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
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>
    </capital>
  </state>

  ...

</states>

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

...</north-america>
Graph Representation

- **states**
  - **provinces**
    - **province**
      - **sname**: "California"
      - **governor**: "Davis" "Ontario" "Harris"
      - **pname**: "Toronto"
      - **cname**: "Sacramento"
      - **pop**: "2.5M"
  - **cities**
    - **cname**: "Toronto" "Ontario"
      - **governor**: "Harris"
      - **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**
  
SSQL’s: LOREL, UnQL, ...
  
XMLQL’s: XML-QL, XSLT, XQL
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 links
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”;

Next

Next

Next

...
Example applications

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

• Need pairs <URL of .ps, Metadata>

Internet


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
    “http://www.cis.upenn.edu/~db/langs/allpapers.html” [here] d,
    d [here] e;
Programmatic Interface

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
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: LI]

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

[Tag: XYZ, Text: This page contains ...]

[Tag: XYZ, 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: U. S.,
  Headline: Canadians win ...
  Url: http://www.cnn... ]

[ Section: World,
  Headline: Fire in Toronto ...
  Url: http://www.cnn... ]
Restructuring the Result into HTML

```csharp
[Tag: "table"/
  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!’
]
```

```
[ Tag: table ]

[Tag: th, Text: Section ]
[Tag: th, Text: Headline ]
[Tag: th, Text: Url ]

[Tag: tr ]

[Tag: td, Text: U. S. ]
[Tag: td, Text: Part-time ...]

[Label: http://www.cnn..., Url: http://www.cnn... ]
```
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>Name</th>
<th>Image Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>T.S. Abdelrahman, MSc, PhD</td>
<td><img src="http://www.cs.toronto.edu/gifs/Faculty/tsa.jpg" alt="T.S. Abdelrahman" /></td>
</tr>
<tr>
<td>R.M. Baecker, MSc, PhD</td>
<td><img src="http://www.cs.toronto.edu/gifs/Faculty/rmb.jpg" alt="R.M. Baecker" /></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

DBMS

File System

Index Server

LDAP

WWW
Computing Page Reputations

(Rafiei and Mendelzon, WWW9)

• Search engine Search-U-Matic just returned 60,000 pages on the query “liver disease.” Where should I start looking?

• We’re spending $200K/year maintaining our web pages. What do people think of them?

• Prof. X, an expert on Icelandic sagas, is up for tenure. I wonder how well known her research is on the Web.

• How is our Internet country music radio station doing, compared to the other 200 out there?

Idea:

• analyze links to find pages that are better/better known/more authoritative than others on some topics
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 = probability, in the limit, of hitting page p

Page Rank is query- and topic- independent
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

A transition is either:

• with probability $d>0$ jump to a random page that contains term $t$, or

• with probability $(1-d)$ follow a random link from the current page

Alternate between:

• make a transition out of the current page into $p$ (*forward visit to $p$*)

• make a transition out of a page $q$ that points to the current page (*backward visit to $q$*)
• $A(p,t) =$ probability of a forward visit to page $p$ when searching for term $t = \textit{Authority rank}$ of page $p$ on term $t$

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

\textbf{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? 

\textit{No}: See Lempel and Moran, WWW9.)
Inverting H&A computation

Topic → H & A → Pages

Page → ? → Topics
Two Solutions

• *Search engine solution*: a large crawl of the web is available. Find authorities on t for each term t

• *Real-time solution*: approximate the search engine solution by starting with some set of pages and the terms that appear in them, and iteratively expanding this set
Search Engine Solution (bottom up)

For every page $p$ and term $t$

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

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

While changes occur

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

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

Set of pages:

Set of terms: all terms $t$ that appear in $p$ or some of the $q_i$’s
Real-time 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 \( q_1 \)

\[
R(p, t) = R(p, t) + \left( \prod_{j=1}^{i} \frac{(1-d)^i}{\prod_{j=1}^{i} \text{Out}(q_i)} \right) \frac{d}{N_t}
\]
TOPIC: A crude approximation

- 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 real-time algorithm with $d = 0.10$, $k = 1$, $\text{Out}(q) = 7.2$
Example

- www.mcleans.ca

1. Maclean’s Magazine
2. macleans
3. Canadian Universities
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

• www.neci.nj.nec.com/homepages/giles.html
  Lee Giles, Neural Networks, Machine learning
Example: Institutional Home Page

www.cs.toronto.edu

- Russian History
- Computer Vision
- University of Toronto
- Hockey
Example: Institutional Home Page

- www.neci.nj.nec.com
  Watermarking
  Search engines
  Computer vision
  Neural networks
  Othello
Example: Institutional Home Page

- www.wins.uva.nl (Univ. of Amsterdam, Faculty of Sciences)

Solaris 2 FAQ
Wiskunde
Frank Zappa
Limitations

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

• Improving the real-time algorithm

• Implementing the search-engine algorithm:
  collaboration with search-engine company
  snapshot from Internet Archive

• Competitive ranking

• Reputation and communities
Summary

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