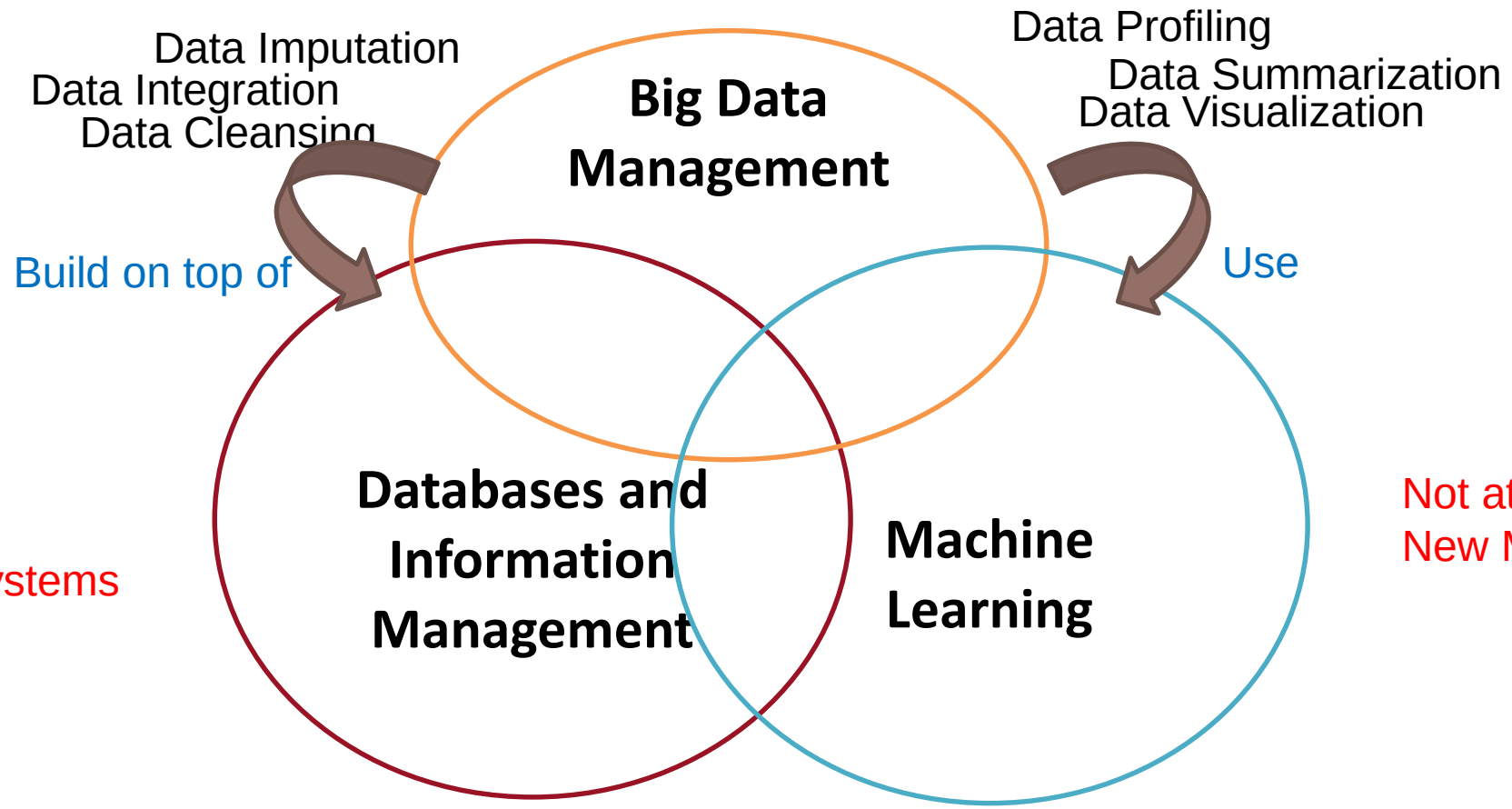
Data Curation in the Wild: Limits and Challenges

Ziawasch Abedjan

Big Data Excellence Workshop 2017

Big Data Management Group (BigDaMa)

Focus: Variety of Big Data



Not so much:
New storage systems

Not at all:
New ML models

Emergence of Data Driven Applications

Big data: The next frontier for innovation, competition, and productivity

McKinsey&Company

Harvard Business Review

DATA

Data Scientist: The Sexiest Job of the 21st Century

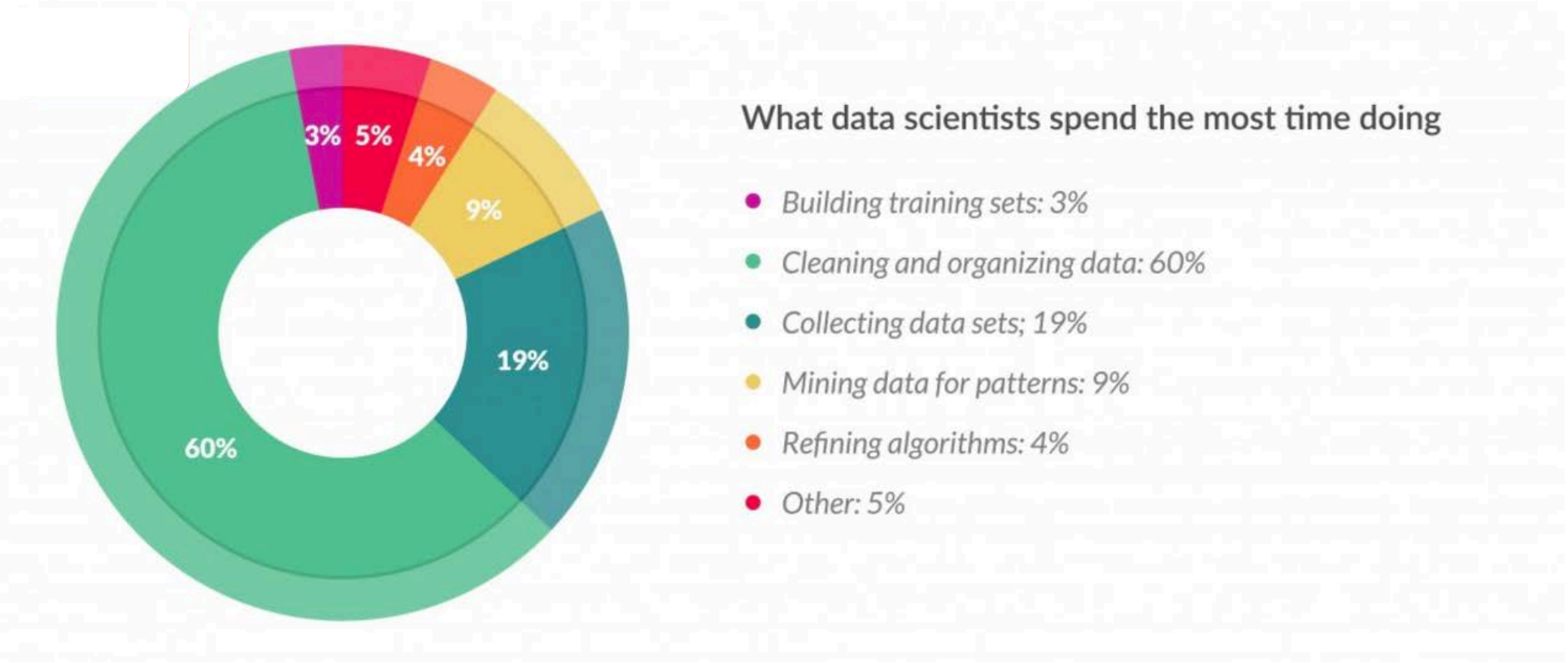
by Thomas H. Davenport and D.J. Patil

FROM THE OCTOBER 2012 ISSUE

But, what do data scientists actually do?

CrowdFlower's Data Science Report 2016

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



"Cleaning Data: Most Time-Consuming, Least Enjoyable Data Science Task", Gil Press, Forbes, March 23rd, 2016

Challenges in Data Curation

- Manual Curation:
  - Infeasible on large datasets (Big Data)
  - Even the user does not anticipate and see all possible problems
- Automatic algorithms (Previous Research):
  - Well-defined models
  - Not general enough to capture all possible data quality problems
  - If end-to-end: User has to know everything

Outline

1. Data cleaning:
   - Error detection case study
2. Data transformation:
   - Semi-automatic transformation discovery
3. Tractable Data integration
   - Data Civilizer
   - Workflow generation

Error Detection

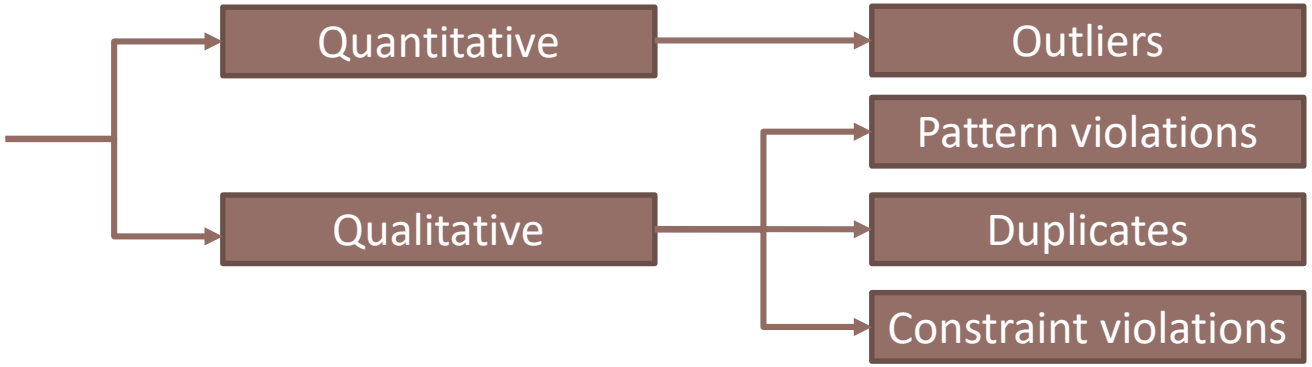
- Extensive previous research on many different cleaning algorithms
  - Usually evaluated on errors injected into clean data
    - Which we find unconvincing (finding errors you injected…)
  - Tools evaluated with tools of the same category
    - Well-defined but narrow error models
- How well do current techniques work “in the wild”?
- What about combinations of techniques?

What we did [PVLDB'16]

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:
  - [Hellerstein 2008, Ilyas&Chu 2015,Kim et al. 2003, Rahm&Do 2000]
- Our model:
  - Error = A value that is different from groundtruth
- General types:



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

Zip =“10001” & City = “NYC”
Vs.
Zip=“10001” & City =”Boston”

mm/dd/yy vs. dd.mm.yyyy

City name = “Peter Pane”

Salary = -1,0000,000 $

$record_1 = record_2$
vs.
$record_1[name] \neq record_2[name]$

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:

	DBoost	DC-Clean	OpenRefine	Trifacta	Pentaho	KNIME	Katara	Tamr
Pattern violations			✓	✓	✓	✓	✓	
Constraint violations		✓						
Outliers	✓			✓				
Duplicates								✓

5 Data Sets

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

	MIT VPF	Merck	Animal	Rayyan Bib	Blackoak
Pattern violations	✓	✓	✓	✓	✓
Constraint violations	✓	✓	✓	✓	✓
Outliers	✓	✓		✓	✓
Duplicates	✓				✓

Assumptions and 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 = $\frac{correctly\ detected\ errors}{marked\ as\ error}$
  - Recall = $\frac{correctly\ detected\ errors}{existing\ errors}$
  - F-Measure = $\frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$

Single Tool Performance: All Datasets

Tools		MIT VPF			Merck			Animal			Rayyan Bib			BlackOak		
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
DC-Clean		.25	.14	.18	**.99**	**.78**	**.87**	.12	**.53**	**.20**	.74	.55	.63	.46	.43	.44
Trifacta		.94	**.86**	**.90**	**.99**	**.78**	**.87**	**1.0**	.03	.06	.71	.59	.65	.96	.93	.94
OpenRefine		**.95**	**.86**	**.90**	**.99**	**.78**	**.87**	.33	.001	.20	**.95**	**.60**	**.74**	**.99**	**.95**	**.97**
Pentaho		**.95**	.59	.73	**.99**	**.78**	**.87**	.33	.001	.20	.71	.58	.64	**1.0**	.66	.79
KNIME		**.95**	**.86**	**.90**	**.99**	**.78**	**.87**	.33	.001	.20	.71	.58	.64	**1.0**	.66	.79
DBoost	Gaussian	.07	.07	.07	.19	.00	.01	.00	.00	.00	.41	.13	.20	.91	.73	.81
	Histogram	.13	.11	.12	.13	.02	.04	.00	.00	.00	.40	.16	.23	.52	.51	.52
	GMM	.14	.29	.19	.17	.32	.22	.00	.00	.00	.53	.39	.44	.38	.37	.38
Katara		.40	.01	.02	--	--	--	.55	.04	.07	.60	.39	.47	.88	.06	.11
Tamr		.16	.02	.04	--	--	--	--	--	--	--	--	--	.41	.63	.50
Union		.24	.93	.38	.33	.85	.48	.13	.58	.21	.47	.85	.61	.39	.99	.56

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.

Dataset	Best effort recall	Upper-bound recall	Remaining errors
MIT VPF	0.92	0.98 (+1,950)	798
Merck	0.85	0.99 (+4,101)	58
Animal	0.57	0.57	592
Rayyan Bib	0.85	0.91 (+231)	347
BlackOak	0.99	0.99	75

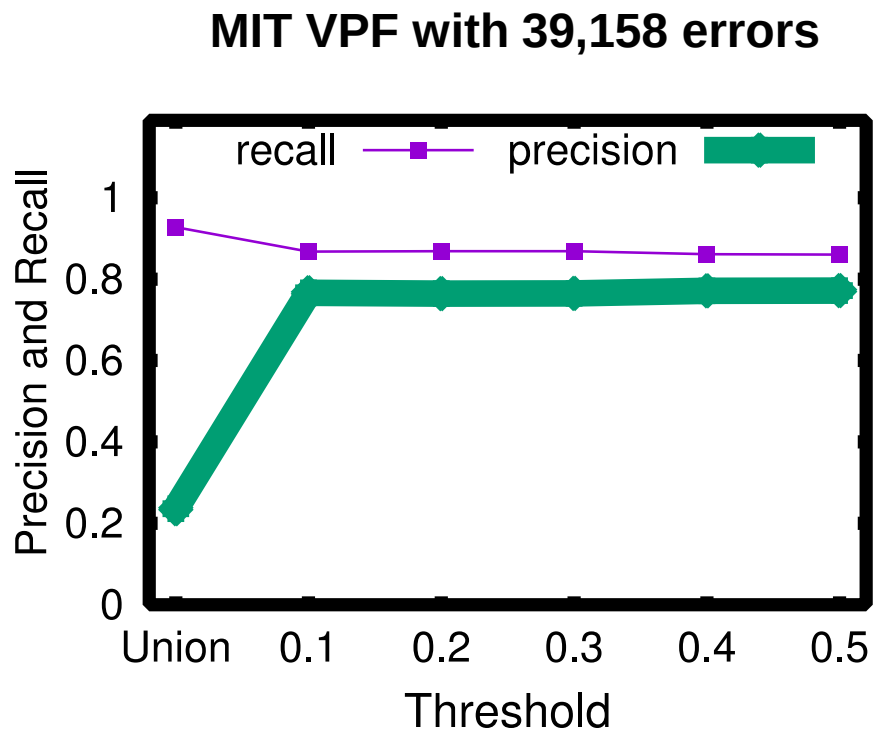
Combined Tool Performance

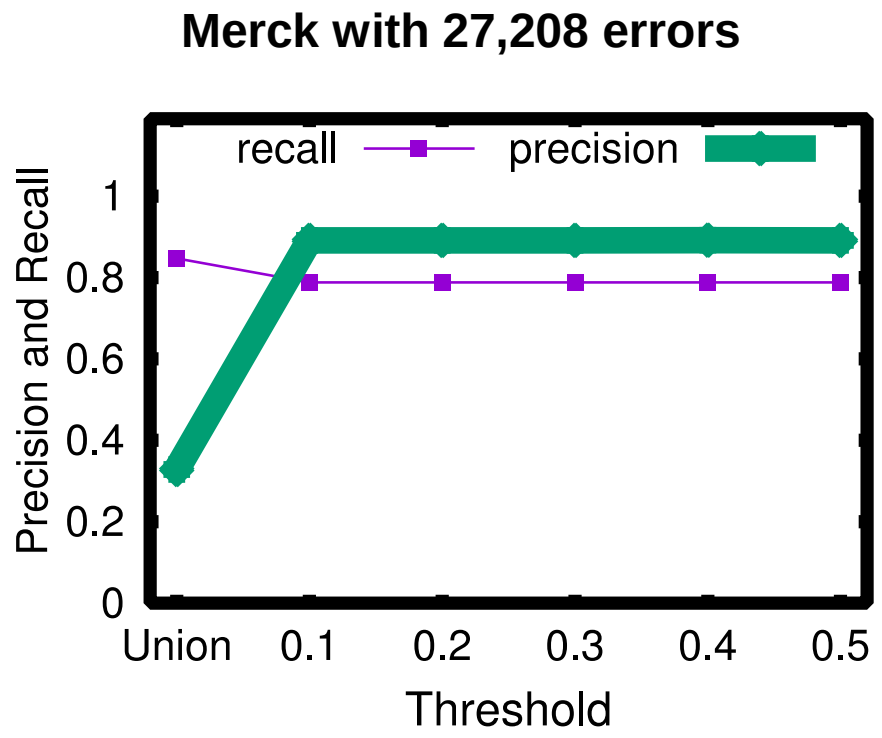
- Naïve approach
  - k tools agree on a value to be an error
    - Typical precision-recall trade-off
- Maximum entropy-based order selection:
  1. Run tools on samples and verify the results
  2. Pick the tool with highest precision
  3. Verify the results
  4. Update precision and recall of other tools accordingly
  5. Repeat step 2

Drop tools with precision below X%

Ordering-based approach

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





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!

Transformation Tasks

Airport code → City

Airport code	City
BOS	Boston
JFK	New York
ORD	Chicago
BER	Berlin
CDG	Paris

- Date formats
  - MM/DD/YYYY → DD/MM/YY
- Currency conversion
  - 1 USD → 0.89 EUR

- Model → Brand
  - Iphone 6 → Apple Inc.
- ISBN→ Title
  - 0-553-57340-3 → “A Game of Thrones”
- Unit conversion
  - 1 Mile → 1.6 km
- Long/Lat → location
- Country → Currency
- …

Problem Statement: Example-based Discovery

Given

Airport	City
BER	Berlin
JFK	New York
ORD	Chicago
HBE	?
IST	?
FRA	?
BOS	?
DFW	?
..	…

Example transformations

Find

Airport	City
BER	Berlin
JFK	New York
ORD	Chicago
HBE	Alexandria
IST	Istanbul
FRA	Frankfurt
BOS	Boston
DFW	Dallas
..	…

Workload from the “Wild”

- 120 Transformation Tasks

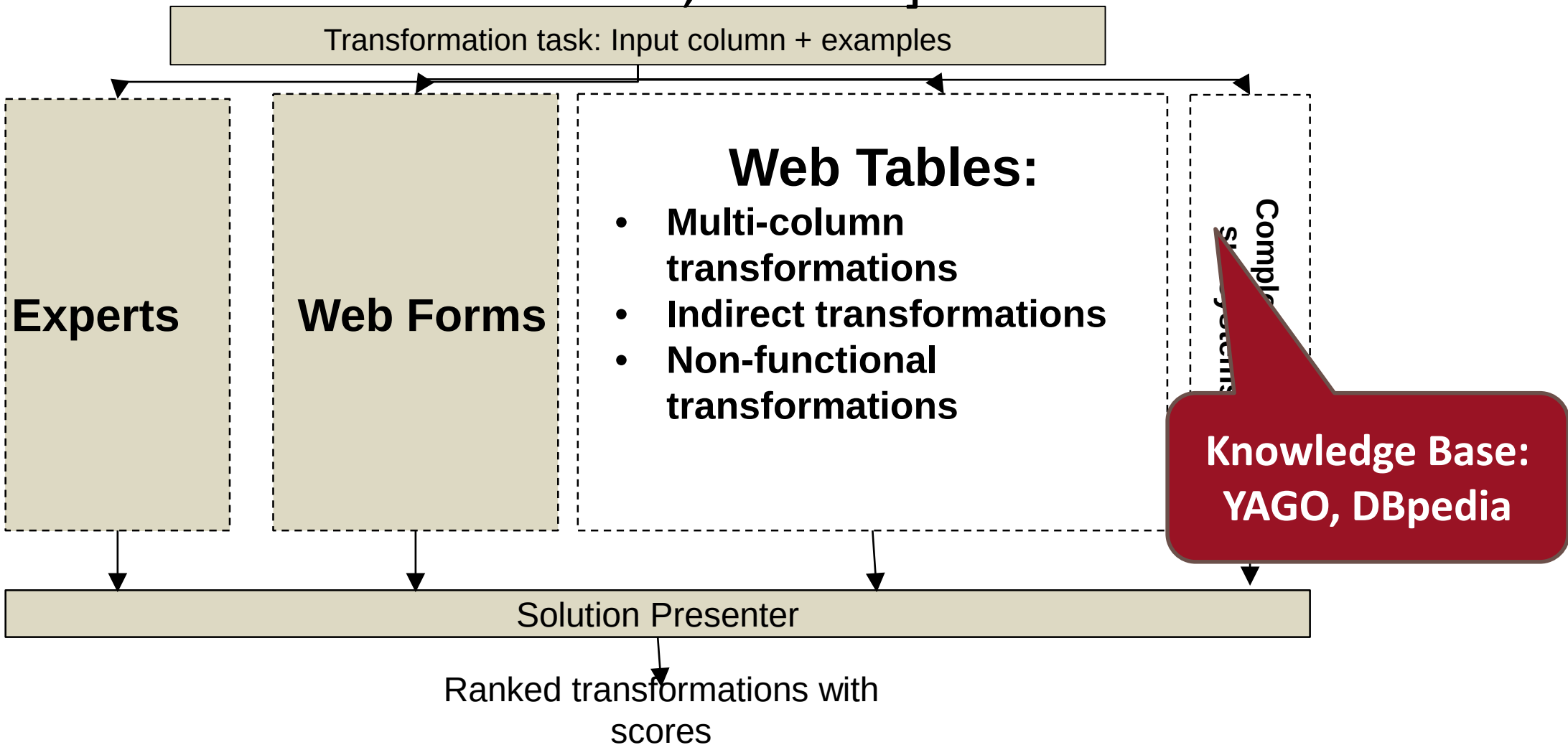
1. Fahrenheit to Celsius
2. miles to km
3. pound to kg
4. USD to EUR
5. zip to state
6. zip to city
7. UPS tracking to address
8. English to German
9. swift code to bank
10. hex to RGB
11. ISBN to publisher
12. ISBN to title
13. ISBN to author
14. ISSN to title
15. ip address to country
16. Domain to 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. country code to country
28. ascii to char
29. car model to brand
30. country to demonym
31. country to language
32. country to currency
33. company to BBGID
34. patent ID to name
35. city to long/lat
36. Entity to Wikipedia link
37. Entity to
38. person to
39. ip to dom
40. company
41. company
42. US standard to metric
43. fractions to decimals
44. country to code
45. State to state abbrv
46. time zone to abbrv
47. city to country
48. airport code to city
49. RGB to color
50. ASCII to Unicode
51. RGB to color
52. ascii to unicode
53. Mountains to range
54. team to manager
55. unesco site to city
56. USD to QAR
57. country
63. Movie to year
64. country to ISO 3 letters
65. country to ISO 2 letters
66. movie+actor to role
67. shoe size
68. us to eur
69. airport code to country
70. title + year to a
71. soccer play
72. soccer player t
73. soccer player t
74. univers to st
75. term to
76. unesco ntry
78.
9.
0.
1. video game to publisher
2. …

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

Player → teams
(31) Non-functional mappings

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

DataXFormer Improvements [CIDR'15, SIGMOD'16,ICDE'16]



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

Transformation task

Airport	City
BER	Berlin
JFK	New York
ORD	Chicago
HBE	?
IST	?
FRA	?
BOS	?
DFW	?
..	…

1

Look-up

Web tables

2

Filter

code	location
FRA	Frankfurt
JFK	New York
ORD	Chicago
BOS	Boston
BER	Berlin

Table 1

apc	city	…
DFW	Dallas	…
HBE	Alexandria	…
IST	Istanbul	…
FRA	Frankfurt	…

Table 4

Table 3

Table 2

……

…	…	airport	city
…	…	FRA	Frankfurt
…	…	DFW	Dallas
…	…	JFK	New York
…	…	BER	Berlin
…	..	…	…

apc	location
JFK	New York
BER	Berlin
ORD	Illinois
FRA	Hessen
…	…

Augment query

FRA	Frankfurt
BOS	Boston
DFW	Dallas

Airport	City
BER	Berlin
JFK	New York
ORD	Chicago
FRA	Frankfurt
BOS	Boston
DFW	Dallas

Augment database

Mappings

3

X	Y	Score	Tables
FRA	Frankfurt	0.83	T1,T2
BOS	Boston	1	T1
DFW	Dallas	0.67	T2
FRA	Hessen	0.67	T4
…	…	…	…

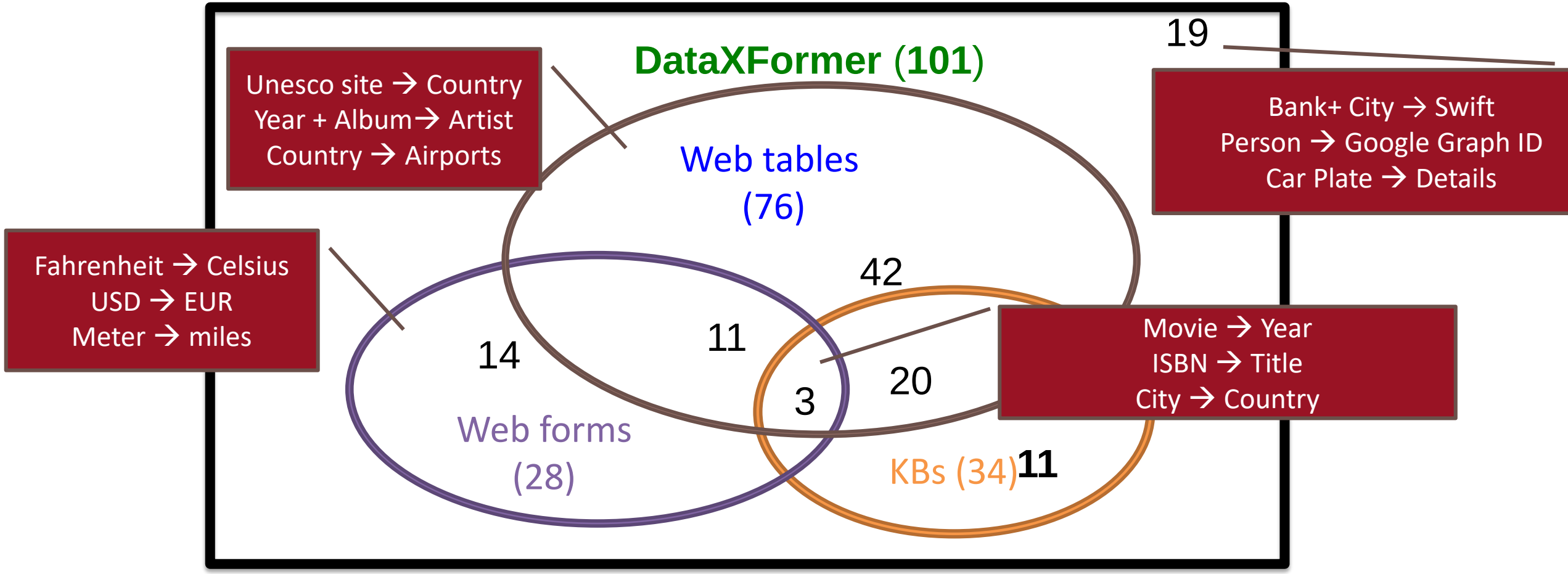
Refine

4

Airport	City
BER	Berlin
JFK	New York
ORD	Chicago
FRA	Frankfurt
BOS	Boston
DFW	Dallas
HBE	?
IST	?
…	…

Coverage

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



Lessons Learned So far

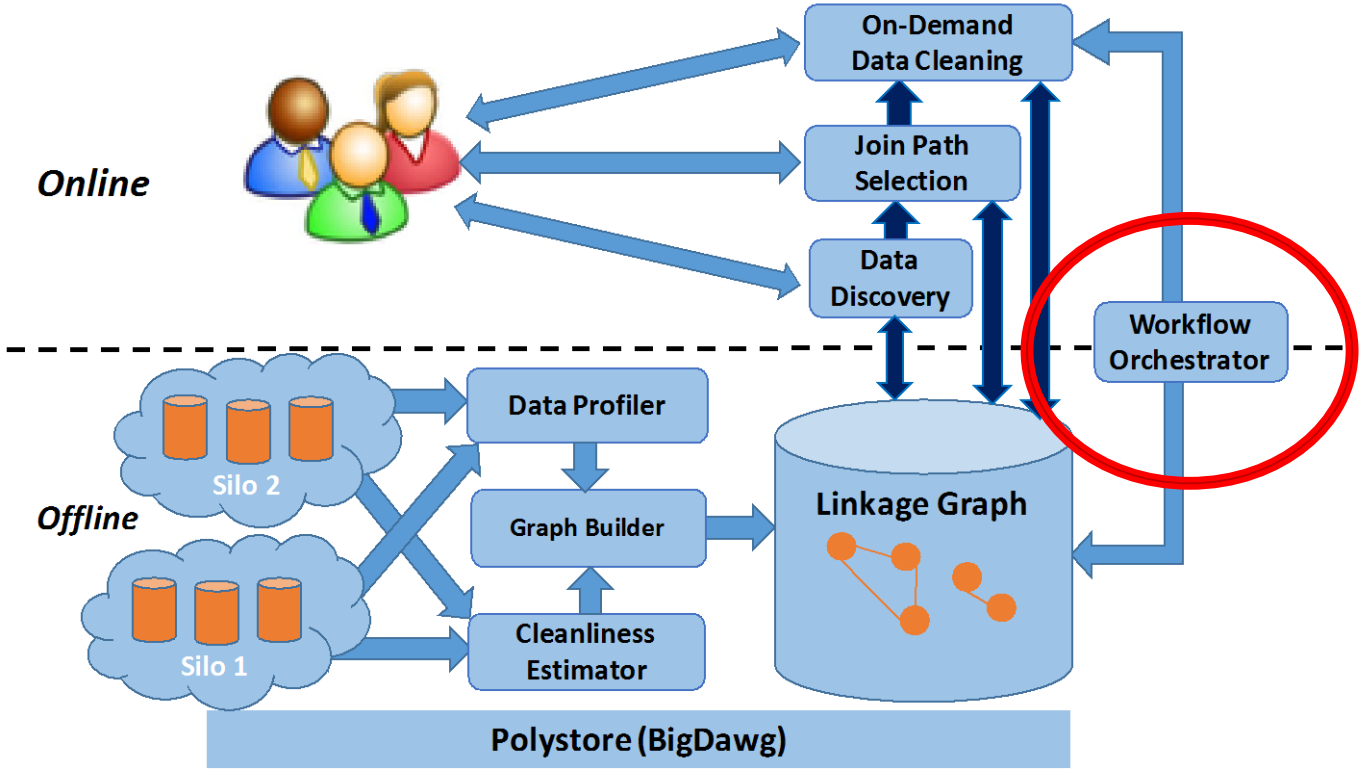
- Error Detection
  - There is no single dominant tool.
  - Improving individual tools has marginal benefit.
  - Picking the right order of tools can reduce the cost of validation by humans.
- DataXFormer
  - Web resources are a treasure trove for finding transformations
  - Domain-specific tasks require domain-specific transformation repositories
  - We want to try DataXFormer inside the firewall

Data Civilizer [CIDR 2017,SIGMOD'17]

- Project with MIT, University of Waterloo, QCRI
  - Data discovery
  - Pull-based data integration
- Use case:
  - Data warehouses (MIT)
  - Scientific data lakes (Merck)

Raul Castro

Dong Deng



Recommending Cleaning Workflows

- Assumption:
  - Similar datasets require similar cleaning efforts
    - Similar structure
    - Similar use case
- What is dataset similarity?
  - Structural similarity
  - Content similarity
  - Similarity in data quality?
    - Similar errors
    - Similar error distributions

Data Profiling [VLDBJ 2015]

Data Cleaning results [PVLDB 2016]

Mohammad Mahdavi
(PhD student who has to build the space ship)

Conclusions

(1) More reasoning on holistic combination of tools

(2) More reasoning on real-world data

(3) Data Civilizer: pull-based data integration

(4) Data-driven cleaning suggestions