

# Social Search and Discovery

## Using a Unified Approach

### Traditional search

Content: Documents  
 Task: Find relevant documents  
 Ranking: Traditional IR techniques, link analysis



Content: "Web 2.0" user-generated content (e.g., blogs), metadata (tags, comments, ratings), person-document relationships  
 Task: Find **relevant people**, tags  
 Ranking: Leverage the "wisdom of the crowd"

= **Social search**

### Technical Approach

- Search space is expanded to include relationships between objects
- Objects and relationships are indexed and used to compute most relevant search results
- Results come from expanded object space:



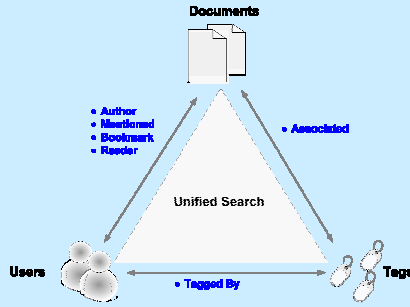
People who are highly related to the topic



Relevant documents, people, blogs, etc.



A tag cloud defining the topic of your query



### Data Sources



**Profiles** - IBM's internal **BluePages** application contains 475,000 profiles. BluePages serves 3.5 million searches per week and 1.5 million profile views per day.



**Communities** - IBM **Communities** hosts 900 communities. IBM Forums contain 147,000 threads and 410,000 messages.



**Blogs** - IBM's **BlogCentral** hosts 27,300 weblogs (420 group blogs) with 62,000 entries, 60,000 comments, and 10,800 distinct tags.



**Bookmarks** - IBM's social-bookmarking system **Dogear** has 327,000 bookmarks from 8,511 users. One-third are intranet links and only 2.5% are private.

### Social Ranking

#### Ranking of documents:

- Traditional **text similarity** relevance
- User-contributed metadata** (e.g., tags) adds text to documents
- Document **static-score** based on its **popularity** (bookmarks, comments, etc.)

#### Ranking of people (and tags):

- A **person is related to a query** if related to documents matching the query
- The **score of the person** is a function of the relevance of the document
- Different **types of person-document relationships** get different weights, e.g., author has higher weight than tagger
- IEF** lowers rank of a person that matches every query, not just this one
- Personalization**: boost people you know

Implemented using an enhanced **faceted-search** engine.

### Enterprise Social Search

#### The user's query

Search took 0.21 seconds.  
 Found 53 results. Showing results 1-10:

- © COW - Web 2.0 Unified Search - W3 Intranet Search Engine COW  
 Haifa's lab web 2.0 unified search. In this demo you can perform unified search in IBM's social bookmarking site, Dogear, and blogging site, BlogCentral. The search engine will find documents, people and tags relevant to your query. The index is based on data from 11130 people, 301799 bookmarks of 193855 pages, and 53293 blog threads.  
 Bookmarked 128 times  
<http://lnx-trec.haifa.ibm.com/2802/>
- © Web 2.0 Unified Search  
 The search engine will find documents, people and tags relevant to your query. In this demo you can perform unified search in IBM's social bookmarking site, Dogear, and blogging site, BlogCentral. -<br />  
 Bookmarked 11 times  
<http://lnx-trec.haifa.ibm.com/2802/search>
- © Web 2.0 Unified Search - IBM W3 Intranet Search...  
 Site describing web2.0 Unified Search. A prototype from Haifa Research Lab.  
 Bookmarked 12 times  
<http://lnx-trec.haifa.ibm.com/2802/about.html>
- © IBM Web 2.0 Unified Search - cool! - Donna Jones, IBM 2.0 evangelist - BlogCentral  
 Blog entry by Donna Jones /Markham/IBM, with 1 comments  
 The Haifa lab's come out with Web 2.0 Unified Search, a nice, clean search engine for finding bookmarks, blog entries, and people. Here's what a search for Second Life looks like:  
 Bookmarked 1 time  
[http://blogs.tap.ibm.com/weblogs/donna.../entry/web\\_2\\_0\\_unified](http://blogs.tap.ibm.com/weblogs/donna.../entry/web_2_0_unified)
- © Web 2.0 Unified Search - Results  
 Bookmarked 1 time  
<http://lnx-trec.haifa.ibm.com/2802/search>
- © Web 2.0 Unified Search - Results Mss  
 Bookmarked 1 time  
<http://lnx-trec.haifa.ibm.com/2802/search2>
- © Cattali - Sandra Morri

**Related people** are people related to the result set: authors, commenters, and/or taggers of one or more documents in the result set. This is a ranked list.

**Related tags** are a "tag cloud" of tags associated with documents in the result set.

The result set includes **relevant documents** (web pages, blogs, person profiles). Ranking is affected by the volume of tags and comments associated with each document.

**Related people**  
 Terra Cogan/Minneapolis/IBM  
 Ruthi Cohn/Haifa/IBM  
 Danny Nave/Haifa/IBM  
 Charlie Novak/UK/IBM  
 Andrew Jones/Raleigh/IBM  
 John Kelly /Markham/IBM  
 Stefano Kerri/France/IBM  
 Roger Mehm/White Plains/IBM  
 Ron Na'aman/Haifa/IBM  
 Laura Little/Dallas/IBM

**Related tags**  
 © COW expertise-locator ibmsweet...  
 research-search-engine search...  
 searchmachines unified unified-search  
 unified\_search web2\_0\_unified\_search web20\_unified\_search

**Related communities**  
 Web 2.0 Search  
 Social Software  
 Lotus Connections Sales

**Narrow search by:**  
 Source  
 Blog (42)  
 Url (33)  
 Date  
 2006 (3)  
 2007 (23)  
 2008 (34)

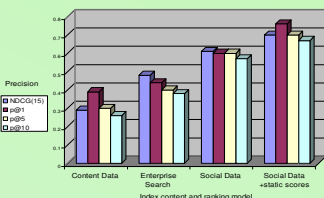
Additional **facets**. Search results can be narrowed by additional dimensions.

### Evaluation

#### Relevant documents

- Standard IR evaluation methodology:
- Picked 50 real users' queries
  - Executed them on several variants of the search engine
  - Relevance level judged by humans

Using social data and static scores based on social data **significantly** improves search accuracy:



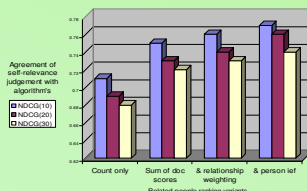
#### Related people

Large-scale **user study**: 600 respondents from 116 IBM locations in 38 countries.

For each query, found the "related people" and asked each to rank their relevance to topics (some believed relevant, some not).

Calculated agreement of each person's and algorithm's judgment (using NDCG).

High agreement shown, improved by more refined algorithms:



#### Related tags

Measure how retrieved "related tags" are related to queries.

Used **Normalized Google Distance**: searching (in Google) for supposedly-related terms, together and separately.

We showed that "related tags" are indeed related to queries, and improve with improved algorithm:

