Data cleansing

Juhani Ojares

3.4.2017

Contents

- Introduction
- Traditional methods
- Big data method: KATARA
- Summary

Introduction

- Why data needs cleaning?
  - 25% critical corporate data is dirty
  - Compromises queries and analysis tasks
    - → wrong price data in retail databases alone costs US consumers $2.5 billion annually

Introduction

- What is data cleaning?
  - making the data *consistent*
  - process of *detecting* and *correcting* errors in data
- What is dirty data?
  - Duplicates
  - Miss-spellings
  - Missing data
  - Outdated data
  - data not obeying business rules

Traditional methods

- Integrity Constraints of database (ICs)
  1) Functional dependencies
     - Example 1: Primary keys, foreign keys
     - Example 2: [CountryCode, AreaCode] -> [City]
  2) Conditional functional dependencies
     - Example: [CountryCode=44, AreaCode] -> [City]
  - Many others

Traditional methods

- Integrity Constraints only *detect* errors – cannot fix them!
  - Domain experts needed to find repair
    - → costly, manual
- Many error-detection algorithms have high time-complexity

  Example
  - Detecting duplicates is combinatorial problem
- NOSQL databases usually don't use ICs!

Traditional methods

- Problem of detecting errors
  - Scalability
    - BigDansing addresses the *scalability* problem
- Problem of correcting errors
  - Accuracy
    - KATARA addresses the *accuracy* problem

KATARA

KATARA: A Data Cleaning System Powered by Knowledge Bases and Crowdsourcing

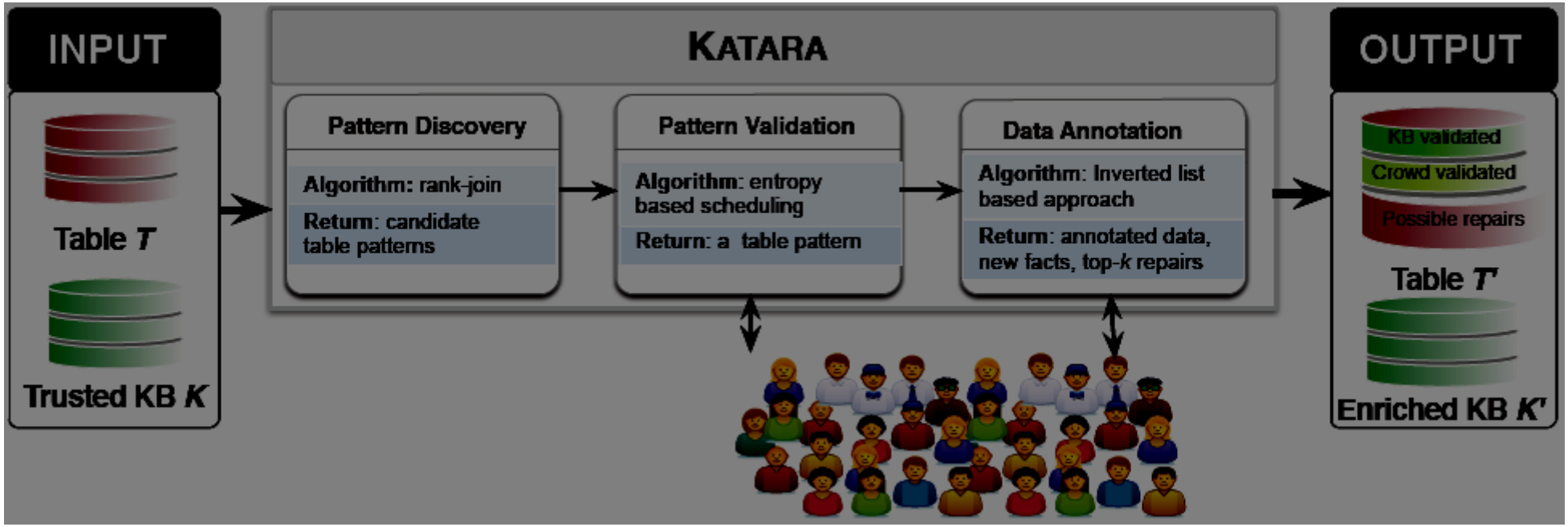
Xu Chu[1*] John Morcos[1*] Ihab F. Ilyas[1*]
Mourad Ouzzani[2] Paolo Papotti[2] Nan Tang[2] Yin Ye[2]

[1]University of Waterloo [2]Qatar Computing Research Institute
{x4chu,jmorcos,ilyas}@uwaterloo.ca {mouzzani,ppapotti,ntang,yye}@qf.org.qa

- Instead of domain experts
  - Knowledge Bases:
    - Example: Yago, DBPedia (Wikipedia)
  - Crowdsourcing

KATARA

- Overview of workflow



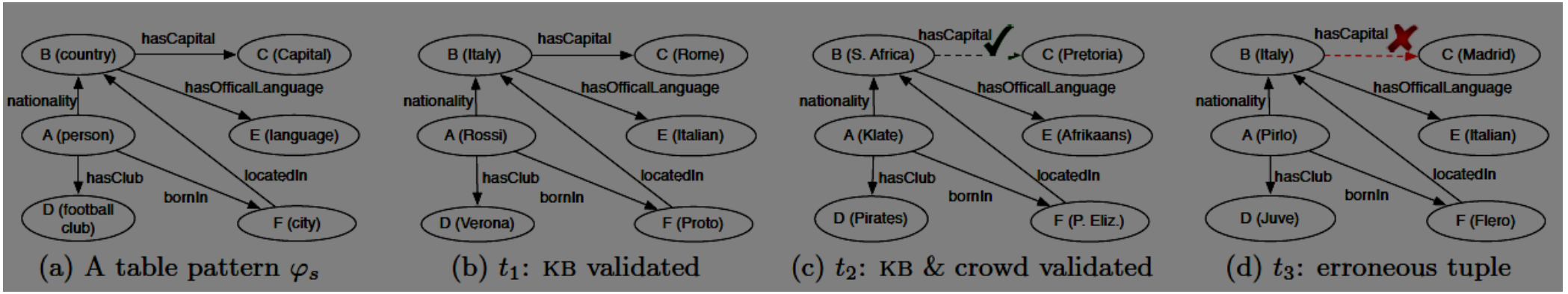
KATARA

- Contributions
  - Table pattern definition and discovery
    - new rank-join based algorithm to efficiently discover table patterns with high scores
  - Table pattern validation via crowdsourcing
    - entropy-based scheduling algorithm
  - Data annotation
    - algorithm to generate top-k possible repairs for those erroneous data

KATARA

Example: Pattern discovery and annotation

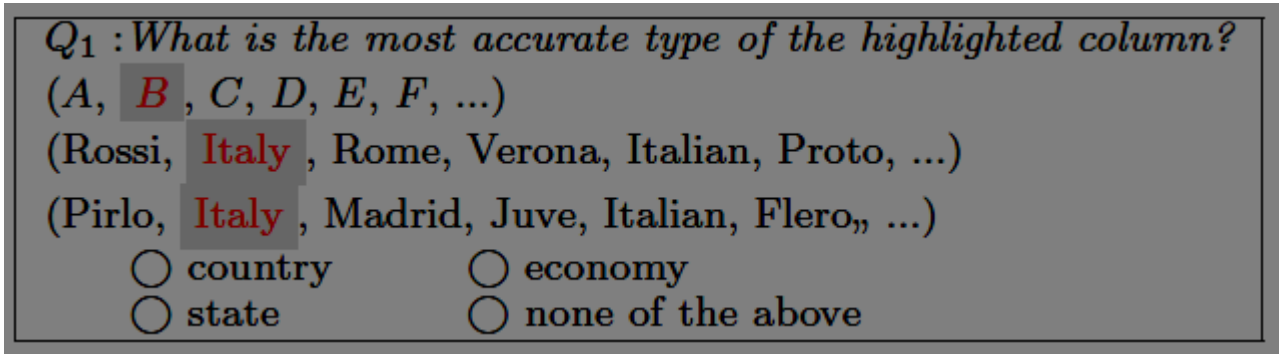
	A	B	C	D	E	F	G
t_1	Rossi	Italy	Rome	Verona	Italian	Proto	1.78
t_2	Klate	S. Africa	Pretoria	Pirates	Afrikaans	P. Eliz.	1.69
t_3	Pirlo	Italy	Madrid	Juve	Italian	Flero	1.77



(a) A table pattern φ_s (b) t_1: KB validated (c) t_2: KB & crowd validated (d) t_3: erroneous tuple

KATARA

- Example: question for the crowd



KATARA

- Experimental study results
  - Pattern discovery

	Support		MaxLike		PGM		RankJoin	
	P	R	P	R	P	R	P	R
WikiTables	.54	.59	.62	.68	.60	.67	.78	.86
WebTables	.65	.64	.63	.62	.77	.77	.86	.84
RelationalTables	.51	.51	.71	.71	.53	.53	.77	.77
Yago								
	P	R	P	R	P	R	P	R
WikiTables	.56	.70	.71	.89	.61	.77	.71	.89
WebTables	.65	.69	.80	.84	.76	.80	.82	.87
RelationalTables	.64	.67	.81	.86	.74	.77	.81	.86
DBPedia								

Table 2: Pattern discovery precision and recall

KATARA

- Experimental study results
  - Effectiveness of possible repairs

	KATARA (Yago)		KATARA (DBPedia)		EQ		SCARE	
	P	R	P	R	P	R	P	R
Person	1.0	0.80	1.0	0.94	1.0	0.96	0.78	0.48
Soccer	N.A.		0.97	0.29	0.66	0.29	0.66	0.37
University	0.95	0.74	1.0	0.18	0.63	0.04	0.85	0.21

Table 6: Data repairing precision and recall (RelationalTables)

	KATARA (Yago)		KATARA (DBPedia)		EQ	SCARE
	P	R	P	R	P/R	P/R
WikiTables	1.0	0.11	1.0	0.30	N.A.	
WebTables	1.0	0.40	1.0	0.46	N.A.	

Table 7: Data repairing precision and recall (WikiTables and WebTables)

KATARA

- Experimental study results
  - Pattern validation

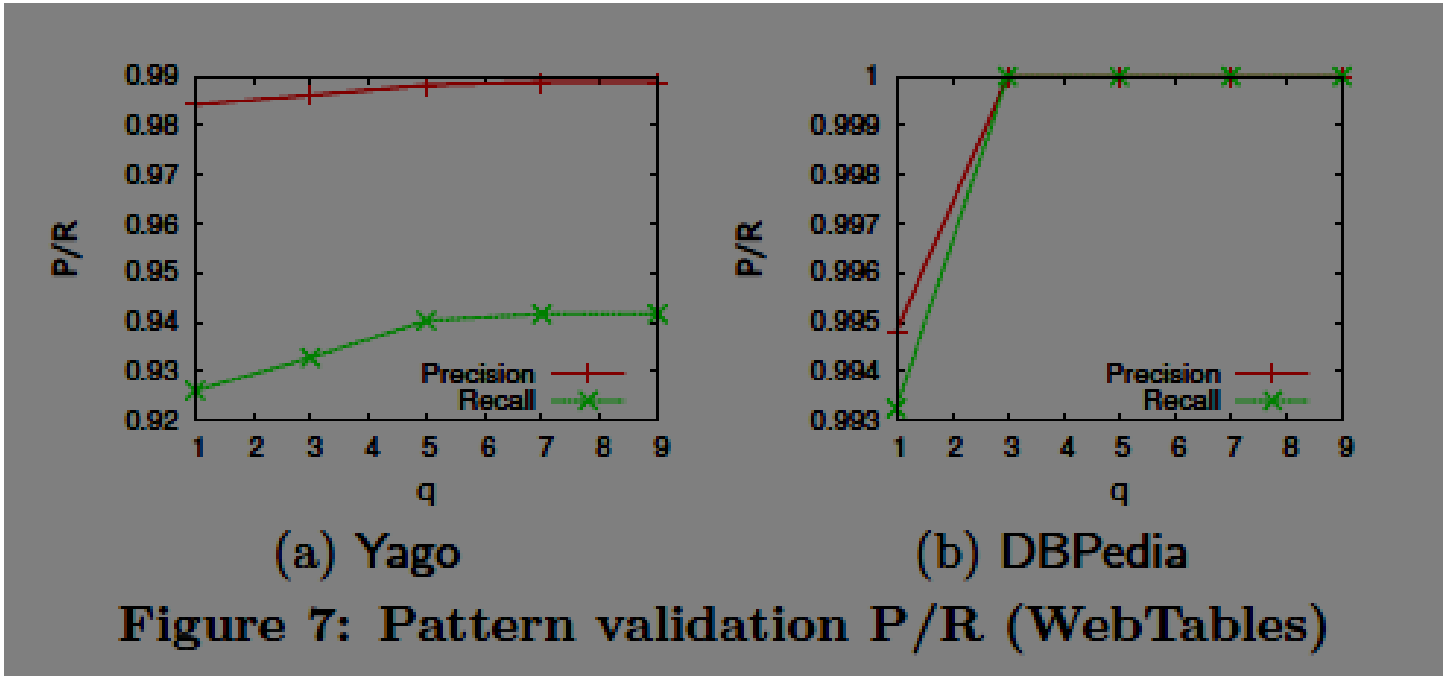


Figure 7: Pattern validation P/R (WebTables)

Discussion

- Strengths:
  - Automatic finding of repairs
  - Easy-to-answer questions presented to crowd
- Weaknesses:
  - If data is special-purpose, no KBs available
  - Not any crowd can be used!

Summary

- Dirty data is a big problem
- Even bigger problem with Big Data
- Traditional methods such as ICs don't scale to Big Data
- Knowledge Bases and crowdsourcing can help finding *accurate* repairs