Data Cleaning for Data Integration

Advanced School on Data Exchange, Integration, and Streams (DEIS)

Ekaterini Ioannou

Tuesday, 9th of Nov. 2010, Schloss Dagstuhl
Problem overview

Data integration:
- Combine data from various sources/applications
- Merge into a single database
- Requires a unified view over the data → cleaning

Challenges:
- Handling the various incoming schemata
- Dealing with the missing data values
- **Entity Resolution**
  → combine the various descriptions or references for the same real world objects
Reasons for Various Descriptions

- **Text variations:**
  - Misspellings
  - Acronyms
  - Transformations
  - Abbreviations
  - etc.
Reasons for Various Descriptions

- **Text variations**
- **Local knowledge:**
  - Each source uses different formats
e.g., person from publication vs. person from email
  - Lack of global coordination for identifier assignment

![Image showing examples of text variations and entity mentions.](image-url)
Reasons for Various Descriptions

- Text variations
- Local knowledge
- Evolving nature of data:
  - Entity alternative names appearing in time
  - Updates in entity data

figure from [RVMB09]

Alternate Names: Jackie Bouvier | Jackie Kennedy | Mrs. John F. Kennedy | Jackie Onassis | Jacqueline Kennedy Onassis | Jacqueline Onassis
Reasons for Various Descriptions

- Text variations
- Local knowledge
- Evolving nature of data

**New functionality:**

- Web page extraction
  
e.g., Calais, Cogito
- Import data collections from various applications
  
e.g., Wikipedia data used in Freebase
- Mashups for easy and fast integration from various source
  
e.g., yahoo pipes
Entity Resolution typical methodology:

- Identify data describing the same real-world objects
- Decide how to merge the data
- Update the data collection

Solutions following various directions
We present them through four categories:

1. Atomic similarity methods
2. Similarity methods for sets
3. Facilitating inner-relationships
4. Methods in uncertain data
Alternative names for Entity Resolution

- entity-relationship
- name-matching
- reference
- match
- learning
- blocking
- similarity
- uncertainty
- survey
- cleansing
- probabilistic
- deduplication
- weights
- probabilistic
- scheme
- lineage
- object
- joins
- adaptive
- real-world
- metrics
- semantic
- matching
- integration
- fuzzy
- links
- heterogeneous
- iterative
- identification
- multi-relational
- ambiguous
- answers
- reconciliation
- measures
- matching
- record
- resolution
- names
- entity-aware
- merge/purge
- link
- cleaning
- detection
- duplicate
- name
- string
- approximate
- processing
- relationships
- grouping
- approach
- active
- distance
- clean relations
- domain-independent
- graph
- multi-source
- cluster
- duplicates
- record
- linkage
- objects
- clean relations
- group
Outline

1. Motivation: Entity Resolution
2. Atomic similarity methods
3. Similarity methods for sets
4. Facilitating inner-relationships
5. Methods in uncertain data
6. Conclusions
Atomic String Similarity

Examples of targeting cases:

- Publication authors: “John D. Smith” vs. “J. D. Smith”

Edit Distance:

- Number of operations to convert from 1\textsuperscript{st} to 2\textsuperscript{nd} string
- Operations in Levenstein distance \footnote{[Lev66]} \[\text{delete, insert, and update a character with cost 1}\]

\[
\begin{array}{l}
\text{ekaterini} \\
\text{katerina}
\end{array}
\]

\[
\begin{array}{l}
\text{cost = 2}
\end{array}
\]
Atomic String Similarity

Gap Distance:
- Overcome limitation of edit distance with shortened strings
- Considers two extra operations [Nav01]
  - open gap, and extend gap (with small cost)

\[
\text{cost} = 1 + o + 8e
\]
Atomic String Similarity

Jaro similarity [Jar89]:

- Small string, e.g., first and last names

\[
\text{JaroSim}(s_1, s_2) = \frac{1}{3} \left( \frac{C}{|s_1|} + \frac{C}{|s_2|} + \frac{C-T}{C} \right)
\]

- \(C\) → common characters in \(s_1\) and \(s_2\)
- \(T\) → transpositions/2 (transposition is a \(k\) in which \(s_1[k] \neq s_2[k]\)

Example: “DEIS” vs. “DESI”

\(C=4, T=2/2\), JaroSim = \[
\frac{1}{3} \left( \frac{4}{4} + \frac{4}{4} + \frac{4-1}{4} \right) = 0.9167
\]

Jaro-Winkler similarity [Win99]:

- Extension that gives higher weight to matching prefix
- Increasing it’s applicability to names
Atomic String Similarity

Soundex:
- Converts each word into a phonetic encoding by assigning the same code to the string parts that sound the same
- Similarity between the corresponding phonetic encodings

Remarks:
- Surveys: [CRF03], [Win06]
- Existing API with these methods:
  - SecondString: http://secondstring.sourceforge.net/
  - SimMetrics: http://www.dcs.shef.ac.uk/~sam/simmetrics.html
1. Motivation: Entity Resolution
2. Atomic similarity methods
3. Similarity methods for sets
4. Facilitating inner-relationships
5. Methods in uncertain data
6. Conclusions
Similarity methods for sets

Database community:
- Each record is an entity
- A simple example:

<table>
<thead>
<tr>
<th>Name</th>
<th>Email</th>
<th>Journal</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1</td>
<td>John D. Smith</td>
<td><a href="mailto:smith@uni.edu">smith@uni.edu</a></td>
</tr>
<tr>
<td>e2</td>
<td>Smith, J.</td>
<td><a href="mailto:smith@uni.edu">smith@uni.edu</a></td>
</tr>
</tbody>
</table>

Merge-purge [HS95],[HS98]:
- Idea: same entities will share information
- Create a key for each record (e.g., email)
- Sort records according to key
- Compare only a limited set of records in each iteration
Similarity methods for sets

Using transformations [TKM02]:

1. Analyze data to generate transformations
   - **Unary transform:**
     - Equality, Stemming, Soundex, Abbreviation (e.g., 3rd or third)
   - **N-ary transformations:**
     - Initial, Prefix, Suffix, Substring, Acronym, Abbreviation, Drop

2. Calculate transformation weights
3. Apply on candidate mappings
Similarity methods for sets

Group Linkage [OKLS07]:

- Considers groups of relational records
  - not individual relational records
- Groups match when:
  1. High similarity between data of individual records
  2. Large fraction of matching records, i.e., no. 1

Some additional methods

→ [DLLH03]

Surveys for methods in this category

→ [DH05], [EIV07], [OS99]
Remarks:
- Methods do not consider semantics of data
- Currently used as a first step of Entity Resolution
Outline

1. Motivation: Entity Resolution
2. Atomic similarity methods
3. Similarity methods for sets
4. Facilitating inner-relationships
5. Methods in uncertain data
6. Conclusions
Facilitating inner-relationships

General idea

- Heterogeneous data
  - Lack of schema information
  - Variations in entity descriptions
  - Incomplete or missing values
- Improve effectiveness by considering data semantics
- Example → Reference Reconciliation
Facilitating inner-relationships

Reference Reconciliation [DHM05]

1. Build a dependency graph

Diagram:
- (p₁, p₄) connected to (“Robert S. Epstein”, “Epstein, R.S.”)
- (p₂, p₅) connected to (“Michael Stonebraker”, “Stonebraker, M.”)
- (p₃, p₆) connected to (“Eugene Wong”, “Wong, E.”)
- (“Distributed…”, “Distributed …”) connected to (a₁, a₂)
- (“169-180”, “169-180”) connected to (c₁, c₂)

Legend:
- Reconciled
- Similar
Facilitating inner-relationships

Reference Reconciliation [DHM05]

1. Build a dependency graph
2. Exploit information and relationships

(`p_1`, `p_4`)  
(`Robert S. Epstein`, “Epstein, R.S.”)

(`p_2`, `p_5`)  
(`Michael Stonebraker`, “Stonebraker, M.”)

(`p_3`, `p_6`)  
(“Eugene Wong”, “Wong, E.”)

(`a_1`, `a_2`)  
(“Distributed…”, “Distributed…”)

(`c_1`, `c_2`)  
(“169-180”, “169-180”)

(`p_1`, `p_4`)  
(“Robert S. Epstein”, “Epstein, R.S.”)

(`p_2`, `p_5`)  
(`Michael Stonebraker`, “Stonebraker, M.”)

(`p_3`, `p_6`)  
(“Eugene Wong”, “Wong, E.”)

(`a_1`, `a_2`)  
(“Distributed…”, “Distributed…”)

(`c_1`, `c_2`)  
(“169-180”, “169-180”)

Reconciled  Similar
Facilitating inner-relationships

Reference Reconciliation [DHM05]

1. Build a dependency graph
2. Exploit information and relationships

(p₁, p₄) → ("Robert S. Epstein", "Epstein, R.S.") → (p₂, p₅) → ("Michael Stonebraker", "Stonebraker, M.") → (p₃, p₆) → ("Eugene Wong", "Wong, E.")

("Distributed…", "Distributed ...") → (a₁, a₂) → (c₁, c₂) → ("169-180", "169-180")

("ACM …", "ACM SIGMOD") → ("1978", "1978")

Reconciled

Similar
Facilitating inner-relationships

Reference Reconciliation [DHM05]

1. Build a dependency graph
2. Exploit information and relationships

Diagram:
- (p₁, p₄) → (“Robert S. Epstein”, “Epstein, R.S.”)
- (p₂, p₅) → (“Michael Stonebraker”, “Stonebraker, M.”)
- (p₃, p₆) → (“Eugene Wong”, “Wong, E.”)
- (“ Distributed…””, “ Distributed…””) → (a₁, a₂)
- (c₁, c₂) → (“169-180”, “169-180”)

Legend:
- Reconciled
- Similar
Facilitating inner-relationships

Reference Reconciliation [DHM05]

1. Build a dependency graph
2. Exploit information and relationships
3. Propagate information → enrich relationships
Analysis of entity-relationship graph [KM06], [KMC05]:

<table>
<thead>
<tr>
<th>Author (clean)</th>
<th>Publication (to be cleaned)</th>
</tr>
</thead>
</table>
Facilitating inner-relationships

Analysis of entity-relationship graph [KM06], [KMC05]:
1. Dataset modeled as a graph

![Entity-relationship graph diagram]

Author table (clean)
- \langle A1, ‘Dave White’, ‘Intel’ \rangle
- \langle A2, ‘Don White’, ‘CMU’ \rangle
- \langle A3, ‘Susan Grey’, ‘MIT’ \rangle
- \langle A4, ‘John Black’, ‘MIT’ \rangle
- \langle A5, ‘Joe Brown’, unknown \rangle
- \langle A6, ‘Liz Pink’, unknown \rangle

Publication table (to be cleaned)
- \langle P1, ‘Databases . . .’, ‘John Black’, ‘Don White’ \rangle
- \langle P2, ‘Multimedia . . .’, ‘Sue Grey’, ‘D. White’ \rangle
- \langle P3, ‘Title3 . . .’, ‘Dave White’ \rangle
- \langle P4, ‘Title5 . . .’, ‘Don White’, ‘Joe Brown’ \rangle
- \langle P5, ‘Title6 . . .’, ‘Joe Brown’, ‘Liz Pink’ \rangle
- \langle P6, ‘Title7 . . .’, ‘Liz Pink’, ‘D. White’ \rangle
Facilitating inner-relationships

Analysis of entity-relationship graph [KM06], [KMC05]:

1. Dataset modeled as a graph
2. Data more strongly connected when sharing relationships

Author table (clean)

- \( \langle A_1, 'Dave White', 'Intel' \rangle \)
- \( \langle A_2, 'Don White', 'CMU' \rangle \)
- \( \langle A_3, 'Susan Grey', 'MIT' \rangle \)
- \( \langle A_4, 'John Black', 'MIT' \rangle \)
- \( \langle A_5, 'Joe Brown', 'unknown' \rangle \)
- \( \langle A_6, 'Liz Pink', 'unknown' \rangle \)

Publication table (to be cleaned)

- \( \langle P_1, 'Databases . . .', 'John Black', 'Don White' \rangle \)
- \( \langle P_2, 'Multimedia . . .', 'Sue Grey', 'D. White' \rangle \)
- \( \langle P_3, 'Title3 . . .', 'Dave White' \rangle \)
- \( \langle P_4, 'Title5 . . .', 'Don White', 'Joe Brown' \rangle \)
- \( \langle P_5, 'Title6 . . .', 'Joe Brown', 'Liz Pink' \rangle \)
- \( \langle P_6, 'Title7 . . .', 'Liz Pink', 'D. White' \rangle \)
Facilitating inner-relationships

Analysis of entity-relationship graph [KM06], [KMC05]:

1. Dataset modeled as a graph
2. Data more strongly connected when sharing relationships
3. Measure the connection strengths (details in paper)
Facilitating inner-relationships

Some additional methods:

- **Relationship-based clustering** [BG04a], [BG04b]:
  - Common references for a match increase our belief
  - For this we need to identify common references
  - Iterative process: common matches → identifying additional matches

- **Incremental & adaptive** [INN08], [MPC+10]:
  - Targets data that are constantly changing and evolving
  - Bayesian network to model entities, relationships, and evidences (possible linkages)
  - Enables flexible update of the network

Surveys for methods in this category

→ [GD05], [KSS06]
Outline

1. Motivation: Entity Resolution
2. Atomic similarity methods
3. Similarity methods for sets
4. Facilitating inner-relationships
5. Methods in uncertain data
6. Conclusions
Methods in uncertain data

General idea:

- Keep conflicting relations, e.g., [AFM06], [RDS07], [DS07a], [DHY07]
  - Lack of resolution rules to correctly resolve and merge relations
  - No merging, but maintain results in the database
  - Relation are alternative representations of the same real world object

- Entity representation with probability – indicates...
  - Reliability of the source
  - Output of the matching process
  - Etc.

<table>
<thead>
<tr>
<th>custId</th>
<th>name</th>
<th>income</th>
<th>prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>c1</td>
<td>John</td>
<td>$120K</td>
</tr>
<tr>
<td>s2</td>
<td>c1</td>
<td>John</td>
<td>$80K</td>
</tr>
<tr>
<td>s3</td>
<td>c2</td>
<td>Mary</td>
<td>$140K</td>
</tr>
<tr>
<td>s4</td>
<td>c2</td>
<td>Marion</td>
<td>$40K</td>
</tr>
</tbody>
</table>
Methods in uncertain data

Clean answers over dirty databases [AFM06]:

- Dirty database represents several possible databases
- Result set for queries should include the entity resolution results
- Query rewriting mechanism with efficient computation of probability for each answer

<table>
<thead>
<tr>
<th>order</th>
<th>id</th>
<th>orderId</th>
<th>custFk</th>
<th>cIdFk</th>
<th>quantity</th>
<th>prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁</td>
<td>o1</td>
<td>11</td>
<td>m1</td>
<td>c1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>t₂</td>
<td>o2</td>
<td>12</td>
<td>m2</td>
<td>c1</td>
<td>2</td>
<td>0.5</td>
</tr>
<tr>
<td>t₃</td>
<td>o2</td>
<td>13</td>
<td>m3</td>
<td>c2</td>
<td>5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>customer</th>
<th>id</th>
<th>custId</th>
<th>name</th>
<th>balance</th>
<th>prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₄</td>
<td>c1</td>
<td>m1</td>
<td>John</td>
<td>$20K</td>
<td>0.7</td>
</tr>
<tr>
<td>t₅</td>
<td>c1</td>
<td>m2</td>
<td>John</td>
<td>$30K</td>
<td>0.3</td>
</tr>
<tr>
<td>t₆</td>
<td>c2</td>
<td>m3</td>
<td>Mary</td>
<td>$27K</td>
<td>0.2</td>
</tr>
<tr>
<td>t₇</td>
<td>c2</td>
<td>m4</td>
<td>Marion</td>
<td>$5K</td>
<td>0.8</td>
</tr>
</tbody>
</table>

\[ D_{1}^{cd} = \{t_1, t_2, t_4, t_6\} \]
\[ D_{2}^{cd} = \{t_1, t_2, t_4, t_7\} \]
\[ D_{3}^{cd} = \{t_1, t_2, t_5, t_6\} \]
\[ D_{4}^{cd} = \{t_1, t_2, t_5, t_7\} \]
\[ D_{5}^{cd} = \{t_1, t_3, t_4, t_6\} \]
\[ D_{6}^{cd} = \{t_1, t_3, t_4, t_7\} \]
\[ D_{7}^{cd} = \{t_1, t_3, t_5, t_6\} \]
\[ D_{8}^{cd} = \{t_1, t_3, t_5, t_7\} \]
Methods in uncertain data

Clean answers over dirty databases [AFM06]:

- **Query rewriting**

  ```sql
  select A_1, ..., A_n 
  from R_1, ..., R_m 
  where W
  ```

  ```sql
  select A_1, ..., A_n, sum(R_1.prob.*...* R_m.prob) 
  from R_1, ..., R_m 
  where W
  group by A_1, ..., A_n
  ```

  - Groups the result by the attributes
  - For each group: sums the product of relation probabilities
  - (applicable only to rewritable queries)
Entity-Aware querying over prob. linkages [INNV10]:

- Not merging the entities using threshold
- Keep probabilistic linkages alongside the original data
- Use them during query processing

Query:

- “J. K. Rowling” movies in “2002”

Assume no linkages:

- zero results

Possible answer with linkages:

- merge($e_1$, $e_2$)
- merge($e_1$, $e_2$, $e_3$)
Entity-Aware querying over prob. linkages [INNV10]:

- Linkage prob. represent several possible $l$-worlds
- Attribute prob. represent several possible worlds

Efficient query processing:
- Analyze query conditions
- Identify the required entity merges
- Decide useful possible $l$-worlds
- Generate possible worlds
- Compute probability

- **title**: Harry Potter and the Chamber of Secrets 0.6
  - starring: Daniel Radcliffe 0.7
  - starring: Emma Watson 0.4
  - writer: J.K. Rowling 0.6
  - genre: Fantasy 0.6

- **title**: Harry Potter and the Chamber of Secrets 0.7
  - date: 2002 0.8
  - starring: Daniel Radcliffe 0.5
  - starring: Emma Watson 0.9

- **title**: Harry Potter and the Chamber of Secrets 0.8
  - genre: Fantasy 0.8
  - author: J.K. Rowling 0.7
Outline

1. Motivation: Entity Resolution
2. Atomic similarity methods
3. Similarity methods for sets
4. Facilitating inner-relationships
5. Methods in uncertain data
6. Conclusions
Conclusions

Discussed methods entity resolution

Four categories of methods

Not presented:

- **Blocking mechanisms:**
  - Split data into blocks and compare inner-block data
  - Improves efficiency for large-size datasets
  - Examples: [WMK+09], [PINF11]

- **Active learning approaches:**
  - Use a subset of the data to learn matching rules
  - Apply the rules to remaining data
  - Examples: [SB02], [CR01]

- **Similarity Joins** [GIJ+1]

- **Schema matching**
  - ....
Bibliography


Bibliography (II)


Bibliography (III)


