Data (and Links) on the Web

Alberto Mendelzon
University of Toronto

http://www.cs.toronto.edu/~mendel

Joint work with Gus Arocena, Attila Barta, George Mihaila, Tova Milo, Davood Rafiei, Greg Keast, Elaine Toms, Joan Cherry
Outline

• Data on the Web
  semistructured data: data models, query languages

• What about links?

• Two link-centric projects
  WebSQL/WebOQL: unstructured/semistructured data + links
  TOPIC: exploiting links to evaluate page reputations

• Future Work
Data on the Web


Excellent survey of semistructured data
Semistructured Data

60’s: Data in files, structure in application programs

70’s, 80’s: Data and structure (schema) in DBMS

90’s: Data on the Web, where is the schema?

“Schemaless”: HTML

“Self-Describing”: XML
Example: an XML document

<north-america>

<states>
    <state id = "s1">
        <sname> California </sname>
        <capital idref="c1" />
        <governor> Gray Davis </governor>
    </state>

    ...
</states>

<provinces>
    <province id = "p1">
        <pname> Ontario </pname>
        <capital idref="c2" />
        <premier> Mike Harris </premier>
    </province>

    ...
</provinces>
<cities>
  <city id = “c1”>
    <cname> Sacramento </cname>
    <state-of idref = “s1”>
  </city>

  <city id = “c2”>
    <cname> Toronto </cname>
    <pop> 2.5M </pop>
    <province-of idref = “p1”>
  </city>

  ...
</cities>
Graph Representation

states

provinces

sname

“California”

governor

“Davis” “Ontario” “Harris”

province

province

pname

“Toronto”

cname

“California”

cname

“Sacramento”

cname

“Ontario”

“Toronto”

cname

“2.5M”

cname

“Harris”

“Toronto”

cname

“Sacramento”

NA

cities

pop

“2.5M”
State of the Art

• Data Models
  Pioneering work: OEM, LORE/LOREL, UnQL
  Data models for XML: XML Schema, DOM, RDF

• Query Languages
  SS QL’s: LOREL, UnQL, ...
  XML QL’s: XML-QL, XSLT, XQL

• Indexing

• Storage Mappings
What about the links?

Entry for link in index of DOTW book:

• pp. 45-46: XLink and XPointer
• pp. 189: “If Web data follows the same patterns as Web documents, then we should expect links to become prevalent.”

*The Web is not just semistructured data: it’s autonomous distributed pieces of unstructured, semistructured, and structured data, interconnected by link*
Some link-aware projects

- Strudel (AT&T)
- Tiramisu (Washington)
- Araneus (Rome)
- AutoWeb (Milan)
- SQUEAL (MIT)
- COIR (NEC)
- FLORID (Freiburg)
- WebSQL/WebOQL (Toronto)
WebSQL: Unstructured data + links

• Integrate *Browsing & Searching*

• Data Model:

  *Document* (URL, title, type, length, text, modif)

  *Anchor* (base, label, href)

• Query Language: SQL + regexps

• Semantics:
  • Materialize a fragment of the database
  • Compute the answer on this fragment
Search Automation

• Find documents about Toronto that reside in servers in Canada

SELECT d.url, d.title
FROM Document d SUCH THAT d MENTIONS “Toronto”
WHERE d.url CONTAINS “.ca$”

• Find documents about WebSQL that point to U of T

DEFINE INDEX “HotBot”; 

SELECT d.url
FROM Document d SUCH THAT d MENTIONS “WebSQL”,
    Anchor a SUCH THAT base = d,
WHERE a.href CONTAINS “toronto.edu”
OR a.href CONTAINS “utoronto.ca”
Documents about “excursions” near WWW9 home page

```
SELECT d.url, d.title
FROM Document d
SUCH THAT “www9.org” (-| ->| ->|- | -|->->-| ) d
WHERE d.text CONTAINS “excursions”
```
Path Regular Expressions

• Alphabet (Link types)
  
  #>    interior link: same document
  ->    local link: same server
  =>    global link: different server
  =      null path

• Regexps Over Link Types
  
  -> | =>    path of length one, either local or global
  ->*      local path of any length
  =>->*    idem, but in other servers
  (->|=>)* the reachable portion of the Web
User-Defined Link Types

DEFINE LINK [next] AS label CONTAINS “Next”;

SELECT d.url 
FROM Document d 
SUCH THAT “http://the.starting.document” [next]* d, 
WHERE d.title CONTAINS “Canada”;

[Diagram showing a sequence of 'Next' links with ellipsis]
Example applications

- Indexing an On-line Manual
- Indexing Publication List
Index of Online Publications

• Need pairs <URL of .ps, Metadata>

**Internet**


```sql
SELECT a.href, a.label
FROM Anchor a
SUCH THAT base = “http://www.cs.utoronto.ca/~mendel/papers.html”
```
A (partial) list of publications


```
DEFINE CONTEXT BEGIN = <LI>, END = <LI>;

SELECT e.href, e.context
FROM Anchor e SUCH THAT
    base = "http://www.math.tau.ac.il/~milo/dept/papers.html"
WHERE e.href CONTAINS " .ps"
```
DEFINE LINK [here] AS label CONTAINS “here”
SELECT e.url, d.text
FROM Document d SUCH THAT
Programmatic Interface

```java
public static void main(String args[]) {
    String query = "SELECT x.url, x.title, x.length, x.date " + " FROM Document x SUCH THAT x MENTIONS\"Java\";";

    try{
        WebSQLServer eng = new WebSQLServer(query, new Mon());
        for (Enumeration e = eng.elements(); e.hasMoreElements(); ) {
            Vector tuple = (Vector) e.nextElement();
            for (int i = 0; i < eng.tupleSize; i++) {
                System.out.print(tuple.elementAt(i));
                System.out.print(" ");
            }
            System.out.println();
        }
    }catch(Exception e){System.out.println("Couldn’t create server.");}
}
```
WebOQL: semistructured data + links

- WebSQL: Web as graph of atomic objects
- WebOQL: Web as graph of structured objects

Query:
- the Web
- a single page
- a set of related pages

Restructure:
- HTML to HTML
- HTML to databases
- Databases to HTML
City Overview

- One of the most attractive aspects of our city is the variety of cultural activities. You can purchase tickets for several theatres from Theatres Online.

- All the hotels on the Web provide discounts to cyber-clients!

- If you are interested in live sports, then you must visit Sports Zone. You can also buy tickets from them.
Data Model

- Records as Labels on Arcs
- Internal and External Arcs
Tree operators

T

T'
Webs
Query: list elements containing “ticket”

[ Tag “UL” /
    select y
    from y in doc !’
    where y’.text ~ “ticket”]

[Tag: UL]

[Tag: LI]

[Label: Theatres Online,
  Url: http://www..., Base: http://www..., Text: One of the ...]

[Tag: XYZ,
  Text: This page contains ...
  Url: http://www..., Base: http://www..., Text: One of the ...]

[Tag: LI]

[Tag: XYZ, Text: If you are...]

[Tag: XYZ, Text: One of the ...]

[Label: Sports Zone,
  Url: http://www..., Base: http://www..., Text: Sports Zone ...]
Extracting CNN’s Headlines

```
select [Section:Y.text, Headline:z.text, Url:z’.url]
from X in “http://www.cnn.com” via ^*[text ~ “T O P”],
    Y in X!!!’ via ^*[tag = “blockquote”],
z in Y!’
```

- [ Section: U. S.,
  Headline: Part-time ...
  Url: http://www.cnn... ]

- [ Section: World,
  Headline: Fire in Toronto ...
  Url: http://www.cnn... ]

- [ Section: U. S.,
  Headline: Canadians win ...
  Url: http://www.cnn... ]
Restructuring the Result into HTML

```csharp
[Tag:“table”/
  [Tag:“th”, Text:“Section”] + [Tag:“th”, Text:“headline”] + [Tag:“th”, Text:“Url”]+
  select [Tag:”tr”/ [Tag:”td”, Text:Y.text] + [Tag:”td”, Text:z.text] +
  [Tag:”td”/ [Label:z’.url, Url:z’.url]]
]
from X in “http://www.cnn.com” via ^*[text ~ “T O P”],
  Y in X!!!’ via ^*[tag = “blockquote”], z in Y!’
]```

---

<table>
<thead>
<tr>
<th>Section</th>
<th>Headline</th>
<th>Url</th>
</tr>
</thead>
<tbody>
<tr>
<td>U. S.</td>
<td>Part-time</td>
<td><a href="http://www.cnn.com">http://www.cnn.com</a></td>
</tr>
</tbody>
</table>
Generating a new Web

Table = [previous query]

select [y'] as y.Text
from x in Table!!!, y in x

creates one page for each Section, with the Section name as URL
Easy to do in WebOQL

Extract all headings
Extract all images
Linearize page hierarchy
Flatten hierarchy into table
Create Web views
Extract pictures of faculty
SCAN
“http://www.cs.toronto.edu/DCS/People/Faculty/index.html”
USING
ANY
  <BODY>
    MANY
      <UL>
        {<LI> <A HREF = MemberPage> MemberName </A> </LI>}
      </UL>
  </BODY>
AND
  MemberPage
USING
  ...<IMG SRC = Jpg “.jpg$”>
GIVING
  <HTML>
    <TABLE>
      {<TR>
        <TD> text(MemberName) </TD>
        <TD> <IMG SRC = Jpg> </TD>
      </TR>}
    </TABLE>
  </HTML>
Generated WebOQL

[Tag:"html"/
 [Tag:"table"/
  select [Tag:"tr"/
   [Tag:"td"/[Text:MemberName.text]] +
   [Tag:"td"/[Src:Jpg.src, Tag:"img"]]
  ]
 from V__ is "http://www/DCS/People/Faculty/index.html",
  V_0 in V__!' via [Tag = "ul"] until true,
  V_1 in V_0',
  MemberName is V_1',
  MemberPage is MemberName,
  V_2 in browse(MemberPage.url)
   via ^*[Src ~ ".[jpg$" and Tag = "img"],
  Jpg is V_2&
  where V__!.Tag = "body" and V_1.Tag = "li" and
 MemberName.Tag = "a"
 ];
<table>
<thead>
<tr>
<th>T.S. Abdelrahman, MSc, PhD</th>
<th><img src="http://www.cs.toronto.edu/gifs/Faculty/tsa.jpg" alt="tsa.jpg" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>R.M. Baecker, MSc, PhD</td>
<td><img src="http://www.cs.toronto.edu/gifs/Faculty/rmb.jpg" alt="rmb.jpg" /></td>
</tr>
<tr>
<td>A. Bonner, MSc, PhD (Erin)</td>
<td>...</td>
</tr>
</tbody>
</table>
System Architecture

Interactive User

Application

API

WebOQL Engine

Wrapper Manager

Query / Web

URL / Tree

Wrapper

Wrapper

Wrapper

Wrapper

Wrapper

DBMS

File System

Index Server

LDAP

WWW
Computing Page Reputations

(Rafiei and Mendelzon, WWW9)

• How do we rank a large number of pages relevant to a query, so the good ones appear first? (search engine company’s problem)
• Given a page and a topic, how good is this page on this topic? (tenure committee’s problem)
• Given a page (or a site), what topics is this page good on? (webmaster’s problem)

Good means reputable, authoritative, well-known, up-to-date,...

Idea:

• Analyze links to compute Rank(p,t) = goodness of page p on t
Page Rank

(Brin and Page 1998, Google; Geller 1978 in bibliometrics)

A page is **good** if lots of **good** pages point to it.

**One level random walk model:**

At each step:

- with prob $p>0$ jump to a random page, or
- with prob $(1-p)$ follow a random link from the current page

Page Rank of page $p = \text{probability, in the limit, of hitting page } p$
Problems with PageRank

- topic-independent: a page may be good for one topic but not another

- good pages may not point to each other: BMW does not point to Mercedes
Hubs and Authorities

(Kleinberg, 1998)

Given a set of pages relevant to topic t:

A page is a good hub for t if it points to good authorities on t.

A page is a good authority on t if good hubs for t point to it.

Algorithm to find authorities on t:

• Issue the query t to a search engine

• Take the first N answers, add pages at distance 1

• Compute authorities for t within this set
A two-level random walk model

• with prob $d > 0$ jump to random page *that contains term t*

• with prob $(1-d)$ follow random link *forward/backward* from the current page, alternating directions

Pages accumulate

• forward visits

• backward visits
• $A(p,t) = \text{probability of a forward visit to page } p \text{ when searching for term } t = \textbf{Authority rank} \text{ of page } p \text{ on term } t$

• $H(p,t) = \text{probability of a backward visit to page } p \text{ when searching for term } t = \textbf{Hub rank} \text{ of page } p \text{ on term } t$

**Theorem** If $d > 0$, the two-level random walk has unique stationary probability distributions $A(p,t)$ and $H(p,t)$.

(Does this model Kleinberg’s algorithm?

*No*: See Lempel and Moran, WWW9, Borodin et al, WWW10)
Does Hubs & Authorities solve our ranking problems?

- Search engine problem: yes
- Tenure committee’s problem: maybe
- Webmaster’s problem: no
Inverting H&A computation

- Topic → H & A → Pages
- Page → ? → Topics
Two Solutions

- *Global solution*: a large crawl of the web is available. Find authorities on each term $t$

- *Local solution*: approximate the global solution by starting with some set of pages and the terms that appear in them, and iteratively expanding this set
Global Solution (bottom up)

For every page p and term t

\[ A(p, t) = H(p, t) = \frac{1}{2N_t}, \text{ if t appears in p} \]

\[ A(p, t) = H(p, t) = 0 \text{ otherwise.} \]

While changes occur

\[ A(p, t) = (1 - d) \sum_{q \rightarrow p} \frac{H(q, t)}{Out(q)} + \begin{cases} \frac{d}{2N_t} & \text{if t appears in page p;} \\ 0 & \text{otherwise} \end{cases} \]

\[ H(p, t) = (1 - d) \sum_{p \rightarrow q} \frac{A(q, t)}{In(q)} + \begin{cases} \frac{d}{2N_t} & \text{if t appears in page p;} \\ 0 & \text{otherwise} \end{cases} \]
Local Solution: (top down)

Set of pages:

Set of terms: all terms $t$ that appear in $p$ or some of the $q_i$'s
Local algorithm (Using the one-level model for simplicity)

\[ R(p, t) = \frac{d}{N_t} \]

For \( i = 1, 2, \ldots, k \)

For each path \( q_1 \rightarrow q_2 \rightarrow \ldots \rightarrow q_i \rightarrow p \),

For each term \( t \) in page \( q_1 \)

\[ R(p, t) = R(p, t) + \left( \frac{1 - d}{i} \right)^i \frac{d}{N_t} \prod_{j=1}^{i} \text{Out}(q_j) \]
TOPIC: Approximating the local algorithm

• Given page p
  • Find 500 pages q that link to p (using Altavista)
  • From each q “snippet,” extract all terms t
  • Remove internal links and duplicate snippets
  • Remove stop words and rare terms
  • Apply the local algorithm with $d = 0.10$, $k = 1$, $\text{Out}(q) = 7.2$
Penetration and Focus

(Mendelzon and Rafiei, IEEE Bull. Data Eng., 2000)

For d and Out(q) constant, the local algorithm reduces to

\[ R(p,t) \sim \frac{I(p,t)}{N_t} \]

where \( I(p,t) \) = number of pages that contain t and point to p, \( N_t \) = number of pages that contain t

\[ R(p,t) = \text{fraction of pages on t that point to p: penetration of page p on topic t} \]

Can also define

\[ F(p,t) = \text{fraction of pages pointing to p that are about p: focus of page p on topic t} \]
Example: authorities on (+censorship +net)

- **www.eff.org**
  
  Anti-censorship, Join the Blue Ribbon, Blue Ribbon Campaign, Electronic Frontier Foundation

- **www.cdt.org**
  
  Center for Democracy and Technology, Communications Decency Act, Censorship, Free Speech, Blue Ribbon

- **www.aclu.org**
  
  ACLU, American Civil Liberties Union, Communications Decency Act
Example: Personal Home Pages

- www.w3.org/People/Berners-Lee
  History of the Internet, Tim Berners-Lee, Internet History, W3C

- www-db.stanford.edu/~ullman
  Jeffrey D. Ullman, Database Systems, Data Mining, Programming Languages
Examples: Institutional Home Pages

- www.db-stanford.edu:
  Database research, data warehousing, database systems, data mining, Stanford

- www.almaden.ibm.com:
  IBM Almaden, search engines, data mining, microscopy, visualization
Examples: Canadian CS Departments

- **www.cs.toronto.edu:**
  Women hockey, computer vision, department of Computer Science, University of Toronto, archive, Russian

- **www.cs.ualberta.ca:**
  University of Alberta, virtual reality, chess, language, artificial

- **www.cs.ubc.ca:**
  confocal, periodic table, anime, Computer Science, manga, Mathematics
TOPIC as search engine ranking method

• Given query t, rank answer pages p by $R(p,t)$

• Experiment: 467 queries obtained from major search engine company. For each query, rerank top 100 engine hits by TOPIC ranking

• Evaluation with human subjects in progress by FIS (Keast et al, ACM/IEEE DL Conf., 2001)
**class action**

1. [01] securities.stanford.edu/ 0.023049
2. [43] members.aol.com/CClass450/0.001814
3. [24] www.classactionlitigation.com/ 0.001374
5. [16] www.acer.com/aac/main/settle.htm 0.000843
6. [09] www.masoniteclaims.com/ 0.000813
7. [11] www.web-access.net/~ahark/frames45.htm 0.000471
9. [06] www.classactionpl.com/ 0.000281
10. [07] www.irbc.com/ 0.000228
12. [50] www.kinsella.com/engle/ 0.000137
13. [41] www.classactionreports.com/ 0.000108
14. [02] www.tappi.org/paperu/classAction/classAction.htm 0.000000
15. [03] philipmorriss.com/engleweb/Lawsuit.asp 0.000000
16. [04] www.women4fatherhood.org/we.html 0.000000
17. [05] locklaw.com/classact/classact.html 0.000000
18. [08] www.internetnews.com/bus-news/article/0,21713_402581,00.html 0.000000
20. [12] www.consumerclassactionlawyers.com/ 0.000000
21. [14] slashdot.org/articles/00/02/02/1655251.shtml 0.000000
22. [15] www.overlawyered.com/topics/class.html 0.000000
23. [17] www.boston.com/dailyglobe2/207/business/Microsoft_wins_5th_class_action_lawsuit+.shtml 0.000000
24. [18] www.100toplawnets.com/Law/classaction/100/35k 0.000000
25. [19] www.classaction.xe.com/ 0.000000
27. [21] www.anicity.com/kansascity/stories/1999/05/31/editorial2.html 0.000000
28. [22] www.wirednews.com/news/politics/0.1283.34063.00.html 0.000000
29. [23] www.thestandard.com/article/display/0,1151,17024,00.html 0.000000
30. [25] www.insure.com/lawsuits/classaction.html 0.000000
Limitations

- Topics vs. terms
- Search engines provide non-random samples
- All links are equal
- Some topics not well-represented on the Web
Current Work

- Implementing the global algorithm (UofA, using Internet Archive snapshot)
- Incorporating TOPIC rank into search engine
- Evaluation of TOPIC as search engine rank function
Summary

• Unstructured data + links: WebSQL
• Semistructured data + links: WebOQL
• Exploiting links for reputation ranking: TOPIC