Data Warehouses and Web Data Integration

Andreas Thor
July 2\textsuperscript{nd}, 2008
Two worlds: Data warehouse & web data

- Data warehouse
  - integrated, centralized data
  - closed world (company-internal information)
  - high data quality
  - stable

- Web data
  - non integrated
  - open world (“data about everything”)
  - diverse data quality
  - volatile

- Web data can enhance DWH data
  - additional dimensions for fact table(s)
  - additional characteristics for existent dimension(s)

→ Increase coverage by preserving high data quality
Example scenarios

- E-Commerce: product sales
  - DWH: products (article information, supplier, purchases, ...)
  - Web: reviews, competitor’s prices, ...
- E-Commerce: customer relationship management
  - DWH: customers (address, channels, ...)
  - Web: socio-economic data, territory risk analysis, ...
- Bibliographic domain
  - DWH: bibliographic information (title, venue, authors, ...)
  - Web: citation counts, institutions, keywords, ...
- Geographic domain
  - DWH: geographic places (name, inhabitants, region, ...)
  - Web: hotel ratings, point of interest, ...
- Bioinformatics: GeWare (?)
- ...
Agenda

- Motivation
- Web entity search architecture
- OCS 2.0: Lessons learned
- Search strategy generation
  - by query relaxation
  - by page analysis
- Conclusion
Finding web entities

- Problem: Find all relevant (web) entities for a given subset of the data warehouse

- „Google approach“
  - crawl the WWW, extract entities, import into warehouse
  - too expensive, even for focused crawling

- Search-based approach ("Use Google approach")
  - use existing (and powerful) search engine technology
  - efficient access to millions of web sources
Architecture / workflow

- Find all relevant (web) entities for a given DWH subset
  - set of fact table instances + associated dimensions
  - set of dimension instances
Entity search

- Entity search engines: Google Scholar, Google Base, ...

- General search engines: Google, Yahoo, ...
  - document extractor is site-specific (e.g., created with Dapper)
  - restrict search to website, e.g., "<query> site:portal.acm.org" searches for <query> within ACM portal
  - "interesting" web sites, e.g., ACM portal, Wikipedia, ...
Search strategies

- Input: n input entities
- Output: m queries (for a given search engine)
- Intention: query execution yields corresponding web entities

- Effectiveness
  - find all relevant entities (recall = 100%)
  - find only relevant entities (precision = 100%) → object matching

- Efficiency
  - minimize the user-/administrator effort for query generation
  - minimize the number of queries (run-time, accessibility, ...)

- Problem: Automatic or manual strategy definition / query generation?
Search strategies: Lessons learned

- Prototype: Online Citation Service (OCS) 2.0
  - user selects set of DBLP publications
  - OCS searches for corresponding publications in other sources (and summarizes the citation counts per source)

- Sources
  - Google Scholar, MS Libra → entity search engine
  - ACM, Citeseer → general search engine

- Application of multiple search strategies
  - OCS allows a flexible integration of new strategies
A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces (VLDB 1998)
Roger Weber, Hans-Jörg Schek, Stephen Blott

A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces

A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces

A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces

A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces

A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces

Stephen Blott: A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces

Algorithms for Mining Distance-Based Outliers in Large Datasets (VLDB 1998)
Edwin M. Knorr, Raymond T. Ng

MindReader: Querying Databases Through Multiple Examples (VLDB 1998)
Yoshiharu Ishikawa, Ravishankar Subramanya, Christos Faloutsos
A survey of approaches to automatic schema matching

E Rahm, PA Bernstein - The VLDB Journal The International Journal on Very Large ...; 2001 - Springer
10.1007/s007780100057 A survey of approaches to automatic schema matching ...
Zitiert durch: 1222 - Ähnliche Artikel - Websuche

[ZITATION] A survey of approaches to automatic schema matching
PA Bernstein, E Rahm - VLDB Journal, 2001
Zitiert durch: 17 - Ähnliche Artikel - Websuche

[ZITATION] A survey of approaches to automatic schema mapping
E Rahm, PA Bernstein - The VLDB Journal, 2001
Zitiert durch: 7 - Ähnliche Artikel - Websuche

[ZITATION] A survey of approaches to semantic schema matching
E Rahm, PA Bernstein - The VLDB Journal 10: 334, 2001
Zitiert durch: 6 - Ähnliche Artikel - Websuche

[ZITATION] A survey of approaches to automatic schema mapping" the VLDB Journal
E Rahm, PA Bernstein - Vol
Zitiert durch: 3 - Ähnliche Artikel - Websuche

E Rahm, PA Bernstein - The International Journal on Very Large Data Bases (VLDB), 2001
Zitiert durch: 2 - Ähnliche Artikel - Websuche

E Rahm, P Bernstein - VLDB Journal
Zitiert durch: 1 - Ähnliche Artikel - Websuche

+ 2 additional
## Entity search engine: Example (2)


<table>
<thead>
<tr>
<th>Search strategy</th>
<th>Query</th>
<th>#Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>title (simple)</td>
<td>a survey of approaches to automatic schema matching</td>
<td>18.2000</td>
</tr>
<tr>
<td>title (simple phrase)</td>
<td>„a survey of approaches to automatic schema matching”</td>
<td>1.040</td>
</tr>
<tr>
<td>title</td>
<td>intitle:a survey of approaches to automatic schema matching</td>
<td>3.530</td>
</tr>
<tr>
<td>title (phrase)</td>
<td>intitle:&quot;a survey of approaches to automatic schema matching“</td>
<td>7</td>
</tr>
<tr>
<td>title pattern</td>
<td>intitle:&quot;survey * approaches ** schema“</td>
<td>13</td>
</tr>
<tr>
<td>author (1)</td>
<td>author:rahm</td>
<td>2.230</td>
</tr>
<tr>
<td>author (2)</td>
<td>author:bernstein author:rahm</td>
<td>44</td>
</tr>
<tr>
<td>author (1) + title pattern</td>
<td>author:rahm intitle:&quot;survey * approaches ** schema“</td>
<td>9</td>
</tr>
<tr>
<td>author (1) + keyword</td>
<td>author:rahm intitle:survey intitle:schema</td>
<td>9</td>
</tr>
</tbody>
</table>
Lessons learned: Entity search engine

- Search strategies can be generated by answering …
  - What attributes should be queried for?
  - What attribute value transformation should be applied?

- Domain 1: Computer science
  - small number of authors
  - title contains common words + acronyms
  - heterogeneous venue names
  → good strategies: „author + keyword“, „title pattern“

- Domain 2: Chemistry
  - high number of authors
  - title contains string representations of chemical formulas
  - homogenous venue names
  → good strategies: „venue (+year)“, „title keywords“, „author group“
Entity search: Model

- Query = $p_1(v_1) \land p_2(v_2) \land \ldots \land p_n(v_n)$
- $p_i = \text{predicates}$
  - realized as search form fields and/or „[predicate name]:“
  - $p_0 = \text{free text}$
- $v_i = \text{search values for predicates}$
  - derived from attribute values
  - may be transformed (keyword selection, pattern, reduction, …)
Query containment

- **A \(\subseteq\) B**: all resulting entities of query A appear in B’s search result
- Containment based on
  - predicates
  - attribute values
Search strategy generation by query relaxation

- Example-based query learning
  - input: n DWH entities + m corresponding web entities
  - output: algorithm for query generation
  - intention: algorithm also works for other DWH entities
- Start with fully specified query, e.g., all predicates with exact attribute value
- Relax query by
  - eliminating predicates
  - generalizing / transforming attribute values
- Determine precision and recall
  - for example pages (training data)
  - for test data
- Find optimum (w.r.t. to minimal number of query size)
  - overfitting
### General search engine: Example

**URL:** http://portal.acm.org/citation.cfm?id=767149.767154  
**Title:** A survey of approaches to automatic schema matching

<table>
<thead>
<tr>
<th>Query</th>
<th>#Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;a survey of approaches to automatic schema matching&quot;</td>
<td>13,200</td>
</tr>
<tr>
<td>initle:&quot;a survey of approaches to automatic schema matching&quot;</td>
<td>63</td>
</tr>
<tr>
<td>initle:&quot;a survey of approaches to automatic schema ...&quot; inurl:citation.cfm site:portal.acm.org</td>
<td>1</td>
</tr>
<tr>
<td>&quot;Erhard Rahm&quot; inurl:citation.cfm site:portal.acm.org</td>
<td>2,530</td>
</tr>
<tr>
<td>inanchor:&quot;Erhard Rahm&quot; inurl:citation.cfm site:portal.acm.org</td>
<td>40</td>
</tr>
<tr>
<td>intitle:schema inanchor:&quot;Erhard Rahm&quot; inurl:citation.cfm site:portal.acm.org</td>
<td>4</td>
</tr>
<tr>
<td>intitle:survey inanchor:&quot;Erhard Rahm&quot; inurl:citation.cfm site:portal.acm.org</td>
<td>1</td>
</tr>
</tbody>
</table>
Lessons learned: General search engine

- Search strategies can be generated by (manually) analyzing web pages

- **Website: ACM Digital Library**
  - publication title \(\approx\) page title (in some cases: substring)
  - author names with hyperlinks (\(\rightarrow\) inanchor:)
  - publication Id (URL parameter) contains venue Id (\(\rightarrow\) inurl:)

- **Website: Citeseer**
  - publication title = page title
  - author names are concatenated (\(\rightarrow\) pattern), no hyperlinks
  - publication Id "useless"
Search strategy generation by page analysis

- Example-based query learning
  - input: n DWH entities + m corresponding web pages (entities)
  - output: algorithm for query generation
  - intention: algorithm also works for other DWH entities

- Analyze example pages regarding “searchable” elements
  - content: keywords, phrases (""), pattern ("a * b")
  - structure: URL (site:), linked pages (link:)
  - content + structure: page title (intitle:), anchor text (inanchor:)

- Determine precision and recall
  - for example pages (training data)
  - for test data

- Find optimum (w.r.t. to minimal number of query size)
  - overfitting
Conclusion and future work

- Web data integration can be crucial for data warehouses
- Automatic and efficient querying for web entities
- Two approaches
  - entity search engine $\rightarrow$ query relaxation
  - general search engine $\rightarrow$ page analysis

- Algorithms
  - elaboration (bulk search)
  - implementation

- Evaluation for different scenarios