Data Quality: Experiences and Lessons from Operationalizing Big Data

Archana Ganapathi, Yanpei Chen

Splunk, Inc. {aganapathi, ychen}@splunk.com

***Abstract*—Data quality issues pose a significant barrier to operationalizing big data. They pertain to the meaning of the data, the consistency of that meaning, the human interpretation of results, and the contexts in which the results are used. Data quality issues arise after organizations have moved past clear-cut technical solutions to early bottlenecks in using data. Left unaddressed, such issues can and have led to high profile missteps, and raise doubts about the data-driven world view altogether. In this paper, we share real-world case studies of tackling data quality challenges across industry verticals. We present initial ideas on how to systematically address data quality issues via technology. The success of operationalizing big data will depend on the quality of data involved, and whether such data causes uncertainty and disruptions, or delivers genuine knowledge and value.**

I. Introduction

Business-critical decisions and processes increasingly rely on big data platforms for analytics. Although good data-driven decisions bring immediate and tangible benefits to day-to-day consumers, poor decisions can harm the business operation, or even cause damage to the business brand and the general public at large. Here is a real-life story the authors experienced.

A technology company discovered that they can use data from an existing system to derive a new metric for customer engagement. Upon back-computing this metric for the past few years, they discovered a sharp and ongoing drop in this metric. The cause of the drop was mysterious to the engineering department, and the sales department was likewise puzzled, as the incoming revenue saw no disruption. Upon investigation, it was discovered that a recent version of the software modified the information recorded, invalidating the calculated metric for customers who have upgraded their software. Newer versions of the software directly observed the metric, instead of relying on indirect calculations.

This story highlights an increasingly common challenge of operationalizing big data: data-informed decisions are fundamentally constrained by the quality of the data. As these decisions become integral to businesses and government organizations, any unanswered questions regarding the meaning, the reliability, or the interpretation of the data can and have caused high profile disruptions and missteps.

Data quality represents an additional dimension to recent academia and industry emphasis on scale and performance [1], [2], [3], [4], [5], [6], new programming paradigms [7], [8], [9], [10], [11], and advanced algorithms [12], [13], [14], [15], [16], [17], [18]. Data quality becomes an issue only because big data platforms are now mature enough to process data at scale, using expressive paradigms, and executing rich algorithms. Addressing data quality involves both technology solutions and business process improvements.

Data quality issues encompass a super-set of data cleaning in traditional relational databases (Section II). Data cleaning efforts focus on defining rigorous rules around "what is clean data", and once such rules are defined, the focus shifts to optimized automated data cleaning tools subject to known theoretical bounds [19], [20], [21]. For big data use cases, this approach fails to capture the diversity, complexity, and rapid evolution of data cleanliness. Consequently, data scientists today typically spend a large amount of time "script-hacking" to clean their data [22], [23].

This paper seeks to highlight an important class of problems that no amount of "script-hacking" can address. We broadly term them "data quality", because they pertain to the meaning of the data, the consistency of that meaning, the human interpretation of results, and the contexts in which results are used. Data quality issues arise after organizations surpass initial data barriers such as not having data, relevant skills, or sufficiently stable data processing platforms (Section III).

We illustrate real-life data quality issues with a number of case studies that expose opportunities for technology improvements. The data quality issues fall into five high-level themes: ad-hoc instrumentation (Section IV), inconsistent data (Section V), unclear ownership of data and analysis (Section VI), unintentional visibility of work-in-progress (Section VII), and disconnect between data engineers and data consumers (Section VIII). The proposed technology improvements can free up precious time to focus on data quality aspects of business processes (Section IX).

The issues we present have appeared in various forms predating big data. Big data disproportionally amplifies these concerns because the vision of big data *is* to instrument everything, to join data across silos, to democratize the data, to focus on knowledge and insight despite diversity, inconsistency, and scale. We hope the case studies here instill a sense of urgency in addressing this problem space. Successfully operationalizing big data will depend on the quality of the data processing platform, the quality of data involved, and whether such data can deliver genuine knowledge and value.

II. Background - Data Cleaning

By definition, dirty data is poor in quality. Data cleaning has received considerable attention both historically and recently. Below, we provide an overview of key perspectives in data cleaning. The interested reader can consult various detailed surveys on the topic [24], [25], [26].

The classical formulation of data cleaning is as follows. Given a dataset D and integrity constraints C, D is inconsistent if any of the constraints are violated. Errors are the set of rows in the relations within D that if repaired, will allow D to be consistent. Repair is a series of operations $R_1 ... R_k$ such that $R_1 \circ R_2 ... \circ R_k(D)$ results in a consistent database. Under such a formulation, it is often possible to derive precise bounds on the number of errors and the cost of repair operations.

Though attractive, this formulation becomes impractical in modern big data use cases. First, data cleanliness is often impossible to define while data itself evolves. Second, error in the data is often inseparable from error in the analysis, and errors are identified only when results are counter intuitive. Recent surveys of big data practitioners indicate that real-life data cleaning is often ad-hoc and insufficiently rigorous [27].

Data cleaning is high on the consciousness of practitioners. Various surveys have identified data cleaning as important and time consuming [28], [22], [23], [29], [30], [31], with claims that data cleaning consumes up to 80% of the overall analysis time. Practitioners have coined the term *data janitor* to highlight the importance of the cleaning process [32], [33].

Preliminary work also indicates that data cleaning is potentially as valuable as advanced algorithms, if not more so. An undergraduate machine learning class tried to classify online movie reviews into comedy or horror. Naive implementations achieved mediocre accuracy; advanced techniques only marginally increased accuracy. Upon cleaning the data, the accuracy was nearly 100% [34]. Although potentially an anomaly due to exceptionally dirty data, this example illustrates that data cleaning is critical.

More recently, there is increased awareness that data cleanliness should be approached differently depending on the type of subsequent analysis [26]. For many linear aggregations such as sums or averages, there is diminishing return to data cleaning, and results can be estimated from small, cleaned samples. For other complex, high dimensional computations involved in various machine learning and statistical model building computations, data cleaning potentially interferes with the analysis in counter-intuitive ways. For example, estimates computed from cleaned samples may show reverse results from estimates from the aggregate. Such behavior results from statistical paradoxes similar to Simpson's paradox [35], where aggregates over mixtures of different populations of data can result in spurious relationships and subtle hidden biases.

Data cleaning is a challenging problem space in its own right. Our case studies represent challenges pertaining to data quality that are encountered after data is already cleaned.

III. Bottlenecks before Data Quality

Various data usage pre-conditions can potentially derail early efforts to operationalize big data. Addressing such bottlenecks is a prerequisite to tackling data quality issues.

A. Not believing data is necessary or useful:

There is increasing recognition of the immense value to be derived from data. However, there are still some enterprises where the data-informed world view has not yet gained traction. Skepticism about data is often expressed in some rephrase of "we are not a data company" or "our customers are not asking for us to be more data driven."

There are two approaches to address these views. (1) Point to success stories in peer industries or direct competitors. The drawback of this approach is that the most compelling success stories are considered trade secrets and withheld specifically to secure competitive advantages. (2) Take inventory of data already available in the enterprise, quantify any missing metrics of "business health" and "customer engagement." This approach helps surface latent opportunities hidden in the enterprise, but requires a deep commitment to understand business operations.

B. Not having data:

This sentiment can arise as a variation of the "we are not a data company" view, or a recognition that data is important, but data visibility is low. This issue can be addressed simply by taking inventory of the existing automated operations in the enterprise. All automation system generate data in the form of direct output, and indirect output such as activity and monitoring logs, providing two different data sources for each system. Given the amount of business process automation existing in every enterprise, there is a large and diverse treasure trove of data awaiting analysis.

C. Lacking data-literate skill sets:

This issue is real, and often magnified disproportionately. Big data success stories are often accompanied by cautionary tales of what can go wrong. Simultaneously, leading work on advanced algorithms and techniques require the support of armies of statisticians and computer scientists.

There are indeed data problems that warrant advanced techniques and careful interpretation. However, simple techniques can often extract a large fraction of the value in data: a count of "interesting events" grouped by "interesting factors" can usually identify critical issues. Specialist insight is needed to, for example, establish a causal relationship, or try to model the change in one variable as a function of the change in another.

Moreover, there is an increasing number of skilled workers trained to manipulate and interpret data. University degree programs and data vendors have both increased the availability of relevant training. Consequently, "data-literacy" is rapidly increasing across all industries.

D. Data processing platform not production-ready:

A data platform must transcend raw scale and performance when it is deployed for business critical use. Here are some common issues that fall directly on the vendors and researchers building data platforms.

Bugs and instability: The more important a use case, the more important it is for the platform to be bug free and stable. Consider a system built for real-time credit card fraud protection, and it crashes for a few minutes. Imagine the countless consumers exposed, the financial cost of a breach during the system downtime, and the risk of permanent brand damage. Instability is a common barrier to technologies moving from experimentation to production use.

Configuration complexity: Many platforms require tuning a large number of parameters to achieve peak performance. The tuning process is often manual and sensitive to the data and computation involved. Further, multiple platforms in the same pipeline need to be tuned together. For P platforms with Q configurable parameters each, there is potentially $O(Q^P)$ combination of parameters. Such setups become a nightmare if the platform lacks good defaults, or has a narrow range of high performing configurations.

Difficult to grow and scale: Successful use cases likely lead to unforeseen demands on the underlying platform, such as additional data sources, human analysts, and computation being performed. The deployment needs to grow quickly and without disruption, a capability where some platforms fall short. For example, if the platform can handle 1PB with 100 users on 1000 nodes, how can it transition to 10PB with 1000 more users on 1000 more nodes? Would there be disruptive downtime, large volume data migration and rebalancing, or temporarily poor caching behavior? Does the cluster configuration need to be completely re-tuned? These questions pose a different perspective on scalability than just "handle X computation on Y data given Z resources."

E. Data processing platform without commercial support:

Proof-of-concept projects have the freedom to employ any technology. Many platforms are "home-built". When used in real-life, their total cost of ownership includes resources for infrastructure, technology, and diverse skill sets needed to develop, maintain, interface, and support the platform. Often, platforms with commercial support end up achieving lower total cost of ownership.

There are also legitimate technology reasons for preferring platforms with commercial support. We established earlier that instability, complexity, and difficulty in scaling all represent barriers to adopting a particular platform for use cases that directly impact business and consumer well-being. Commercial support entails some obligation that such issues will be promptly addressed. Without such obligation, it falls on each enterprise to address their own issues. This model is inefficient. If N enterprises encounter the same problem, there will be N duplicate fixes. Additionally, each enterprise sees only their own use cases. Therefore, a vendor offering commercial support can observe entire classes of problems across different enterprises, and provide more general and optimized solutions.

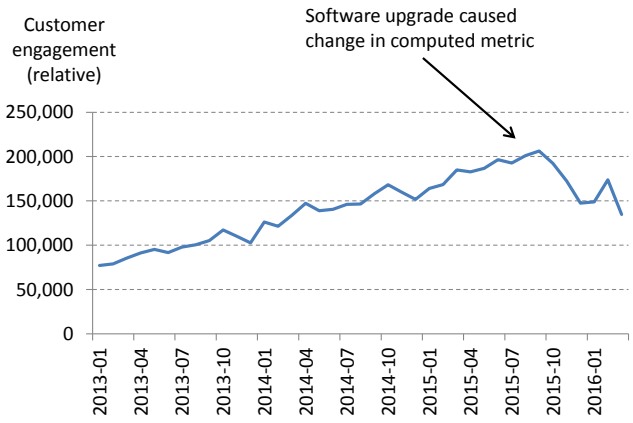

Fig. 1. Ad-hoc instrumentation with an indirect metric. The metric became invalid upon a software upgrade, causing confusion regarding whether the data exposed a critical issue with customer engagement.

The case studies in this paper have already overcome these bottlenecks. Data quality truly is the next challenge to tackle.

IV. Ad-Hoc Instrumentation

A common problem we have encountered is that the systems instrumenting and generating the data are unaware of the downstream consumers of the data. Consequently, the data is often an ad-hoc proxy to behaviors and metrics that are being observed. Further, if data format is changed, data consumers often find out only when they see unexpected anomalies in their analysis results. Either way, a long process ensues of validating whether the observations are valid, and that the data can actually serve its intended purpose.

The example in the introduction involving a perceived drop in customer activity was caused by ad-hoc instrumentation (Figure 1). The drop in product usage was a red herring as the logging format changed between software versions. The actual data instrumentation came from a tool originally intended to check for software upgrades. Apparently the information required to check for software upgrades could be repurposed to compute a customer engagement metric. The logging format change did not affect the primary purpose of the tool, but it invalidated the metric calculation.

In a second example, reports that show activity by geographical region suddenly showed empty charts. The field that was used to split by geographical region remained in the data but had empty values for the latest quarter. Apparently the use of that field was deprecated and the information was encapsulated in a new field with an entirely different set of values that expressed an indirect categorization by geography.

In this case, the format change came from a change in business process rather than a change in the data instrumentation tool. The effect is the same - a report that is closely monitored to gauge the health of the business is disrupted.

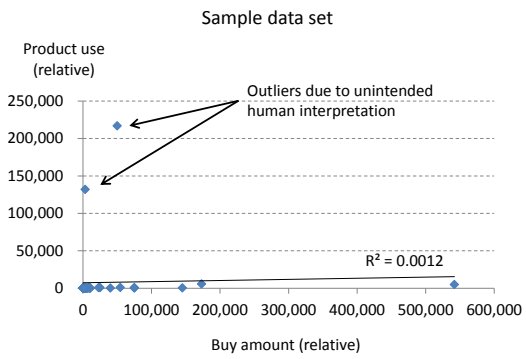

Fig. 2. Ad-hoc instrumentation that leads to unintended interpretation. The mis-interpretation leads to outliers that overwhelm statistical contribution from other data points.

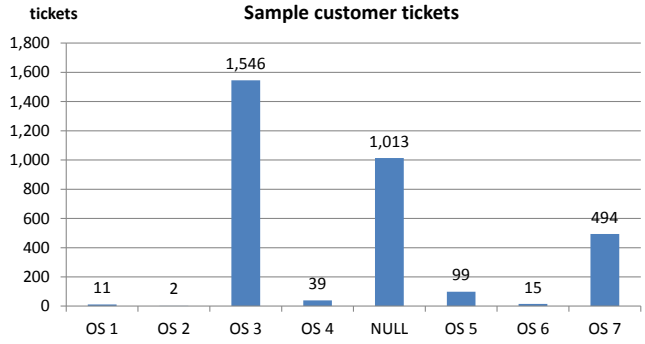

Fig. 3. Inconsistent data where NULL-filling is not possible. NULL could mean an operating system (OS) is not relevant, unknown, or omitted. Each situation required a different treatment in the analysis.

In another example, a model seeking to translate between customer product use and buying patterns found that some dimensions of product use did not have an anticipated correlation with buying patterns (Figure 2). In the absence of automated instrumentation, the data input into the model was gathered from a survey. A particular survey question was phrased in a way that could lead to two possible human interpretations. The survey designers did not foresee this issue, and indeed almost all respondents followed the intended interpretation. However, a few respondents supplied numerical responses that were magnified by a constant factor, which was large enough to overwhelm the statistical contribution from all of the other survey participants, resulting in garbled data in that dimension.

Solutions: The best solution is to build automated, direct instrumentation into systems. This avoids any non-uniform measurement due to either human processes being involved or different measurement contexts introducing bias in any computed/indirect metrics.

Developing direct instrumentation is not always feasible, because many big data use cases today involve cyber-physical systems in healthcare, manufacturing, retail, finance, and other industries where the pace of system replacement has inherent physical constraints. In other words, insights have to be derived from whatever data is available, while awaiting deployment of physical systems with improved instrumentation.

In addition, it is crucial to follow a disciplined approach to data generation where changes are mindful of the data consumers. Changes to existing formats should accommodate older fields and values, with the older format deprecated gradually. The data processing platform should ideally maintain a lineage of consumers for particular data sources, and ensure that as systems associated with the data sources are upgraded or changed, the downstream analysis can be appropriately "tested" for compatibility.

V. Inconsistent Data

When humans are involved in the data creation process, one must account for human behavior and consequent biases introduced in the data. The interface presented for human data input can influence data accuracy and usefulness. Furthermore, default values and parameters, such as whether a field is required or optional, can significantly impact data quality.

As an example, a software development organization tried to quantify development discipline and efficiency by analyzing change logs in the JIRA issue tracking system over time. The analysis proved overwhelming, because even limiting the analysis to a small, predetermined field list yielded an unhelpful and messy combination of field values (Table I). For this use case, there are 6 different priorities, 27 different statuses, 32 different issue types. Having so many options for each of these fields proved unwieldy, and became a source of inconsistency for the software development process. For this particular project, the process issues identified turned out to be a meta-finding that was helpful for the organization.

As another example, a technology company tried to analyze how their support tickets varied among different operating systems. The data source involved had an optional text box indicating the operating system. This caused two issues. (1) Each operating system has a variety of ways it is expressed, requiring much human effort in cleaning the data. (2) The optional nature of the text box led to a large number of NULL values (Figure 3). A NULL value could indicate either that an operating system is not relevant to the ticket, or that it is unknown or omitted. Each situation required a different treatment in the analysis. Some alternate data sources were used to correlate and NULL-fill where possible. However, there was still considerable uncertainty, and the results had to be accompanied with a detailed analysis on error bounds.

Solutions: Many business critical use cases involve a combination of machine and human generated data. Thoughtful user interface design can help reduce the data inconsistency associated with human generated data.

Interfaces must allow for efficient data entry and also manage the degrees of freedom when manual data creation is involved. For example, drop downs with pre-defined options are often better than free-text boxes, both in terms of data

Priority values	Status values				
Major	New	Production Ready	Pending Confirmation	To Do	In Review
Critical	Untriaged	Production QA	Staging Ready	UAT Ready	Under Investigation
Minor	Open	Researching	Pending Fix	Change Control	UAT in Progress
Blocker	In Progress	Staging QA	Rationalized	Code Review Pending	
Trivial	Done	QA In Progress	Ready for Certification	Waiting For Third Party	
Cosmetic	Reopened	QA Ready	Submitted	Monitoring	

TABLE I
EXAMPLE OF INCONSISTENT DATA. PRIORITIES AND STATUS VALUES IN A BUG TRACKING SYSTEM, WITH UNCLEAR MEANINGS FOR EACH.

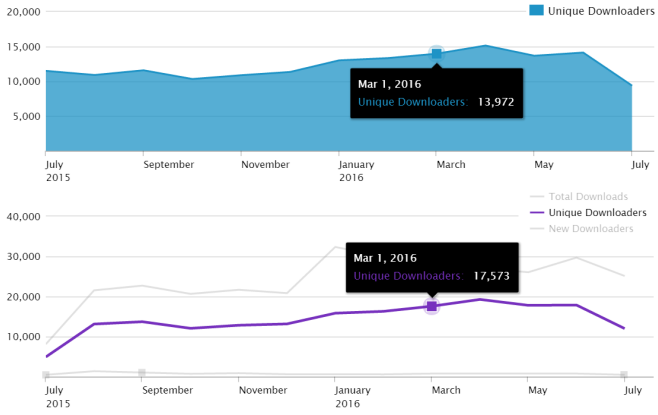

Fig. 4. Consequence of unclear ownership - two measures of "unique downloaders" from different departments. The graphs have similar shapes but different numerical values.

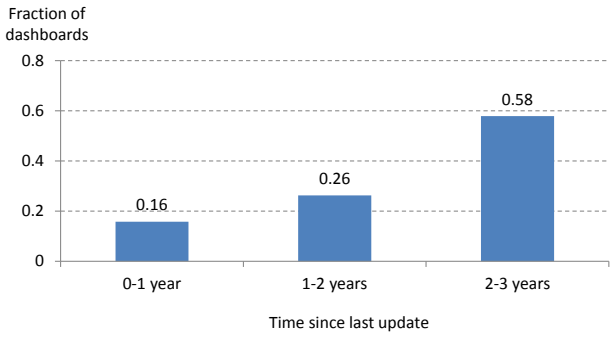

Fig. 5. Potential for orphaned dashboards on an production system. Many dashboards have not been updated for >2 years, potentially creating unnecessary load and impacting user experience.

consistency and the efficiency of data entry. Additionally, careful choice of default values and required versus optional fields can reduce the need for manual data clean up, and lead to more streamlined business process. The order of options affect data consistency as well - humans frequently select the first or the default value, prioritizing time spent rather than correctness, especially if the data creators are not the data consumers.

Furthermore, data platforms that aim to store data at historical time scales should provide easy mechanisms to update or re-clean the data. As business processes evolve, deprecation or alteration of historical fields and values often can be expressed as first order logic. For organizations disciplined enough to track how their internal processes evolve, the data platform should make it easy for them to reflect these changes as programmatic updates to the associated data.

VI. Unclear Ownership of Data and Analysis

Analysis results and reports are often propagated across multiple forwarding layers and via various channels, e.g. slides, spreadsheets, emails. There are few methods to associate the analysis with its original creator as well as the "owner" of the data involved. Additionally, modifications to the analysis and subsequent versions of the same report are not always performed by the same person. It is easy to lose track of what is the most recent and accurate analysis, as well as the precise quantitative meaning of the data involved.

For example, an organization tried to get a business wide view on some "health of the business" metrics. Previously, different departments created separate dashboards that tried to capture the same set of indicators. The underlying computation for each dashboard used slightly different conditions for filtering and grouping the data. There were no annotations regarding when and why certain filters were modified and which set of filters is current and most appropriate for the analysis being performed. Some of the dashboards had unambiguous caretakers who could explain the context behind them. Other dashboards were created by former employees, or did not have a clear owner. The problem was compounded by the fact that there were no audit reports to determine frequent viewers of each dashboard. To this day, the clarification of the precise meaning of these metrics remain a work in progress at this organization (Figure 4).

As another example, a large consumer goods company observed significant performance issues on their company-wide data processing platform. The system's administrators indicated there had been no configuration changes or upgrades and that the number of users had also not changed sufficiently to cause a performance impact. Upon time-consuming and costly investigation it was found that several orphaned computations were consuming significant system resources. Additionally, these computations were unoptimized and were scheduled to run very frequently, e.g., read an archival dataset in full every five minutes and export via a low-bandwidth API to an external system, with data filtering and summarization done after the export. While the company derived significant

business value from opening their platform to a large number of users, in this case a small number of orphaned computations contributed to an unnecessary performance issue for their entire organization.

Many organization-wide platforms accumulate knowledge artifacts over time, with only a small fraction being modified recently (Figure 5). Orphaned artifacts are a potentially hidden problem at many organizations, both in terms of the background computation load they create, and the meaning of the computation being lost over time.

Solutions: These problems have technology-centric solutions.

Knowledge artifacts such as automated reports and scheduled queries should have default "time-to-live" behavior that can be extended based on frequency of usage. This is a way to ensure platform hygiene and enforce usage discipline - if certain artifacts have been neglected for a configurable time, then the artifact owners should be automatically asked whether the artifact remains relevant. If no human input is given, there should be default behavior to transition the artifacts to an archival state that consumes less system resources, or the artifacts should be removed altogether.

Additionally, data processing platforms have the ability to perform versioning and have a concept of lineage and ownership. Data lineage features are already required for many systems to satisfy legal auditing and compliance requirements for various industry verticals such as healthcare, finance, and government. Such lineage features should be extended to track creators and owners of datasets and analysis reports, the time and person associated with the last modification, the list of users consuming or viewing a report, and have an owner "inheritance" capability if certain users leave the organization. Such capabilities, combined with traditional lineage features tracking the sequence of data sources and computations, allows organizations to build trust in the data and the analysis.

Data set and knowledge artifacts lineage can also help track changes. Specifically, any rename, relocate, redefine operations can be propagated to downstream data sets and knowledge artifacts. This saves considerable time in locating human owners to facilitate updating pointers whenever such operations take place.

Lineage pointers are also essential to enable good default "time-to-live" behavior. The decision to remove "expired" data sets and knowledge artifacts must consider pointers to dependent data sets and knowledge artifacts. Policies need to evaluate what can be removed conditioned on the liveliness of the downstream artifacts.

Further, the data processing platform should have self monitoring capabilities with thoughtful bounds on per-user or per-computation resource consumption limits. Such capabilities allow the platform to be accessible to a broad class of users, while reducing the risk of a single user significantly affecting the overall platform behavior. In cases of legitimate computations requiring a large amount of resources, administrators can identify the users involved, and work with them to optimize the computation or resource consumption limits.

The data processing platform should also accommodate multiple levels of credentials associated with increasing capabilities for content creation. This allows organizations to require new users to complete platform usage training and gradually unlock more capabilities of the platform.

VII. Unintentional Visibility for Work-in-Progress

With increasing attention being paid to big data, and increasing deployment of enterprise-wide data platforms, partial or exploratory work could inadvertently become highly visible. Over-reaction from across the organization requires much effort to explain and reset expectations.

For example, an engineering team performed for-fun analysis projecting customer activity based on various metrics. Although logically such analysis should not succeed, by statistical accident, the models successfully predicted some activity, fulfilling the original "for-fun" purpose of the analysis. However, the dashboard with the model was left visible on the enterprise-wide platform, and noticed by the sales team, who took customer-facing action based on the result, necessitating considerable effort to explain the analysis and undo the actions.

In another case, an analyst created a report looking at historical customer trends. The report was a work in progress. The analyst was focusing on the late-stage computations and visualization layout, while delaying work on the early-stage data filtering and summarization. The work-in-progress was noticed, and forwarded to a large number of recipients. This created a large load on the platform, as well as a poor experience for the recipients. Fortunately, the issue was quickly resolved, as the performance optimization was obvious, and merely planned for later at the analyst's discretion.

Solutions: These issues reveal two technology problems.

The first issue is around content visibility. Data processing platforms should have mechanisms that allow clear annotations of "work in progress" or "exploratory work". Knowledge content creators should be given the option to set initial visibility. Depending on the organizational culture, default visibility could be set low to prevent unintended action, or set high to encourage collaboration.

A second issue is around creating a conceptual "sandbox" for developmental work. Analysts want the freedom to explore various data sources and computations without having to worry that their work will accidentally impact the organization-wide platform. This is a common process, where considerations of "what to compute" precede optimizing for "how to compute it efficiently." Fortunately, most platforms already have some sort of resource management capability to implement various resource sharing/limiting policies. The effectiveness of those mechanisms remains an open research question.

VIII. Disconnect between Data Engineers and Data Consumers

The consumer of data and analyses is typically a domain expert who understands what computations must be performed

and how to leverage results for a business decision. In many organizations, typically a different person implements the analysis and manipulates the data. As a result, we commonly find miscommunications that lead to under-thinking the problem and performing incorrect analysis, or overthinking the problem and creating unnecessarily complex solutions.

A large consumer company hired consultants to build dashboards for their marketing department based on data aggregated across various sources. Upon completion, marketing users found the dashboards slow and severely hindered their business process. The problem came from the dashboards loading all historical data spanning multiple years. The actual business decision only required data from the most recent month. This is an example of over-engineering based on miscommunicated or misunderstood requirements.

As another example, a specific department wanted to use a particular visualization tool to view their data. They proposed standing up a new data repository to regularly export all data from their existing platform. It was unclear why cloning the entire data set was necessary or why they could not connect the data visualization tool to the existing data platform. None of the stakeholders had an end-to-end view on the goals and the data architecture. A complex and inefficient architecture resulted, involving data passing through multiple systems.

Solutions: On the surface, these seem to be purely non-technology problems addressable by proper communication and business processes. This view obscures a fundamental technology problem - the disconnect often arises because the underlying platforms remain too difficult to use. Specialized skills are needed to operate the platforms and manipulate the data, leading to data consumers passing-on their requirements to dedicated data teams.

Many big data platforms today leave much to be desired in terms of usability. Various tools still expose command line interfaces as the primary method of interacting with the data. Often, specialized languages and programming paradigms need to be mastered. The analysis results are returned in table or even raw text form, and need to be exported to additional tools for visualization. These limits severely impact the breadth of use of the data platforms, not to mention performance inefficiencies associated with frequent export/import.

We argue that there is great value in democratizing direct use of big data platforms beyond the small number of "expert users". All participants in the analysis pipeline should be able to self-service their needs. A useful analogy comes from web search, in some ways the earliest big data use case. Imagine a "search engine" that requires users to write MapReduce jobs or SQL queries to express their searches. Such a search engine would have limited users, and search intent can often get "lost in translation." Contrast that with search engines today, where *everyone* can directly type search terms into a text box. That should be the usability goal for all big data platforms.

IX. Improve Data Quality via Human Processes

Thus far, the paper focused on technology solutions to alleviate data quality issues. As technologists, we would like technology to be the entirety of the solution. However, human processes inevitably form a part of the solution.

There are various straightforward actions that all organizations can take to improve communication and exploration around data. Below we discuss several less obvious topics, to provide a starting point for organizations to simultaneously improve technology and business process around their data.

A. Elevate data as an organization-wide asset:

Data is increasingly taken out of departmental silos, and made accessible for broad use. This "data openness" unlocks the ability to derive organization-wide metrics. Most organizations are still in the early stages of interconnecting previously siloed data: ad-hoc instrumentation and inconsistent data are issues only because we have extended existing data beyond its original intended use.

This organization-wide view is essential. Some organization-wide metrics inherently draw upon data associated with different departments. Additionally, for business critical decisions, the analysis result needs to be validated against multiple data sources. While each data source could suffer data quality issues and systematic bias, the combined view can be convincing: If multiple independent data sources corroborate the same story, it is unlikely they all suffer from the same bias. In our experience, organizations that successfully elevate data as an organization-wide asset derive disproportionally greater value from their data.

B. Address business process issues that affect data quality:

Some data quality issues reflect issues in related business processes. For example, lack of direct measurement of key metrics could indicate that decisions are based on insufficient reasoning and are susceptible to hidden bias. Inconsistent data could suggest that existing processes are suboptimal or not diligently followed. Disconnect between data engineers and data consumers could reflect misaligned priorities between different functions.

Open-minded organizations recognize that such issues need to be surfaced without risk or blame. In our experience, data quality can serve as a rational, objective channel to spur improvements in related business processes.

C. Being data-informed without overly relying on data:

Data inherently captures a snapshot of the current state of an entity. Business and technology vision inherently project onto the future. An over-reliance on data can shackle decision making, or worse, become an excuse for inertia.

This concern arises only after successful examples of data-driven business critical decisions. Subsequent projects are rightly held to a high standard. However, data hardly ever tells the "complete story". Data scientists and analysts need to be rigorous with all available data. Decision makers need to understand what level of uncertainty is acceptable for the decision at hand, whether numerical rigor is required, or directional guidance suffices. Where uncertainties exist, the data should be cross-checked against human experience and

proven heuristics. A couple of the use cases outlined earlier have led to decisions that run counter to the data, precisely because the organizations involved have competing concerns about broader business impact and direction. Data analysis helped clarify the priority.

D. A role for centralized data teams?

Organizations tend to start their adoption of big data with independent efforts, addressing urgent business needs of individual departments. This pattern often results in department specific data processing platforms, with data specialist expertise distributed throughout the organization and tied to domain expertise in each functional area. Over time, the data processing platforms tend to consolidate and analysis needs tend to expand across domains, often with the emergence of centralized data science teams.

This is a positive adoption path. However, some organization-wide issues still persist without clear ownership. Addressing the data quality issues identified here require technical interactions with data platform administrators and existing data scientists, both in centralized teams and in each functional area. Cross-organizational expectations need to be clarified, and a set of "organizational APIs and SLAs" need to be understood. Where appropriate, disconnects in process and communication should be alleviated. These efforts require a charter and skill set broader than that of typical centralized data science teams and platform administration teams. Who and how to tackle these challenges remain questions for organizations looking to take their data operations beyond early pilot and into mainstream.

X. Summary

Data quality issues pose a significant barrier to operationalizing big data. They encompass issues related to data cleaning, a topic that has received much attention. The appearance of data quality issues signify that organizations have moved past clear-cut technical solutions to early bottlenecks in using data. We presented real-life data quality challenges from organizations across industry verticals, and seeded initial ideas on how to address them via technology. The issues here present an opportunity for researchers to tackle an important problem space. As technologists, our vision is that data platforms should be designed to prevent, or at least minimize, data quality issues to the fullest extent possible. Such data platforms would free up precious human effort to focus on non-technology aspects of the solution.

Acknowledgment

The authors would like to thank the following people for providing feedback on an early draft of the paper: Steve Zhang, Pete Sicilia, Gaurav Agarwal, Matt Green.

References

[1] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, January 2008.

[2] Sanjay Ghemawat et al., "The Google File System," in *SOSP 2003*.

[3] G. DeCandia et al., "Dynamo: Amazon's Highly Available Key-value Store," in *SOSP 2007*.

[4] M. Kornacker et al., "Impala: A Modern, Open-Source SQL Engine for Hadoop," in *CIDR 2015*.

[5] M. Zaharia et al., "Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing," in *NSDI 2012*.

[6] H. Li et al., "Tachyon: Reliable, Memory Speed Storage for Cluster Computing Frameworks," in *SOCC 2014*.

[7] M. Isard et al., "Dryad: Distributed data-parallel programs from sequential building blocks," in *EuroSys 2007*.

[8] G. Malewicz et al., "Pregel: A system for large-scale graph processing," in *SIGMOD 2010*.

[9] Y. Low et al., "Distributed graphlab: A framework for machine learning and data mining in the cloud," in *VLDB 2012*.

[10] M. Zaharia et al., "Discretized streams: Fault-tolerant streaming computation at scale," in *SOSP 2013*.

[11] "Apache Kafka," http://kafka.apache.org/.

[12] T. Kraska et al., "MLbase: A Distributed Machine Learning System," in *CIDR 2013*.

[13] S. Agarwal et al., "Blinkdb: Queries with bounded errors and bounded response times on very large data," in *EuroSys 2013*.

[14] J. Dean et al., "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," Google Research Whitepaper, http://research.google.com/pubs/archive/45166.pdf, 2015.

[15] Raluca Popa et al., "Cryptdb: Protecting confidentiality with encrypted query processing," in *SOSP 2011*.

[16] C. Ding et al., "Convex and semi-nonnegative matrix factorizations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, 2010.

[17] Yann LeCun et al., "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[18] A. Kleiner, A. Talwalkar, P. Sarkar, and M. I. Jordan, "A scalable bootstrap for massive data," *Journal of the Royal Statistical Society, Series B*, vol. 76, 2014.

[19] I. F. Ilyas and X. Chu, "Trends in cleaning relational data: Consistency and deduplication," *Foundations and Trends in Databases*, vol. 5, no. 4, pp. 281–393, 2015.

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

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

[22] Sandy Ryza et al., *Advanced Analytics with Spark*. Sebastopol, California: O'Reilly Media, Inc., 2015.

[23] J. Grus, *Data Science from Scratch*. Sebastopol, California: O'Reilly Media, Inc., 2015.

[24] E. Rahm and H. H. Do, "Data cleaning: Problems and current approaches," *IEEE Data Engineering Bulletin*, vol. 23, no. 4, 2000.

[25] T. Johnson and T. Dasu, "Data quality and data cleaning: An overview," in *SIGMOD 2003*.

[26] Xu Chu et al., "Data cleaning: Overview and emerging challenges," in *SIGMOD 2016 Tutorials*.

[27] Sanjay Krishnan et al., "Towards reliable interactive data cleaning: A user survey and recommendations," in *HILDA 2016*.

[28] S. Kandel, A. Paepcke, J. M. Hellerstein, and J. Heer, "Enterprise data analysis and visualization: An interview study," *IEEE Transactions on Visualization and Computer Graphics*, December 2012.

[29] G. Press, "Cleaning big data: Most time-consuming, least enjoyable data science task, survey says," Forbes, March 23, 2016.

[30] S. Lohr, "For big-data scientists, janitor work is key hurdle to insights," New York Times, August 17, 2014.

[31] T. H. Davenport and D. Patil, "Data scientist: The sexiest job of the 21st century," *Harvard Business Review*, 10 2012.

[32] A. Popescu, "Data Scientists are Like Forrest Gump, Scrubbing Data with Toothbrushes," Sillicon Angle, February 28, 2013.

[33] Wikipedia, "Data Janitor," https://en.wikipedia.org/wiki/Data_janitor.

[34] S. Krishnan, "Relational data cleaning: A statistical perspective," Invited academia exchange seminar, Splunk, 2016.

[35] E. H. Simpson, "The interpretation of interaction in contingency tables." *Journal of the Royal Statistical Society*, 1951.