My Background

2005 2010 2013 2014 2016 now

Hasso Plattner Institut
BSc, MSc, PhD
Prof. Dr. Felix Naumann
• Mining Open Data
• Dependency Discovery
• Data Visualization

Intern

2010

Prof. Dr. Felix Naumann

2013

PostDoc
Prof. Michael Stonebraker

2014

• Data Cleaning
• Data Transformation

2016

Assistant Professor

• Data Integration
• Data Profiling
• Data Visualization

MIT
Emergence of Data Driven Applications

But, what do data scientists actually do?

Data Curation in the Wild: Limits and Challenges
Data Scientist, the sexiest job of 21st century requires a mixture of multidisciplinary skills ranging from an intersection of mathematics, statistics, computer science, communication and business. Finding a data scientist is hard. Finding people who understand who a data scientist is, is equally hard. So here is a little cheat sheet on who the modern data scientist really is.

**Math & Statistics**
- Machine learning
- Statistical modeling
- Experiment design
- Bayesian inference
- Supervised learning: decision trees, random forests, logistic regression
- Unsupervised learning: clustering, dimensionality reduction
- Optimization: gradient descent and variants

**Programming & Database**
- Computer science fundamentals
- Scripting language e.g. Python
- Statistical computing packages, e.g. R
- Databases: SQL and NoSQL
- Relational algebra
- Parallel databases and parallel query processing
- MapReduce concepts
- Hadoop and Hive/Pig
- Custom reducers
- Experience with xaaS like AWS
CrowdFlower’s Data Science Report 2016

*Data preparation accounts for about 80% of the work of data scientists*

Data Preparation is basically Data Integration

Data Integration:

- Data cleaning
- Schema matching/mapping
- Data transformation
- Entity resolution

Mediated Schema
Challenges in Data Integration

• Manual Curation:
  • Infeasible on large datasets (Big Data)
  • Even the user does not anticipate and see all possible problems

• Automatic algorithms (Previous Research):
  • Require well-defined input models
  • Not general enough to capture
  • If end-to-end: User has to know everything
Outline

1. Data cleaning:
   • Error detection benchmark
2. Data transformation:
   • Semi-automatic transformation discovery
3. Tractable Data integration
   • Workflow generation
Error Detection

• Extensive previous research on many different cleaning algorithms
  • Usually evaluated on errors injected into clean data
    • Which we find unconvincing (finding errors you injected...)
  • Tools evaluated with tools of the same category
    • Well-defined but narrow error models

• How well do current techniques work “in the wild”? 
• What about combinations of techniques?
What we did

1. Analyzed 5 different datasets
   • Identified general error types that can be discovered by tools
2. Selected 8 different error detection systems
3. Measured
   • effectiveness of each single system
   • combined effectivity
   • upper-bound recall
4. Analyzed impact of enrichment
5. Tried out domain specific cleaning tools
Error Types

- Literature:
- Our model:
  - Error = A value that is different from groundtruth
- General types:

```
  Quantitative
    └── Outliers
    └── Pattern violations
  Qualitative
    └── Duplicates
    └── Constraint violations
```
Error Detection Strategies

1. Rule-based detection algorithms
   • Detecting violation of constraints, such as Zip Code → City

2. Pattern verification and enforcement tools
   • Syntactical patterns, such as date formatting
   • Semantical patterns, such as location names

3. Quantitative algorithms
   • Statistical outliers

4. Deduplication
   • Discovering conflicting attribute values in duplicates
Tool Selection

• Premise:
  • Tool is State-of-the-Art
  • Tool is sufficiently general
  • Tool is available
  • Tool covers at least one of the 4 error types:

<table>
<thead>
<tr>
<th></th>
<th>DBoost</th>
<th>DC-Clean</th>
<th>OpenRefine</th>
<th>Trifacta</th>
<th>Pentaho</th>
<th>KNIME</th>
<th>Katara</th>
<th>Tamr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern violations</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Constraint violations</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Outliers</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Duplicates</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Data Curation in the Wild: Limits and Challenges
## 5 Data Sets continued

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Description</th>
<th># columns</th>
<th># rows</th>
<th>Ground truth</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIT VPF</td>
<td>Procurement dataset</td>
<td>42</td>
<td>24K</td>
<td>13k (partial)</td>
<td>6.7%</td>
</tr>
<tr>
<td>Merck</td>
<td>List of IT Services</td>
<td>61</td>
<td>2262</td>
<td>2262</td>
<td>19.7%</td>
</tr>
<tr>
<td>Animal</td>
<td>Documentation of animal captures</td>
<td>14</td>
<td>60k</td>
<td>60k</td>
<td>0.1%</td>
</tr>
<tr>
<td>Rayyan Bib</td>
<td>Consolidated literature references</td>
<td>11</td>
<td>1M</td>
<td>1k (partial)</td>
<td>35%</td>
</tr>
<tr>
<td>BlackOak</td>
<td>Address data (artificial errors)</td>
<td>12</td>
<td>94k</td>
<td>94k</td>
<td>34%</td>
</tr>
</tbody>
</table>

### Pattern violations
- MIT VPF ✔
- Merck ✔
- Animal ✔
- Rayyan Bib ✔
- BlackOak ✔

### Constraint violations
- MIT VPF ✔
- Merck ✔
- Animal ✔
- Rayyan Bib ✔
- BlackOak ✔

### Outliers
- MIT VPF ✔
- Merck ✔
- Animal ✔
- Rayyan Bib ✔
- BlackOak ✔

### Duplicates
- MIT VPF ✔
- Merck ✔
- Animal ✔
- Rayyan Bib ✔
- BlackOak ✔
Evaluation Methodology

• We have the same knowledge as the data owners about the data:
  • Quality constraints, business rules
• Best effort in using all capabilities of the tools
  • However: No heroics, i.e., embedding custom java code within a tool
• KPIs:
  • Precision = \( \frac{\text{correctly detected errors}}{\text{marked as error}} \)
  • Recall = \( \frac{\text{correctly detected errors}}{\text{existing errors}} \)
  • F-Measure = \( \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \)
# Single Tool Performance: All Datasets

<table>
<thead>
<tr>
<th>Tools</th>
<th>MIT VPF</th>
<th>Merck</th>
<th>Animal</th>
<th>Rayyan Bib</th>
<th>BlackOak</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F</td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>DC-Clean</td>
<td>.25</td>
<td>.14</td>
<td>.18</td>
<td>.99</td>
<td>.78</td>
</tr>
<tr>
<td>Trifacta</td>
<td>.94</td>
<td>.86</td>
<td>.90</td>
<td>.99</td>
<td>.78</td>
</tr>
<tr>
<td>OpenRefine</td>
<td>.95</td>
<td>.86</td>
<td>.90</td>
<td>.99</td>
<td>.78</td>
</tr>
<tr>
<td>Pentaho</td>
<td>.95</td>
<td>.59</td>
<td>.73</td>
<td>.99</td>
<td>.78</td>
</tr>
<tr>
<td>KNIME</td>
<td>.95</td>
<td>.86</td>
<td>.90</td>
<td>.99</td>
<td>.78</td>
</tr>
<tr>
<td>Gaussian</td>
<td>.07</td>
<td>.07</td>
<td>.07</td>
<td>.19</td>
<td>.00</td>
</tr>
<tr>
<td>Histogram</td>
<td>.13</td>
<td>.11</td>
<td>.12</td>
<td>.13</td>
<td>.02</td>
</tr>
<tr>
<td>GMM</td>
<td>.14</td>
<td>.29</td>
<td>.19</td>
<td>.17</td>
<td>.32</td>
</tr>
<tr>
<td>Katara</td>
<td>.40</td>
<td>.01</td>
<td>.02</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Tamr</td>
<td>.16</td>
<td>.02</td>
<td>.04</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Union</td>
<td>.24</td>
<td>.93</td>
<td>.38</td>
<td>.33</td>
<td>.85</td>
</tr>
</tbody>
</table>
Combined Tool Performance

• Naïve approach
  • k tools agree on a value to be an error
    • Typical precision-recall trade-off

• Maximum entropy-based order selection:
  1. Run tools on samples and verify the results
  2. Pick the tool with highest precision
  3. Verify the results
  4. Update precision and recall of other tools accordingly
  5. Repeat step 2

Drop tools with precision below 10%
Ordering-based approach

- Precision and recall depending on different minimum precision thresholds (compared to union)
Maximum possible recall

- Manually checked each undetected error
- Reasoned whether the error could have been detected by a better variant of a tool, e.g. a more sophisticated rule or transformation.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Best effort recall</th>
<th>Upper-bound recall</th>
<th>Remaining errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIT VPF</td>
<td>0.92</td>
<td>0.98 (+1,950)</td>
<td>798</td>
</tr>
<tr>
<td>Merck</td>
<td>0.85</td>
<td>0.99 (+4,101)</td>
<td>58</td>
</tr>
<tr>
<td>Animal</td>
<td>0.57</td>
<td>0.57</td>
<td>592</td>
</tr>
<tr>
<td>Rayyan Bib</td>
<td>0.85</td>
<td>0.91 (+231)</td>
<td>347</td>
</tr>
<tr>
<td>BlackOak</td>
<td>0.99</td>
<td>0.99</td>
<td>75</td>
</tr>
</tbody>
</table>
Enrichment and Domain-specific tools

• Enrichment
  • Manually append more columns by joining to other tables of the database
    ➢ Improves performance of rule-based and duplicate detection systems

• Domain-specific tool:
  • Used a commercial address cleaning service
    ➢ High precision on the specific domain
    ➢ But did not lead to the increase of overall recall
What about Repair?

- **Automatic repair of errors:**
  - Similarity join with a dictionary
  - Sophisticated probabilistic approaches
    - Truth discovery [Yin, Han, Yu. KDD’07]
  - Conflict resolution
    - Data Fusion [Bleiholder, Naumann. ACM Surveys’09]
  - Crowd-sourcing
    - CrowdER [Wang et al. PVLDB’12])

- **Semi-automatic:**
  - User-defined transformation
    - Already hard enough!
Booking a flight

![Flight Booking](image)
### Different value representations

<table>
<thead>
<tr>
<th>Departure</th>
<th>Destination</th>
<th>Cabin</th>
<th>Time</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Berlin</td>
<td>Berlin – New York NY - J.F. Kennedy</td>
<td>Economy Classic</td>
<td>13:00</td>
<td>1.139,42 €</td>
</tr>
<tr>
<td>TXL</td>
<td>JFK</td>
<td>Choice</td>
<td>1:00 pm</td>
<td>2518 $</td>
</tr>
<tr>
<td>Berlin DE</td>
<td>Berlin, DE (TXL - Tegel)</td>
<td>Economy (lowest)</td>
<td>08:45 am</td>
<td>$2991</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Departure</th>
<th>Destination</th>
<th>Cabin</th>
<th>Time</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Berlin (TXL)</td>
<td>New York (JFK)</td>
<td>Economy</td>
<td>1:00p</td>
<td>$1454</td>
</tr>
<tr>
<td>Berlin (TXL)</td>
<td>New York (JFK)</td>
<td>Economy</td>
<td>1:00p</td>
<td>$2518</td>
</tr>
<tr>
<td>Berlin (TXL)</td>
<td>New York (JFK)</td>
<td>Economy</td>
<td>8:45a</td>
<td>$2991</td>
</tr>
</tbody>
</table>
Transformation Tasks

Airport code $\rightarrow$ City

<table>
<thead>
<tr>
<th>Airport code</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOS</td>
<td>Boston</td>
</tr>
<tr>
<td>JFK</td>
<td>New York</td>
</tr>
<tr>
<td>ORD</td>
<td>Chicago</td>
</tr>
<tr>
<td>BER</td>
<td>Berlin</td>
</tr>
<tr>
<td>CDG</td>
<td>Paris</td>
</tr>
</tbody>
</table>

• Model $\rightarrow$ Brand
  • Iphone 6 $\rightarrow$ Apple Inc.

• ISBN $\rightarrow$ Title
  • 0-553-57340-3 $\rightarrow$ “A Game of Thrones”

• Unit conversion
  • 1 Mile $\rightarrow$ 1.6 km

• Long/Lat $\rightarrow$ location

• Country $\rightarrow$ Currency
  • ...
Problem Statement: Example-based Discovery

<table>
<thead>
<tr>
<th>Given</th>
<th>Find</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Airport</strong></td>
<td><strong>Airport</strong></td>
</tr>
<tr>
<td>BER</td>
<td>BER</td>
</tr>
<tr>
<td>JFK</td>
<td>JFK</td>
</tr>
<tr>
<td>ORD</td>
<td>ORD</td>
</tr>
<tr>
<td>HBE</td>
<td>HBE</td>
</tr>
<tr>
<td>IST</td>
<td>IST</td>
</tr>
<tr>
<td>FRA</td>
<td>FRA</td>
</tr>
<tr>
<td>BOS</td>
<td>BOS</td>
</tr>
<tr>
<td>DFW</td>
<td>DFW</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>City</strong></th>
<th><strong>City</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Berlin</td>
<td>Berlin</td>
</tr>
<tr>
<td>New York</td>
<td>New York</td>
</tr>
<tr>
<td>Chicago</td>
<td>Chicago</td>
</tr>
<tr>
<td>?</td>
<td>Alexandria</td>
</tr>
<tr>
<td>?</td>
<td>Istanbul</td>
</tr>
<tr>
<td>?</td>
<td>Frankfurt</td>
</tr>
<tr>
<td>?</td>
<td>Boston</td>
</tr>
<tr>
<td>?</td>
<td>Dallas</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Example transformations
DataXFormer: Initial Prototype

Transformation task: Input column + examples

Experts Web Forms Web Tables

Initial study with 50 transformations from the “wild”: Prototype covered 82% with average precision/recall above 80%

Solution Presenter

Ranked transformations with scores
New Workload from the “Wild”

• 120 Transformation Tasks

1. Fahrenheit to Celsius
2. miles to km
3. pound to kg
4. USD to EUR
5. zip to state
6. zip to city
7. UPS tracking to address
8. English to German
9. swift code to bank
10. hex to RGB
11. ISBN to publisher
12. ISBN to title
13. ISBN to author
14. ISSN to title
15. ip address to country
16. Domain to primary ip
17. sentence to language
18. text to encoding
19. Gregorian to Hijri
20. CUSIP to company
21. CUSIP to ticker
22. symbol to company
23. iban to bank name
24. Location to temperature
25. location to humidity
26. car plate to details
27. city to country
28. city to long/lat
29. Entity to Wikipedia link
30. person to title
31. ip to domain
32. company to parent company
33. company to employee
34. country to sqkm
35. team to coach
36. mountains to feet
37. mountains to meter
38. unescosite to country
39. team to arena
40. soccer player to national team
41. bank to country
42. Video game to publisher
43. Movie to year
44. country to code
45. State to state abbrev
46. time zone to abbrev
47. us to eur
48. airport code to country
49. title + year to artist
50. soccer player to team
51. soccer player to industry
52. university to state
53. term to abbreviation
54. unescosite to country
55. country to ISO 3 letters
56. country to ISO 2 letters
57. movie+actor to role
58. shoe size
59. us to eur
60. domain to ip
61. company to CEO
62. company to industry
63. US standard to metric
64. fractions to decimals
65. country to code
66. temperature to location
67. humidity to location
68. person to twitter id
69. ipto domain
70. company to address
71. multi-column transformations
72. multi-column transformations
73. multi-column transformations
74. multi-column transformations
75. multi-column transformations
76. multi-column transformations
77. multi-column transformations
78. multi-column transformations
79. multi-column transformations
80. multi-column transformations
81. multi-column transformations
82. multi-column transformations
83. multi-column transformations
84. multi-column transformations
85. multi-column transformations
86. multi-column transformations
87. multi-column transformations
88. multi-column transformations
89. multi-column transformations
90. multi-column transformations
91. multi-column transformations
92. multi-column transformations
93. multi-column transformations
94. multi-column transformations
95. multi-column transformations
96. multi-column transformations
97. multi-column transformations
98. multi-column transformations
99. multi-column transformations
100. multi-column transformations
101. multi-column transformations
102. multi-column transformations
103. multi-column transformations
104. multi-column transformations
105. multi-column transformations
106. multi-column transformations
107. multi-column transformations
108. multi-column transformations
109. multi-column transformations
110. multi-column transformations
111. multi-column transformations
112. multi-column transformations
113. multi-column transformations
114. multi-column transformations
115. multi-column transformations
116. multi-column transformations
117. multi-column transformations
118. multi-column transformations
119. multi-column transformations
120. multi-column transformations

Initial prototype covered only 52%

Unesco site ➔ Country
For some transformations input and output do not co-occur in any table.

Title + year ➔ artist
(10) Multi-column transformations

Player ➔ teams
(31) Non-functional mappings

Data Curation in the Wild: Limits and Challenges
DataXFormer Improvements

Transformation task: Input column + examples

Experts

Web Tables:
• Multi-column transformations
• Indirect transformations
• Non-functional transformations

With the new functionality:
Covered again 84%

Solution Presenter

Ranked transformations with scores

Knowledge Base:
YAGO, DBpedia

Data Curation in the Wild: Limits and Challenges
Challenges with Web Tables

- Many irrelevant tables
  - 120 million in total (TU Dresden corpus)

- Overcome fragmentation
  - Average row count = 12

- Dirty and heterogeneous

Filter and Refine approach

- Multiple iterations
- Example augmentation
- Indirect mappings

Rate transformations based on example hits and reconciliation
Storage and Indexing

```
SELECT colX1.tableid, colX2.colid, [colX2.colid, ...], colYid
FROM
  (SELECT tableid, colid
   FROM Cells
   WHERE term_tokenized IN (X1)
   GROUP BY tableid, colid
   HAVING COUNT(DISTINCT tokenized) >= tau) AS colX1,

  (SELECT tableid, colid
   FROM Cells
   WHERE term_tokenized IN (X2)
   GROUP BY tableid, colid
   HAVING COUNT(DISTINCT tokenized) >= tau) AS colX2,

  (SELECT tableid, colid
   FROM Cells
   WHERE term_tokenized IN (Y)
   GROUP BY tableid, colid
   HAVING COUNT(DISTINCT tokenized) >= tau) AS colY

WHERE colX1.tableid = colY.tableid
  [AND colX1.tableid = colX2.tableid AND ...]
AND colX1.colid <> colY.colid
  [AND colX1.colid <> colX2.colid AND colX2.colid <> colY.colid
  AND ...]
```
Indirect transformations

- No table with the mapping:
  - Unesco site → Country

- However:
  - There are join paths

- Leverage FD relationships to prune the search space

<table>
<thead>
<tr>
<th>Unesco Site</th>
<th>Location</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hahoe Folk Village</td>
<td>Hahoe</td>
<td>...</td>
</tr>
<tr>
<td>Old Havana and its Fortifications</td>
<td>La Habana</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>City</th>
<th>...</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hahoe</td>
<td>...</td>
<td>South Korea</td>
</tr>
<tr>
<td>La Habana</td>
<td>...</td>
<td>Cuba</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Non-functional Transformations

• Functional transformations:
  • Only one mapping needed
  • Pick the transformation with the highest score

• Non-functional transformations:
  • Multiple mappings
  • Which results and how many should be picked?
Non-functional Transformations with User Feedback

- Automatic clustering has very low precision
  - Agglomerative clustering based on score
  - Bayesian network (Truth Discovery)
- The user has to identify **counter examples**
- Avoid validating the whole result set
- Tested various sampling strategies
  - Score-based
  - Frequency-based
  - Diversity based

<table>
<thead>
<tr>
<th>Approach</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agglomerative clustering</td>
<td>0.45</td>
<td>0.25</td>
<td>0.3</td>
</tr>
<tr>
<td>Bayesian network</td>
<td>0.35</td>
<td><strong>0.63</strong></td>
<td>0.45</td>
</tr>
<tr>
<td>Supervised methods</td>
<td><strong>0.86</strong></td>
<td>0.58</td>
<td><strong>0.69</strong></td>
</tr>
</tbody>
</table>

- Tested various sampling strategies
  - Score-based
  - Frequency-based
  - Diversity based

- Data Curation in the Wild: Limits and Challenges
Coverage

120 transformations, average precision: 87% average recall: 76%

DataXFormer (101)

Web tables (76)

Web forms (28)

KBs (34)

Unesco site → Country
Year + Album → Artist
Country → Airports

Fahrenheit → Celsius
USD → EUR
Meter → miles

Bank+ City → Swift
Person → Google Graph ID
Car Plate → Details

Movie → Year
ISBN → Title
City → Country

Data Curation in the Wild: Limits and Challenges
## Coverage

<table>
<thead>
<tr>
<th>Approach</th>
<th># Queries</th>
<th>Tables</th>
<th>Forms</th>
<th>KB</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single column</td>
<td>79</td>
<td>50 (63%)</td>
<td>28 (23%)</td>
<td>17 (22%)</td>
<td>68 (86%)</td>
</tr>
<tr>
<td>Multi Column</td>
<td>10</td>
<td>7 (70%)</td>
<td>0</td>
<td>0</td>
<td>7 (70%)</td>
</tr>
<tr>
<td>Non-Functional</td>
<td>31</td>
<td>19 (61%)</td>
<td>0</td>
<td>17 (55%)</td>
<td>25 (81%)</td>
</tr>
<tr>
<td>Total</td>
<td>120</td>
<td>76 (63%)</td>
<td>28 (23%)</td>
<td>34 (28%)</td>
<td>101 (84%)</td>
</tr>
</tbody>
</table>
Lessons Learned So far

• Error Detection
  • There is no single dominant tool.
  • Improving individual tools has marginal benefit.
  • Picking the right order of tools can reduce the cost of validation by humans.

• DataXFormer
  • Web resources are a treasure trove for finding transformations
  • Domain-specific tasks require domain-specific transformation repositories
  • We want to try DataXFormer inside the firewall
Ideal Solution: Tractable Data Integration

• A system that
  • Embeds various data integration techniques
  • Extracts useful meta-data
  • Semi-automatically generates integration workflows
  • Enables the user to track and influence workflow steps and rationales

Efficient Profiling Techniques
Data Visualization & Summarization
Workflow Generator
The Tractable Data Integration System

User

Data Summarizer & Visualizer

Profiler

Workflow Generator

Workflow Repository

Input Data

Generated Workflow

Alg 1

Alg 2

Alg k-1

Alg k

Result k

Results 1

Results 2

Results k-1
Open Challenges

- Mapping meta-data to curation tasks
  - E.g., functional dependencies vs. rule-based systems
  - Different types of meta-data: we need holistic profiling algorithms
- Can we learn from previous data curation workflows?
  - How to index previously performed workflows?
  - Can we optimize?
- Involving the user
  - What kind of feedback can be incorporated?
  - Which meta-data to visualize?
  - Which sample should the user see?
Conclusions

(1) More reasoning on holistic combination of tools

(2) More reasoning on real-world data

(3) Interactive dashboard: Multiple iterations are inevitable