pages
  – The similarity among annotations can be identified by similar web pages they are assigned to
  – The similarity among web pages can be identified by similar annotations they are annotated with
- Proposed iterative algorithm to compute these similarities and use them to improve ranking

Algorithm 1: SocialSimRank (SSR)

Step 1 ***Init:*** ***Let*** $S_A^{\,0}(a_i, a_j) = 1$ for each $a_i = a_j$ otherwise 0

$S_P^{\,0}(p_i, p_j) = 1$ for each $p_i = p_j$ otherwise 0

Step 2 ***Do*** {

For each annotation pair (a_i, a_j) ***do***

Similarity of annotations a_i and a_j

Sum over all pairs of pages annotated with a_i or a_j

$$S_A^{k+1}(a_i, a_j) = \frac{C_A}{|P(a_i)||P(a_j)|} \sum_{m=1}^{|P(a_i)|} \sum_{n=1}^{|P(a_j)|} \frac{\min(M_{AP}(a_i, p_m), M_{AP}(a_j, p_n))}{\max(M_{AP}(a_i, p_m), M_{AP}(a_j, p_n))} S_P^k(P_m(a_i), P_n(a_j)), \quad (2)$$

For each page pair (p_i, p_j) ***do***

Similarity of pages p_i and p_j

Sum over all pairs of annotations assigned to a_i or a_j

$$S_P^{k+1}(p_i, p_j) = \frac{C_P}{|A(p_i)||A(p_j)|} \sum_{m=1}^{|A(p_i)|} \sum_{n=1}^{|A(p_j)|} \frac{\min(M_{AP}(a_m, p_i), M_{AP}(a_n, p_j))}{\max(M_{AP}(a_m, p_i), M_{AP}(a_n, p_j))} S_A^{k+1}(A_m(p_i), A_n(p_j)), \quad (3)$$

}***Until*** $S_A(a_i, a_j)$ converges.

Step 3 ***Output:*** $S_A(a_i, a_j)$

Using Annotation Similarity for Ranking

- Given a query $q=\{q_1,\ldots,q_n\}$, a page p, and a set of annotations $A(p)=\{a_1,\ldots,a_m\}$, "social similarity" of q and p can be computed as follows:

$$sim_{SSR}(q,p)=\sum_{i=1}^{n}\sum_{j=1}^{m}S_A(q_i,a_j)$$

- Combine different ranking features using RankSVM (Joachims 02)

DocSimilarity	Similarity between query and page content
TermMatching (TM)	Similarity between query and annotations using the term matching method.
SocialSimRank (SSR)	Similarity between query and annotations based on SocialSimRank.

***See (Xu 07) for how to use annotation similarity in a Language Modeling framework**

Experimental Results

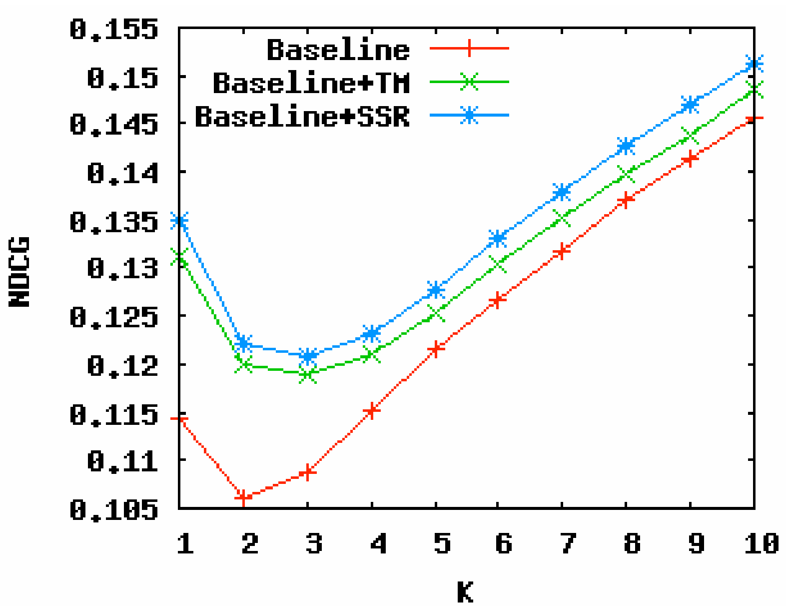
- Data from Delicious: 1,736,268 pages, 269,566 different annotations

Example:
Top 4 related annotations for different categories

Technology related:	
dublin	metadata, semantic, standard, owl
debian	distribution, distro, ubuntu, linux
Economy related:	
adsense	sense, advertise, entrepreneur, money
800	number, directory, phone, business
Entertainment related:	
album	gallery, photography, panorama, photo
chat	messenger, jabber, im, macosx
Entity related:	
einstein	science, skeptic, evolution, quantum
christian	devote, faith, religion, god

Experimental Results

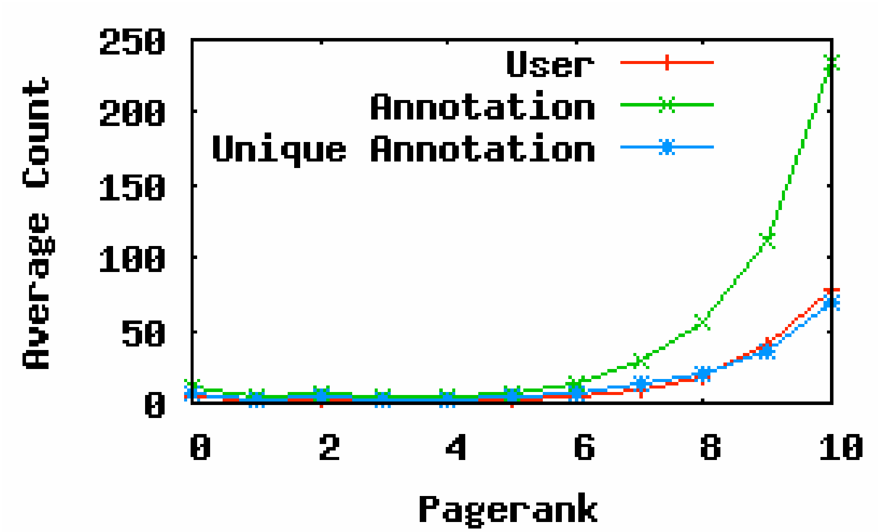
- Two query sets:
  - MQ50: 50 queries manually generated by students
  - AQ3000: 3000 queries auto-generated from ODP
- Measure NDCG and MAP:



Method	MQ50	AQ3000
Baseline	0.4115	0.1091
Baseline +TM	0.4341	0.1128
Baseline +SSR	**0.4697**	**0.1147**

What about PageRank?

- <u>Observation</u>: popular web pages attract hot social annotations and bookmarked by up-to-date users



- Use these properties to estimate popularity of pages (SocialPageRank)

Algorithm 2: SocialPageRank (SPR)

Step 1 ***Input:***

Association matrices M_{PU}, $M_{AP,}$ and M_{UA} and the random initial SocialPageRank score P_0

Step 2 ***Do:***

Page-User association matrix

$$U_i = M_{PU}^T \cdot P_i \qquad (5.1)$$

User-Ann. association matrix

$$A_i = M_{UA}^T \cdot U_i \qquad (5.2)$$

Ann.-Page association matrix

$$P_i' = M_{AP}^T \cdot A_i \qquad (5.3)$$

$$A_i' = M_{AP} \cdot P_i' \qquad (5.4)$$

$$U_i' = M_{UA} \cdot A_i' \qquad (5.5)$$

$$P_{i+1} = M_{PU} \cdot U_i' \qquad (5.6)$$

(5)

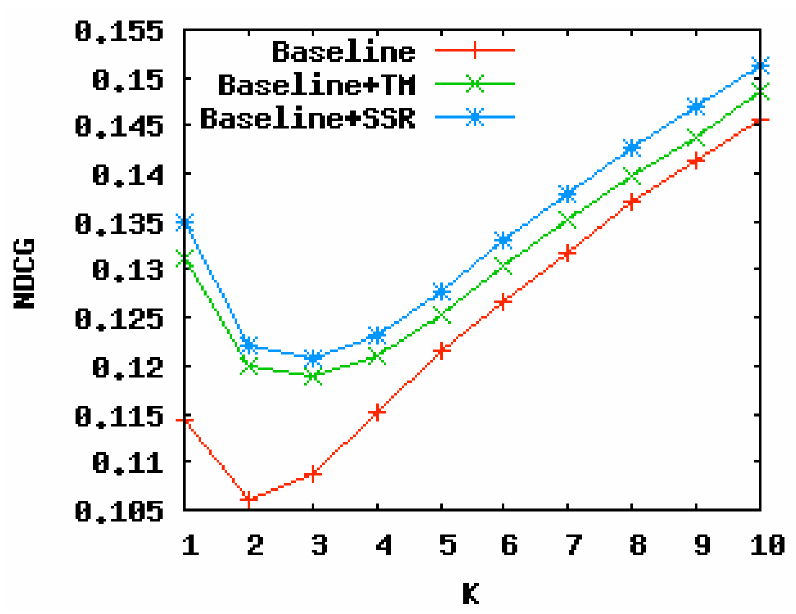
Until P_i converges.

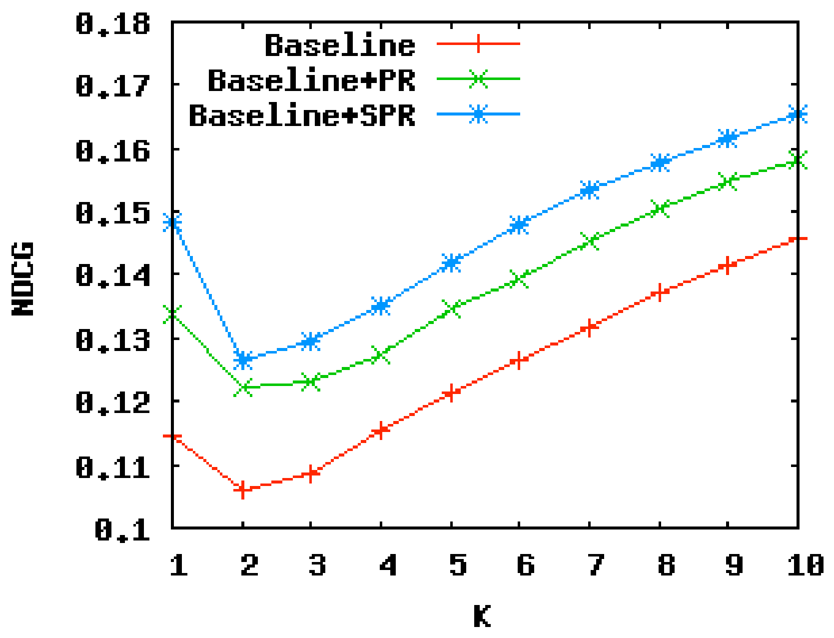
Step 3: ***Output:***

P^*: the converged SocialPageRank score.

Experimental Results

- Using SocialPageRank significantly improves both MAP and NDCG mesures:





Exploring Folksonomy for Personalized Search

Shengliang Xu*
Shanghai Jiao Tong University
Shanghai, 200240, China
slxu@apex.sjtu.edu.cn

Shenghua Bao*
Shanghai Jiao Tong University
Shanghai, 200240, China
shhbao@apex.sjtu.edu.cn

Ben Fei
IBM China Research Lab
Beijing, 100094, China
feiben@cn.ibm.com

Zhong Su
IBM China Research Lab
Beijing, 100094, China
suzhong@cn.ibm.com

Yong Yu
Shanghai Jiao Tong University
Shanghai, 200240, China
yyu@apex.sjtu.edu.cn

- Observation: social annotations characterize well topics of pages and interests of users
- Rank query results for query q, page p, user u as follows:

$$r(u, q, p) = \gamma \cdot r_{term}(q, p) + (1 - \gamma) \cdot r_{topic}(u, p)$$

- Compute $r_{topic}(u,p)$ as cosine similarity between annotations of u and annotations of p

Experimental Results

Data Set	Num. Users	Max. Tags	Min. Tags	Avg. Tags	Max. Pages	Min. Pages	Avg. Pages
Delicious	9813	2055	1	56.04	1790	1	40.35
Dogear	5192	2288	1	47.43	4578	1	46.78
DEL.gt500	31	1133	74	464.42	1790	506	727.55
DEL.80-100	100	456	2	107.51	100	80	88.43
DEL.5-10	100	64	1	18.53	10	5	7.44
DOG.gt500	92	2147	42	543.87	4578	500	999.04
DOG.80-100	85	295	9	126.96	100	80	89.32
DOG.5-10	100	41	2	16.11	10	5	6.99

- Observed 75%-250% improvement in MAP for all datasets
- Improvement is larger for the datasets where users who own less bookmarks, because typically their annotations are semantically richer

Summary

- Social Annotations (tags) can help improve search quality in the Enterprise
- While they can be directly used as features for the ranking function, exploiting their collaborative properties helps to further improve search quality
- Annotations can also be used to infer users' interests and provide personalized search results

Users’ Browsing Traces

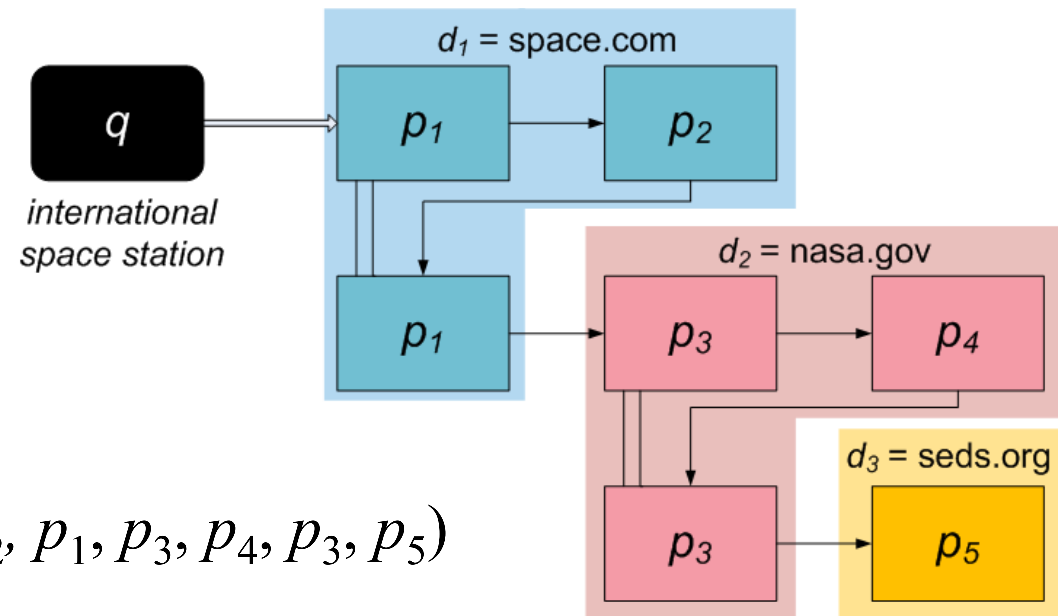
Mining the Search Trails of Surfing Crowds: Identifying Relevant Websites From User Activity

Mikhail Bilenko
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
mbilenko@microsoft.com

Ryen W. White
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
ryenw@microsoft.com

- Observe users' browsing behavior after entering a query and clicking on a search result
- Rank web sites for a new query based on how heavily they were browsed by users after entering same or similar queries
- Use it as a feature in search ranking algorithm

Search Trails



$q \rightarrow (p_1, p_2, p_1, p_3, p_4, p_3, p_5)$

- Start with a search engine query
- Continue until a terminating event
  - Another search
  - Visit to an unrelated site (social networks, webmail)
  - Timeout, browser homepage, browser closing

Using Search Trails for Ranking

- Approach 1: Adapt BM25 scoring function

$$w_{d_i,t_j} = QTF_{i,j} \cdot IQF_j =$$

$$= \frac{(\lambda+1)n(d_i,t_j)}{\lambda((1-\beta)+\beta\frac{n(d_i)}{\bar{n}(d_i)})+n(d_i,t_j)} \cdot \log\frac{N_d - n(t_j)+0.5}{n(t_j)+0.5}$$

Instead of term frequency in a document use sum of logs of dwell times on d_i from queries containing t_j

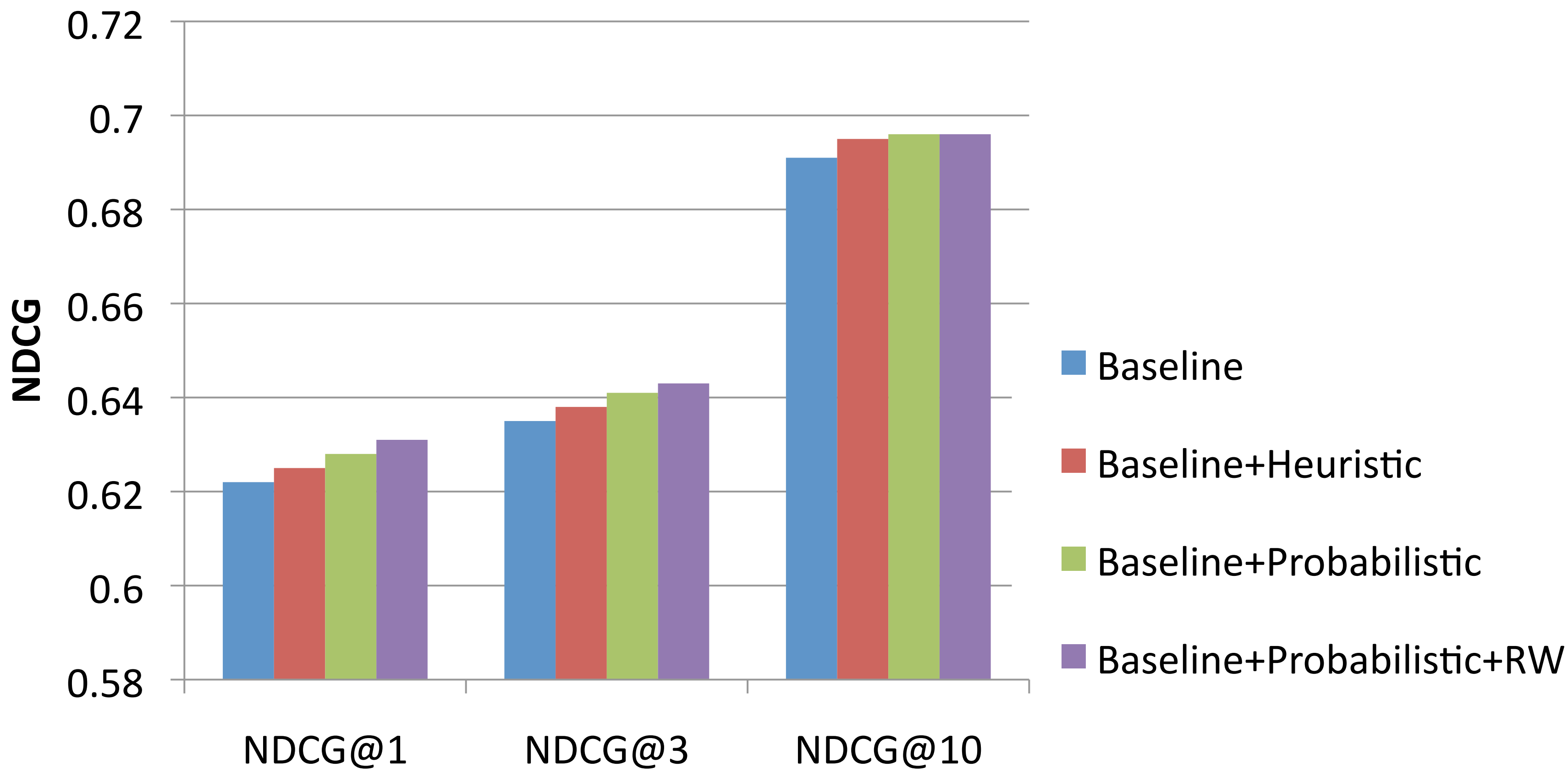
Instead inverse doc frequency use #docs for which queries leading to them include t_j

- Approach 2: Probabilistic model

$$Rel_P(d_i,\hat{q}) = p(d_i|\hat{q}) = \sum_{\hat{t}_j \in q} p(\hat{t}_j|\hat{q})p(d_i|\hat{t}_j)$$

Experimental Results

- Dataset: 140 million search trails; 33,150 queries with 5-point scale human judgments (site gets highest relevance score of all its pages)
- Add the web site rank feature to RankNet (Burges 05)
- Measure improvement in NDCG



Implicit Link Analysis for Small Web Search

Gui-Rong Xue[1] Hua-Jun Zeng[2] Zheng Chen[2] Wei-Ying Ma[2] Hong-Jiang Zhang[2] Chao-Jun Lu[1]

[1]Computer Science and Engineering
Shanghai Jiao-Tong University
Shanghai 200030, P.R.China
grxue@sjtu.edu.cn, cj-lu@cs.sjtu.edu.cn

[2]Microsoft Research Asia
5F, Sigma Center, 49 Zhichun Road
Beijing 100080, P.R.China
{i-hjzeng, zhengc, wyma, hjzhang}@microsoft.com

- Use all users' browsing traces to infer "implicit links" between pairs of web pages
- Intuitively, there is an implicit link between two pages if they are visited together on many browsing paths
- Construct a graph with pages as nodes and implicit links as edges and use it to calculate PageRank

Implicit Link Generation

- Use gliding window to move over each browsing path generating all ordered pairs of pages and counting occurrence of each pair

$$(w_{i1},\ w_{i2},\ w_{i3},\ \ldots,\ w_{ik})$$

$$(i1,\ i2),\ (i1, i3),\ \ldots,\ (i1,\ ik),\ (i2, i3),\ \ldots,\ (i2,\ ik),\ \ldots$$

- Select pairs which have frequency > *t* as implicit links

Using Implicit Links in Ranking

- Calculate PageRank based on the web graph with implicit links
- Combine PageRank and content-based similarity using a weighted linear combination
- Approach 1: use raw scores

$$Score(w) = \alpha Sim + (1- \alpha)\ PR \quad (\alpha \in [0, 1])$$

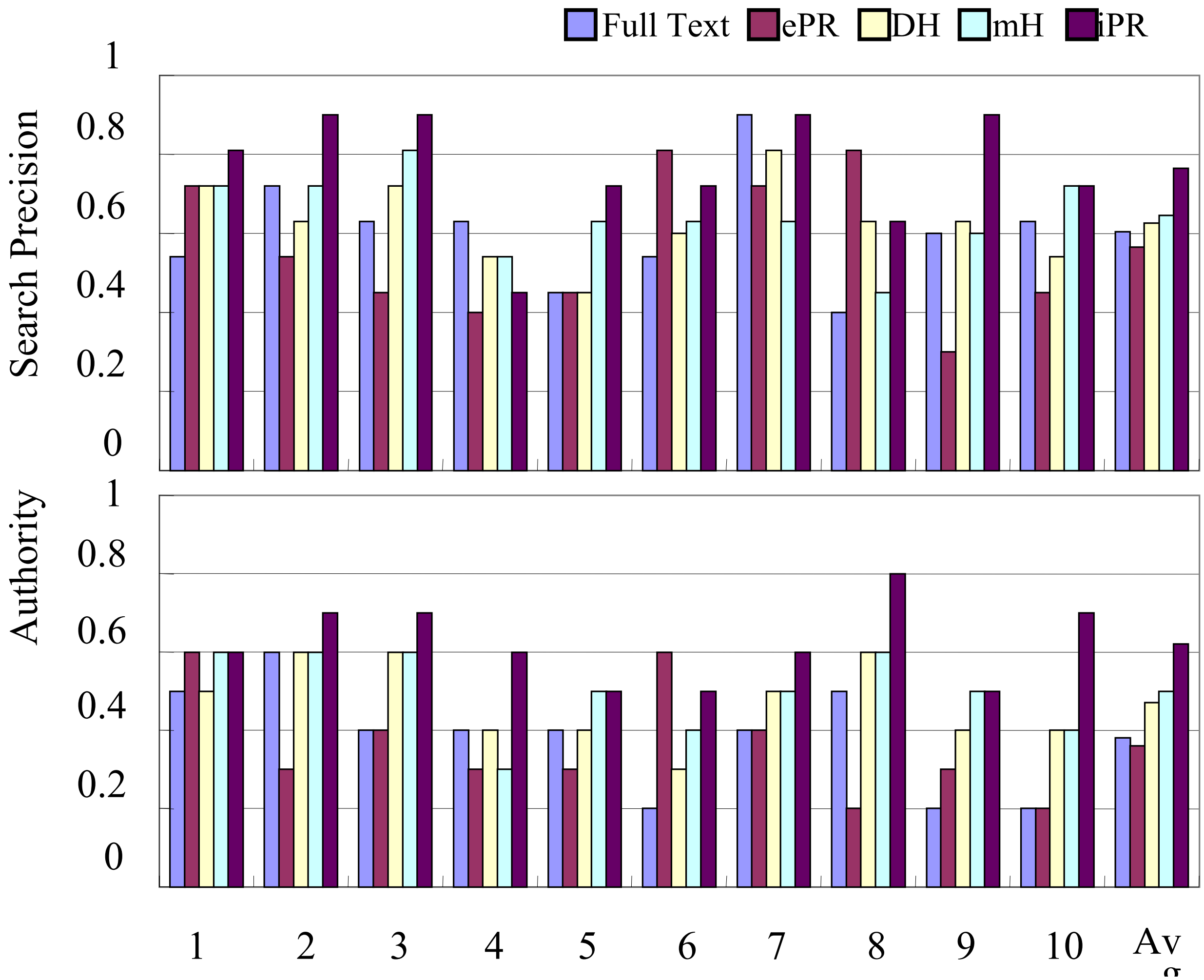
- Approach 2: use ranks instead of scores

$$Score(w) = \alpha O_{Sim} + (1- \alpha) O_{PR} \quad (\alpha \in [0, 1])$$

Experimental Results

- Dataset: 4-months logs from www.cs.berkeley.edu (300,000 traces; 170,000 pages; 60,000 users)
- 216,748 explicit links; 336,812 implicit links (11% are common to both sets)
- 10 queries; volunteers identify relevant pages and 10 most authoritative pages for each query out of top 30 results
- Measure “Precision @ 30” and “Authority @ 10”

Experimental Results



Summary

- User browsing traces can be collected easily in the Enterprise
- Two types of traces:
  - Traces starting from search engine queries
  - Arbitrary traces
- Traces are very useful for calculating authoritativeness of web pages and web sites, and can be successfully used to improve search ranking

Short-term User Context
and Eye-tracking based Feedback

Context-Sensitive Information Retrieval Using Implicit Feedback

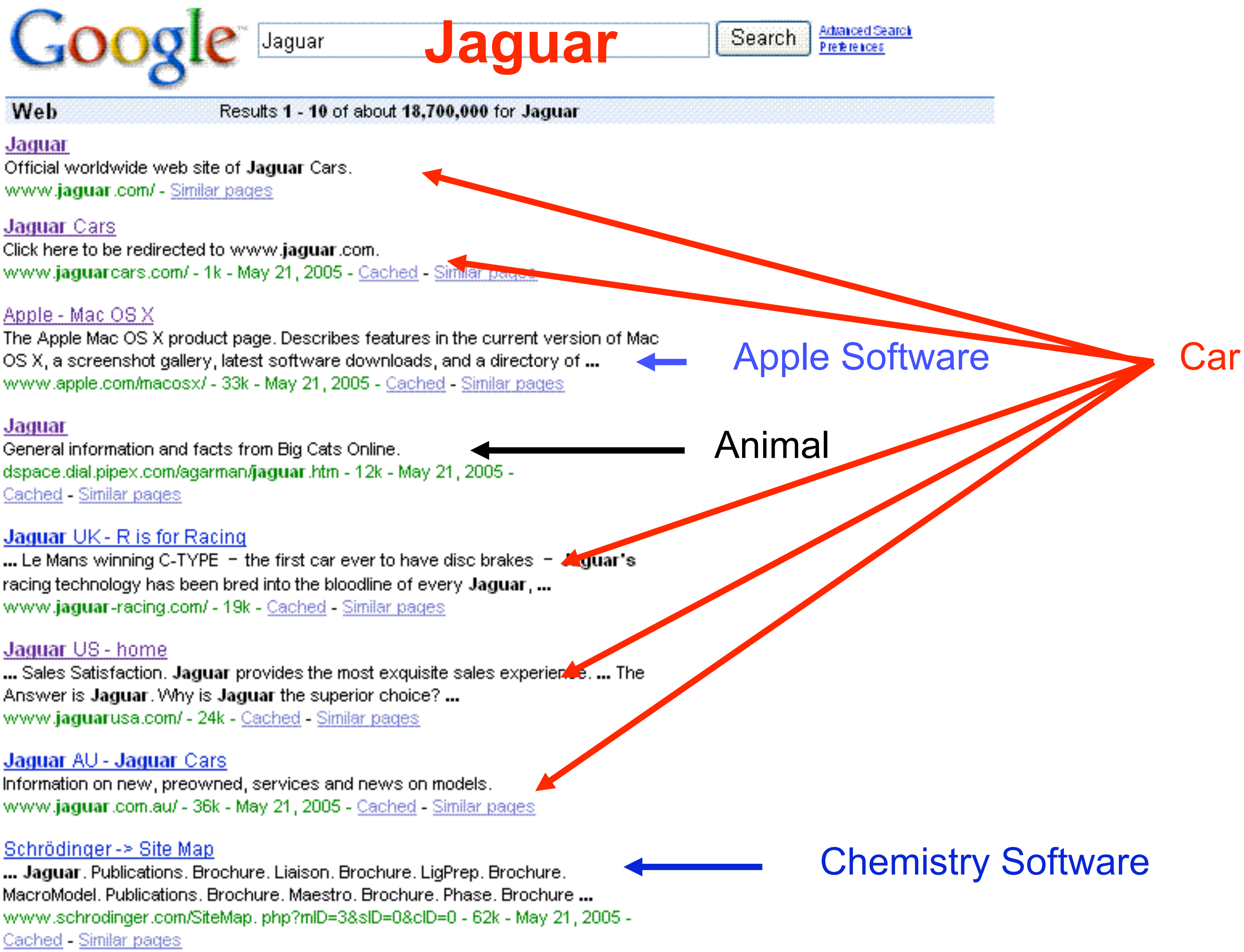
Xuehua Shen
Department of Computer Science
University of Illinois at Urbana-Champaign

Bin Tan
Department of Computer Science
University of Illinois at Urbana-Champaign

ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign

- Two types of user context information:
  - Short-term context
  - Long-term context
- Long-term context:
  - User's topics of interest, department and position, accumulated query history, desktop context, etc.
- Short-term context:
  - Queries and clicks in the same session, the text user has read in the past 5 min, etc.

Problem of Context-Independent Search



Putting Search in Context

Apple software

Query History

Clickthrough

Other Context Info:

- **Dwelling time**
- **Mouse movement**
- **Hobby**
- **…**

Short-term Contexts

- Will look at 2 types of short-term contexts:
  - *Session Query History*: preceding queries issued by the same user in the current session
  - *Session Clicked Summary*: concatenation of the displayed text about the clicked urls in the current session
- Will use language modeling framework to incorporate the above data into the ranking function

Using Short-term Contexts for Ranking

- Basic Retrieval Model:
  - For each document D build a unigram language model θ_D, specifying $p(\omega|\theta_D)$
  - Given a query Q, build a query language model θ_Q, specifying $p(\omega|\theta_Q)$
  - Rank the documents according to the KL divergence of the two models:

$$D(\theta_Q \,||\, \theta_D) = \sum_{\omega} P(\omega \,|\, \theta_Q) \log \frac{P(\omega \,|\, \theta_Q)}{P(\omega \,|\, \theta_D)}$$

- Assuming user already issued k-1 queries $Q_1,..,Q_{k-1}$, want to estimate the "context query model" θ_k specifying $p(\omega|\theta_k)$ for the current query Q_k to use instead of θ_Q

Using Short-term Contexts for Ranking

- Fixed Coefficient Interpolation:

$$
\begin{aligned}
p(w|Q_i) &= \frac{c(w,Q_i)}{|Q_i|} \\
p(w|H_Q) &= \frac{1}{k-1}\sum_{i=1}^{i=k-1} p(w|Q_i) \\
p(w|C_i) &= \frac{c(w,C_i)}{|C_i|} \\
p(w|H_C) &= \frac{1}{k-1}\sum_{i=1}^{i=k-1} p(w|C_i) \\
p(w|H) &= \beta p(w|H_C) + (1-\beta)p(w|H_Q) \\
p(w|\theta_k) &= \alpha p(w|Q_k) + (1-\alpha)p(w|H)
\end{aligned}
$$

Query history model

Click summary model

$$p(w|\theta_k) = \alpha p(w|Q_k) + (1-\alpha)[\beta p(w|H_C) + (1-\beta)p(w|H_Q)]$$

Using Short-term Contexts for Ranking

- Problem with Fixed Coefficient Interpolation is that the coefficients are the same for all queries. Want to trust the current query more if it is longer and less if it is shorter
- Bayesian Interpolation:

$$p(w|\theta_k) = \frac{c(w, Q_k) + \mu p(w|H_Q) + \nu p(w|H_C)}{|Q_k| + \mu + \nu}$$

$$= \frac{|Q_k|}{|Q_k| + \mu + \nu} p(w|Q_k) + \frac{\mu + \nu}{|Q_k| + \mu + \nu} [\frac{\mu}{\mu + \nu} p(w|H_Q) + \frac{\nu}{\mu + \nu} p(w|H_C)]$$

Coefficients depend on the query length

Experimental Results

- Dataset: TREC Associated Press set of news articles (~250,000 articles)
- Select 30 most difficult topics, have volunteers issue 4 queries for each topic and record query reformulation and clickthrough information
- Measure MAP and Precision@20

Experimental Results

- Results show that incorporating contextual information significantly improves the results

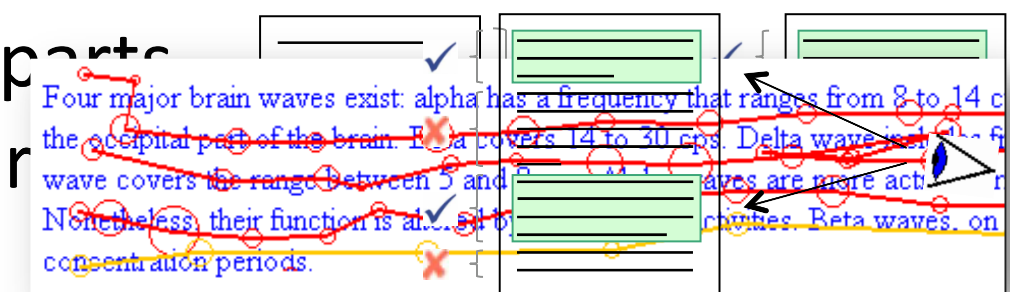
Query	FixInt ($\alpha = 0.1, \beta = 1.0$)		BayesInt ($\mu = 0.2, \nu = 5.0$)	
	MAP	pr@20docs	MAP	pr@20docs
q_1	0.0095	0.0317	0.0095	0.0317
q_2	0.0312	0.1150	0.0312	0.1150
$q_2 + H_Q + H_C$	0.0324	0.1117	0.0345	0.1117
Improve.	3.8%	-2.9%	10.6%	-2.9%
q_3	0.0421	0.1483	0.0421	0.1483
$q_3 + H_Q + H_C$	0.0726	0.1967	0.0816	0.2067
Improve	72.4%	32.6%	93.8%	39.4%
q_4	0.0536	0.1933	0.0536	0.1933
$q_4 + H_Q + H_C$	0.0891	0.2233	0.0955	0.2317
Improve	66.2%	15.5%	78.2%	19.9%

- Additional experiments showed that improvement is mostly due to using Session Clicked Summaries

Attention-Based Information Retrieval

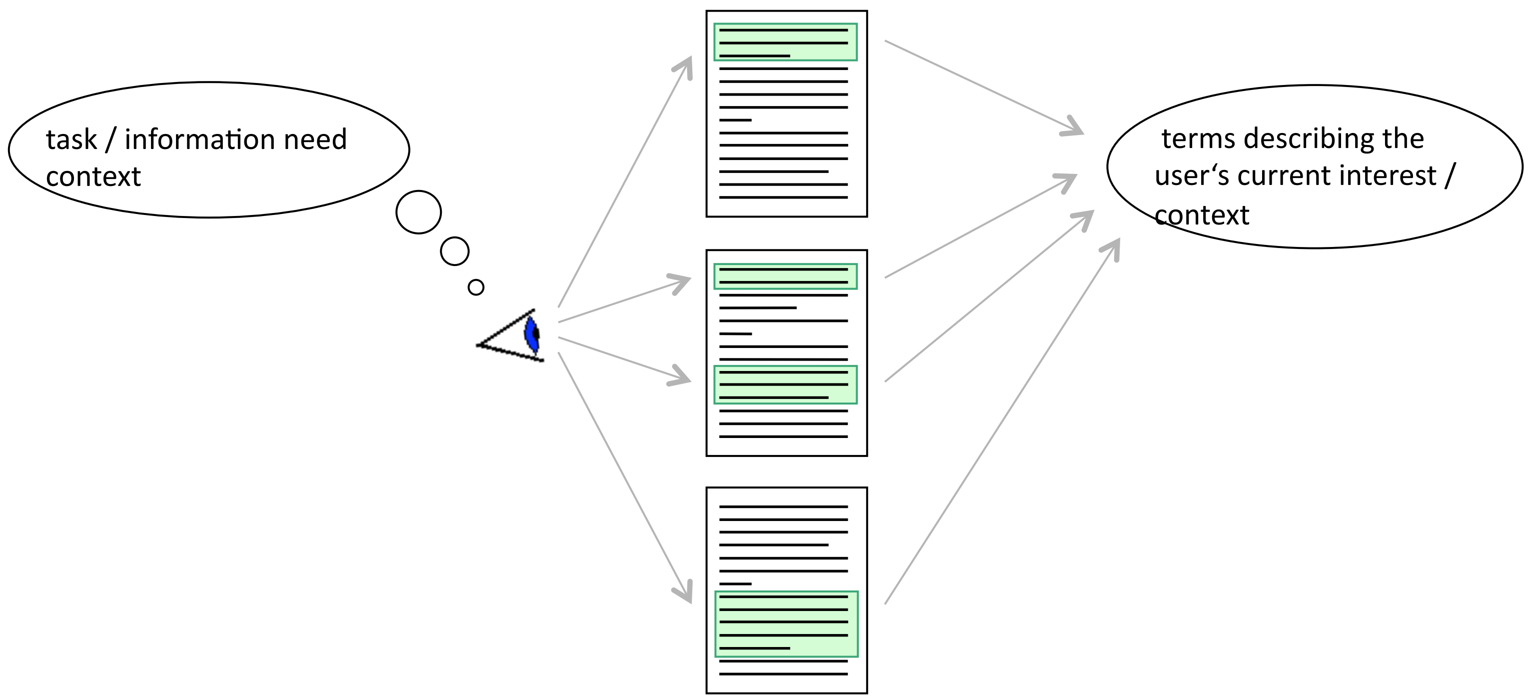
Georg Buscher
German Research Center for Artificial Intelligence (DFKI)
Kaiserslautern, Germany
georg.buscher@dfki.de

- Feedback on sub-document level should allow for better retrieval improvements
- Use an eye-tracker to automatically detect which portions of the displayed document were read or skimmed
- Determine which part
  of the document ar



How can we use this?

- For each page, can aggregate the “visual annotations” across the users of the enterprise
- Can construct a precise short-term user context



Summary

- Using short-term user context to improve search quality is a new and very promising direction of research
- Initial results show that it can be very effective
- Using eye tracking can help to improve the quality and increase the amount of the context data
- Many unexplored applications: on-the-fly reranking, abstract personalization, etc.

Interesting Problems and Promising Research Directions

- Applying the techniques we talked about to improve Enterprise Web search, extending them to better suit Enterprise environment
- Models for the Enterprise Web which take into account its complex structure and allow for expressing different usage data
- Personalization in the Enterprise Web search (usage data + employee personal info)
- Using context (recent history + desktop info) to improve Enterprise Web search

References

- [Fagin 03] Fagin. R., Kumar, R., McCurley, K.S., Novak, J., Sivakumar, D., Tomlin, J.A., Williamson, D.P. “Searching the Workplace Web”. WWW Conference, May 2003, Budapest, Hungary.
- [Hawking 04] Hawking, D. “Challenges in Enterprise Search”. ADC Conference, Dunedin, NZ.
- [Dmitriev 06] Dmitriev, P., Eiron, N., Fontoura, M., Shekita, E. “Using Annotation in Enterprise Search”. WWW Conference, May 2006, Edinburgh, Scotland.
- [Poblete 08] Poblete, B., Baeza-Yates, R. “Query-Sets: Using Implicit Feedback and Query Patterns to Organize Web Documents”. WWW Conference, April 2008, Beijing, China.
- [Joachims 02] Joachims, T. Optimizing Search Engines Using Clickthrough Data. KDD Conference, 2002.
- [Radlinski 05] Radlinski, F., Joachims, T. “Query Chains: learning to rank from implicit feedback”. KDD Conference, 2005, New York, USA.
- [Broder 00] Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., Wiener, J. “Graph Structure in the Web”. WWW Conference, 2000.
- [Dwork 01] Dwork, C., Kumar, R., Naor, M., Sivkumar, D. “Rank Aggregation Methods for the Web”. WWW Conference, 2001.
- [Shen 05] Shen, X., Tan, B., Zhai, C. “Context-Sensitive Information Retrieval Using Implicit Feedback”. SIGIR Conference, 2005.

References

- [Fagin 03-1] Fagin, R., Lotem, A., Naor, M. "Optimal Aggregation Algorithms for Middleware". Journal of Computer and Systems Sciences, 66:614-656, 2003.
- [Chirita 07] Chirita, P.-A., Costache, S., Handschuh, S., Nejdl, W. "P-TAG: Large Scale Generation of Personalized Annotation TAGs for the Web". WWW Conference, 2007.
- [Bao 07] Bao, S., Wu, X., Fei, B., Xue, G., Su, Z., Yu, Y. "Optimizing Web Search Using Social Annotations". WWW-Conference, 2007.
- [Xu 07] Xu, S., Bao, S., Cao, Y., Yu, Y. "Using Social Annotations to Improve Language Model for Information Retrieval". CIKM Conference, 2007.
- [Millen 06] Millen, D.R., Feinberg, J., Kerr, B. "Dogear: Social Bookmarking in the Enterprise". CHI Conference, 2006.
- [Bilenko 08] Bilenko, M., White, R.W. "Mining the Search Trails of Surfing Crowds: Identifying Relevant Web Sites from User Activity". WWW Conference, 2008.
- [Xue 03] Xue, G.-R., Zeng, H.-J., Chen, Z., Ma, W.-Y., Zhang, H.-J., Lu, C.-J. "Implicit Link Analysis for Small Web Search". SIGIR Conference, 2003.
- [Burges 05] Burges, C.J.C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.N. "Learning to Rank Using Gradient Descent". ICML Conference, 2005.
- [Buscher 07] Buscher, G. "Attention-Based Information Retrieval". Doctoral Concortium, SIGIR Conference, 2007.