Objekt-Matching zur Datenintegration

Erhard Rahm
H. Köpcke, A. Thor

http://dbs.uni-leipzig.de

DB-Stammtisch, Dresden
Jan. 16, 2008

Object matching problem
• Identify semantically equivalent (matching) objects
  • within one data source or between different sources
  • to integrate (merge) them, compare them, eliminate duplicates, etc.
• Most previous work for structured data (relational databases)

<table>
<thead>
<tr>
<th>CID</th>
<th>Name</th>
<th>Street</th>
<th>City</th>
<th>Sex</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Kristen Smith</td>
<td>2 Hurley Pl</td>
<td>South Fork, MN 48503</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>Christian Smith</td>
<td>Hurley St 2</td>
<td>S Fork MN</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cno</th>
<th>LastName</th>
<th>FirstName</th>
<th>Gender</th>
<th>Address</th>
<th>Phone/Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>Smith</td>
<td>Christoph</td>
<td>M</td>
<td>23 Harley St, Chicago IL, 60633-2394</td>
<td>333-222-6542 / 333-222-6599</td>
</tr>
<tr>
<td>493</td>
<td>Smith</td>
<td>Kris L.</td>
<td>F</td>
<td>2 Hurley Place, South Fork MN, 48503-5998</td>
<td>444-555-6666</td>
</tr>
</tbody>
</table>
Matching objects in web sources

A survey of approaches to automatic schema matching - group of 25

E Rahm, PA Bernstein - The VLDB Journal The International Journal on Very Large Data Bases, 2001 - Springer

... In the next section, we summarize some example applications of schema matching ... E Rahm, PA Bernstein - The VLDB Journal, 2001

Duplicates within (integrated) sources

Duplicates due to:
- Order of authors
- Extraction error (title)
- Different titles
- Typos (author name)
- Additional authors (!)
Classification of data quality problems*

Data Quality Problems

**Single-Source Problems**
- **Schema Level**
  - (Lack of integrity constraints, poor schema design)
- **Instance Level**
  - (Data entry errors)

**Multi-Source Problems**
- **Schema Level**
  - (Heterogeneous data models and schema designs)
- **Instance Level**
  - (Overlapping, contradicting and inconsistent data)

**Object matching**
- Uniqueness
- Referential integrity
- Misspellings
- Redundancy/duplicates
- Contradictory values
- Naming conflicts
- Structural conflicts


---

**Agenda**

- **Motivation**
- **Data Integration Architectures**
  - DB & IR approaches
  - Workflow-based data integration
  - Use of Mashups
- **iFuice & MOMA**
  - iFuice data integration platform
  - MOMA framework
  - Mashup-like data integration (OCS)
- **Summary**
**Traditional DB approaches**

**Physical Integration** (Data Warehousing)
- Data Marts
- Data Warehouse
- Import (ETL)
- Operational Systems
- Metadata
- Analysis Tools

**Virtual Integration** (query mediators)
- Client 1
- Mediator
- Wrapper 1
- Source 1
- ... Wrappers
- ... Sources

---

**"IR World"**

General search engine architecture

_Arasu et. al.: Searching the Web. ACM Trans. Internet Techn, 2001_
Integration Alternatives

<table>
<thead>
<tr>
<th>Data quality issues for many sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup time too high: crawling; schema integration  …</td>
</tr>
<tr>
<td>Current data integration approaches are query-focussed: search engine queries, query mediators, warehouse access</td>
</tr>
<tr>
<td>Queries are not enough: complex data integration problems cannot easily be solved in one query / search</td>
</tr>
<tr>
<td>What is the most cited VLDB paper so far?</td>
</tr>
<tr>
<td>Which famous scientists lived close to this building?</td>
</tr>
<tr>
<td>Data quality for heterogeneous/dirty web data and query results</td>
</tr>
<tr>
<td>Execution time for dynamic fusion of larger data sets</td>
</tr>
</tbody>
</table>
Workflow-like data integration

- "You only have three hours - how far can you go to solve a data integration problem?"
- Reuse + Combine existing (data) services within **data integration workflows**
  - Reuse existing services
  - Reuse existing data integration systems, e.g. search engines, query mediators, warehouses
  - Combine query/service results within a workflow
  - Perform data cleaning and data transformation
  - Perform data analysis
- **Must be supported by a flexible data integration framework**
- **Workflow-like data integration complements query-based data integration**

Workflow-based Integration

- **Examples**
  - **Offline**: ETL processes for Data Warehouses
  - **Online**: Workflows for analyzing biological data

- **New aspects:**
  - Combine (on-demand) ETL and analysis workflows
  - Share and reuse existing data services and tools
  - Reuse existing (entity) search engines
  - Easy development and use of workflows (→ Mashups)
Mashups - a light-weight data integration approach

- "A web mashup is a web page or application that combines data from two or more external online sources." (ProgrammableWeb)

- "A mashup is a web application that combines data from more than one source into an integrated experience." (Wikipedia)

- "Mashups are an exciting genre of interactive Web applications that draw upon content retrieved from external data sources to create entirely new and innovative services." (Merrill: Mashups: The new breed of Web app)

Mashup Example: Forbes List

Ranking by Forbes List of best paid celebrities

YouTube video

Hometown displayed by Google Maps

http://www.mubazaar.com/top100celebrities
Mashups: Driving forces

- AJAX (Asynchronous Javascript and XML)
  - Desktop-like look-and-feel of Web applications
- Development tools, e.g. Google Web toolkit
- Visual development tools without programming need
- Increasing number of Web services (APIs)
  - Easy access to "interesting" content and services
  - 50% of mashups use Google Maps

ProgrammableWeb
#Mashups >2500
#Mashups/Day 3
#APIs >500

Top APIs

- GoogleMaps (49%)
- Flickr (11%)
- Amazon (7%)
- YouTube (6%)
- VirtualEarth (4%)
- YellowMaps (4%)
- eBay (4%)
- AirSync (3%)
- delicious (3%)
- Yahoo (3%)

The Big Picture: Mashup Fabric*

* Jhingran: Enterprise Information Mashups: Integrating Information, Simply. Keynote at VLDB '06
Mashup Tools: Overview

Mashup Builders

Managing mashup components, e.g., maps, feeds

Data Transformation/Data Aggregation

Data transformation workflows

Source Wrappers

Information Extraction

Mashups: Characteristics

- Easy and fast development
  - Visual programming (drag & drop) or integration with power development environments (e.g., Eclipse)
- Service-oriented paradigm
  - Sharing and reuse of web services
- Web 2.0 interfaces
- XML-based data formats, e.g., RSS, SOAP/REST (data exchange)
- Simple workflows
- Simple instance-based data integration
  - Geographical coordinates
  - Keywords (e.g., names)
- Simple keyword queries dominate (no query transformation)
- Limited result postprocessing (primarily merge instead of match)
Mashups: Open Problems

- Data quality
  - Typos, missing/wrong attribute values (e.g., due to extraction errors)
  - Duplicates, i.e., sources contain multiple instances for the same (real world) object
- More complex queries, e.g. for heterogeneous entity search engines (-> query transformation)
- Performance for large data volume (automatic optimization)
- Semantic repository of services
  - Service description & service discovery
- Support for business applications, e.g. security restrictions

Agenda

- Motivation
- Data Integration Architectures
  - DB & IR approaches
  - Workflow-based data integration
  - Use of Mashups
- iFuice & MOMA
  - iFuice data integration platform
  - MOMA framework
  - Mashup-like data integration (OCS)
- Summary
Information Fusion with iFuice [RTA+05]

- Generic data integration platform for structured and unstructured data sources
  - Query / search / id-based data access
- Workflow-like data integration with operator-based programming model
  - Generic high-level operators for use within script programs
  - Example: query, traverse, map, union, aggregate
- Utilization of instance-level mappings
  - Correspondences between object instances
  - Represent semantic relationship ("is same", "is associated to")
- Metadata repository for data sources and services
  - Semantic object (e.g., Author, Publication) and mapping types
- Object matching (MOMA framework)
- Iterative query strategies

iFuice Framework: Architecture

[Diagram showing the architecture of iFuice with various components like Mashup Interface, Mashup Definition, Mashup Execution, and Applications/Extensions.]
Data sources

- **Physical data source (PDS)**
  - Web data (DBLP), local data (files), ...
  - Split into logical data sources

- **Logical data source (LDS)**
  - Refers to one object type
  - Contains object instances

**Object instance**
- Refers to real-world entity
- Set of attributes
- One attribute is \textit{id}

---

Mappings

- **Directed binary relationship between LDS**
- **Mappings have a semantic mapping type**
  - e.g., "publications of author"

**Same mappings vs. association mappings**
- same = "equality" relationship typically between PDS
  - e.g., DBLP publication (id) $\rightarrow$ ACM publication (id)

**Instance correspondences**
- Materialized: mapping tables
- Determined on-the-fly: execution result (e.g., from web service or query)
Objects and instance-based mappings

<table>
<thead>
<tr>
<th>Schema and ontology matching with COMA++</th>
<th>Publication@ACM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full text</td>
<td>1066157.1066283</td>
</tr>
<tr>
<td>Source</td>
<td>International Conference on Management of Proceeding of the 2005 ACM SIGMOD inter</td>
</tr>
<tr>
<td>Authors</td>
<td>David Aumueller, Hong Hai Do, Sabine Massmann, Erhard Rahm</td>
</tr>
<tr>
<td></td>
<td>University of Leipzig, Leipzig, Germany</td>
</tr>
<tr>
<td></td>
<td>University of Leipzig, Leipzig, Germany</td>
</tr>
<tr>
<td></td>
<td>University of Leipzig, Leipzig, Germany</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Association-Mapping</th>
<th>Publication@ACM</th>
<th>Author@ACM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1066157.1066283</td>
<td>P729451</td>
</tr>
<tr>
<td></td>
<td>1066157.1066283</td>
<td>P707877</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Same-Mapping</th>
<th>Publication@ACM</th>
<th>Publication@DBLP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1066157.1066283</td>
<td>conf/sigmod/AumuellerDMR05</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Metadata model

- **Source/mapping model**
- **Domain model** indicates available object types and relationships
Operators

- Query language capabilities + scripting support
- **Set-oriented operators**
  - Input: set of object or mapping instances + parameters / query specification
  - Output: set of object / mapping instances
  
  ⇒ Can be combined within scripts

Operators overview

- **Object instances (OI)**
  - OI → OI: `getInstances, traverse, traverseSame`
  - Query → OI: `queryInstances, queryMatch, attrTransf`

- **Aggregated objects (AO)**
  - OI → AO: `agg, disagg, fuseAttributes`
  - AO → AO: `aggregateSame, aggregateTraverse, aggregateMap`

- **Generic**
  - `union, diff, intersect`
  - `domain, range, compose`
MOMA Overview [TR07]

- **MOMA** = **Mapping based Object Matching**
- **Object consolidation framework**
  - Matching objects from 2 sources
  - Generation of instance mappings (correspondences)
  - Special case: duplicate detection within 1 source (generation of self-mapping)
- **Key features**
  - Extensible matcher library
  - Mapping combination
  - Construction of match workflows
  - Storage of mappings for reuse in other match problems

\[
\begin{array}{ccc}
\text{LDS}_A & \text{LDS}_B & \text{Sim} \\
\hline
a_1 & a'_1 & 1 \\
a_2 & a'_1 & 0.9 \\
a_3 & a'_2 & 0.8 \\
\end{array}
\]

**LDS** A

**LDS** A

**LDS** B

**LDS** B

**Matcher Library**

Matcher implementation (e.g., Attribut based)

Match Workflows

**Matcher n**

**Matcher 2**

**Matcher 1**

**Mapping Cache**

**Mapping Operator**

**Same Mapping**

**Mapping Repository**

**Mapping Combiner**

**Match Workflow**

**Match Workflows**
Object matching approaches

Value-based

Single attribute

Multiple attributes

unsupervised

supervised

Context-based

• Hierarchies:
  • Ananthakrishna et al. (VDBL 2002)
  • Graphs:
  • Bhattacharya, Getoor (DMKD 2004)
  • Dong et al. (SIGMOD 2005)
  • Ontologies

• Aggregation function with threshold
• User-specified Rules:
  • Hernandez et al. (SIGMOD 1995)
  • Clustering
  • Monge, Elkan (DMKD 1997)
  • Mc Callum et al. (SIGKDD 2000)
  • Cohen, Richman (SIGKDD 2002)

• Decision trees
• Verykios et al. (Information Sciences 2000)
• Tejada et al. (Information Systems 2001)
• Support Vector Machine
  • Bilenko, Mooney (SIGKDD 2003)
  • Minton et al. (2005)

Match Workflows

- Coordinated execution of matchers and combination of mappings
  - single-attribute matcher (e.g. based on specific string similarity function)
  - multi-attribute matcher (hybrid matcher)
  - context matcher ...

- Example: Independent matcher execution

DBLP

Matcher 1

Matcher 2

$M1 := \text{attrMatch} ($DBLP$, $ACM$, ["title"], \text{TFIDF}, 0.9);

$M2 := \text{attrMatch} ($DBLP$, $ACM$, ["year"], \text{EditDistance}, 0.7);

$\text{Union} := \text{union} ($M1$, $M2$, \text{avg});

$\text{Result} := \text{select} ($\text{Union}, 0.8);
Match Strategies: Merge & Compose

1. Merge

- Overcome short-comings (e.g., recall)

2. Compose

- Efficient re-use of mappings

Match Strategies: Neighborhood

- Combine same- & association mappings
  - Attribute matching may suffer from highly different values
  - Use information stored in associated objects
  - Bibliographic example: Conference@DBLP - Conference@ACM
  - “Two conferences are the same if they share a significant number of publications.”
  - Reuse of publication same mapping

- Very good results for 1:N relationship (e.g., Venue-Publication)
- Restriction of matching space for N:1 (Publication-Venue) and N:M (Author-Publication)
Neighborhood Matcher (2)

- Compose 3 mappings
  - Conf@DBLP → Pub@DBLP → Pub@ACM → Conf@ACM
    - Association-Map (1:N)
    - Same-Map (1:1)
    - Association-Map (N:1)

- reuse pub.same-mapping to determine conf.same-mapping

<table>
<thead>
<tr>
<th>DBLP</th>
<th>ACM</th>
<th>Similarity Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLDB 2001</td>
<td>VLDB 2001</td>
<td>0.8 (4 out of 5 paths)</td>
</tr>
<tr>
<td>VLDB 2001</td>
<td>VLDB J. 2002</td>
<td>0.2 (1 out of 5 paths)</td>
</tr>
</tbody>
</table>

Evaluation

- Real data sources: DBLP, ACM, GoogleScholar (GS)
- DBLP:
  - 2616 publications, 3319 authors
  - 130 venues from 10 years: 20 conferences (Sigmod, VLDB), 110 journal issues (TODS, VLDBJ, Sigmod Record)
- Match problems
  - publications: DBLP-ACM, DBLP-GS, GS-ACM
  - authors: DBLP-ACM
  - venues: DBLP-ACM
- Perfect mapping: manually determined
Evaluation: Matcher combination

- Match: DBLP – ACM publications
- Combination of independent matchers: Title vs. Author-Matcher

Combination outperforms initial matchers

![Graph showing performance metrics for Title, Authors, Union, and Intersect]

Evaluation: Compose same-mappings

- Publication same mappings
  - (1) DBLP-GS: Query with title + trigram
  - (2) DBLP-ACM: Attribute matcher
  - (3) GS-ACM: Link extraction

Compose quality depends on initial mappings
Combination (consolidate) of direct and compose mapping retains overall quality

![Graph showing F-Measure for DBLP-GS, DBLP-ACM, and GS-ACM]
Evaluation: Neighborhood matcher

- Venue@DBLP – Venue@ACM via Publications: 98.8% F-Measure
- DBLP – ACM authors (N:M case)
  - Neighborhood: Publication same mappings
  - Combination with attribute matcher using intersect

Mashup Example

- On-demand citation service (OCS, [TAR07]) *
  - What are the most cited papers of conference X?
  - What is the average citation number of publications from author Y?
  - Frequent changes, i.e., new publications & new citations
- Idea: Combine publication lists, e.g. from DBLP or Pubmed, with citation counts, e.g. from Google Scholar, Citeseer or Scopus
  - DBLP, Pubmed: high bibliographic data quality
  - GS: large coverage of citations counts
- Query problem: Given a set of DBLP publications → How to find the corresponding GS publications?
  - Query GS and match DBLP-GS

* http://labs.dbs.uni-leipzig.de/ocs
Online Citation Service: Result overview

<table>
<thead>
<tr>
<th>Title</th>
<th>Authors</th>
<th>Venue</th>
<th>Year</th>
<th>Citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A survey of approaches to automatic schema matching</td>
<td>Erhard Rahm, Philip A. Bernstein</td>
<td>VLDB J</td>
<td>2001</td>
<td>1078</td>
</tr>
<tr>
<td>Generic Schema Matching with Cupid</td>
<td>Jyotsna Madhevan, Philip A. Bernstein, Erhard Rahm</td>
<td>VLDB</td>
<td>2001</td>
<td>659</td>
</tr>
<tr>
<td>COIMA - A System for Flexible Combination of Schema Matching Approaches</td>
<td>Hong Hai Do, Erhard Rahm</td>
<td>VLDB</td>
<td>2002</td>
<td>306</td>
</tr>
<tr>
<td>Data Cleaning: Problems and Current Approaches</td>
<td>Erhard Rahm, Hong Hai Do</td>
<td>IEEE Data Eng Bull</td>
<td>2000</td>
<td>234</td>
</tr>
</tbody>
</table>

Bibliographic data from DBLP

Sum of GS citations

Corresponding GS publications

OCS example: Top conference papers

OCS result for venue BTW 2003

- Found 36 GS publications for 31 DBLP publications.
- No GS publications found for 17 DBLP publications.
- Over 40 DBLP publications having 392 citations.
- Average: 5.9 citations per publication.
- H-index: 9

<table>
<thead>
<tr>
<th>Title</th>
<th>Authors</th>
<th>Venue</th>
<th>Year</th>
<th>Citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Services: Distributed Applications Without Limits</td>
<td>Frank Leymann</td>
<td>BTW</td>
<td>2003</td>
<td>74</td>
</tr>
<tr>
<td>Type Checking in XBIDE</td>
<td>Martin Kempe, Volker Linnemann</td>
<td>BTW</td>
<td>2003</td>
<td>36</td>
</tr>
<tr>
<td>Data Integration: A Status Report</td>
<td>Alon Y. Halevy</td>
<td>BTW</td>
<td>2003</td>
<td>29</td>
</tr>
<tr>
<td>The Paradigm of Relational Indexing: a Survey</td>
<td>Hans-Peter Kriegel, Martin Pfeifer, Marco Pödke, Thomas Seidl</td>
<td>BTW</td>
<td>2003</td>
<td>15</td>
</tr>
<tr>
<td>Nutzerverdefinierte Replikation zur Realisierung neuer mobiler Datenbankanwendungen</td>
<td>Christoph Gollnick</td>
<td>BTW</td>
<td>2003</td>
<td>13</td>
</tr>
<tr>
<td>Executing Nested Queries</td>
<td>Goetz Graefe</td>
<td>BTW</td>
<td>2003</td>
<td>11</td>
</tr>
<tr>
<td>SQL/MM Spatial - The Standard to Manage Spatial Data in a Relational Database System</td>
<td>Knut Stolze</td>
<td>BTW</td>
<td>2003</td>
<td>10</td>
</tr>
<tr>
<td>Kombinierte Personalisierung von Web Services</td>
<td>Markus Kießl, Stefan Seltscham, Christof König, Alton Kemper</td>
<td>BTW</td>
<td>2003</td>
<td>9</td>
</tr>
</tbody>
</table>
OCS Match Strategies

Interactive approach, i.e., user selects match thresholds

<table>
<thead>
<tr>
<th>Title</th>
<th>Year</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>80%</td>
<td>+/- two years</td>
<td>50%</td>
</tr>
<tr>
<td>85%</td>
<td>+/- one year</td>
<td>60%</td>
</tr>
<tr>
<td>90%</td>
<td>equal year</td>
<td>70%</td>
</tr>
<tr>
<td>95%</td>
<td></td>
<td>80%</td>
</tr>
<tr>
<td>100%</td>
<td></td>
<td>90%</td>
</tr>
</tbody>
</table>

Aggregated result is adjusted automatically based on match definition

OCS: Iterative Queries

1st query strategy: name (#queries = 1)
- Query: author’s/venue’s name, e.g., 'author:E-Rahm'
- Goal: find as many relevant pub’s with only one query

2nd query strategy: title pattern (#qu.=#pubs/10)
- Query: Disjunction of title patterns, e.g., 'intitle:"survey * approaches ** schema matching" OR ...
- Goal: Precise search for a limited set of publications

3rd query strategy: title keywords (#queries = #pubs)
- Query: publication title, e.g., 'A survey of approaches to automatic schema matching'
- Goal: Find a certain publication at the cost of many irrelevant search results
Illustration of OCS mashup execution

User selects author from list

01: $\text{DBLPPubs} := \text{query} (\text{DBLP}, \text{"author=\{name\}"})$
02: $\text{GSPubs1} := \text{query} (\text{GS}, \text{"author=\{name\}"})$
03: $\text{Result1} := \text{fuse} (\text{DBLPPubs/Pub/}, \text{GSPubs1/Entry/})$
04: $\text{Result1} := \text{aggregate} (\text{Result1}, \text{"[DBLP/Pub/NoOfCit\]", \text{sum([./Entry/Citations])"}})$

Result 1 is displayed to the user

Start next script automatically

05: $\text{GSPubs2} := \text{query} (\text{GS}, \text{[DBLP/Pub/Titlepattern]"})$
06: $\text{Result2} := \text{union} (\text{fuse} (\text{DBLPPubs/Pub/}, \text{GSPubs2/Entry/}), \text{Result1})$
07: $\text{Result2} := \text{aggregate} (\text{Result2}, \text{"[DBLP/Pub/NoOfCit\]", \text{sum([./Entry/Citations])"}})$

Result 1 is replaced by Result 2
If user wants exhaustive search (e.g., by button click) → start next script

08: $\text{DBLPPubs3} := \text{query} (\text{Result2}, \text{"count([./Entry/Citations]=0")}$
09: $\text{GSPubs3} := \text{query} (\text{GS}, \text{DBLPPubs3, "intitle:[DBLP/Pub/Title]"})$
10: $\text{Result3} := \text{union} (\text{fuse} (\text{DBLPPubs/Pub/}, \text{GSPubs3/Entry/}), \text{Result2})$
11: $\text{Result3} := \text{aggregate} (\text{Result3}, \text{"[DBLP/Pub/NoOfCit\]", \text{sum([./Entry/Citations])"}})$

Agenda

- Motivation
- Data Integration Architectures
  - DB & IR approaches
  - Workflow-based data integration
  - Use of Mashups
- iFuice & MOMA
  - iFuice data integration platform
  - MOMA framework
  - Mashup-like data integration (OCS)

Summary
Summary

- **iFuice**: generic data integration platform
  - workflow-based data integration
  - powerful operators and script facility to build data transformation and analysis workflows
  - utilizes existing **instance correspondences** (mappings)
  - support for advanced mashups
  - successful adoption in different domains

- **MOMA framework** for object matching
  - combined use of several matchers
  - reuse of existing and previously determined instance mappings
  - flexible match strategies, e.g. neighborhood matching

Ongoing / Future Work

- **Optimization of match strategies** (self-tuning)
  - Which matchers?
  - Which attributes? What are the best thresholds?
  - Use of training-data
  - Quality - performance tradeoffs for interactive apps

- **Evaluation and optimization of interactive query strategies**
  - Example: 'name' strategy often good for authors but not for venues
  - Dependency on instance values "E-Rahm" 😊 vs. "J-Smith" 😕
  - Can we automatically determine the relevant attributes (and their transformations) that should appear in the queries?
References