Towards Reliable Interactive Data Cleaning: A User Survey and Recommendations

Sanjay Krishnan, Daniel Haas, Michael J. Franklin, Eugene Wu [†]
AMPLab, UC Berkeley [†]Columbia University
{sanjay,dhaas,franklin}@cs.berkeley.edu, ewu@cs.columbia.edu

ABSTRACT

Data cleaning is frequently an iterative process tailored to the requirements of a specific analysis task. The design and implementation of iterative data cleaning tools presents novel challenges, both technical and organizational, to the community. In this paper, we present results from a user survey ($N = 29$) of data analysts and infrastructure engineers from industry and academia. We highlight three important themes: (1) the iterative nature of data cleaning, (2) the lack of rigor in evaluating the correctness of data cleaning, and (3) the disconnect between the analysts who query the data and the infrastructure engineers who design the cleaning pipelines. We conclude by presenting a number of recommendations for future work in which we envision an interactive data cleaning system that accounts for the observed challenges.

1. INTRODUCTION

Preparing and cleaning datasets prior to analysis is a perennial challenge in data analytics. As it has become easier to acquire and store ever larger datasets, the challenges associated with large-scale *data cleaning*, wherein issues caused by incorrect, missing, and duplicate data are identified and repaired, has become a subject of intense interest in the academic community e.g., [12,19,26]. While there has been significant progress in the design and implementation of data cleaning algorithms, data cleaning remains expensive and time-consuming in terms of analyst effort [32]. Almost all data cleaning software requires some level of analyst supervision, on a spectrum from defining data quality rules to actually manually identifying and fixing errors. Consequently, this paper presents explores how data analysts use such tools, and what changes must be made to make data cleaning faster and more reliable.

Traditionally, data cleaning routines, sequences of transformations such as deduplication or outlier removal that convert raw data into a format useful for analysis, have been viewed as static components that fit into data integration or Extract-Transform-Load (ETL) pipelines and are executed once on new data entering the system [1, 2,5,15]. However, this perspective fails to take into account that data cleaning is frequently an iterative process tailored to the requirements and semantics of a specific analysis task. As a result, several systems have been developed recently to support the iterative specification and refinement of data cleaning workflows [6,19, 22,38]. These human-in-the-loop cleaning systems are inherently interactive, and their design and implementation presents novel problems at the intersection of human factors and database research.

The data cleaning community has long studied abstractions for modeling data error and designing large scale cleaning systems, and we believe the time is ripe to focus attention towards usability and interactivity. We conducted a survey and interview study of 29 data analysts, data engineers, and others who work heavily with data. Though the number of participants is insufficient to draw statistically significant conclusions, we nevertheless present a qualitative selection of our initial survey results that expose several important themes in the workflows, methodologies, and challenges faced by practitioners today. Driven by these insights and building on our collective prior work on this subject, we present a number of recommendations for future of data cleaning systems.

In particular, our survey results highlight three main themes: (1) analysts clean data in a non-linear and iterative process interleaving analysis and cleaning, (2) debugging and validating data cleaning is a major concern, and (3) the disconnect between the analysts who query the data and the infrastructure engineers who design the data cleaning routines serves as a major bottleneck. Based on these results, we propose a series of recommendations to better match academic data cleaning systems research with industrial practice. We describe simplification of data cleaning operators through high-level language design to streamline systems that both infrastructure and analysis staff can use, the opportunities for joint optimization over cleaning and query processing that such a system will create, and a better suite of tools to track lineage, debug, and validate data cleaning.

2. BACKGROUND AND RELATED WORK

Over the last two decades, data cleaning has been a key area of database research (see surveys [14], [34], and tutorial [11]). Even so, precisely defining data cleaning has been a challenging problem because there is a gap between the theoretical abstractions of data quality research (e.g., constraint-based cleaning) and the prevalent script-hacking done by data scientists. To understand data cleaning practice, Kandel et al. conducted a seminal interview study of industry to identify the key challenges in data analytics [25]. Our paper revisits three conclusions from [25]: (1) analysts engaged in a non-linear and iterative processes, (2) analysts often work closely with IT staff to acquire and clean data, and (3) existing tools make it difficult to communicate assumptions, i.e., which data have been removed.

Since Kandel et al. there have been several new developments in data analytics such as the growing adoption of Machine Learn-

HILDA'16, June 26 2016, San Francisco, CA, USA
© 2016 ACM. ISBN 978-1-4503-4207-0/16/06…$15.00
DOI: http://dx.doi.org/10.1145/2939502.2939511

ing in industry and the proliferation of in-memory, low-latency data processing frameworks such as Apache Spark. Our survey focuses on these points and how they affect the three aforementioned themes. Since no existing systems address the end-to-end iterative data cleaning process, our analysis provides useful guidance for the design of systems in the future.

Existing systems fall into two major categories. First, extract-transform-load (ETL) systems [1,2,5] require developers to manually write data cleaning rules and execute them as long batch jobs, and constraint-driven tools allow analysts to define "data quality rules" and automatically propose corrections to maximally satisfy these rules [13]. These frameworks are largely aimed towards IT/DevOPs staff and do not provide the opportunity for analyst iteration or user feedback– inhibiting the user's ability to rapidly prototype different data cleaning solutions. On the other hand, projects such as Wrangler [6,23] and OpenRefine [38] support iteration with spreadsheet-style interfaces that enable the user to compose data cleaning sequences by directly manipulating a sample of the data and applying these sequences to the full dataset. However, they are limited to specific cleaning tasks (extraction) and sit outside the critical path of the main data cleaning routine. Ideally, we would like a framework that is processing data as it arrives, like the ETL tools, but supports the interactivity of tools like Trifacta.

While human-in-the-loop data cleaning systems have been extensively studied [10,12,17,19,33,37,41], a key missing piece is a detailed study of how data analysts interact with such systems. In prior work, we have studied a number of aspects of this problem, including reducing the latency of human operations [20], understanding quality in macrotasks [18], reducing latency with sampling [29], mitigating cognitive biases [27], addressing privacy concerns [30], and defining statistical correctness for data cleaning [28, 31]. With this survey, we contextualize these research questions and propose recommendations for the future.

3. DATA CLEANING SURVEY

Between April 2015 and Dec 2015, we conducted two sets of surveys and interviews with engineers, data analysts, and scientists who self-reported that they directly work with data in their organizations (N=29).

3.1 Methods

The surveys consisted of a series of quantitative questions about tool/language preference and job description, and several open-ended questions about the participants' organization's data management challenges. The interviews followed the script of the surveys and audio was recorded. It is important to note that we conducted two separate surveys to ensure that our survey questions were properly calibrated. The first survey was conducted with a preliminary set of 53 questions in June 2015, and we collected 5 written responses and 4 in-person/phone interviews. In November 2015 we conducted a revised second survey with 18 more focused questions, and collected 21 written responses. Most of the participants were contacted personally by the authors and were often acquaintances. However, the authors took care to ensure that the participants were not informed of any of the quantitative hypotheses or conclusions of the study before taking the survey. Other participants were reached through forums frequented by data analysts[1], and all participants had the option to take the survey anonymously.

It is important to note that this survey is not meant to be a statistically representative sample of industrial practice, and the number of participants is insufficient to draw statistically significant conclusions. Therefore, we present qualitative results around the insights learned from the participants, and contextualize these insights based on the participants' self-reported demographics. The questions used in surveys are available at [7,8].

Size	Number	Job Desc.	Number
Small	7	Infrastructure	10
Large	17	Analysis	12
N/A	5	Both	7

Table 1: Categorized responses to the question "Describe your company/organization and your role there." We defined a large organization as one with > 100 employees. To determine the job description, there was a clarifying question "I develop infrastructure to process incoming and historical data at scale for use by other business units.".

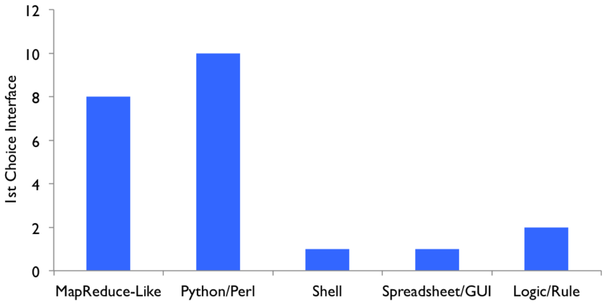

Figure 1: Top ranked responses to: "Which of the following tools/interfaces/languages do data scientists at your organization prefer for manipulating data, including extraction, schema transformations, and outlier removal, to make analysis easier or more reliable. Please Rank."

3.1.1 *Participant Demographics*

The survey asked a series of questions about the participants' job descriptions, expertise, and use of certain tools/interfaces. We briefly summarize the results.

Job Descriptions: We requested participants to provide a job description and details about their organizations. We categorized the participants by their reported organization size and their roles. The participants were mostly from larger organizations (defined as > 100 employees). We also found that they were mostly evenly split between infrastructure and data analytics. A surprisingly large number (7/30) reported that they performed both roles in their organizations. The results are summarized in Table 1.

Data Products: Machine Learning is an increasingly popular use-case for large datasets. We asked participants who self-reported as data analysts whether they work with Machine Learning. We found that 11/17 "data analyst" participants reported working with machine learning models.

Tools/Interfaces For Cleaning: Next, we asked participants about the existing tools and interfaces they used for data cleaning. This set of questions was only asked in the second survey. Figure 1 shows the results. We find that most of the participants responded that they used Python/Perl or MapReduce-like frameworks (clarified in the survey to be Spark/Hadoop etc.) to manipulate data before analysis. A minority of participants responded that they used graphical interfaces or rule-based interfaces to clean data.

[1] http://reddit.com/r/datascience

3.2 Themes

We highlight several of the themes we discovered from the survey and the interviews: data cleaning methodology and the tension between infrastructure engineers and data analysts.

3.2.1 Data Cleaning Methodology

Responses from participants who self-reported as data analysts to questions about data cleaning methodology revealed two important themes: the iterative nature of data cleaning, and the lack of rigor in evaluating cleaning workflows.

Data cleaning is iterative: Confirming the findings of Kandel et al., we found that many participants reported data cleaning to be an interactive and iterative process. For example, as one participant noted,

> *[It's an] iterative process, where I assess biggest problem, devise a fix, re-evaluate. It is dirty work.*

We broadly interpreted iteration to mean that analysts alternate between cleaning data and analysis, and using the analysis to guide future cleaning cleaning results. A natural concern raised by this approach is over-fitting: cleaning data until the output of a specific analysis is achieved. We asked questions to explore how this iterative process may affect results, especially in the context of confirmation bias.

Evaluating data cleaning is ad-hoc: The responses to the question "How do you determine whether the data is sufficiently clean to trust the analysis?" made it clear that many analysts had not thought hard about how they go about evaluating their cleaning workflows:

> *Other than common sense we do not have a procedure to do this.*

> *We usually do not do rigorous validation of data cleaning. We typically clean our data until the desired analytics works without error. This is not desirable but practical since in most cases data error is probably overshadowed by errors/inaccuracies in the models themselves.*

Iteration coupled with a lack of evaluation methodology is particularly worrisome. In their defense, some analysts were aware of the potential of such problems, but did not have a solution. For example, One analyst suggested comparing to other published results as a sanity check:

> *We typically cross-reference data with other published materials to make sure it is in the right ballpark.*

This solution may work in some cases, but many data analysis projects in industry are one-of-a-kind, and for most there likely does not exist a publicly available gold standard against which to validate results. These results emphasize that the data cleaning community needs to have a better answer to this problem, especially in the backdrop of the reproducibility crisis in science [3].

3.2.2 Analysts vs. Infrastructure

Another important theme that we discovered in the data was the divide between infrastructure engineers and analysts in how these groups address data quality problems. In particular, we see a difference in the way that these two groups of participants conceptualize dirty data, the solutions, and repair procedures.

One of the most significant tensions between the infrastructure engineers and analysts is about the definition of dirty data. While the infrastructure engineers are in charge of the data ingest pipelines, ETL, and other pre-processing steps, it is often the analysts who get to "define" what is dirty. One infrastructure engineer noted the frustration about being caught in the middle between the business units that generated the data and the analysts querying the data:

> *There are often long back and forths with senior data scientists, devs, and the business units that provided the data on data quality. It is almost never a smooth process. The vast majority of problems are in turning semi-structured data into features. What placeholder value is sensible to use for a missing value, do we replace it with the mean or nearest neighbor; or is a variable ordinal or categorical? These are tough questions that often can only be answered by the business unit themselves. We try to get them to do some of this work but it inevitably falls on us esp. if it is a big unit.*

The tension between the infrastructure engineers and the data analysts seems to stem from the semantics of data and who knows these semantics. Our responses also suggest that the definition of data error is highly analysis-dependent. In response to how she defines dirty data, one analyst responded,

> *Domain expertise, I guess. I wish there were a more rigorous way to do this but we look at the models and guess if the data are correct.*

This is in contrast to how infrastructure engineers want to handle data quality problems. Several infrastructure engineers noted that they saw dirty data as symptomatic of an error in the processing pipeline (i.e., a software bug or incorrect schema). Their goal was to rectify the bug or drop the corrupted data with a minimal impact on system SLOs:

> *There are software bugs in the application such as edge cases that are not handled or changes to the services by programmers that have unintended consequences. Fixing data errors in a high-availability setting is challenge as it may require shutting off services.*

Most of the infrastructure engineers surveyed used sampling or unit testing to detect obvious problems in the data processing pipeline, for example,

> *We have checks for file consistency, if the end of files look like the write was interrupted early, whether the size of a file seems significantly bigger or smaller than usual, etc. we check data with some standard queries to make sure the files have the expected range of values, ie for every country, and every minute of the day there was data.*

Such approaches require a clear definition of dirty data that is independent of the downstream analysis. Even worse, one consultant from a large database vendor noted that errors might be found well after some result is reported:

> *Most of these errors are subtle enough that the analysis will go through e.g., with standard null value semantics of SQL, but give an incorrect answer. Usually is only caught weeks later after someone notices something like...well the Wilmington branch cannot have 1M sales in a week.*

4. FUTURE DIRECTIONS

Our initial survey results highlighted several bottlenecks that add friction to the data cleaning and analysis process. The primary finding reiterates the observation that data cleaning is highly contextual – users intimately familiar with the downstream applications are needed to direct the data cleaning process. In other words, these domain experts are the most important "humans in the data cleaning loop".

In practice, however, we found that the data cleaning process is commonly split across multiple organizational units: the IT de-

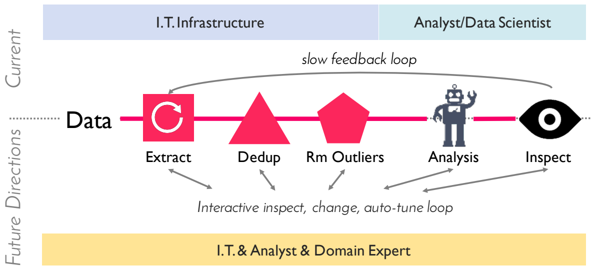

Figure 2: Current and proposed iterative feedback loop for data cleaning. The top shows the current slow feedback loop, where the data cleaning (magenta) and analysis (robot) steps are split between I.T. and data scientists, and feedback is obtained from the data scientists. The bottom shows the potential unified approach, where the cleaning can be inspected or modified by any user at any step throughout the pipeline. This can happen through manual changes or auto-tuned optimizations.

partment performs data cleaning, and sends the processed data to application developers and data scientists, who in turn perform additional data transformation and analysis (top of Figure 2). This separation inhibits the ability to experiment and test different cleaning procedures and tune their parameters—feedback about the data cleaning process is often delayed until the downstream application developer (e.g., visually) inspects the application results. We believe that these limitations need not exist, and envision a highly interactive data cleaning and analysis process, wherein infrastructure engineers, data analysts and domain experts can design, evaluate, and modify (with automated support) any stage of the data cleaning workflow (bottom of Figure 2). To this end, we present a series of technical challenges spanning HCI, statistics, and data management that must be overcome in order to support a truly interactive data cleaning system.

Developing High-level Language for Domain Experts: Data cleaning is an involved process that involves extraction, schema/ontology matching, value imputation, de-duplication, and other processes. In addition, each of these operations encapsulates dozens of specialized algorithms such as machine learning, clustering, or rule-based procedures. It is both difficult for domain experts to navigate through the zoo of options, and easy for those implementing data cleaning operations to become married to a specific algorithmic choice. In addition, these parties must interact, and it is important to facilitate the coordination between the two. There is a need for a high level language for domain experts to describe the data cleaning goals at a logical level (e.g., providing de-duplication examples, descriptions of outliers) that also enables physical implementation choices to be guided by either automated tools or the technical experts that are tasked with implementing data cleaning at scale [16,19].

Usability and Interactivity: The need to focus on usability and visual interaction has been reiterated across many domains: Wrangler [24] (commercialized as Trifacta [6]) enables domain experts to perform complex text extraction tasks at scale, and Polaris [36] (commercialized as Tableau [4]) helps business analysts perform data-cube analysis through a visual interface. These systems place emphasis on the end-to-end process by reducing bottlenecks that stem from human interaction and decision making. We must similarly lift a high level cleaning language into the interactive domain [21] in order to tighten the feedback loop between the user and cleaning process.

Application-oriented Cleaning: A recurring theme amongst our survey participants was the observation that data cleaning is driven by the needs of the downstream application. We found surprisingly little evidence of data cleaning as a discrete and isolated process. Systems such as SampleClean [29] and ActiveClean [31] have already shown the potential for leveraging application knowledge to reduce data cleaning costs by an order of magnitude or more compared to application agnostic approaches. However these systems have been tailored to specialized use cases (individual aggregation queries and convex models, respectively), and support for other more complex operations as well as multi-stage sequences of analyses is needed.

Human-Computer Symbiosis: Some participants described tweaking cleaning operations and running the downstream analysis in order to visually inspect the results. However, this form of manual configuration and parameter tuning does not make the best use of the domain expert's resources. There is potential to introduce automation, and we have already seen examples of this. For instance, active learning is used as part of crowd-sourced label acquisition [17,20] to optimize an operation such as de-duplication; Wrangler automatically generates string extraction rules so that the user only needs to pick from a set of options; and TuPAQ [35] performs automatic hyper-parameter tuning for machine learning models. There is similar opportunity to identify additional data cleaning operations that can be automatically tuned, as well as to propose modifications to the sequence of operations itself [19]. Ultimately, our goal should be to let experts do what they do best, while machines do the rest.

Testing and Debugging: In order to develop automated optimization and tuning procedures, there must be a metric to optimize. This can be quite challenging—one common measure of data cleaning effectiveness among survey participants was simply whether or not the downstream process compiled and ran! This clearly falls short of the standards needed for sophisticated automation, and methods for introducing metrics throughout the cleaning and analysis process are needed. For example, one might use performance on gold examples of known clean data to evaluate a cleaning operator. Such data could be acquired up front, or adaptively collected from the analyst herself or crowd workers throughout the cleaning process.

As the data analyst inspects different parts of the cleaning process, it will be increasingly important to provide tools to summarize and explain [9,39,40] the intermediate results in a way to goes beyond print statement outputs or row data entries.

In addition, to accelerate data cleaning research, there is a need for a cleaning benchmark analogous to industry standard transaction and analytical data processing benchmarks. Existing systems are often evaluated on synthetically generated errors that are may not reflect reality or on application-specific errors that are too specialized to serve as a standard benchmark across systems.

Combating Over-fitting: Despite their importance, data cleaning procedures are often under-reported when presenting the results of data analysis. This is problematic since the data cleaning operations have a potential to introduce analyst biases, i.e., favoring a certain outcomes, into the analysis process. This sentiment is corroborated by our survey results and the results of Kandel et al. [25]. In a sense, this problem is analogous to statistical over-fitting, where cleaning decisions based on strong assumptions over a small sample of data (or a specialized analysis) may not apply to future evolutions of the application. An important challenge is designing data cleaning tools that: (1) allow analysts to communicate assumptions (e.g., which records have been removed) when presenting results, (2) automatically determine when an assumption is risky (e.g., correlates with the tested hypothesis), and (3) manages a "paper trail" of data transformations.

5. CONCLUSION

We have presented initial results from a study of industry users of data analysis software that confirms the recent shift in data cleaning processes towards iterative workflows. Our survey results highlight the issues that frustrate current workflows, and motivate our proposal of the research challenges central to the design of unified systems that can alleviate these issues. In summary, though the data cleaning community is in the early days of highly interactive data cleaning and preparation, there are clear opportunities for systems that facilitate and automate rapid human-in-the-loop interactivity.

This research is supported in part by NSF CISE Expeditions Award CCF-1139158, DOE Award SN10040 DE-SC0012463, and DARPA XData Award FA8750-12-2-0331, and gifts from Amazon Web Services, Google, IBM, SAP, The Thomas and Stacey Siebel Foundation, Apple Inc., Arimo, Blue Goji, Bosch, Cisco, Cray, Cloudera, Ericsson, Facebook, Fujitsu, HP, Huawei, Intel, Microsoft, Pivotal, Samsung, Schlumberger, Splunk, State Farm and VMware.

6. REFERENCES

[1] Apache falcon. http://falcon.apache.org.

[2] Informatica. https://www.informatica.com.

[3] A reproducibility crisis? http://www.apa.org/monitor/2015/10/share-reproducibility.aspx.

[4] Tableau. https://www.tableau.com.

[5] Talend. https://www.talend.com/solutions/etl-analytics.

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

[7] Data cleaning survey 1. https://www.surveymonkey.com/r/data-cleaning, 2015.

[8] Data cleaning survey 2. https://www.surveymonkey.com/r/YT8RP3R, 2015.

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

[10] Z. Chen and M. Cafarella. Integrating spreadsheet data via accurate and low-effort extraction. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1126–1135. ACM, 2014.

[11] X. Chu, I. Ilyas, S. Krishnan, and J. Wang. Data cleaning: Overview and emerging challenges. In *SIGMOD Tutorial*, 2016.

[12] 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.

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

[14] T. Dasu and T. Johnson. *Exploratory Data Mining and Data Cleaning*. John Wiley & Sons, Inc., 2003.

[15] A. Ebaid, A. Elmagarmid, I. F. Ilyas, M. Ouzzani, J.-A. Quiane-Ruiz, N. Tang, and S. Yin. Nadeef: A generalized data cleaning system. *VLDB*, 2013.

[16] H. Galhardas, D. Florescu, D. Shasha, and E. Simon. Ajax: an extensible data cleaning tool. In *ACM Sigmod Record*, volume 29, page 590, 2000.

[17] 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.

[18] D. Haas, J. Ansel, L. Gu, and A. Marcus. Argonaut: Macrotask crowdsourcing for complex data processing. *PVLDB*, 8(12):1642–1653, 2015.

[19] D. Haas, S. Krishnan, J. Wang, M. J. Franklin, and E. Wu. Wisteria: Nurturing scalable data cleaning infrastructure. *Proc. VLDB Endow.*, 8(12):2004–2007, Aug. 2015.

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

[21] J. Heer, J. M. Hellerstein, and S. Kandel. Predictive interaction for data transformation. In *CIDR*, 2015.

[22] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Wrangler: Interactive visual specification of data transformation scripts. In *ACM Human Factors in Computing Systems (CHI)*, 2011.

[23] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Wrangler: interactive visual specification of data transformation scripts. In *CHI*, pages 3363–3372, 2011.

[24] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Wrangler: Interactive visual specification of data transformation scripts. In *SIGCHI*, 2011.

[25] S. Kandel, A. Paepcke, J. Hellerstein, and H. Jeffrey. Enterprise data analysis and visualization: An interview study. *VAST*, 2012.

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

[27] 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.

[28] 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.

[29] 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.*, 2015.

[30] S. Krishnan, J. Wang, K. Goldberg, M. Franklin, and T. Kraska. Privateclean: Data cleaning and differential privacy. In *SIGMOD Conference*, 2016.

[31] 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.

[32] S. Lohr. For big-data scientists, janitor work is key hurdle to insights. 2014.

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

[34] E. Rahm and H. H. Do. Data cleaning: Problems and current approaches. *IEEE Data Engineering Bulletin*, 23:2000, 2000.

[35] E. R. Sparks, A. Talwalkar, D. Haas, M. J. Franklin, M. I. Jordan, and T. Kraska. Automating model search for large scale machine learning. In *SOCC*, 2015.

[36] C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *TVCG*, 2002.

[37] 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.

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

[39] X. Wang, A. Meliou, and E. Wu. Qfix: Diagnosing errors through query histories. *arXiv preprint arXiv:1601.07539*, 2016.

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

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