Data Cleaning: Overview and Emerging Challenges

Xu Chu* Ihab F. Ilyas* Sanjay Krishnan# Jiannan Wang$
*University of Waterloo #UC Berkeley $Simon Fraser University
*{x4chu,ilyas}@uwaterloo.ca #sanjaykrishnan@berkeley.edu $jnwang@sfu.ca

ABSTRACT

Detecting and repairing *dirty* data is one of the perennial challenges in data analytics, and failure to do so can result in inaccurate analytics and unreliable decisions. Over the past few years, there has been a surge of interest from both industry and academia on data cleaning problems including new abstractions, interfaces, approaches for scalability, and statistical techniques. To better understand the new advances in the field, we will first present a taxonomy of the data cleaning literature in which we highlight the recent interest in techniques that use constraints, rules, or patterns to detect errors, which we call qualitative data cleaning. We will describe the state-of-the-art techniques and also highlight their limitations with a series of illustrative examples. While traditionally such approaches are distinct from quantitative approaches such as outlier detection, we also discuss recent work that casts such approaches into a statistical estimation framework including: using Machine Learning to improve the efficiency and accuracy of data cleaning and considering the effects of data cleaning on statistical analysis.

1. INTRODUCTION

It is becoming easier for enterprises to store and acquire the large amounts of data. These data sets can facilitate improved decision making, richer analytics, and increasingly, provide training data for Machine Learning. However, data quality remains to be a major concern, and *dirty data* can lead to incorrect decisions and unreliable analysis. Examples of common errors include missing values, typos, mixed formats, replicated entries of the same real-world entity, and violations of business rules. Analysts must consider the effects of dirty data before making any decisions, and as a result, data cleaning has been a key area of database research (see Johnson and Dasu [44] and Rahm and Do [63]).

Over the past few years, there has been a surge of interest from both industry and academia on different aspects of data cleaning including new abstractions [10, 30, 77, 22, 33, 14, 73], interfaces [1, 26], approaches for scalability [75, 45, 3, 53, 67], and crowdsourcing techniques [35, 69, 62, 18, 82, 23, 37, 60, 61, 76, 71, 78]. One of the key differentiating factors is how to *define* data error (i.e., error detection). Quantitative techniques, largely used for outlier detection, employ statistical methods to identify abnormal behaviors and errors (e.g., *"a salary that is three standard deviation away from the mean salary is an error"*). On the other hand, qualitative techniques use constraints, rules, patterns to detect errors (e.g., *"there cannot exist two employees of the same level, the one who is located in NYC is earning less than the one not in NYC"*). Once the errors are detected, repair can be performed using scripts, human crowds or experts, or a hybrid of both. Quantitative data cleaning techniques have been extensively covered in multiple surveys [2, 65, 40] and tutorials [48, 17], but there have been fewer surveys of qualitative data cleaning [44]. Accordingly, this tutorial focuses on the subject of qualitative data cleaning (in terms of both detection and repair), and we argue that much of the recent interest in data cleaning has a similar focus [14, 22, 33, 26, 73, 21, 82, 23, 10, 30, 77].

In the first part of the tutorial, we overview qualitative data cleaning with a taxonomy of error detection and error repairing approaches. We will describe the state-of-the-art techniques and also highlight their limitations with a series of illustrative examples. This section will focus on rule-based data cleaning techniques, where integrity constraints (ICs) are used to express data quality rules; any part of the data that does not conform to a given set of ICs is considered erroneous (also known as a violation of ICs). These rules can capture a wide variety of errors including duplication, inconsistency, and missing values. We conclude by discussing the challenges raised by "big data" era, and recent proposals for scalable data cleaning techniques. Most of the materials in the first part of the tutorial come from our survey in *Foundations and Trends in Databases* [41].

In the second part of the tutorial, we describe a statistical perspective on qualitative data cleaning, where approaches either use techniques from Machine Learning to improve accuracy or efficiency or consider the effects of cleaning on subsequent numerical queries. We present these approaches within the same overall taxonomy of data cleaning and show that many qualitative techniques are amenable to such statistical analysis. By considering the qualitative models in a rigorous statistical framework, we can understand the trade-off between cleaning and the ultimate accuracy of inferences made from the data. The materials for this section are inspired by our work on the SampleClean project [51].

Tutorial Structure: The intended audience are members of the academic and industrial research community. We will not require any prior background knowledge about data cleaning research, but assume familiarity with database research concepts. A basic understanding of the concepts and concerns in modern data analytics (e.g., training v.s. test data) will also be helpful. The tutorial is 3 hours split into two 1.5 hour sections.

SIGMOD'16, June 26-July 01, 2016, San Francisco, CA, USA

DOI: http://dx.doi.org/10.1145/2882903.2912574

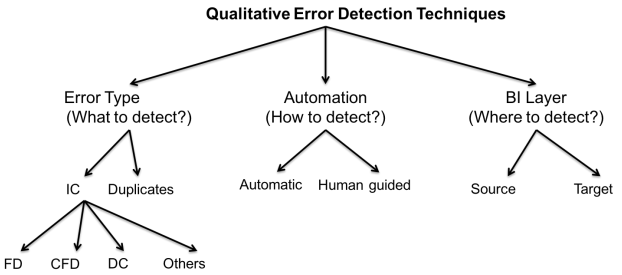

Figure 1: Classification of qualitative error detection techniques.

	Error Type What		Automation How		BI Layer Where	
	IC	Data deduplication	Automatic	Human involved	Source	Target
FDs value modification [13]	✓		✓		✓	
Holistic data cleaning [22]	✓		✓		✓	
CrowdER [74]		✓		✓	✓	
Corleone [35]		✓		✓	✓	
Causality Analysis [58]	✓			✓		✓
Scorpion [79]	✓			✓		✓
DBRx [16]	✓		✓			✓

Table 1: A sample of qualitative error detection techniques.

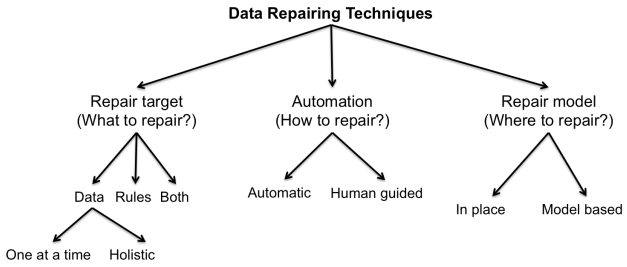

Figure 2: Classification of data repairing techniques.

2. DATA CLEANING OVERVIEW

Since data cleaning usually consists of two stages: error detection and error repairing, we will discuss a variety of techniques for qualitative error detection (Section 2.1), as well as various techniques for error repairing (Section 2.2). These techniques will be explained with a motivating example highlighting many different data quality problems, such as duplicates, missing values, integrity constraints violations, and outliers.

2.1 Qualitative Error Detection

Given a dirty database instance, the first step is to detect anomalies or errors. Figure 1 illustrates our taxonomy of qualitative error detection. There are three main questions that every technique needs to address: (1) "What to Detect", (2) "How to Detect", and "Where to Detect".

• *Error Type (What to Detect?)* Qualitative error detection techniques can be classified according to which type of errors are captured. In other words, what languages are used to describe patterns or constraints of a legal data instance. A large body of work uses integrity constraints (ICs), a fractional of first order logic, to capture data quality rules that the database should conform to, including functional dependencies (FDs) [13], and denial constraints (DCs) [22]. While duplicate records can be considered a violation of an integrity constraint (key constraint), we recognize the large body of work that focuses on this problem and we discuss it as a separate error type from other types of integrity constraints.

Manual designing such ICs or patterns require great domain expertise, and is time-consuming, automatic discovery techniques are essential and have been proposed for various ICs [21]. We classify the IC discovery techniques into schema-driven approaches and instance driven approaches, and we will discuss and compare these two approaches.

• *Automation (How to Detect?)* We classify proposed approaches according to whether and how humans are involved in the error detection process. Most techniques are fully automatic, for example, detecting violations of functional dependencies [13], while other techniques involve humans, for example, to identify duplicate records [74].

• *Business Intelligence Layer (Where to Detect?)* Errors can happen in all stages of a business intelligence (BI) stack, for example, errors in the source database are often propagated through the data processing pipeline. While most error detection techniques detect errors in the original database, some errors can only be discovered much later in the data processing pipeline [16], where more semantics and business logic are available, for example, constraints on total budget can only be enforced after aggregating cost and expenses.

Table 1 shows a sample of detection techniques, which cover all categories of the proposed taxonomy. We will give one or more example error detection techniques in each category in detail in the tutorial.

2.2 Error Repairing

Given a relational database instance I of schema R and a set of data quality requirements expressed in a variety of ways, data repairing refers to the process of finding another database instance I' that conforms to the set of data quality requirements. This problem has been extensively studied and Figure 2 depicts our taxonomy of the proposed data repairing techniques. Similar to error detection, there are three main questions that every technique needs to address: (1) "What to Repair?", (2) "How to Repair?", and (3) "Where to Repair?". In the following, we classify the techniques on these axes, and discuss the impact on the design and efficiency of the techniques.

• *Repair Target (What to Repair?)* Repairing algorithms make different assumptions about the data and the quality rules: (1) trusting the declared integrity constraints, and hence, only data can be updated to remove errors [22]; (2) trusting the data completely and allowing the relaxation of the constraints [36], for example, to address schema evolution and obsolete business rules; and finally (3) exploring the possibility of changing both the data and the constraints [11]. For techniques that trust the rules, and change only the data, they can be further divided according to the driver to the repairing exercise, that is, what types of errors they are targeting. A majority of techniques repair the data with respect to one type of errors only (one at a time), while other emerging techniques consider the interactions among multiple types of errors and provide a holistic repair of the data (holistic).

• *Automation (How to Repair?)* We classify proposed approaches with respect to the tools used in the repairing process. More specifically, we classify current repairing approaches according to whether and how humans are involved. Some techniques are fully automatic, for example, by modifying the database, such that the distance between the original database I and the modified database I' is minimized according to some cost function. Other techniques

	Repair target What				Automation How		Repair model Where	
	Data - One at a time	Data - Holistic	Rules	Both	Automatic	Human involved	In place	Model based
FDs value modification [13]	✓				✓		✓	
FDs hypergraph [46]	✓				✓		✓	
CFDs value modification [25]	✓				✓		✓	
Holistic data cleaning [22]		✓			✓		✓	
LLUNATIC [33]		✓			✓		✓	
Record matching and data repairing [31]		✓			✓		✓	
NADEEF [26]		✓			✓		✓	
Generate optimal tablaux [36]			✓		✓		✓	
Unified repair [19]				✓	✓		✓	
Relative trust [11]				✓	✓		✓	
Continuous data cleaning [73]				✓	✓		✓	
Potter's Wheel [64]	✓					✓	✓	
GDR [82]	✓					✓	✓	
KATARA [23]	✓					✓	✓	
DataTamer [69]	✓					✓	✓	
Editing rules [30]	✓					✓	✓	
Sampling FDs repairs [10]	✓				✓			✓
Sampling Duplicates [12]	✓				✓			✓

Table 2: A sample of data repairing techniques.

involve humans in the repairing process either to verify the fixes, to suggest fixes, or to train machine learning models to carry out automatic repairing decisions [82].

• *Repair Model (Where to Repair?)* We classify proposed approaches based on whether they change the database in-situ, or build a model to describe the possible repairs. Most proposed techniques repair the database in-situ, thus destructing the database. For none in-situ repairs, a model is often built to describe the different ways to repair the underlying database. Queries are answered by these models using, for example, sampling from all possible repairs and other probabilistic query answering mechanisms [10].

Table 2 shows a sample of data repairing techniques using the taxonomy. We will discuss one or more example error repairing techniques in each category in detail in the tutorial.

3. DATA CLEANING FROM A STATISTICAL PERSPECTIVE

As analytics become increasingly complex, it is important to understand the statistical implications of data cleaning. In this part of the tutorial, we will discuss approaches that either use techniques from Machine Learning to improve accuracy or efficiency, or consider the effects of cleaning on subsequent numerical queries. We will focus on deduplication (Duplication Error), repairing missing and incorrect values (Attribute Error), and the removal of erroneous or irrelevant data (Relevant Error). We will build on the taxonomy presented in the previous part, and overview recently proposed data cleaning algorithms and systems based on their relationship with statistics (Table 3).

3.1 Data Cleaning with Statistics

There are several techniques to improve the efficiency or accuracy of data cleaning algorithms using statistical methods such as Machine Learning.

Active Learning in Crowdsourcing: Crowdsourcing is widely applied in industry for data cleaning [56]. In academia, there is a growing consensus that crowds are difficult to scale [35, 69, 62, 18, 82, 23, 37, 60, 61], and as a result several recent works employ Active Learning to prioritize queries to the crowd [38, 35, 59]. The basic idea is to formulate the human-input to data cleaning as labels for a supervised learning technique (such as an SVM or Random Forest), and Active Learning is a class of algorithms that select the most informative labels to acquire.

Other Statistical Methods: There have also been several works that use statistical techniques to more accurately clean a dirty database, and this section of the tutorial will highlight a few of the seminal results. The Eracer project showed how data cleaning on dirty relations could be posed as a two-step learning problem: first learning a graphical model to represent the relation and message-passing algorithm to resolve inconsistencies [57]. Furthermore, in the sensor network literature (refer to a 2007 survey [6] and a 2010 survey [54]) there are several examples of statistical outlier detection and mitigation techniques. Finally, Yakout et al. [81] employ Machine Learning to improve the reliability of data cleaning.

3.2 Data Cleaning for Statistical Analysis

While larger datasets can facilitate training more sophisticated ML models, systematic data errors, i.e., corruption that affect particular records disproportionately, can make model training unreliable. It has repeatedly been found that ML problems are highly sensitive to dirty data, even when using robust techniques [52, 55, 49, 80], and the high-dimensionality of these models lead to counter-intuitive effects when trained after some types of data cleaning procedures [52]. For example, in one fraud prediction example, simply merging inconsistent attributes before SVM model training improved true positive detection probabilities from 62% to 91% [52]. This tutorial section studies the link between data cleaning and the subsequent analytics, and surveys works that try to analyze the effects of data cleaning on the analytics.

Aggregate Queries: The tutorial will first overview recent results on cleaning samples of data to estimate aggregate query results. SampleClean [75, 51] notes that for aggregates such as `sum`, `count`, and `avg` there are diminishing returns for data cleaning, and it often suffices to clean small samples of data to estimate results with high accuracy. This problem was further extended to study aggregate queries on materialized views [50]. We will also briefly describe the related fields of query-driven cleaning and consistent query answering [9, 7, 4] as these works have studied the problem of how inconsistencies affect individual queries.

Machine Learning: SampleClean was extended to study data cleaning that precedes Machine Learning model training in a system called ActiveClean [52]. ActiveClean employs selection techniques for the most valuable data and techniques to incrementally update ML models given newly clean data. One of the interesting findings of this work is that progressive data cleaning and model training do not commute in an expected way. Suppose $k \ll N$ records are cleaned, but all of the remaining dirty records are retained in the dataset. Aggregates over mixtures of different populations of data can result in spurious relationships due to the well-known phenomenon called Simpson's paradox [68]. Some of these problems are not apparent in 1D analytics such as `sum`, `count`, and `avg`, but can lead to subtle biases in higher-dimensional statistical analysis.

Adaptive Data Analysis: In a recent development, the statistics community has studied some of these problems in a field called "Adaptive Data Analysis", and we will also overview some of that work [66, 29]. Concepts such as Multiple Hypothesis Testing and False Discovery Rate are highly relevant to the design of analysis tools [66, 29]. These works consider the problem of false discoveries, where analysts discover non-existent trends in a dataset due to statistical chance. In the tutorial, we will discuss recent results and new opportunities for the database community to build these techniques into new tools.

Type of Error	Operation	Uses Statistics	For Statistics
Attribute Error	Value Imputation	[57, 72, 8, 81, 62, 23]	[75, 52, 43, 24]
	Data Repairing	[1, 57, 8, 37]	[75, 52]
Duplication Error	Entity Resolution	[39, 24, 81, 69]	[35, 37]
Relevance Error	Error/Outlier Removal	[27, 70, 28, 15, 34]	[52]

Table 3: We categorize the capabilities of recently proposed data cleaning algorithms and systems (some may fit in multiple categories). "Uses Statistics" are the ones that use a statistical model (e.g., a Probabilistic Graphical Model) to identify and correct errors. "For Statistics" are the ones that are explicitly designed to support data cleaning for aggregate analytics and advanced statistical analytics.

4. NEW CHALLENGES

We highlight many emerging trends in data cleaning research. Both Part 1 and Part 2 of the tutorial will go through some of them.

Scalability. Scaling data cleaning techniques to the large and rapidly growing datasets of the Big Data era will be an important challenge. Current techniques include blocking for duplicate detection [5], sampling for data cleaning [75], and distributed data cleaning [47, 45, 20].

User Engagement. Although much research has been done about involving humans to perform data deduplication, for example, through active learning, involving humans in other data cleaning tasks, such as repairing IC violations, and taking user feedback in discovering of data quality rules, is yet to be explored.

Semi-structured and unstructured data. A significant portion of data is residing in semi-structured formats, e.g., JSON, and unstructured formats, e.g., text documents. Data quality problems for semi-structured and unstructured data remain largely unexplored.

New Applications for Streaming Data. There is a renewed interest in considering data collected from vast collections of sensors and mobile devices. Gartner estimates that there will be 26 billion devices on the Internet-of-Things (IoT) by 2020 [32]. 5-10 years ago the data management and quality challenges from distributed sensors was an important research topic, e.g., [54, 27, 42, 43]. However, most of the prior work on data cleaning in this domain has relied on many quantitative such as outlier detection. One avenue for future work is to consider how qualitative data cleaning approaches will work on distributed streams of data.

Growing Privacy and Security Concerns. Finally, there are significant concerns about data privacy as increasingly more individual data are collected by enterprises. Data cleaning is by nature a task that requires examining and searching through raw data, which may be restricted in some domains including finance and medicine. An important challenge will be to reconcile the need for data provenance, access to unaggregated data, and privacy.

5. CONCLUSION

Detecting and repairing *dirty* data is one of the perennial challenges in data analytics, and failure to do so can result in inaccurate analytics and unreliable decisions. Over the past few years, there has been a surge of interest from both industry and academia on different aspects of this problem including new abstractions, interfaces, and approaches for scalability. This tutorial focused on qualitative data cleaning which uses constraints, rules, or patterns to detect errors. While this subject has traditionally been distinct from quantitative statistical approaches for cleaning, we described the growing relationship between the two branches of literature. Most of the materials for this tutorial can be found in *Foundations and Trends in Databases* [41], and in an overview of the *SampleClean* project [51].

6. REFERENCES

[1] Trifacta. http://www.trifacta.com.

[2] C. C. Aggarwal. *Outlier Analysis*. Springer, 2013.

[3] Y. Altowim, D. V. Kalashnikov, and S. Mehrotra. Progressive approach to relational entity resolution. *PVLDB*, 7(11), 2014.

[4] H. Altwaijry, S. Mehrotra, and D. V. Kalashnikov. Query: A framework for integrating entity resolution with query processing. *PVLDB*, 9(3):120–131, 2015.

[5] R. Ananthakrishna, S. Chaudhuri, and V. Ganti. Eliminating fuzzy duplicates in data warehouses. In *PVLDB*, pages 586–597, 2002.

[6] M. Balazinska, A. Deshpande, M. J. Franklin, P. B. Gibbons, J. Gray, M. H. Hansen, M. Liebhold, S. Nath, A. S. Szalay, and V. Tao. Data management in the worldwide sensor web. *IEEE Pervasive Computing*, 6(2):30–40, 2007.

[7] M. Bergman, T. Milo, S. Novgorodov, and W. C. Tan. Query-oriented data cleaning with oracles. In *SIGMOD*, 2015.

[8] L. Berti-Equille, T. Dasu, and D. Srivastava. Discovery of complex glitch patterns: A novel approach to quantitative data cleaning. In *ICDE*, pages 733–744, 2011.

[9] L. E. Bertossi. Consistent query answering in databases. *SIGMOD Record*, 35(2):68–76, 2006.

[10] G. Beskales, I. F. Ilyas, and L. Golab. Sampling the repairs of functional dependency violations under hard constraints. *PVLDB*, 3(1-2):197–207, 2010.

[11] G. Beskales, I. F. Ilyas, L. Golab, and A. Galiullin. On the relative trust between inconsistent data and inaccurate constraints. In *ICDE*, pages 541–552, 2013.

[12] G. Beskales, M. A. Soliman, I. F. Ilyas, and S. Ben-David. Modeling and querying possible repairs in duplicate detection. *PVLDB*, pages 598–609, 2009.

[13] P. Bohannon, W. Fan, M. Flaster, and R. Rastogi. A cost-based model and effective heuristic for repairing constraints by value modification. In *SIGMOD*, pages 143–154. ACM, 2005.

[14] P. Bohannon, W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. Conditional functional dependencies for data cleaning. In *ICDE*, pages 746–755, 2007.

[15] L. Cao, D. Yang, Q. Wang, Y. Yu, J. Wang, and E. A. Rundensteiner. Scalable distance-based outlier detection over high-volume data streams. In *ICDE*, pages 76–87, 2014.

[16] A. Chalamalla, I. F. Ilyas, M. Ouzzani, and P. Papotti. Descriptive and prescriptive data cleaning. In *SIGMOD*, pages 445–456, 2014.

[17] S. Chawla and P. Sun. Outlier detection: Principles, techniques and applications. In *PAKDD*, 2006.

[18] Z. Chen and M. Cafarella. Integrating spreadsheet data via accurate and low-effort extraction. In *KDD*. ACM, 2014.

[19] F. Chiang and R. J. Miller. A unified model for data and constraint repair. In *ICDE*, pages 446–457, 2011.

[20] X. Chu, I. F. Ilyas, and P. Koutris. Distributed Data Deduplication. Technical Report CS-2016-02, University of Waterloo, 2016.

[21] X. Chu, I. F. Ilyas, and P. Papotti. Discovering denial constraints. *PVLDB*, 6(13):1498–1509, 2013.

[22] X. Chu, I. F. Ilyas, and P. Papotti. Holistic data cleaning: Putting violations into context. In *ICDE*, pages 458–469, 2013.

[23] X. Chu, J. Morcos, I. F. Ilyas, M. Ouzzani, P. Papotti, N. Tang, and Y. Ye. KATARA: A data cleaning system powered by knowledge bases and crowdsourcing. In *SIGMOD*, pages 1247–1261, 2015.

[24] Y. Chung, M. L. Mortensen, C. Binnig, and T. Kraska. Estimating the impact of unknown unknowns on aggregate query results. *CoRR*, abs/1507.05591, 2015.

[25] G. Cong, W. Fan, F. Geerts, X. Jia, and S. Ma. Improving data quality: Consistency and accuracy. In *PVLDB*, pages 315–326. VLDB Endowment, 2007.

[26] M. Dallachiesa, A. Ebaid, A. Eldawy, A. Elmagarmid, I. F. Ilyas, M. Ouzzani, and N. Tang. Nadeef: a commodity data cleaning system. In *SIGMOD*, pages 541–552, 2013.

[27] A. Deligiannakis, Y. Kotidis, V. Vassalos, V. Stoumpos, and A. Delis. Another outlier bites the dust: Computing meaningful aggregates in sensor networks. In *ICDE*, pages 988–999, 2009.

[28] A. Deshpande, C. Guestrin, S. Madden, J. M. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *PVLDB*, pages 588–599, 2004.

[29] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. L. Roth. Preserving statistical validity in adaptive data analysis. In *STOC*, pages 117–126, 2015.

[30] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. Towards certain fixes with editing rules and master data. *PVLDB*, 3(1-2):173–184, 2010.

[31] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. Interaction between record matching and data repairing. In *SIGMOD*, pages 469–480. ACM, 2011.

[32] Gartner. Forecast: The internet of things, worldwide. https://www.gartner.com/doc/2625419/forecast-internet-things-worldwide-.

[33] F. Geerts, G. Mecca, P. Papotti, and D. Santoro. The llunatic data-cleaning framework. *PVLDB*, 6(9):625–636, 2013.

[34] D. Georgiadis, M. Kontaki, A. Gounaris, A. N. Papadopoulos, K. Tsichlas, and Y. Manolopoulos. Continuous outlier detection in data streams: an extensible framework and state-of-the-art algorithms. In *SIGMOD*, pages 1061–1064, 2013.

[35] C. Gokhale, S. Das, A. Doan, J. F. Naughton, N. Rampalli, J. Shavlik, and X. Zhu. Corleone: Hands-off crowdsourcing for entity matching. In *SIGMOD*, 2014.

[36] L. Golab, H. Karloff, F. Korn, D. Srivastava, and B. Yu. On generating near-optimal tableaux for conditional functional dependencies. *PVLDB*, 1(1):376–390, 2008.

[37] D. Haas, S. Krishnan, J. Wang, M. J. Franklin, and E. Wu. Wisteria: Nurturing scalable data cleaning infrastructure. *PVLDB*, 8(12), 2015.

[38] D. Haas, J. Wang, E. Wu, and M. J. Franklin. Clamshell: Speeding up crowds for low-latency data labeling. *PVLDB*, 9(4):372–383, Dec. 2015.

[39] A. Heise, G. Kasneci, and F. Naumann. Estimating the number and sizes of fuzzy-duplicate clusters. In *CIKM Conference*, 2014.

[40] J. M. Hellerstein. Quantitative data cleaning for large databases. *United Nations Economic Commission for Europe (UNECE)*, 2008.

[41] I. F. Ilyas and X. Chu. Trends in cleaning relational data: Consistency and deduplication. *Foundations and Trends in Databases*, 5(4):281–393, 2015.

[42] S. R. Jeffery, G. Alonso, M. J. Franklin, W. Hong, and J. Widom. A pipelined framework for online cleaning of sensor data streams. In *ICDE*, 2006.

[43] S. R. Jeffery, M. N. Garofalakis, and M. J. Franklin. Adaptive cleaning for RFID data streams. In *Proceedings of the 32nd International Conference on Very Large Data Bases, Seoul, Korea, September 12-15, 2006*, pages 163–174, 2006.

[44] T. Johnson and T. Dasu. Data quality and data cleaning: An overview. In *SIGMOD*, page 681, 2003.

[45] Z. Khayyat, I. F. Ilyas, A. Jindal, S. Madden, M. Ouzzani, P. Papotti, J. Quiané-Ruiz, N. Tang, and S. Yin. Bigdansing: A system for big data cleansing. pages 1215–1230, 2015.

[46] S. Kolahi and L. V. S. Lakshmanan. On approximating optimum repairs for functional dependency violations. In *ICDT*, pages 53–62, 2009.

[47] L. Kolb, A. Thor, and E. Rahm. Dedoop: efficient deduplication with hadoop. *PVLDB*, 5(12):1878–1881, 2012.

[48] H.-P. Kriegel, P. Kröger, and A. Zimek. Outlier detection techniques. In *Tutorial at SIGKDD*, 2010.

[49] S. Krishnan, J. Patel, M. J. Franklin, and K. Goldberg. A methodology for learning, analyzing, and mitigating social influence bias in recommender systems. In *RecSys*, 2014.

[50] S. Krishnan, J. Wang, M. J. Franklin, K. Goldberg, and T. Kraska. Stale view cleaning: Getting fresh answers from stale materialized views. *PVLDB*, 8(12), 2015.

[51] S. Krishnan, J. Wang, M. J. Franklin, K. Goldberg, T. Kraska, T. Milo, and E. Wu. Sampleclean: Fast and reliable analytics on dirty data. *IEEE Data Eng. Bull.*, 38(3):59–75, 2015.

[52] S. Krishnan, J. Wang, E. Wu, M. J. Franklin, and K. Goldberg. Activeclean: Interactive data cleaning while learning convex loss models. In *Arxiv: http://arxiv.org/pdf/1601.03797.pdf*, 2015.

[53] Z. Li, S. Shang, Q. Xie, and X. Zhang. Cost reduction for web-based data imputation. In *Database Systems for Advanced Applications*, pages 438–452. Springer, 2014.

[54] S. Madden. Database abstractions for managing sensor network data. *Proceedings of the IEEE*, 98(11):1879–1886, 2010.

[55] J. Mahler, S. Krishnan, M. Laskey, S. Sen, A. Murali, B. Kehoe, S. Patil, J. Wang, M. Franklin, P. Abbeel, and K. Y. Goldberg. Learning accurate kinematic control of cable-driven surgical robots using data cleaning and gaussian process regression. In *CASE*, 2014.

[56] A. Marcus and A. Parameswaran. Crowdsourced data management: Industry and academic perspectives. *Foundations and Trends in Databases*, 6(1-2):1–161, 2013.

[57] C. Mayfield, J. Neville, and S. Prabhakar. ERACER: a database approach for statistical inference and data cleaning. In *SIGMOD*, 2010.

[58] A. Meliou, W. Gatterbauer, S. Nath, and D. Suciu. Tracing data errors with view-conditioned causality. In *SIGMOD*, pages 505–516, 2011.

[59] B. Mozafari, P. Sarkar, M. J. Franklin, M. I. Jordan, and S. Madden. Scaling up crowd-sourcing to very large datasets: A case for active learning. *PVLDB*, 8(2), 2014.

[60] A. Parameswaran, H. Garcia-Molina, H. Park, N. Polyzotis, A. Ramesh, and J. Widom. Crowdscreen: Algorithms for filtering data with humans.

[61] A. Parameswaran and N. Polyzotis. Answering queries using humans, algorithms and databases. 2011.

[62] H. Park and J. Widom. Crowdfill: collecting structured data from the crowd. In *SIGMOD*, 2014.

[63] E. Rahm and H. H. Do. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4), 2000.

[64] V. Raman and J. M. Hellerstein. Potter's wheel: An interactive data cleaning system. In *VLDB*, pages 381–390, 2001.

[65] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*, volume 589. John Wiley & Sons, 2005.

[66] D. Russo and J. Zou. Controlling bias in adaptive data analysis using information theory. *CoRR*, abs/1511.05219, 2015.

[67] G. Simoes, H. Galhardas, and L. Gravano. When speed has a price: Fast information extraction using approximate algorithms. *PVLDB*, 6(13):1462–1473, 2013.

[68] E. H. Simpson. The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1951.

[69] M. Stonebraker, D. Bruckner, I. F. Ilyas, G. Beskales, M. Cherniack, S. B. Zdonik, A. Pagan, and S. Xu. Data curation at scale: The data tamer system. In *CIDR*, 2013.

[70] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos. Online outlier detection in sensor data using non-parametric models. In *PVLDB*, pages 187–198, 2006.

[71] Y. Tong, C. C. Cao, C. J. Zhang, Y. Li, and L. Chen. Crowdcleaner: Data cleaning for multi-version data on the web via crowdsourcing. In *ICDE*, pages 1182–1185, 2014.

[72] R. Verborgh and M. De Wilde. *Using OpenRefine*. Packt Publishing Ltd, 2013.

[73] M. Volkovs, F. Chiang, J. Szlichta, and R. J. Miller. Continuous data cleaning. In *ICDE*, pages 244–255, 2014.

[74] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. Crowder: Crowdsourcing entity resolution. *PVLDB*, 5(11), 2012.

[75] J. Wang, S. Krishnan, M. J. Franklin, K. Goldberg, T. Kraska, and T. Milo. A sample-and-clean framework for fast and accurate query processing on dirty data. In *SIGMOD*, 2014.

[76] J. Wang, G. Li, T. Kraska, M. J. Franklin, and J. Feng. Leveraging transitive relations for crowdsourced joins. In *SIGMOD*, pages 229–240, 2013.

[77] J. Wang and N. Tang. Towards dependable data repairing with fixing rules. In *SIGMOD*, pages 457–468. ACM, 2014.

[78] S. E. Whang, P. Lofgren, and H. Garcia-Molina. Question selection for crowd entity resolution. *PVLDB*, 6(6):349–360, Apr. 2013.

[79] E. Wu and S. Madden. Scorpion: Explaining away outliers in aggregate queries. *PVLDB*, 6(8):553–564, 2013.

[80] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli. Is feature selection secure against training data poisoning? In *ICML*, 2015.

[81] M. Yakout, L. Berti-Equille, and A. K. Elmagarmid. Don't be scared: use scalable automatic repairing with maximal likelihood and bounded changes. In *SIGMOD*, 2013.

[82] M. Yakout, A. K. Elmagarmid, J. Neville, M. Ouzzani, and I. F. Ilyas. Guided data repair. *PVLDB*, 4(5):279–289, 2011.

Biographies

Xu Chu

Xu Chu is a PhD student in the Cheriton School of Computer Science at the University of Waterloo. His main research interests are data quality and data cleaning. He won the prestigious Microsoft Research PhD fellowship in 2015. Xu has also received Cheriton Fellowship from the University of Waterloo 2013-2015.

Ihab F. Ilyas

Ihab Ilyas is a professor in the Cheriton School of Computer Science at the University of Waterloo. He received his PhD in computer science from Purdue University, West Lafayette. His main research is in the area of database systems, with special interest in data quality, managing uncertain data, rank-aware query processing, and information extraction. Ihab is a recipient of the Ontario Early Researcher Award (2009), a Cheriton Faculty Fellowship (2013), an NSERC Discovery Accelerator Award (2014), and a Google Faculty Award (2014), and he is an ACM Distinguished Scientist. Ihab is a co-founder of Tamr, a startup focusing on large-scale data integration and cleaning. He serves on the VLDB Board of Trustees, and he is an associate editor of the ACM Transactions of Database Systems (TODS).

Sanjay Krishnan

Sanjay Krishnan is a Computer Science PhD candidate in the Algorithms, Machines, and People Lab (AMPLab) and in the Berkeley Laboratory for Automation Science and Engineering at UC Berkeley. His research studies techniques for data analytics on dirty data and data representation problems in physical systems.

Jiannan Wang

Jiannan Wang is an Assistant Professor of Computing Science at Simon Fraser University. His research is focused on developing algorithms and systems for extracting value from dirty data. Prior to that, he was a postdoc in the AMPLab at UC Berkeley. He obtained his PhD from the Computer Science Department at Tsinghua University. During his PhD, he has been a visiting scholar at Chinese University of Hong Kong and UC Berkeley, and an intern at Qatar Computing Research Institute. His PhD research work was supported from a Google PhD Fellowship, a Boeing Scholarship, and a New PhD Researcher Award by Chinese Ministry of Education. His PhD dissertation won the China Computer Federation (CCF) Distinguished Dissertation Award. His similarity-join algorithm won first place of EDBT String Similarity Search/Join Competition.