Toward Large Scale Integration: Building a MetaQuerier over Databases on the Web

K. C.-C. Chang, B. He, and Z. Zhang
Presented by: M. Hossein Sheikh Attar

Background

- Deep (hidden) Web
  - Searchable online databases
  - 450,000 databases on the Internet
  - Growing fast
  - Invisible to users and current crawlers
    - Accessible through query interfaces

Problem Statement

- Currently users have difficulties in
  - Finding the right sources
    - E.g., What is a good source for finding apartments in Waterloo?
  - Querying them
    - Each source supports different query capabilities
- The goal of MetaQuerier
  - Make deep Web systematically accessible
  - Make it uniformly usable
Challenges

- Somewhat similar to the traditional information integration problem
- However
  - The scale is much larger
  - Dynamic discovery
    - No pre-selected sources
  - On-the-fly semantic discovery
    - Ad-hoc queries
    - No pre-configured per-source knowledge

Summary of Observations

- Survey and observe (do some "reality checks")
  - Helps make right assumptions
- Online databases are NOT arbitrarily complex
  - Convergence
  - Regularity
- Reason
  - Influence by peers
    - Amazon effect
Architecture of MetaQuerier

- 7 Subsystems
- Plan
  - Study and implement each subsystem individually
  - Integrate them
- 5 subsystems implemented so far

Subsystem 1: Database Crawler
**Database Crawler (DC)**
- Focused crawler for finding query interfaces
- Survey shows that the query interfaces are close to root page of Web sites

**Subsystem 2: Interface Extraction**

**Interface Extraction (IE)**
- Input: HTML query interface
- Output: query capabilities
  - constraint templates:
    - [attribute, operator, value]
  - E.g., [title, contains, $v$]
IE – Hidden Syntax Hypothesis

- Different interfaces share similar patterns
- Regularities
  - Presentation conventions

Hypothetic hidden syntax across sources
- Using this hidden syntax, we can interpret an interface unseen before
- Principles algorithmic framework
  - Using a grammar for pattern specification
  - Using a parser for pattern recognition

### Grammar

<table>
<thead>
<tr>
<th>Rule</th>
<th>Productions</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Start</td>
</tr>
<tr>
<td>1</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>D</td>
</tr>
<tr>
<td>4</td>
<td>F</td>
</tr>
<tr>
<td>5</td>
<td>H</td>
</tr>
</tbody>
</table>

Productions of the 2P grammar.
Ambiguities - 2P Grammar

- 2P grammar
  - A set of **productions** to capture conventional hidden patterns
  - A set of **preferences** to capture hidden priority conventions
- Best effort parser
  - multiple parse trees
  - incomplete parse trees
  - Merging trees at the end

Subsystem 3: Schema Matching
**Schema Matching**

- **Input:**
  - Query capabilities from several extracted forms in a domain

- **Output:**
  - Semantic correspondence (matching) among attributes

---

**Schema Matching**

- Existing method do not scale well to our problem
- Large scale is both a challenge and an opportunity
  - Holistic schema matching
  - Explore context information across all schemas
  - Assumes the existence of a hidden generative schema model

---

**Attribute frequencies in Books domain.**
Schema Matching

- Abstract the problem as **correlation mining**
  - Mining for *positive and negative* correlations

**Examples:**
- \{first name, last name\}, author

---

**Schema Matching – DCM framework**

**Data Preparation**
- Form Extraction
- Text Recognition
- Symbol Merging

**Correlation Mining**
- Group Discovery
- Matching Discovery
- Matching Selection

From matching to mining: the DCM framework.
Integrating Subsystems

System Integration

- Challenges
  - Accuracy problems
    - IE delivers 85-90% accuracy
    - Not accurate enough for SM

- Opportunities
  - feedback

Ensemble Framework

- Ensemble Framework
  - Accuracy problems mainly because of noisy input
  - **Sampling** and **voting** techniques
Feedback

- Feedback: Domain Statistics
- Example
  - Conflict between
    - last name; contain; $val$
    - e.g. Mike; contain; $val$
  - SM notices that the first one is much more frequent

Feedback

- IE processes one interface at a time
- SM has holistic domain statistics
- Feedback from SM can help IE resolve conflicts
- Another example that large scale is both curse and blessing
Feedback

- 3 types of domain statistics
  - Type of attributes
  - Frequency of attributes
  - Correlation of attributes

Summary

- Large scale integration involves challenges and opportunities
- Integrating subsystems also involves challenges and opportunities
- Holistic Integration insights
  - Hidden regularity
  - Peer majority