Extracting Schema from Semi-structured Data

Svetlozar Nestorov
Serge Abiteboul
Rajeev Motwani

Stanford University

“Inferring Structure in Semi-structured Data”

Outline

- Introduction
- Lattice Approach
- Program Approach
- Experiment
- Conclusion
Introduction

- Motivation
  - homepages of group members
    - name, e-mail, address, photo
    - similar information, but irregular schema
  - perfect typing
    - too large size
    - query optimization, graphical query interface

- Goal
  - approximate typing
    - tradeoff between the quality of a typing and its compactness

Introduction

- Preliminary
  - data model
    - Object Exchange Model (OEM)
    - labeled directed graph
  - notation
    - attribute(o)={labels on outgoing edges}
    - role(o)={labels on incoming edges}
    - at(S)=| {o} attribute(o)= S |
    - above(S)=| {o} attribute(o)⊇⊇⊇⊇ S |
  - definition
    - jump(S)=at(S)/above(S)
Introduction

- OEM Database D

Lattice Approach

- Algorithm
  - identify candidate types
  - select types from candidates and organize them into a type hierarchy
  - derive the typing rules
  - validate the type hierarchy against the data

- Counting lattice: at(S)
  - significant jump: jump(S) ≥ θ
Lattice Approach

- **Counting lattice**
  - $\text{jump}\{\text{Addr Age Name Sal Sex}\} = 1$
  - $\text{jump}\{\text{Addr Empl Name Sub}\} = 1$
  - $\text{jump}\{\text{Addr Empl Name}\} = 0.8$
  - $\text{jump}\{\text{Name Sal Sex}\} = 0.6$

Lattice Diagram:
- Addr Age Name Sal Sex (2000)
- Addr Empl Name Sub (10)
- Addr Name Sal Sex (3000)
- Addr Age Name Sex (1000)
- Addr Empl Name (40)
- Addr Name Sex (1500)
- Addr Name (100)

Lattice Approach

- **Type hierarchy construction**
  - primary role: $p\text{-role}(S)$
    - the most frequent label in $\text{role}(o)$ for all candidates
  - select candidate $T$ if there does not exist $T'$ such that $T' \subseteq T$ and $p\text{-role}(T') = p\text{-role}(T)$
Program Approach

- Base relations
- Typing rule
  - c(X):-A1&…&An
- Type links
  - link(Y,X,l)&c′(Y)
  - link(X,Y,l)&c′(Y)
  - link(X,Y,l)&atomic(Y,Z)
- Example typing rule
  - person(X):-link(X,Y,is-manager-of)&
    firm(Y)&link(X,Y’,name)&atomic(Y’,Z)
    ⇔ person(X):-is-manager-of firm, name atomic

program approach

Example typing program

```
project : t1 = Project, Project, Project, Project, Member, Project, Member,
         Name, Home Page
publication : t2 = Publication, Publication, Author, Name, Conference, Postscript
db-person : t3 = Project, Member, Group, Member, Project, Birthday, Degree, 
               Years, At, Stanford, Email, Home Page, Title,
               Name, Original, Home, Personal, Interest, Research, Interest
student : t4 = Project, Member, Student, Group, Member, Project, Advisor,
           Email, Title, Home Page, Name, Nickname
birthday : t5 = Birthday, Name, Month, Day, Year
degree : t6 = Degree, Major, School, Name, Year
```
Program Approach

- **Defect**
  - excess
  - deficit

- **Example**
  - $\text{type 1: } a^2$
  - $\text{type 2: } \overrightarrow{a}^1, \overrightarrow{b}^0, \overrightarrow{c}^0$
  - $\text{type 3: } \overrightarrow{b}^0, \overrightarrow{d}^0$
  - $\text{type assignment: } o_4$
    - $\text{type 2: excess}=1, \text{deficit}=1$
    - $\text{type 3: excess}=0, \text{deficit}=1$

---

- **Minimal perfect typing**
  - step 1
    - $\text{type 1: } a^2, a^3, a^4$
    - $\text{type 2: } \overrightarrow{a}^1, \overrightarrow{b}^0$
    - $\text{type 3: } \overrightarrow{a}^1, \overrightarrow{b}^0$
    - $\text{type 4: } \overrightarrow{a}^1, \overrightarrow{b}^0, \overrightarrow{c}^0$
  - step 2
    - equivalent class: $\text{type 2} = \text{type 3}$
  - step 3
    - $\text{type 1: } a^2, a^3$
    - $\text{type 2: } \overrightarrow{a}^1, \overrightarrow{b}^0$
    - $\text{type 3: } \overrightarrow{a}^1, \overrightarrow{b}^0, \overrightarrow{c}^0$
Program Approach

- Clustering
  - merge similar types
    - type1: \(a^0, b^3\)
    - type2: \(a^0, b^4\)
    - type3: \(a^0, b^1\)
    - type4: \(a^0, b^2\)
  - the order of coalescing has a significant effect on the quality of the results
  - distance function
    - symmetric difference

- Recasting

Experiment

<table>
<thead>
<tr>
<th>Synthetic Data</th>
<th>Typing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Perfect Types</td>
</tr>
<tr>
<td>DB No</td>
<td>Bipartite</td>
</tr>
<tr>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>Y</td>
</tr>
<tr>
<td>4</td>
<td>Y</td>
</tr>
<tr>
<td>5</td>
<td>N</td>
</tr>
<tr>
<td>6</td>
<td>N</td>
</tr>
<tr>
<td>7</td>
<td>N</td>
</tr>
<tr>
<td>8</td>
<td>N</td>
</tr>
</tbody>
</table>
Experiment

Conclusion

- Contribution
  - an algorithm for approximate typing of semi-structured data
  - each object may have more than one roles