Detecting Data Errors: Where are we and what needs to be done?

Paolo Papotti
Arizona State University

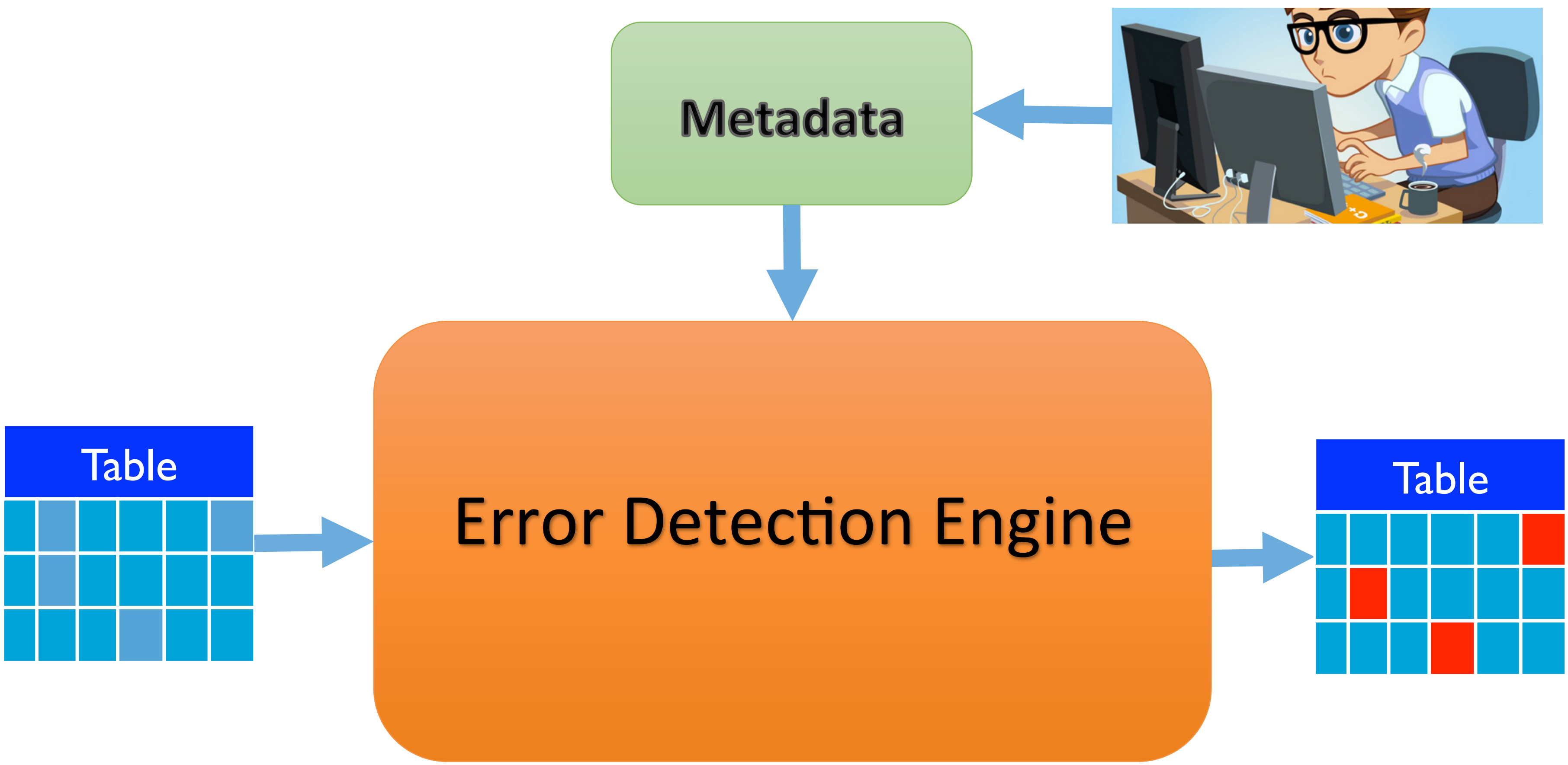
11th International Workshop on Information Search, Integration, and Personalization (ISIP 2016)

Detecting Data Errors

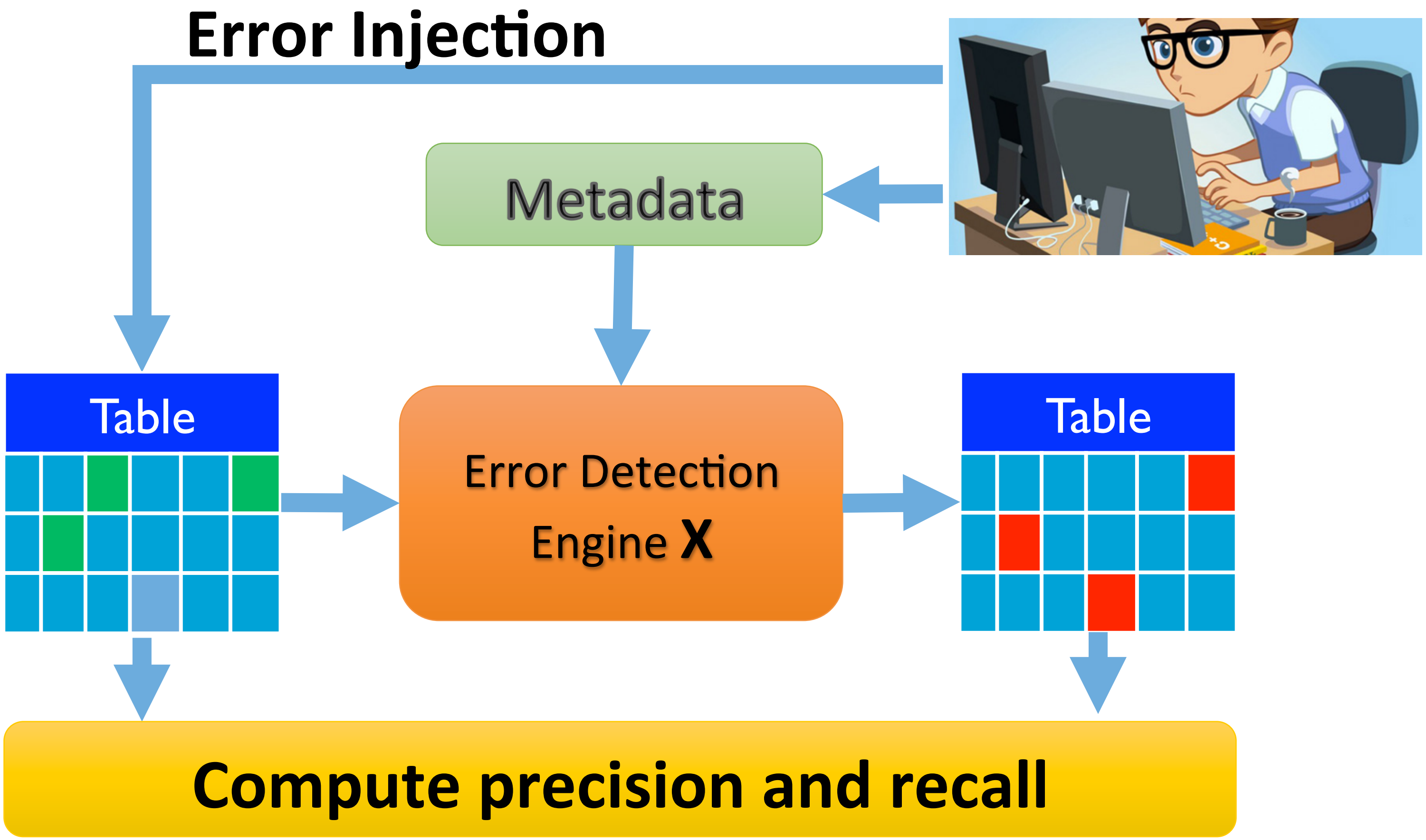
- Where are we?
  - Motivation
  - Error Types, Tools, Data sets
  - Results: single tool, union, min-k, extra mile
- What ~~needs to~~ **can** be done?
  - Ordering
  - Discovering and Exploration

Error = A value that is different from ground truth

Ideal error detection

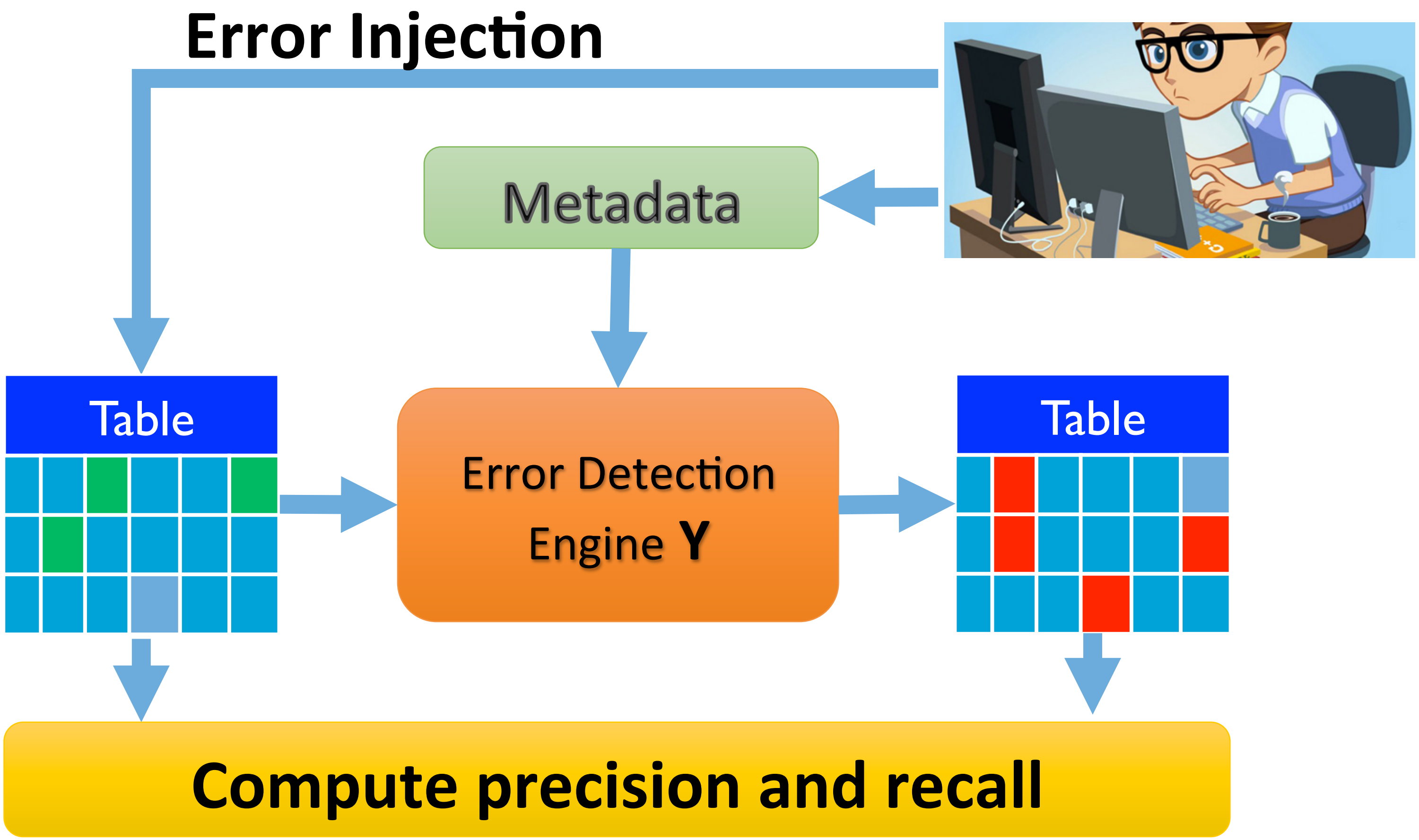


Qualitative Evaluation 1



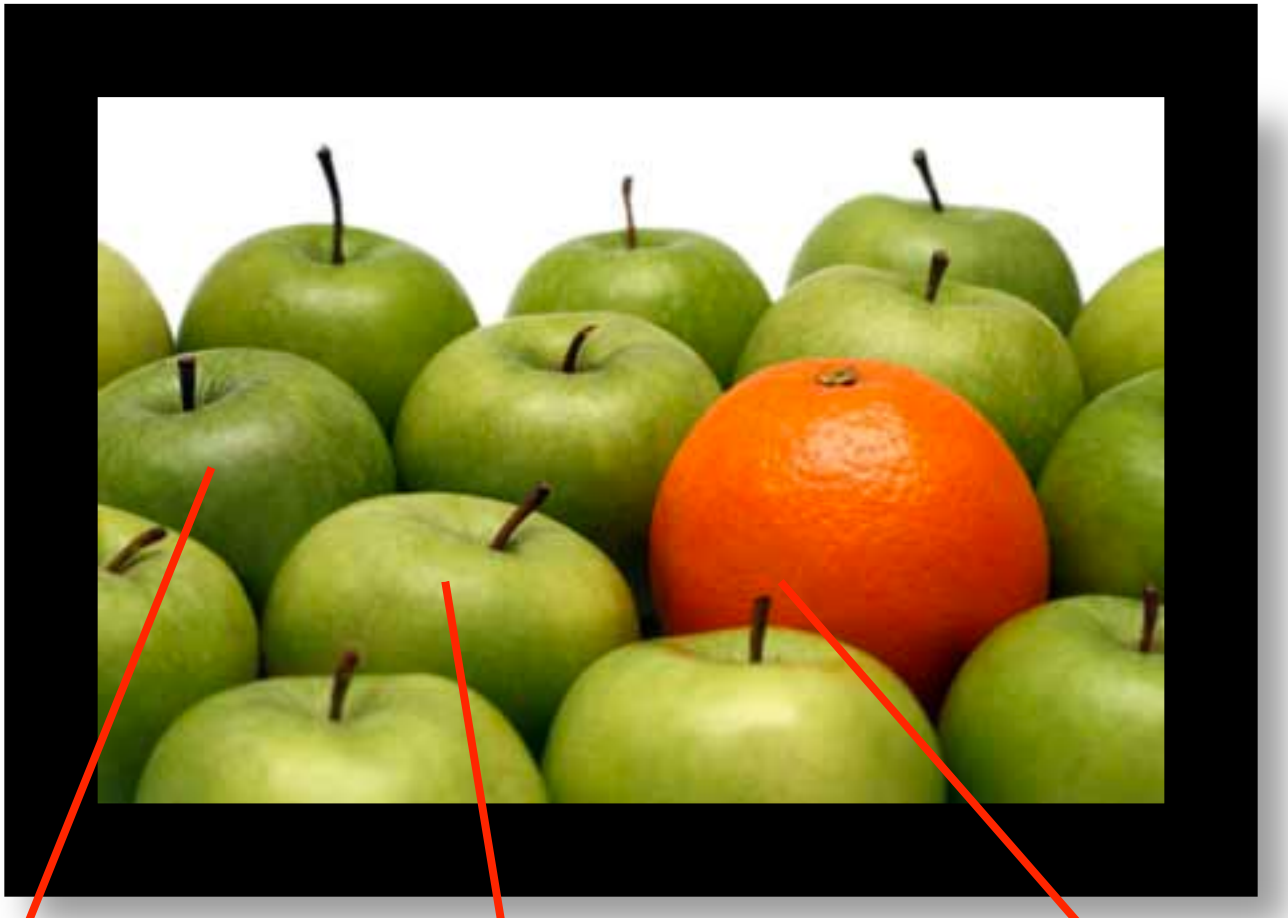
P= 0.66   R= 0.66

Qualitative Evaluation 1



P= 0.25   R= 0.33

Qualitative Evaluation 2



Rule detect X

Rule detect Y

Outlier detect Z

Motivation

- Extensive research on cleaning algorithms
  1. Usually evaluated on errors injected into clean data
     - Good to evaluate algorithms, but <u>does not measure real recall</u>
  2. Tools evaluated against tools of the same category
     - Well-defined but <u>narrow scope</u>

- How well do techniques work “in the wild”?
- What about combinations of techniques?

This study is not about finding the best/better tools!

What we did [PVLDB16 – Exp track]

1. Analyzed 5 different real 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. Tested impact of enrichment and domain specific tools

Error Types

Constraint violation

ID	FN	LN	ROLE	ZIP	ST	SAL
105	Anne	Nash	Emp	85281	NY	110
211	Mark	White	Man	15544	NY	80
386	Mark	Lee	M	85281	AZ	75
215	Anna	Smith Nash	Emp	85283		1

Duplicates

Pattern Violation

Outliers

Error Detection Strategies

- Rule-based detection algorithms
  - Detecting violation of constraints, such as (conditional) functional dependencies, denial constraints, …
- Pattern verification and enforcement tools
  - Syntactical patterns, such as date formatting
  - Semantical patterns, such as location names
- Quantitative algorithms
  - Statistical outliers
- Deduplication
  - Discovering conflicting attribute values in duplicates

Tool Selection

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

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

5 Data Sets

1. MIT VPF
   - Procurement dataset containing information about **suppliers**
   - Attributes include names, contact data, and business flags, etc.
2. Merck
   - List of **IT-services** and software
   - Attributes include location, number of end users, business flags, etc.
3. Animal
   - On field information about **capture of animals**
   - Attributes include tags, sex, weight, etc.
4. Rayyan Bib
   - **Literature references** collected from various sources
   - Attributes include author names, publication titles, ISSN, etc.
5. BlackOak
   - **Address** dataset
   - Attributes include names, addresses, birthdate, etc.

5 Data Sets - continued

Dataset	# columns	# rows	# rows ground truth	Errors
MIT VPF	42	24K	13k (partial)	6.7%
Merck	**61**	2262	2262	19.7%
Animal	14	60k	60k	0.1%
Rayyan Bib	11	**1M**	1k (partial)	**35%**
BlackOak	12	94k	**94k**	34%

	MIT VPF	Merck	Animal	Rayyan Bib	BlackOak
Pattern violations	✔	✔	✔	✔	✔
Constraint violations	✔	✔	✔	✔	✔
Outliers	✔	✔		✔	✔
Duplicates	✔				✔

Evaluation Methodology

- We obtained **knowledge** about the data from **the data owners**:
  - Quality constraints, business rules, distributions
- **Best effort** in using all capabilities of the tools
  - However: **No heroics**
    i.e., no embedding custom Java/Python code in a tool
  - Complete? More on this later
- Metrics:
  - Precision, Recall, F-Measure

Computing Precision for Detection

Constraint violation
P= 1/4

ID	FN	LN	ROLE	ZIP	ST	SAL
105	Anne	Nash	Emp	85281	**NY**	110
211	Mark	White	Man	15544	NY	80
386	Mark	Lee	**M**	85281	AZ	75
215	Anna	Smith Nash	Emp	85283		1

Pattern Violation
P = 1/1

Single Tool Performance: All Datasets

Tools		MIT VPF P	MIT VPF R	MIT VPF F	Merck P	Merck R	Merck F	Animal P	Animal R	Animal F	Rayyan Bib P	Rayyan Bib R	Rayyan Bib F	BlackOak P	BlackOak R	BlackOak 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
DBoost	Histogram	.13	.11	.12	.13	.02	.04	.00	.00	.00	.40	.16	.23	.52	.51	.52
DBoost	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**

Combining Tools

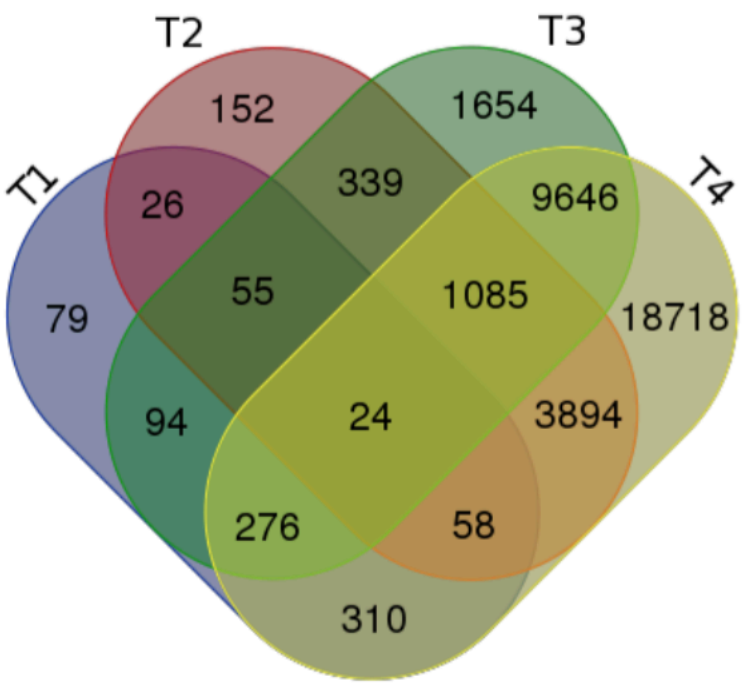
- Naïve approach
  - At least ***k*** tools **agree** on a value to be identified as error
    - Expected precision-recall trade-off (k=1 is Union)

k	MIT VPF			Merck			Animal		
	P	R	F	P	R	F	P	R	F
1	0.24	**0.93**	0.38	0.33	**0.84**	0.47	0.128	**0.575**	**0.209**
2	0.48	0.90	**0.63**	0.889	0.789	0.834	0.241	0.030	0.053
3	0.58	0.41	0.48	0.996	0.787	**0.879**	**1.0**	0.001	0.002
4	0.79	0.09	0.16	0.997	0.280	0.438	0	0	0
5	0.76	0.03	0.06	0.993	0.015	0.029	0	0	0
6	**0.90**	0.00	0.01	**1.0**	0.000	0.000	0	0	0

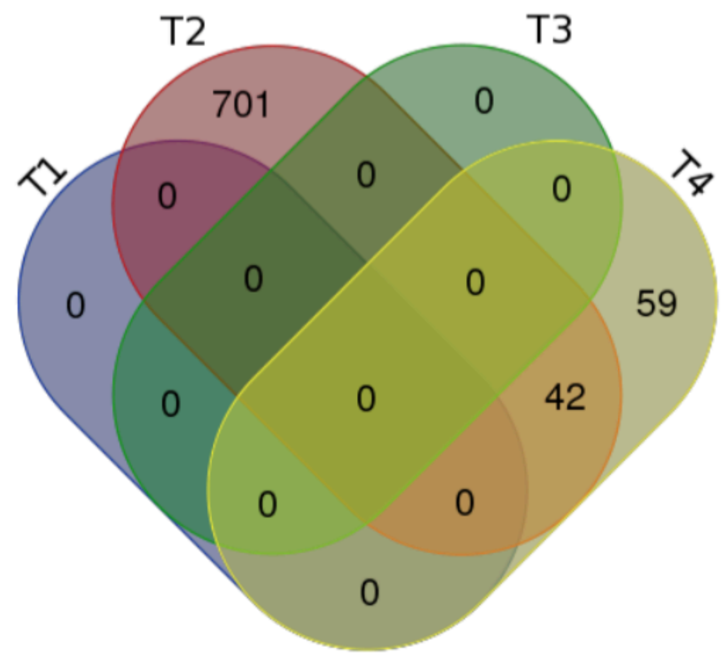
Given labelled data

Combining Tools

T1: Duplicates, T2: Constraint Violations, T3: Outliers, T4: Pattern Violations



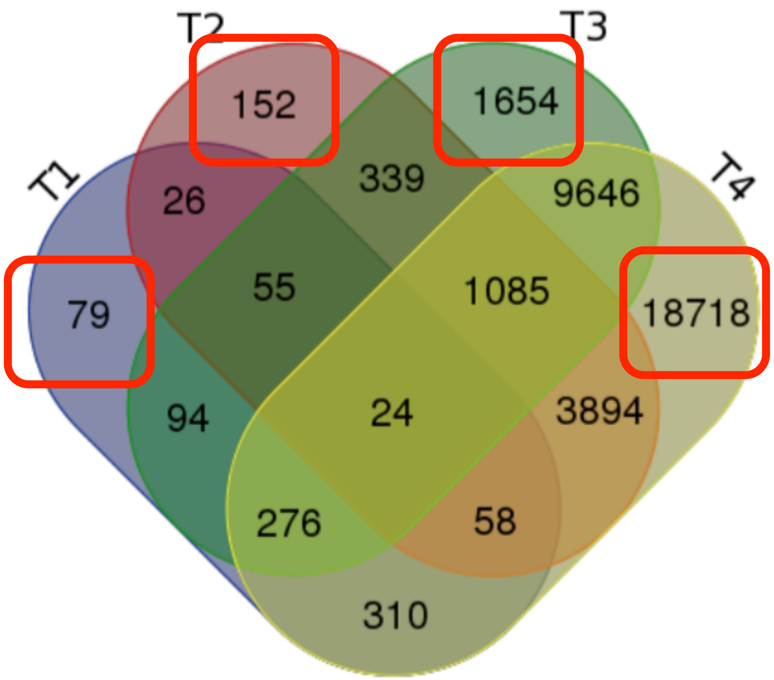
(d) MIT VPF: 36,410 out of 39,158 errors



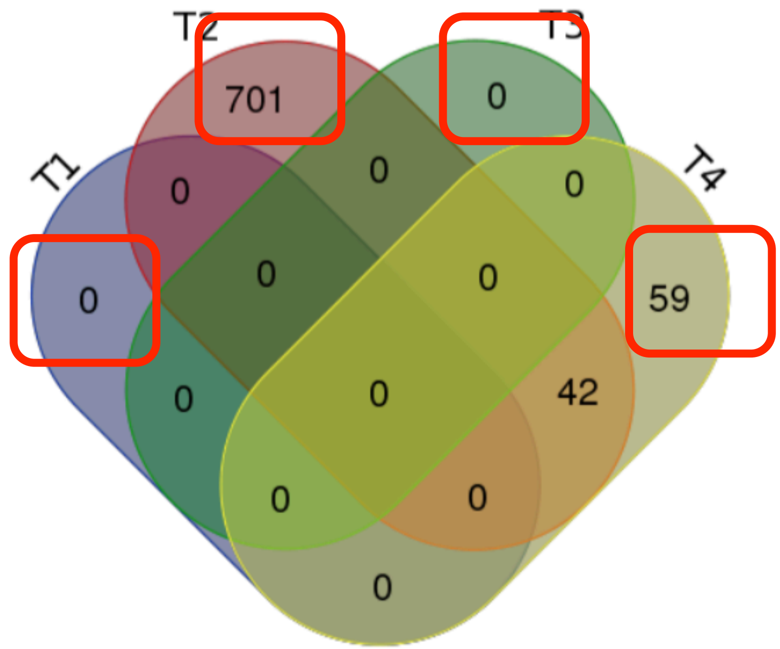
(b) Animal: 802 out of 1,394 errors

Combining Tools k=1 (approx)

T1: Duplicates, T2: Constraint Violations, T3: Outliers, T4: Pattern Violations



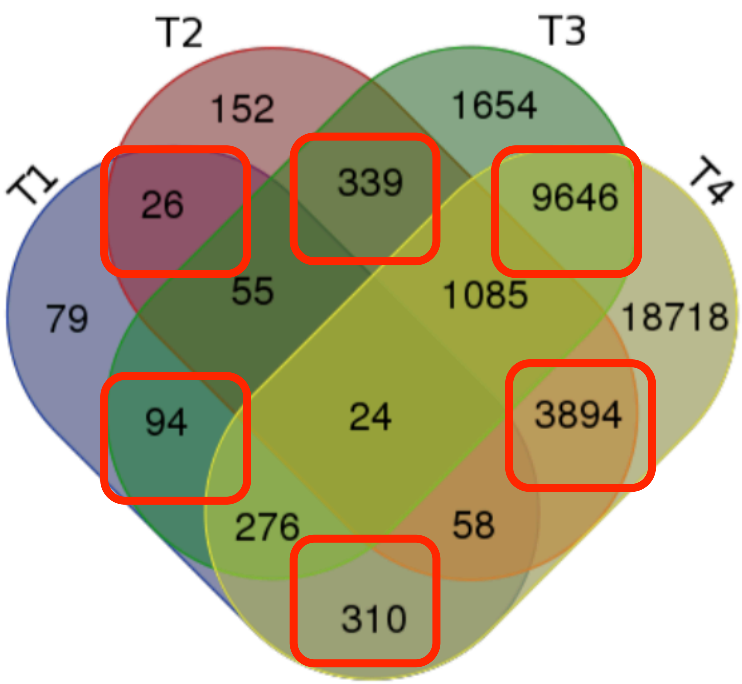
(d) MIT VPF: 36,410 out of 39,158 errors



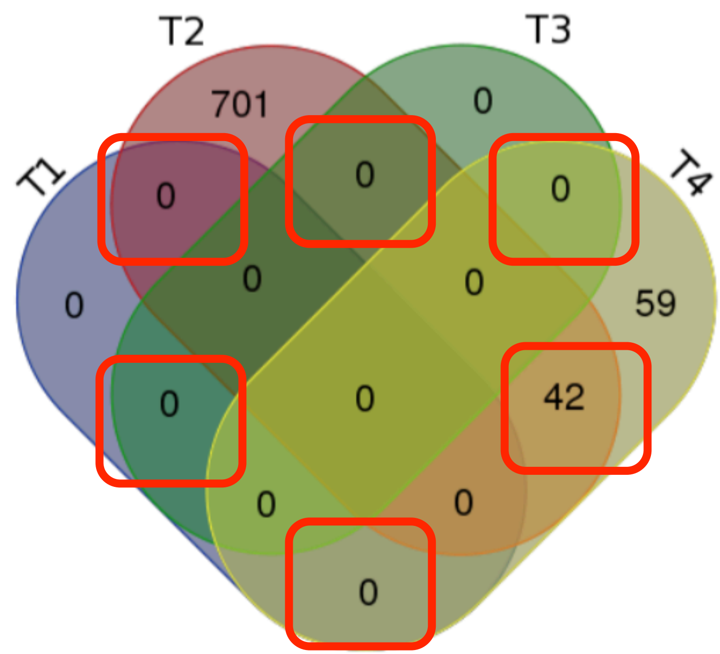
(b) Animal: 802 out of 1,394 errors

Combining Tools k=2 (approx)

T1: Duplicates, T2: Constraint Violations, T3: Outliers, T4: Pattern Violations



(d) MIT VPF: 36,410 out of 39,158 errors



(b) Animal: 802 out of 1,394 errors

Maximum Possible Recall

- Manually checked each undetected error
- Reasoned whether the error could have been detected by a **refinement** of the tool's input, 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

Enrichment and Domain-specific tools

- Enrichment
  - Manually append new columns joining other tables
    - ➢ Improves rule-based and duplicate detection tools

Data set	Rule-based		Duplicates	
	P	R	P	R
MIT VPF	(+6%) 0.31	(+6%)0.20	(+2%) 0.18	(+1%) 0.03
BlackOak	0.46	0.43	0.41	(+5%) 0.68

- Domain-specific tool
  - Tested a commercial address cleaning service
    - ➢ High precision on the specific domain
    - ➢ Very low increase of overall recall
      - ➢ **2** (**13**) new errors detected for MIT VPF (BlackOak)

“Where are we?” Conclusions

(1) There is no single dominant tool

(2) Improving individual tools has marginal benefit

→ We need a combination of tools

Detecting Data Errors

- Where are we?
  - Motivation
  - Error Types, Tools, Data sets
  - Results: single tool, union, min-k, extra mile
- What ~~needs to~~ can be done?
  - Ordering
  - Discovering and Exploration

Combining Tools

Labelled data

- Naïve approach
  - At least ***k*** tools agree on a value to be an error
    - Expected precision-recall trade-off (k=1 is Union)

k	MIT VPF			Merck			Animal		
	P	R	F	P	R	F	P	R	F
1	0.24	**0.93**	0.38	0.33	**0.84**	0.47	0.128	**0.575**	**0.209**
2	0.48	0.90	**0.63**	0.889	0.789	0.834	0.241	0.030	0.053
3	0.58	0.41	0.48	0.996	0.787	**0.879**	**1.0**	0.001	0.002
4	0.79	0.09	0.16	0.997	0.280	0.438	0	0	0
5	0.76	0.03	0.06	0.993	0.015	0.029	0	0	0
6	**0.90**	0.00	0.01	**1.0**	0.000	0.000	0	0	0

Combining Tools

Unlabelled data

- Naïve approach
  - At least ***k*** tools agree on a value to be an error
    - Expected precision-recall trade-off (k=1 is Union)

k	MIT VPF			Merck			Animal		
	P	R	F	P	R	F	P	R	F
1									
2									
3									
4									
5									
6									

1. What is the right ***k*** for a given dataset?

Combining Tools

Labelled data

- Naïve approach
  - At least ***k*** tools agree on a value to be an error
    - Expected precision-recall trade-off (k=1 is Union)

k	MIT VPF			Merck			Animal		
	P	R	F	P	R	F	P	R	F
1	0.24	**0.93**	0.38	0.33	**0.84**	0.47	0.128	**0.575**	**0.209**
2	0.48	0.90	**0.63**	0.889	0.789	0.834	0.241	0.030	0.053
3	0.58	0.41	0.48	0.996	0.787	**0.879**	**1.0**	0.001	0.002
4	0.79	0.09	0.16	0.997	0.280	0.438	0	0	0
5	0.76	0.03	0.06	0.993	0.015	0.029	0	0	0
6	**0.90**	0.00	0.01	**1.0**	0.000	0.000	0	0	0

1. What is the right k for a given dataset?
2. <u>Validate thousands</u> values: up to 87% are **correct**!

Combining Tools

Unlabelled data

- Naïve approach
  - At least ***k*** tools agree on a value to be an error
    - Expected precision-recall trade-off (k=1 is Union)

k	MIT VPF			Merck			Animal		
	P	R	F	P	R	F	P	R	F
1									
2									
3									
4									
5									
6									

1. What is the right k for a given dataset?
2. <u>Validate thousands</u> values: **How to minimize effort**?

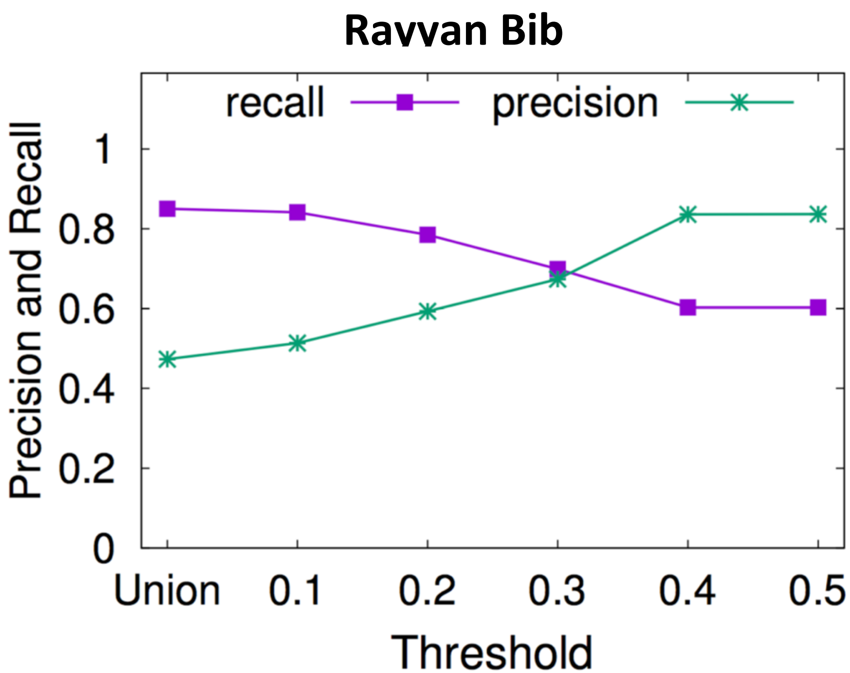
Combining Tools – unlabelled data

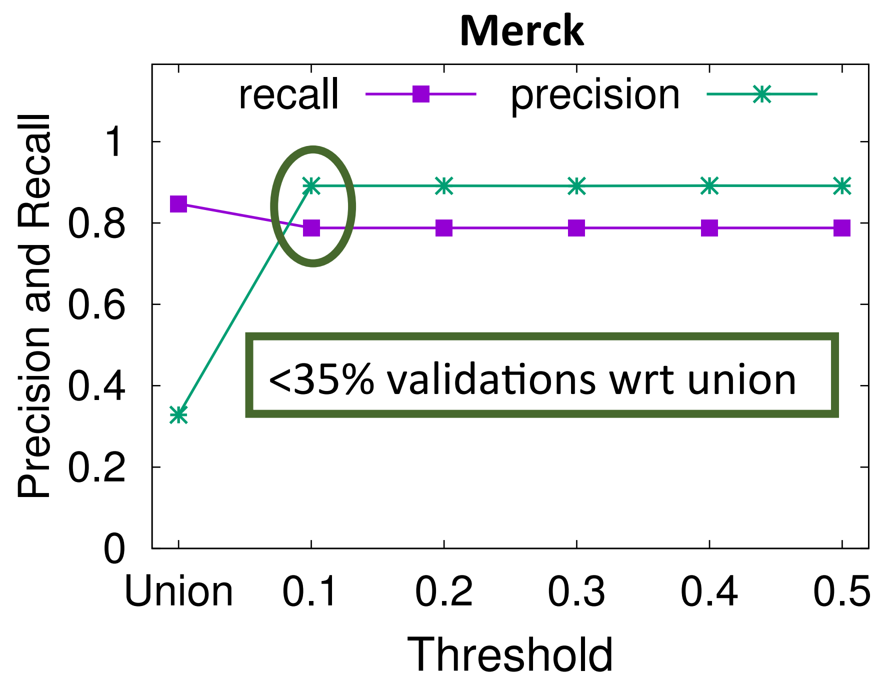
- Minimize validation of possible errors
- Maximum entropy-based <u>order selection</u>:
  1. Run **all** tools on **samples** and verify the results
  2. Pick the tool with **highest precision**
  3. **Verify** the results
  4. **Update** precision and recall of other tools accordingly (implicitly exploits k overlap)
  5. Repeat step 2

Drop tools with precision below threshold (e.g., 10%)

Ordering-based approach

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





5% of tuples sampled to bootstrap algorithm

Which tools are adopted?

OpenRefine / OpenRefine

Google

Watch	485	Star	4,043	Fork	712

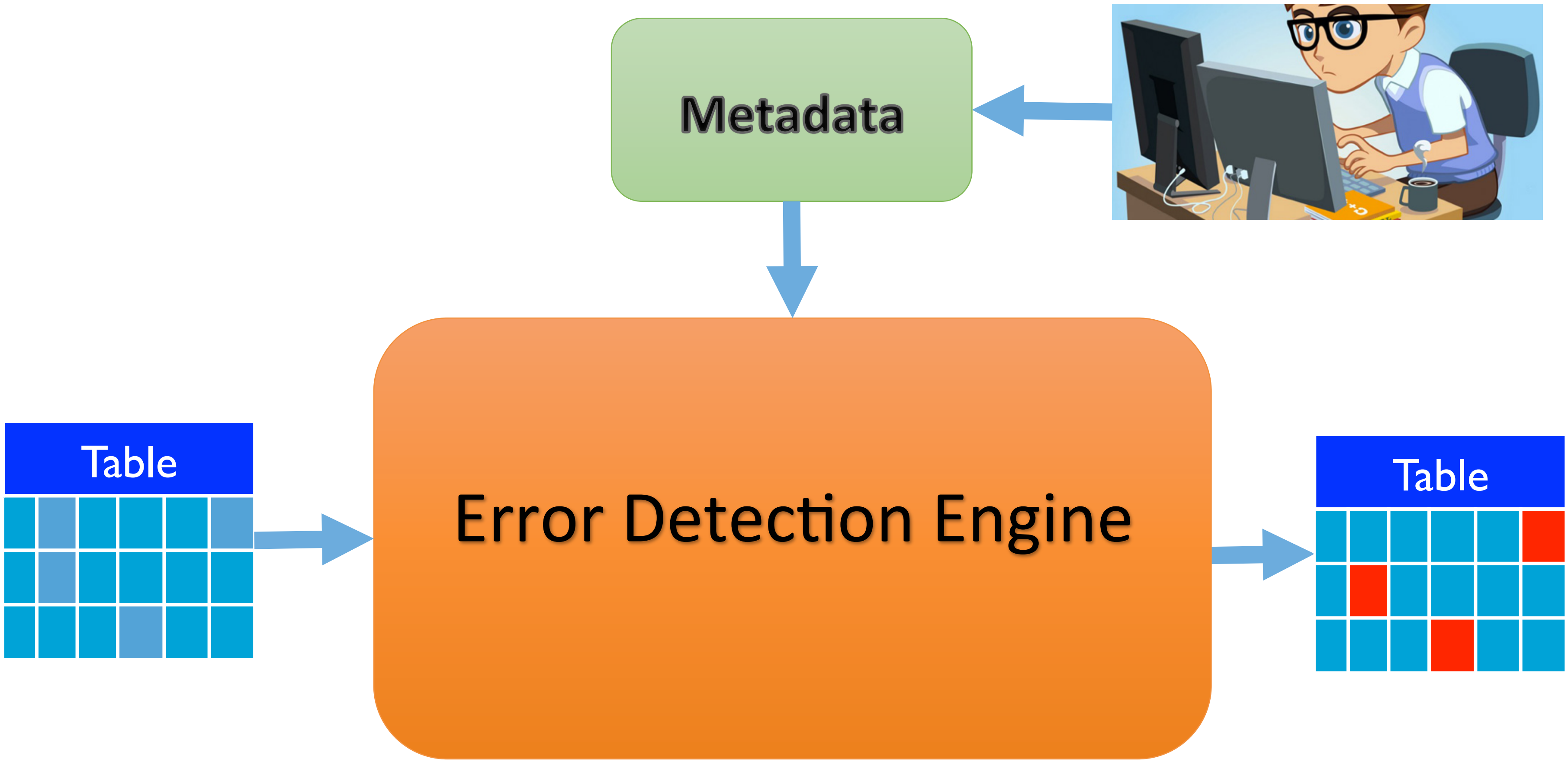
Trifacta is the Data Wrangling Solution
for Over 4,000 Companies in 132 Countries

tamr

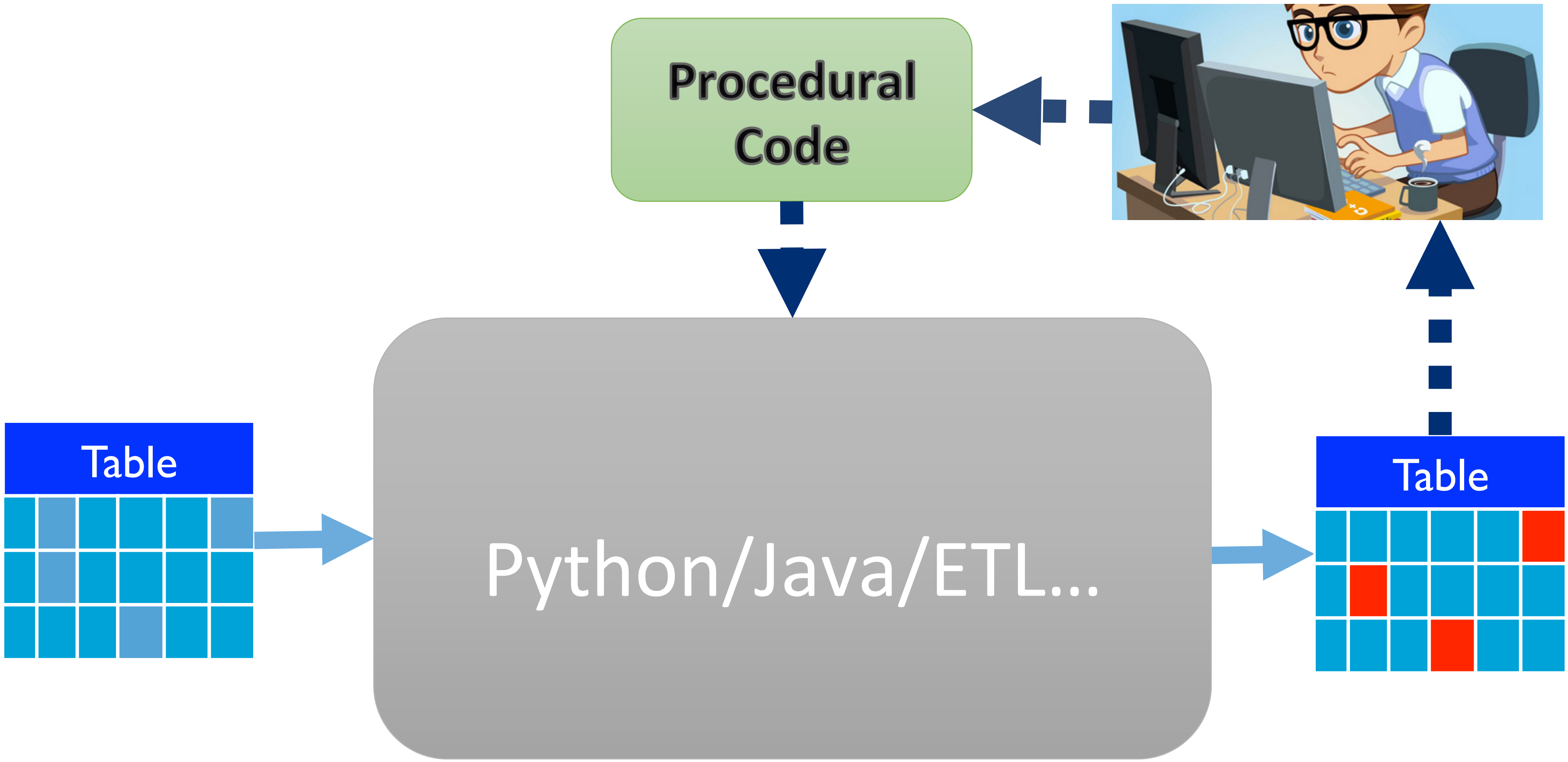
Driving Smarter Analytics at Some of the World's Largest Brands

CISCO, GE, HUAWEI, TOYOTA, PHILIPS, hp, THOMSON REUTERS, NOVARTIS, AMGEN, Roche, MERCK

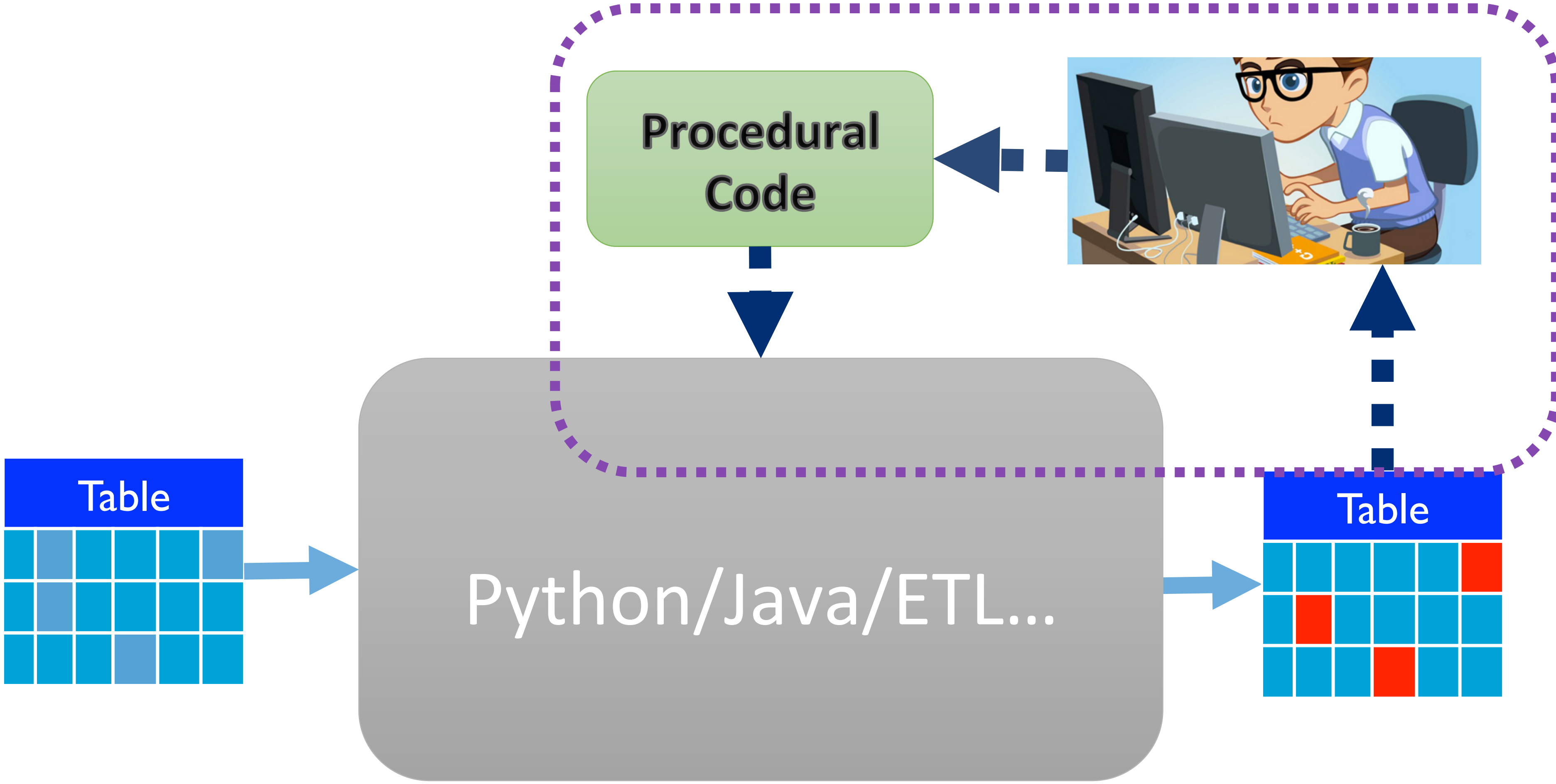
Ideal error detection



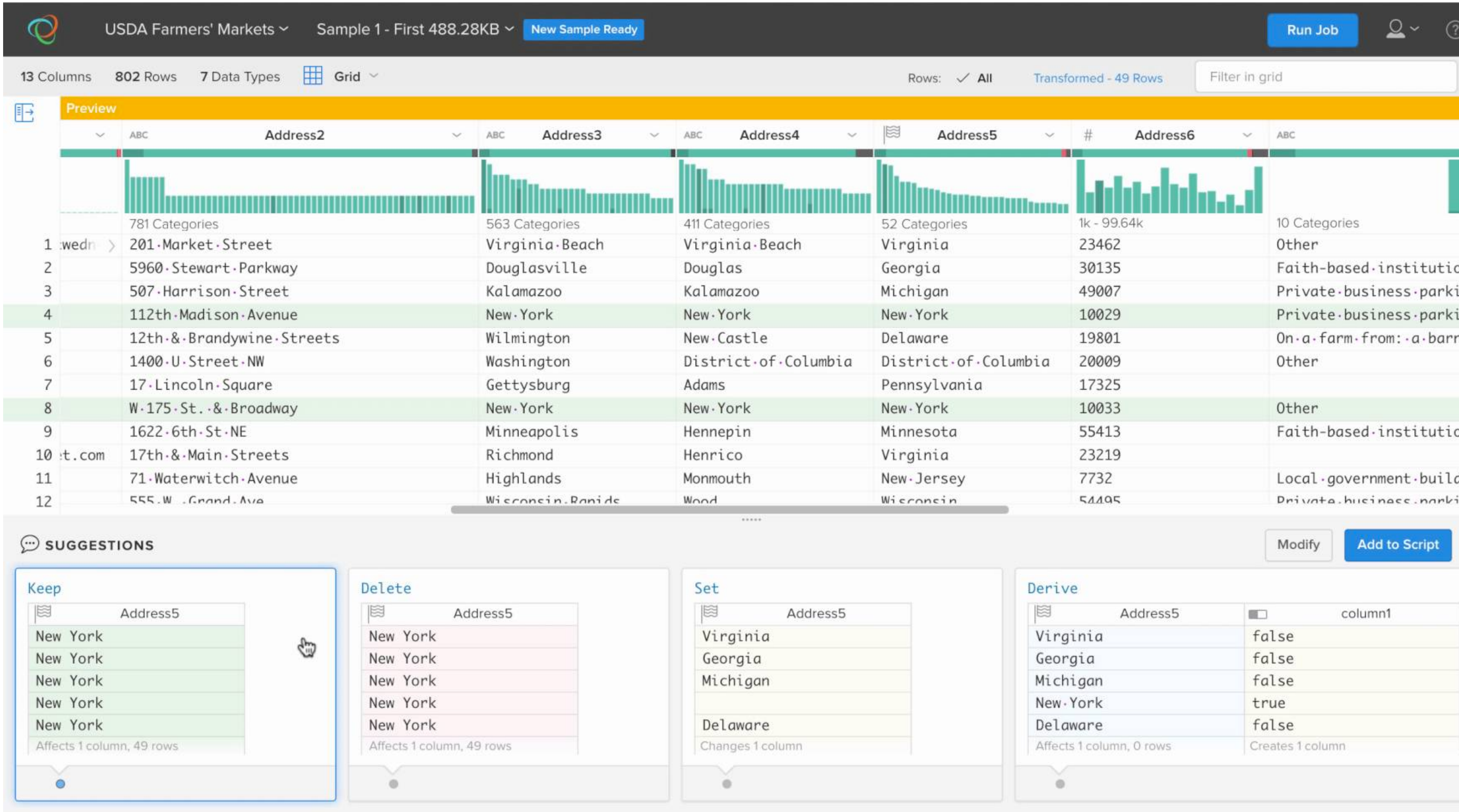
Real error detection



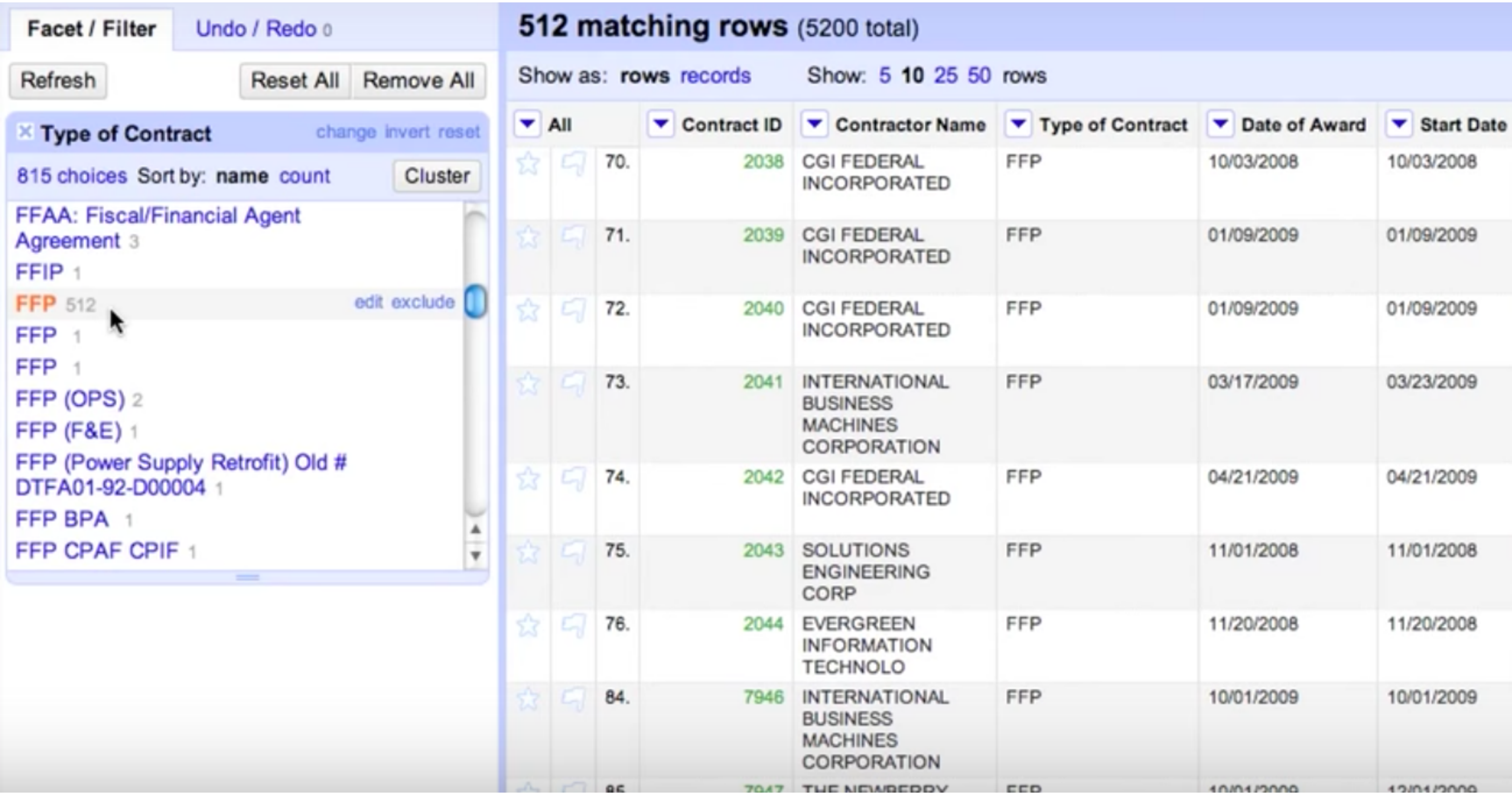
Real error detection



Trifacta Wrangler

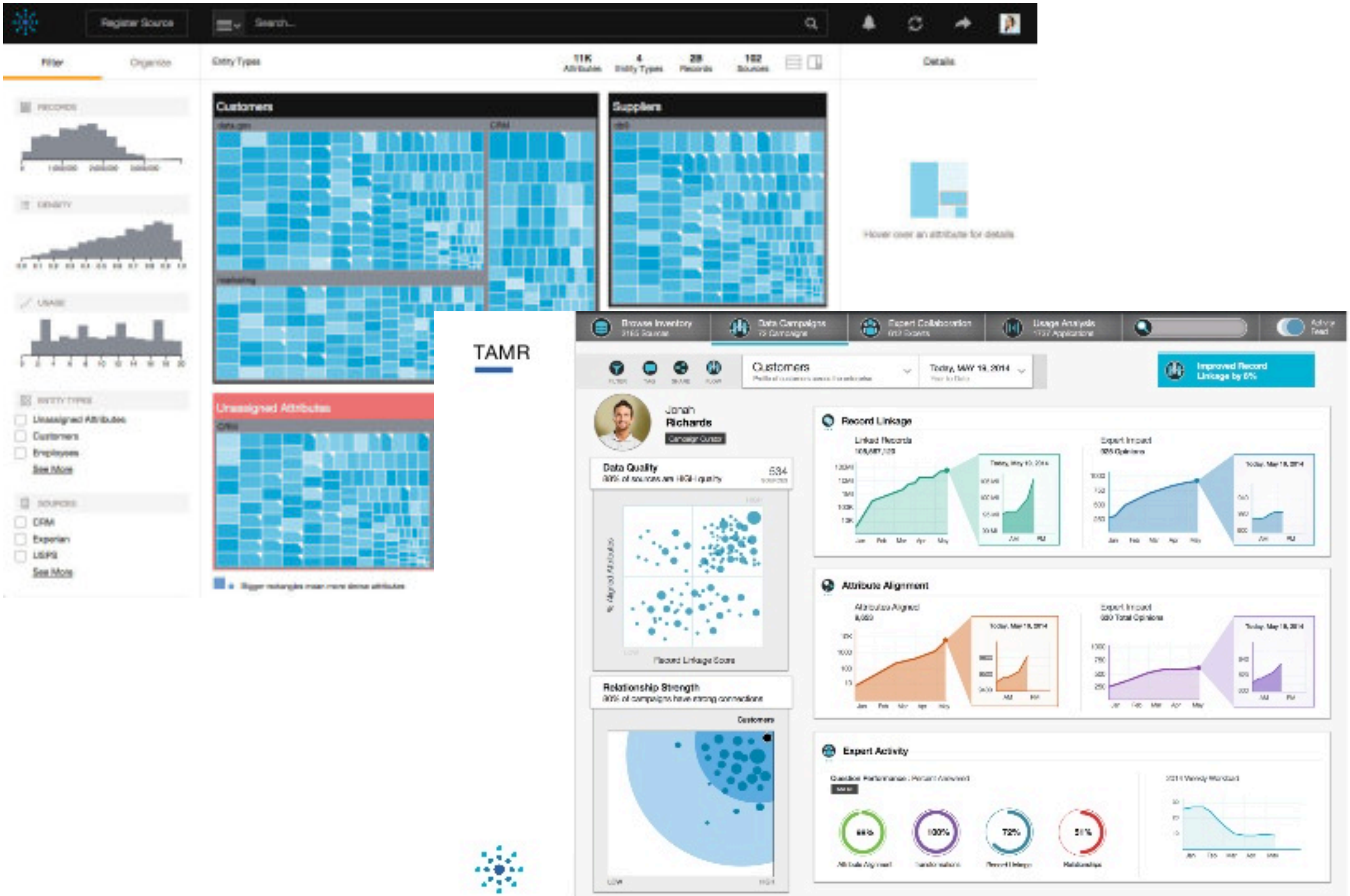


Open Refine



Facet / Filter
Undo / Redo 0
Refresh
Reset All
Remove All
Type of Contract
change invert reset
815 choices Sort by: name count
Cluster
FFAA: Fiscal/Financial Agent Agreement 3
FFIP 1
FFP 512
edit exclude
FFP 1
FFP 1
FFP (OPS) 2
FFP (F&E) 1
FFP (Power Supply Retrofit) Old # DTFA01-92-D00004 1
FFP BPA 1
FFP CPAF CPIF 1
512 matching rows (5200 total)
Show as: rows records
Show: 5 10 25 50 rows
All
Contract ID
Contractor Name
Type of Contract
Date of Award
Start Date
70. 2038 CGI FEDERAL INCORPORATED FFP 10/03/2008 10/03/2008
71. 2039 CGI FEDERAL INCORPORATED FFP 01/09/2009 01/09/2009
72. 2040 CGI FEDERAL INCORPORATED FFP 01/09/2009 01/09/2009
73. 2041 INTERNATIONAL BUSINESS MACHINES CORPORATION FFP 03/17/2009 03/23/2009
74. 2042 CGI FEDERAL INCORPORATED FFP 04/21/2009 04/21/2009
75. 2043 SOLUTIONS ENGINEERING CORP FFP 11/01/2008 11/01/2008
76. 2044 EVERGREEN INFORMATION TECHNOLO FFP 11/20/2008 11/20/2008
84. 7946 INTERNATIONAL BUSINESS MACHINES CORPORATION FFP 10/01/2009 10/01/2009

Tamr



Register Source
Search...
Filter
Organize
Entry Types
Details
Customers
Suppliers
Unassigned Attributes
Hover over an attribute for details
TAMR
Browse Inventory
Data Campaigns
Expert Collaboration
Usage Analysis
Customers
Improved Record Linkage by 6%
Jonah
Richards
Data Quality
Relationship Strength
Record Linkage
Attribute Alignment
Expert Activity

Discovery and Exploration

- Successful for simple patterns
- More challenging for complex rules
  - **Pair-wise** comparisons
    - Quadratic in the number of tuples (DCs)
  - **All attributes** subsets
    - Exponential in relation's arity (lattice helps)
  - Mining not robust to **noise**
    - Approximate rules with >10% errors are useless or buried in thousands of candidates
  - Sampling makes problem much harder!

“What can be done?” Conclusions

(1) Picking the **right order** in applying the tools can improve the precision and help reduce the cost of validation by humans

- *Algorithms for optimal solution: threshold that maximizes F-measure and minimize user’s validations*
  - *Budget version of the problem? How to better use overlap?*

(2) Data exploration and **metadata discovery** is key for adoption and real impact

- *Efficient and robust interactive mining: call for ML solutions*

Thanks

Detecting Data Errors: Where are we and what needs to be done?

Paolo Papotti

ppapotti@asu.edu
Arizona State University

11th International Workshop on Information Search, Integration, and Personalization (ISIP 2016)