Web Data Management: Powering the New Web

Raghu Ramakrishnan
Chief Scientist for Audience, Yahoo!
Research Fellow, Yahoo! Research
(On leave, Univ. of Wisconsin-Madison)
Further Reading

• Content, Metadata, and Behavioral Information: Directions for Yahoo! Research, *The Yahoo! Research Team*, IEEE Data Engineering Bulletin, Dec 2006 (Special Issue on Web-Scale Data, Systems, and Semantics)

• Systems, Communities, Community Systems, on the Web, *Community Systems Group at Yahoo! Research*, SIGMOD Record, Sept 2007

• Towards a PeopleWeb, R. Ramakrishnan and A. Tomkins, IEEE Computer, August 2007 (Special Issue on Web Search)
Outline

• Trends on the Web
  – Search, online communities, hosted systems
• Community Information Management
• Hosted Data Management
Trends in Search
“seafood san francisco”

Category: restaurant  
Location: San Francisco

Reserve a table for two tonight at SF’s best Sushi Bar and get a free sake, compliments of OpenTable!

Category: restaurant  
Location: San Francisco

Alamo Square Seafood Grill - (415) 440-2828  
803 Fillmore St, San Francisco, CA - 0.93mi - map  
Category: restaurant  
Location: San Francisco
Supplying Structured Search Content

• Semantic Web?
• Unleash community computing—PeopleWeb!
• Three ways to create semantically rich summaries that address the user’s information needs:
  – Editorial, Extraction, UGC

Challenge: Design social interactions that lead to creation and maintenance of high-quality structured content
Evolution of Online Communities
Rate of content creation

• Estimated growth of content
  – Published content from traditional sources: 3-4 Gb/day
  – Professional web content: ~2 Gb/day
  – User-generated content: 8-10 Gb/day
  – Private text content: ~3 Tb/day (200x more)
  – Upper bound on typed content: ~700 Tb/day
Metadata

• Estimated growth of metadata
  – Anchortext: 100Mb/day
  – Tags: 40Mb/day
  – Pageviews: 100-200Gb/day
  – Reviews: Around 10Mb/day
  – Ratings: <small>

This is in spite of the fact that interactions on the web are currently limited by the fact that each site is essentially a silo
PeopleWeb: Site-Centric  →  People-Centric

- **Common web-wide id for objects (incl. users)**
  - Even common attributes? (e.g., *pixels* for camera objects)
  - **Recent announcement: Yahoo! supports OpenID**

- As users move across sites, their personas and social networks will be carried along

- **Increased semantics on the web through community activity**
  (another path to the goals of the Semantic Web)
  (Towards a PeopleWeb, Ramakrishnan & Tomkins, IEEE Computer, August 2007)
Facebook Apps, Open Social

• Web site provides canvas
  – Third party apps can paint on this canvas
  – “Paint” comes from data on and off-network
    • Via APIs that each site chooses to expose

What is the core asset of a web portal?

• What are the computational implications?
  – App hosting and caching
  – Dynamic, personalized content
  – Searching over “spaghetti” information threads
Tags / jaguar / clusters

- car, cars, auto, etype, automobile, classic, vintage, autoshow, red, show
  See more in this cluster...

- zoo, animal, cat, animals, bigcat, seattle, woodlandparkzoo, sleep, edinburgh, caged
  See more in this cluster...

- guitar, fender
  See more in this cluster...

- aircraft, raf
  See more in this cluster...

These are the most recent photos tagged with jaguar. See more...
“In newsgroups, conversations disappear and you have to ask the same question over and over again. The thing that makes the real difference is the ability for customers to collaborate and have information be persistent. That’s how we found QUIQ. It’s exactly the philosophy we’re looking for.”

“Tech support people can’t keep up with generating content and are not experts on how to effectively utilize the product ... Mass Collaboration is the next step in Customer Service.”

– Steve Young, VP of Customer Care, Compaq
How It Works

- Customer
- Knowledge Base
- Self Service
- Answer
- Support Agent
- Partner Experts
- Customer Champions
- Employees

Question

Answer added to power self service
Timely Answers

77% of answers provided within 24h

- 40% (2,057) Answers provided in 3h
- 65% (3,247) Answers provided in 12h
- 77% (3,862) Answers provided in 24h
- 86% (4,328) Answers provided in 48h

- 74% answered

- No effort to answer each question
- No added experts
- No monetary incentives for enthusiasts
Power of Knowledge Creation

- 80% Self-Service *)
- 5-10% Agent Cases

*) Averages from QUIQ implementations
Users who on average provide only 2 answers provide 50% of all answers.
Interesting Problems

• Question categorization
• Detecting undesirable questions & answers
• Identifying “trolls”
• Ranking results in Answers search
• Finding related questions
• Estimating question & answer quality

(Byron Dom: SIGIR talk)
Software as a Service: Hosted Applications Come of Age
The Web: A Universal Bus

- People to people
  - Social networks
- People to apps/data
  - Email
- Apps to Apps/data
  - Web services, mash-ups
A Yahoo! Mail Example

• No. 1 web mail service in the world
  • Based on ComScore & Media Metrix
    – More than 227 million global users
    – Billions of inbound messages per day
    – Petabytes of data
• Search is a key for future growth
  – Basic search across header/body/attachments
  – Global support (21 languages)
Search Views

User can change "View" of current results set when searching.

Shows all Photos and Attachments in Mailbox.
Photo View turns the user's mailbox into a Photo album

Ability to quickly save one or multiple photos to the desktop

Hovering over subject provides additional information: filename, sender, date, etc.

Clicking photo thumbnails takes user to high resolution photo

Refinement Options still apply to Photo View

(Courtesy: Raymie Stata)
The Bigger Picture

• Software-as-a-service
  – E.g., Salesforce.com

• Cloud Computing
  – IBM, Microsoft, Google, Yahoo are all investing heavily in “cloud computing”
    • Grid Computing—e.g., Condor: Well-established, CPU-centric
    • Hosted “OLAP”—e.g., Map-Reduce, Hadoop
    • Hosted “OLTP”—e.g., Amazon’s Dynamo, S3, EC2, SimpleDB; Yahoo’s Sherpa suite

• Web application development
  – Ning, Ruby-on-rails
An Example Web App

Heavy use of simple database operations

Updates

uploads

tags

as “flower”

Queries

» Your Photos

» Photos tagged as “flower”

» Friend activity

Sonja uploaded
Brandon tagged a photo
The Problem

What does it take to build the next big app?

- flickr
- Facebook
- Digg
- del.icio.us
A Case for Hosted Infrastructure

• What does it take to get the Next Great Thing off the ground?
  • **Now:**
    – Set up multiple replicas of a clustered data store
    – Set up a system for indexing
    – Set up a system for caching
    – Set up auxiliary DBMS instances for reporting, etc.
    – Set up the feeds and messaging between them
    – Write the application logic
    – Fairly complex system at first line of new code
  
• **Our vision:**
  – Write the application logic
  – Use a hosted infrastructure to store and query your data
  
  – Or, as Joshua Shachter puts it: “The next cool thing shouldn’t take a team of 30, it should be three guys, PHP and a long weekend”
Why Hosted?

- No maintenance worries for application
- Single ops team
- Resource sharing leads to savings
Web Data Management: Community-Driven Information Extraction and Management
Better Search via Information Extraction

- Extract, then exploit, structured data from raw text:

For years, Microsoft Corporation CEO Bill Gates was against open source. But today he appears to have changed his mind. "We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying...

<table>
<thead>
<tr>
<th>Name</th>
<th>Title</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bill Gates</td>
<td>CEO</td>
<td>Microsoft</td>
</tr>
<tr>
<td>Bill Veghte</td>
<td>VP</td>
<td>Microsoft</td>
</tr>
<tr>
<td>Richard Stallman</td>
<td>Founder</td>
<td>Free Soft..</td>
</tr>
</tbody>
</table>

Select Name
From PEOPLE
Where Organization = ‘Microsoft’

Bill Gates
Bill Veghte

(from Cohen’s IE tutorial, 2003)
Community Information Management (CIM)

- Many real-life communities have a Web presence
  - Database researchers, movie fans, stock traders
- Each community = many data sources + people
- Members want to query and track at a semantic level:
  - Any interesting connection between researchers X and Y?
  - List all courses that cite this paper
  - Find all citations of this paper in the past one week on the Web
  - What is new in the past 24 hours in the database community?
  - Which faculty candidates are interviewing this year, where?
DBLife

- Integrated information about a (focused) real-world community
- Collaboratively built and maintained by the community
- Semantic web via extraction & community
DBLife

- Faculty: AnHai Doan & Raghu Ramakrishnan
- Students: P. DeRose, W. Shen, F. Chen, R. McCann, Y. Lee, M. Sayyadian
- Prototype system up and running since early 2005
- Plan to release a public version of the system in Spring 2007
- 1164 sources, crawled daily, 11000+ pages / day
- 160+ MB, 121400+ people mentions, 5600+ persons
- See DE overview article, CIDR 2007 demo
DBLife Papers


DBLife

Data Sources

Click on the + to show details, and the - to hide them again.

+ Colloquia
+ Conference homepage
+ Database group page
+ Dbworld
+ Department homepage
+ Event
+ Project page

Crawled daily, 11000+ pages = 160+ MB / day
Entity Extraction and Resolution

Raghu Ramakrishnan

co-authors = A. Doan, Divesh Srivastava,
Research Resulting ER Graph

“Proactive Re-optimization

Shivnath Babu

David DeWitt

Pedro Bizarro

Jennifer Widom

SIGMOD 2005

advise

coauthor

write

write

coauthor

advise

PC-Chair

PC-member
Challenges

• Extraction
  – Domain-level vs. site-level extraction “templates”
    • Compositional, customizable approach to extraction planning
  – Blending extraction with other sources (feeds, wiki-style user edits)

• Maintenance of extracted information
  – Managing information Extraction
  – Incremental maintenance of “extracted views” at large scales
  – Mass Collaboration—community-based maintenance

• Exploitation
  – Search/query over extracted structures in a community
  – Search across communities—Semantic Web through the back door!
  – Detect interesting events and changes
Example Entity Resolution Plans

\(d_1\): Gravano’s Homepage

L. Gravano, K. Ross.
Text Databases. SIGMOD 03
L. Gravano, J. Sanz.
Packet Routing. SPAA 91

\(d_2\): Columbia DB Group Page

Members
L. Gravano
K. Ross
J. Zhou

L. Gravano, J. Zhou.
Text Retrieval. VLDB 04

\(d_3\): DBLP

Luis Gravano, Kenneth Ross.
Digital Libraries. SIGMOD 04
Luis Gravano, Jingren Zhou.
Fuzzy Matching. VLDB 01
Luis Gravano, Jorge Sanz.
Packet Routing. SPAA 91

\(d_4\): Chen Li’s Homepage

C. Li.
Machine Learning. AAAI 04
C. Li, A. Tung.
Entity Matching. KDD 03

\(s_0\) matcher: Two mentions match if they share the same name.

\(s_1\) matcher: Two mentions match if they share the same name and at least one co-author name.
Continuous Entity Resolution

- What if Entity/Link database is continuously updated to reflect changes in the real world? (E.g., Web crawls of user home pages)
- Can use the fact that few pages are new (or have changed) between updates. Challenges:
  - How much belief in existing entities and links?
  - Efficient organization and indexing
    - Where there is no meaningful change, recognize this and minimize repeated work
Continuous ER and Event Detection

- The real world might have changed!
  - And we need to detect this by analyzing changes in extracted information
Mass Collaboration

- We want to leverage user feedback to improve the quality of extraction over time.
  - Maintaining an extracted “view” on a collection of documents over time is very costly; getting feedback from users can help.
  - In fact, distributing the maintenance task across a large group of users may be the best approach.
Mass Collaboration: A Simplified Example

Picture is removed if enough users vote “no”.
Mass Collaboration Meets Spam

Jeffrey F. Naughton swears that this is David J. DeWitt
A. Gupta, D. Smith, Text mining, SIGMOD-06

System extracted “Gupta, D” as a person name

System extracted “Gupta, D” using rules:

(R1) David Gupta is a person name
(R2) If “first-name last-name” is a person name, then “last-name, f” is also a person name.

Knowing this, system can potentially improve extraction accuracy.

(1) Discover corrective rules
(2) Find and fix other incorrect applications of R1 and R2

A general framework for incorporating feedback?
Collaborative Editing

- Users should be able to
  - Correct/add to the imported data
  - E.g., User imports a paper, system provides bib item

- Challenges
  - Incentives, reputation
  - Handling malicious/spam users
  - Ownership model
    - My home page vs. a citation that appears on it
  - Reconciliation
    - Extracted vs. manual input
    - Conflicting input from different users
Understanding Extracted Information

• The extraction process is riddled with errors
  – How should these errors be represented?
  – Individual annotators are black-boxes with an internal probability model and typically output only the probabilities. While composing annotators how should their combined uncertainty be modeled?

• Lots of work
  – Fuhr-Rollecke; Imielinski-Lipski; ProbView; Halpern; …
  – Recent: See March 2006 Data Engineering bulletin for special issue on probabilistic data management (includes Green-Tannen survey)
  – Tutorials: Dalvi-Suciu Sigmod 05, Halpern PODS 06
Understanding Extracted Information

• Users want to “drill down” on extracted data
  – We need to be able to explain the basis for an extracted piece of information when users “drill down”.
  – Many proof-tree based explanation systems built in deductive DB / LP /AI communities (Coral, LDL, EKS-V1, XSB, McGuinness, …)
  – Studied in context of provenance of integrated data (Buneman et al.; Stanford warehouse lineage, and more recently Trio)

• But … concisely explaining complex extractions (e.g., using statistical models, workflows, and reflecting uncertainty) is hard
  – And especially useful because users are likely to drill down when they are surprised or confused by extracted data (e.g., due to errors, uncertainty).
Provenance and Collaboration

• Provenance/lineage/explanation becomes a key issue if we want to leverage user feedback to improve the quality of extraction over time.
  – Explanations must be succinct, from end-user perspective—not from derivation perspective
  – Maintaining an extracted “view” on a collection of documents over time is very costly; getting feedback from users can help
  – In fact, distributing the maintenance task across a large group of users may be the best approach
A. Gupta, D. Smith, Text mining, SIGMOD-06

Incorrect. But why?

System extracted “Gupta, D” using these rules:

(R1) David Gupta is a person name
(R2) If “first-name last-name” is a person name, then “last-name, f” is also a person name.

Knowing this, system builder can potentially improve extraction accuracy.

One way to do that:
(S1) Detect a list of items
(S2) If A straddles two items in a list
⇒ A is not a person name
The Purple SOX Project

(SOCial eXtraction)

Application Layer

Shopping, Travel, Autos

Academic Portals (DBLife/Me Yahoo)

Enthusiast Platform

...and many others

Extraction Management System
(e.g Vertex, Societek)

Operator Library
Extraction Management System Goals: 50,000-foot view

- Support **Scalable** and **Social** Information Extraction

  - “**Scalable**” in the
    - number and kind of operators
    - number and kind of applications
    - data and processing (commodity hardware)

  - “**Social**” in that community input is solicited to
    - define the goals of the application—customizability
    - bootstrap information extraction with examples of various kinds
    - refine information extraction with quality feedback
Operator Library Goals: 10,000 foot view

- Provide high quality operators for common extraction tasks under realistic settings for a variety of domains
  - record extraction, entity deduping, relation extraction
  - different levels and types of supervision, transfer
    across domains, active/passive modes
  - academic extraction, local search, etc.

- Build core infrastructure for creation & deployment of PSOX operators
  - feature engine, evaluation engine
  - assembling existing resources

- Develop and evaluate fundamentally new methods over target domains
  - semi-supervised, active & transfer learning approaches
Web Data Management: Massively Distributed Hosted Systems
Two Key Subsystems

- **Serving system**
  - Takes queries and returns results
- **Content system**
  - Gathers input of various kinds (including crawling)
  - Generates the data sets used by serving system
- Both highly parallel

Goal: speedup. Hardware increments speed computations.

Goal: scaleup. Hardware increments support larger loads.
Data Serving Systems
The Big Picture: Sherpa Data Services

Applications

PNUTS Services
- Query planning and execution
- Index maintenance

Distributed infrastructure for tabular data
- Data partitioning
- Update consistency
- Replication

YCA: Authorization

YDHT FS
- Hash tables

YDOT FS
- Ordered tables

YMB
- Pub/sub messaging

Zookeeper
- Consistency service
Guiding Principles

Reliable and robust storage
- Replication for fault tolerance
- Predictable consistency guarantees

Simple to use
- Simple and elegant operations set
- Minimal client configuration
- Service-level authentication
- Flexible schemas

Highly Scalable / Performant
- Partitioning data over many machines
- Horizontal scaling at every level
- Data is local to its usage
- Predictable performance via quality of service levels
- Predicates evaluated on back end
- Cheaper consistency guarantees than full ACID

Multiple rich access methods
- Hash and ordered table types
- System-maintained secondary indexes
- Optimization for complex access patterns

Rapid provisioning of new storage
- Simple, automated cluster growth
- Cheap table creation
- Pay as you grow, grow big as you need

Operationally Cheap
- Automated failover
- Automated load balancing
- No single points of failure
- Hosted platform
Sherpa Components

• PNUTS servers
  – Query planning and execution
  – Index maintenance
  – Gather statistics to aid planning

• Routers
  – Layer of indirection to hide physical data placement
  – May be same machines as PNUTS servers

• Storage units
  – Physical storage and serving of data

• Tablet controller
  – Manages data placement
  – Detects and initiates recovery after failures

• YMB (Y! Message Broker)
  – Reliable wide-area messaging and replication
CREATE TABLE Parts (  
    ID VARCHAR,  
    StockNumber INT,  
    Status VARCHAR  
)
Sherpa Architecture

Clients

PNUTS API

PNUTS servers

YDHT/YDOT API

YMB

SU API

Storage units

Cluster 1

Cluster 2

Tablet controller

Tablet map

Load balancer

Server monitor

Routers

Server monitor
• Scalable distributed record store
  – Optimized for small reads and writes
  – Focus on ease of operations, multi-region redundancy, organic scalability
  – Storage as a service
  – YDOT: Ordered primary store complementing YDHT
• Shared storage service
  – Storage “feels” local
  – You get the QoS you pay for
• Minimize operational costs
  – Easy provisioning of machines
  – Focus on simple and correct components
• Reliable
  – Persistence for writes to an individual record
  – Automated redundancy and failover
• High performance
  – Fast reads/writes by primary key
• Massively scalable
  – Scale-out: add servers to scale
  – Minimize performance hotspots via automated load balancing
• Building block for more complex storage solutions
  – Reliable and scalable underlying record/block store
Ways to use YDHT

• As a primary store

• As a materialized view/cache

• As part of PNUTS!
Asynchronous replication
Basic consistency model

• Record lifecycle
  1. Record inserted with a given primary key
  2. Record’s non-primary key attributes updated
     • Primary key cannot be updated
  3. Record deleted
  4. Another record with the same primary key may subsequently be inserted

• What happens to a record with primary key “Brian”?

![Diagram showing record lifecycle and generations over time]
Data Analysis Systems
Data Analysis Platforms

- Exploratory analysis over massive data sets
  - Challenges: Analyze shared, evolving social networks of users, content, and interactions to learn models of individual preferences and characteristics; community structure and dynamics; and to develop robust frameworks for evolution of authority and trust; extracting and exploiting structure from web content …

- Examples:
  - Map-Reduce, Hadoop, PIG
PIG Overview

( SQL )

Pig

Hadoop

Kryptonite

user

Distributed file system + Map/Reduce layer

automatic rewrite + optimize

or

or

- 73 -
Example

Find queries for which the highest-pagerank page in the result set did not appear among the top 5 results.

<table>
<thead>
<tr>
<th>query</th>
<th>position</th>
<th>url</th>
</tr>
</thead>
<tbody>
<tr>
<td>“news”</td>
<td>1</td>
<td>cnn.com</td>
</tr>
<tr>
<td>“news”</td>
<td>2</td>
<td>bbc.com</td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“pig”</td>
<td>1</td>
<td>pigwheels.com</td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>url</th>
<th>pagerank</th>
</tr>
</thead>
<tbody>
<tr>
<td>bbc.com</td>
<td>0.9</td>
</tr>
<tr>
<td>cnn.com</td>
<td>0.9</td>
</tr>
<tr>
<td>pigwheels.com</td>
<td>0.3</td>
</tr>
<tr>
<td>…</td>
<td></td>
</tr>
</tbody>
</table>
Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method

- Efficient Evaluation Method
Summary: Web Meets DB—And Neither is Ready!

- Interpreting and integrating information: structured “web views”
  - From site-centric to people-centric Web
    - Models of trust, ownership, incentives?
  - The promise of Community Information Mgmt:
    - Extraction + mass collaboration = semantics
- Beyond search—Web as app-delivery channel
  - Customizable hosted apps! Desktop ➔ Web-top
    - Data-driven services, not DBMS software
  - Scalable serving of data/relationships
    - Multi-tenancy, QoS, auto-admin, performance