The Data Analytics Group at the Qatar Computing Research Institute

George Beskales Gautam Das Ahmed K. Elmagarmid
Ihab F. Ilyas Felix Naumann Mourad Ouzzani
Paolo Papotti Jorge Quiane-Ruiz Nan Tang
Qatar Computing Research Institute (QCRI), Qatar Foundation, Doha, Qatar
Web Site: http://www.da.qcri.qa/
{gbeskales,gdas,aelmagarmid,ikaldas,fnaumann,mouzzani,ppapotti,jquianeruiz,ntang}@qf.org.qa

1. DATA ANALYTICS AT QCRI

The Qatar Computing Research Institute (QCRI), a member of Qatar Foundation for Education, Science and Community Development, started its activities in early 2011. QCRI is focusing on tackling large-scale computing challenges that address national priorities for growth and development and that have global impact in computing research. QCRI has currently five research groups working on different aspects of computing, these are: Arabic Language Technologies, Social Computing, Scientific Computing, Cloud Computing, and Data Analytics.

The data analytics group at QCRI, DA@QCRI for short, has embarked in an ambitious endeavour to become a premiere world-class research group by tackling diverse research topics related to data quality, data integration, information extraction, scientific data management, and data mining. In the short time since its birth, DA@QCRI has grown to now have eight permanent scientists, two software engineers and around ten interns and postdocs at any given time. The group contributions are starting to appear in top venues.

2. RESEARCH FOCUS

DA@QCRI has built expertise focusing on three core data management challenges: extracting data from its natural digital habitat, integrating a large and evolving number of sources, and robust cleaning to assure data quality and validation.

We are focusing on the interaction among these three core data management challenges, which we call the "Data Trio". We are investigating multiple new directions, including: handling unstructured data; interleaving extraction, integration, and cleaning tasks in a more dynamic and interactive process that responds to evolving datasets and real-time decision-making constraints; and leveraging the power of human cycles to solve hard problems such as data cleaning and information integration.

In this report, we describe sample research projects related to the data trio as well some initial results. In the first couple of years, we have been mainly focusing on data quality management.

3. DATA QUALITY

It is not surprising that the quality of data is becoming one of the differentiating factors among businesses and the first line of defence in producing value from raw input data. As data is born digitally and is directly fed into stacks of information extraction, data integration, and transformation tasks, insuring the quality of the data with respect to business and integrity constraints has become more important than ever. Due to these complex processing and transformation layers, data errors proliferate rapidly and sometimes in an uncontrolled manner, thus compromising the value and high-order information or reports derived from data.

Capitalizing on our combined expertise in data quality [3, 4, 10, 11, 13, 15–18, 21, 22, 27, 29], we have launched several projects to overcome different challenges encountered in this area.

3.1 NADEEF - A Commodity Data Cleaning System

While data quality problems can have crippling effects, there is no *end-to-end off-the-shelf* solution to (semi-)automate error detection and correction *w.r.t.* a set of *heterogeneous* and *ad-hoc* quality rules. In particular, there is no commodity platform similar to general purpose DBMSs that can be easily customized and deployed to solve application-specific data quality problems. To address this critical requirement, we are building NADEEF, a prototype for an extensible and easy-to-deploy data cleaning system that leverages the separability of two main tasks: (1) specifying integrity constraints

and how to repair their violations in isolation; and (2) developing a core platform that holistically applies these routines in a consistent, and a user-guided way. More specifically, we are tackling the following challenges for emerging applications:

Heterogeneity. Business and integrity constraint-based data quality rules are expressed in a large variety of formats and languages(*e.g.,* [2,5,7,12,14,26]) from rigorous expressions (as in the case of functional dependencies), to plain natural language rules enforced by code embedded in the application logic itself (as in most practical scenarios). This diversity hinders the creation of one uniform system to accept heterogenous data quality rules and enforces them on the data within the same framework. For example, data collected by organizations, such as Qatar Statistics Authority, is checked against several constraint types, such as range constraints, not-null constraints, inclusion dependencies, as well as other sophisticated constraints (*e.g.,* the age difference between a person and his/her father should be greater than 15). Additionally, data may come from different sources and with different formats. Thus, we need to revisit how *heterogenous* quality rules can be specified on and applied to *heterogeneous* data.

Interdependency. Even when we consider a single type of integrity constraints, such as functional dependencies, computing a consistent database while making a minimum number of changes is an NP-hard problem [5]. Due to the complexity and interdependency of various data quality rules, solutions have usually been proposed for a single type of rule. Considering multiple types of rules at the same time is considered an almost impossible task. While some attempts have recently tried to consider multiple *homogenous* sets of rules that can be expressed in a unified language [6, 16], the problem is still far from being solved.

Deployment. A large number of algorithms and techniques have been proposed (*e.g.,* [5, 16, 20, 29]), each requiring its own setting and staging of the data to be cleaned. Hence, it is almost impossible to download one of them from a software archive and run it on the data without a tedious customization task, which in some cases is harder than the cleaning process itself.

Data custodians. Data is not born an orphan. Real customers have little trust in the machines to mess with the data without human consultation. For example, at Qatar Statistics Authority, all data changes must be justified and reviewed by domain experts before being committed. Due to the limited processing power of humans, scalability of (semi-)manual techniques is very limited and does not speak to the requirements of today's large-scale applications. Several attempts have tackled the problem of including humans in the loop (*e.g.,* [13,17,23,29]). Unfortunately, these attempts still suffer from the aforementioned problems, but provide good insights on including humans in effective and scalable ways.

Metadata management. Cleaning data requires collecting and maintaining a massive amount of metadata, such as data violations, lineage of data changes, and possible data repairs. In addition, users need to understand better the current health of the data and the data cleaning process through summarization or samples of data errors before they can effectively guide any data cleaning process. Providing a scalable data cleaning solution requires efficient methods to generate, maintain, and access such metadata. For example, we need specialized indices that facilitate fast retrieval of similar tuples or limit pairwise comparisons to specific data partitions.

Incremental cleaning. Data is evolving all the time. The simplistic view of stopping all transactions, and then cleaning and massaging the data is limited to historical and static datasets. Unfortunately, theses settings are becoming increasingly rare in practice. Data evolution suggests a highly incremental cleaning approach. The system has to respond to new evidences as they become available and dynamically adjusts its belief on the quality and repairing mechanisms of the data.

To achieve the separability between quality rule specification that uniformly defines *what* is wrong and (possibly) *why*; and the core platform that holistically applies these routines to handle *how* to identify and clean data errors, we introduce an *interface* class *Rule* for defining the semantics of data errors and possible ways to fix them. This class defines three functions: $\mathsf{vio}(s)$ takes a single tuple s as input, and returns a set of problematic cells. $\mathsf{vio}(s_1, s_2)$ takes two tuples s_1, s_2 as input, and returns a set of problematic cells. $\mathsf{fix}(set\langle cell\rangle)$ takes a nonempty set of problematic cells as input, and returns a set of suggested expressions to fix these data errors.

The overall functioning of NADEEF is as follows. NADEEF first collects data and rules defined by the users. The rule compiler module then compiles these *heterogeneous* rules into homogeneous constructs. Next, the violation detection module finds what data is erroneous and why they are as such, based on user provided rules. After identifying errors, the data repairing module handles

the *interdependency* of these rules by treating them holistically. NADEEF also manages metadata related to its different modules. These metadata can be used to allow domain experts and users to actively interact with the system. As we progress in this project, we will post new developments at http://da.qcri.org/NADEEF.

3.2 Holistic Data Cleaning

The heterogeneity and the interdependency challenges mentioned above motivated the study of novel repair algorithms aiming at automatically producing repairs of high quality and for a large class of constraints. Our holistic data cleaning algorithm [6] tackles the two problems by exploiting a more general language for constraint definition and by introducing a holistic approach to their repair.

As a first step toward generality, we define quality rules by means of denial constraints (DCs) with ad-hoc predicates. DCs subsume existing formalisms and can express rules involving numerical values, with predicates such as "greater than" and "less than".

To handle interdependency, violations induced by the DCs are compiled into a conflict hypergraph in order to capture the interaction among constraints as overlaps of the violations on the data. The proposed mechanism generalizes previous definitions of hypergraphs for FD repairing. It is also the first proposal to treat quality rules with different semantics and numerical operators in a unified artifact. Such holistic view of the conflicts is the starting point for a novel definition of repair context, which allows automatic repairs with high quality and scalable execution time *w.r.t.* the size of the data. The repair algorithm is independent of the actual cost model. Experiments on heuristics aiming at cardinality and distance minimality show that our algorithm outperforms previous solutions in terms of the quality of the repair.

3.3 Guided Data Repair

GDR, a Guided Data Repair framework [28, 29] incorporates user feedback in the cleaning process with the goal of enhancing and accelerating existing automatic repair techniques while minimizing user involvement. GDR consults the user on the updates that are most likely to be beneficial in improving data quality. GDR also uses machine learning methods to identify and to apply the correct updates directly to the database without the actual involvement of the user on these specific updates. To rank potential updates for consultation by the user, GDR first groups these repairs and quantifies the utility of each group using the decision-theory concept of value of information (VOI). An active learning module orders updates within a group based on their ability to improve the learned model. The user is solicited for feedback, which is used to repair the database and to adaptively refine the training set for the model.

3.4 Crowd-Cleaning

A main limitation of GDR is interacting with a single user to clean the data. While this is sufficient for a small number of violations, it is definitely a bottleneck when the number of violations rises to the order of thousands or millions. We propose a data cleaning approach that is based on *crowd-sourcing*. That is, we use thousands of users to resolve the violations found in data. Crowdsourcing has been successfully used in other data management contexts to process large amounts of data (*e.g.,* [19]).

Consulting a large number of users raises various challenges such as the need for partitioning the data in an efficient and balanced way, assigning individual partitions to the best-matching human-cleaners, and resolving conflicts among their feedbacks. In order to partition the data in an effective way, we first detect existing violations as well as the potential violations that might appear during the course of data cleaning. Additionally, we keep track of the previously solved violations. The data is partitioned based on the obtained violations through standard graph clustering algorithms. Each partition is assigned to a human-cleaner such that a global objective function is maximized. This function reflects a load balancing criterion as well as the quality of matching between each partition and the expertise of the assigned human-cleaner.

3.5 Large-Scale Deduplication in Data Tamer

Recently, we introduced SCADD, a *SCAlable DeDuplication system*, to enable scalable data deduplication to a large number of nodes. SCADD is part of a large data integration system named Data Tamer, which we are currently developing in collaboration with MIT [24]. One of the goals of SCADD is to learn a deduplication classifier that (i) carefully selects which attributes to consider, (ii) successfully handles missing values, and (iii) aggregates the similarities between different attributes. To devise a new system to perform data deduplication at a large scale, we have to deal with several research challenges, including:

Large data volume. Existing deduplication techniques are not suitable for processing the sheer

amount of data that is processed by modern applications. Scalable deduplication is challenging not only because of the large amount of records, but also because of the significant amount of data sources. For example, in web site aggregators, such as Goby.com, tens of thousands of web sites need to be integrated with several sites that are continuously added every day. On average, each source contains tens of attributes and thousands of records.

Data heterogeneity and errors in data. Large-scale data management systems usually consist of heterogenous datasets that have different characteristics. For example, Goby.com collects data about hundreds of different entity types, such as golf courses, restaurants, and live music concerts. Some entities might have unique attributes (*e.g.,* the number of holes in a golf course) and different distributions of common attributes (*e.g.,* the range of vertical drops in downhill skiing sites vs. the range of vertical drops in water parks). In such scenarios, datasets experience a large amount of noise in attributes, such as non-standard attribute names, non-standard formats of attribute values, syntactical errors, and missing attribute values.

Continuously adding new data sources and new user feedback. Most of the modern applications, such as Goby.com, continuously collect data over time and integrate the newly arrived data into a central database. Data usually arrives at relatively high rates (*e.g.,* a few sources need to be integrated daily at Goby.com). The deduplication decisions depend on a number of factors, such as training data, previous deduplication results from legacy systems, and explicit deduplication rules set by expert users. Such an evidence is expected to evolve over time to (i) reflect better understanding of the underlying data or (ii) accommodate new data sources that might be substantially different from the existing sources. Clearly, it is infeasible to rerun the deduplication process from scratch in these cases. We need to provide efficient methods to update the deduplication results when new data or new deduplication rules arrive.

Distributed environment. For the large amounts of data gathered by current applications, it is inevitable to use multiple computing nodes to bring down the computational complexity. However, parallelising the data deduplication process causes another set of challenges: (i) it increases the overhead of shipping data between nodes and (ii) it requires coordinating the data deduplication task across multiple computing nodes.

As a result, SCADD has a significant emphasis on the incremental aspect of the problem by allowing every task in the deduplication process to be efficiently reevaluated when new data arrives or deduplication rules are changed. Furthermore, one of the main insights for improving the scalability of deduplication is partitioning the input data into categories (*e.g.,* possible categories in data collected by Goby.com are golf clubs, museums, and skiing sites). Such categorization allows for obtaining high-quality deduplication rules that are suitable for each specific category. Categorization also allows for reducing the number of records that need to be considered when running the data deduplication process (*i.e.,* records that belong to different categories cannot be duplicates and hence can be safely ignored).

4. DATA PROFILING

An important step before any kind of data-management, -cleaning, or -integration that can be well performed is data profiling, *i.e.,* determining various metadata for a given (relational) dataset. These metadata include information about individual columns, such as uniqueness, data type, and value patterns, and information about combinations of columns, such as inclusion dependencies or functional dependencies. With more and more and larger and larger datasets, especially from external sources and non-database sources, (big) data profiling is becoming ever more important. Research faces two principle challenges: efficiency and new functionality.

The efficiency challenge of data profiling arises from both the volume of data and the additional complexity in the size of the schema, for instance when verifying the uniqueness of all column combinations [1]. Especially for the sizes that "big data" promises/threatens, distributed computation is unavoidable. We are currently investigating how to scale conventional data profiling tasks to a large number of nodes. In this quest, we hope to exploit the fact that many intermediate calculations for various metadata are the same, so that the overall cost of calculating a "profile" is lower than the sum of the costs for the individual methods.

New kinds of data demand new profiling functionality: many tools and research methods concentrate on relational data, but much data now comes in other forms, such as XML, RDF, or text. While some profiling tasks and methods can be carried over, others either do not apply or must be newly developed. For example, when examining a *linked data* source, it is often useful to precede any further work by a topical analysis to find out what the source is about and by a graph analysis to find

out how well the source is interlinked internally and externally to other sources.

5. DATA ANALYTICS WITH THE WORLD BANK

We are building an analytics stack on unstructured data to leverage the vast amount of information available in news, social media, emails and other digital sources. We give one example of a recent collaboration with the World Bank. The World Bank implements several projects worldwide to provide countries with financial help. The details of those projects are documented and published along with financial details about the aid. The success of a project is usually reflected in the local social growth indicators where this project was implemented. A first step in analyzing the success of projects is to create a map that displays project locations to determine where aid flows are directed within countries; a process called geo-tagging. Currently, the task of extracting project locations and geo-tagging is executed by a group of volunteers who study project documents and manually locate precise activity locations on a map, a project called Mapping for Results.

We have developed a tool to retrieve the documents and reports relevant information to the World Bank projects. We also run various classifiers and natural language processing tools on those text documents to extract the mentioned locations. Those locations are then geo-coded and displayed on a map. The developed tool combines these locations with other financial data (*e.g.,* procurement notices and contract awards) of projects into a map, thus providing a holistic view about the whereabouts of projects expenses. The technologies used in this project include the UIMA extraction framework and a just-in-time information extraction stack [9].

6. ANALYTICS AND MINING OF WEB AND SOCIAL MEDIA DATA

Online data content is increasing at an explosive rate, as seen in the proliferation of websites, collaborative and social media, and hidden web repositories (the so-called deep web). To cope with this information overload, designing efficient ways of analyzing, exploring and mining such data is of paramount importance. In collaboration with the social computing group at QCRI, we are developing mining algorithms for crawling, sampling, and analytics of such online repositories. Our methods address the challenges of scale and heterogeneity (e.g., structured and unstructured) of the underlying data, as well as access restrictions such as proprietary query views offered by hidden databases and social networks.

As a specific example, we are developing a system for addressing the problem of data analytics over collaborative rating/tagging sites (such as IMDB and Yelp). Such collaborative sites have become rich resources that users frequently access to (a) provide their opinions (in the forms of ratings, tags, comments) of listed items (e.g., movies, cameras, restaurants, etc.), and (b) also consult to form judgments about and choose from among competing items. Most of these sites either provide a confusing overload of information for users to interpret all by themselves, or a simple overall aggregate of user feedback. Such simple aggregates (e.g., average rating over all users who have rated an item, aggregates along pre-defined dimensions, a tag cloud associated with the item) is often too coarse and cannot help a user quickly decide the desirability of an item. In contrast, our system allows a user to explore multiple carefully chosen aggregate analytic details over a set of user demographics that meaningfully explain user opinions associated with item(s) of interest. Our system allows a user to systematically explore, visualize and understand user feedback patterns of input item(s) so as to make an informed decision quickly. Preliminary work in this project has been published in [8, 25].

7. REFERENCES

[1] Z. Abedjan and F. Naumann. Advancing the discovery of unique column combinations. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, 2011.

[2] M. Arenas, L. E. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. *Theory and Practice of Logic Programming (TPLP)*, 3(4-5), 2003.

[3] G. Beskales, I. Ilyas, L. Golab, and A. Galiullin. On the relative trust between inconsistent data and inaccurate constraints. In *Proceedings of the International Conference on Data Engineering (ICDE)*, 2013.

[4] G. Beskales, M. A. Soliman, I. F. Ilyas, and S. Ben-David. Modeling and querying possible repairs in duplicate detection. *Proceedings of the VLDB Endowment*, 2009.

[5] P. Bohannon, W. Fan, M. Flaster, and R. Rastogi. A cost-based model and effective heuristic for repairing constraints by value modification. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, 2005.

[6] X. Chu, I. Ilyas, and P. Papotti. Holistic data cleaning: Putting violations into context. In *Proceedings of the International Conference on Data Engineering (ICDE)*, 2013.

[7] G. Cong, W. Fan, F. Geerts, X. Jia, and S. Ma. Improving data quality: Consistency and accuracy. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, 2007.

[8] M. Das, S. Thirumuruganathan, S. Amer-Yahia, G. Das, and C. Yu. Who tags what? an analysis framework. *Proceedings of the VLDB Endowment*, 5(11):1567–1578, 2012.

[9] A. El-Helw, M. H. Farid, and I. F. Ilyas. Just-in-time information extraction using extraction views. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, pages 613–616, 2012.

[10] M. G. Elfeky, A. K. Elmagarmid, and V. S. Verykios. TAILOR: A record linkage tool box. In *Proceedings of the International Conference on Data Engineering (ICDE)*, 2002.

[11] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 19(1), 2007.

[12] W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. Conditional functional dependencies for capturing data inconsistencies. *ACM Transactions on Database Systems (TODS)*, 33(2), 2008.

[13] W. Fan, F. Geerts, N. Tang, and W. Yu. Inferring data currency and consistency for conflict resolution. In *Proceedings of the International Conference on Data Engineering (ICDE)*, 2013.

[14] W. Fan, X. Jia, J. Li, and S. Ma. Reasoning about record matching rules. *Proceedings of the VLDB Endowment*, 2(1), 2009.

[15] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. Cerfix: A system for cleaning data with certain fixes. *Proceedings of the VLDB Endowment*, 4(12):1375–1378, 2011.

[16] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. Interaction between record matching and data repairing. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, 2011.

[17] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. Towards certain fixes with editing rules and master data. *VLDB Journal*, 21(2), 2012.

[18] W. Fan, J. Li, N. Tang, and W. Yu. Incremental detection of inconsistencies in distributed data. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 318–329, 2012.

[19] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. CrowdDB: answering queries with crowdsourcing. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, 2011.

[20] H. Galhardas, D. Florescu, D. Shasha, E. Simon, and C.-A. Saita. Declarative data cleaning: Language, model, and algorithms. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, 2001.

[21] B. Marnette, G. Mecca, and P. Papotti. Scalable data exchange with functional dependencies. *Proceedings of the VLDB Endowment*, 3(1), 2010.

[22] F. Naumann. *Quality-Driven Query Answering for Integrated Information Systems*, volume 2261 of *LNCS*. Springer, 2002.

[23] V. Raman and J. M. Hellerstein. Potter's Wheel: An interactive data cleaning system. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, 2001.

[24] M. Stonebraker, D. Bruckner, I. F. Ilyas, G. Beskales, M. Cherniack, S. Zdonik, A. Pagan, and S. Xu. Data curation at scale: The Data Tamer system. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, 2013.

[25] S. Thirumuruganathan, M. Das, S. Desai, S. Amer-Yahia, G. Das, and C. Yu. Maprat: Meaningful explanation, interactive exploration and geo-visualization of collaborative ratings. *Proceedings of the VLDB Endowment*, 5(12):1986–1989, 2012.

[26] J. Wijsen. Database repairing using updates. *ACM Transactions on Database Systems (TODS)*, 30(3), 2005.

[27] M. Yakout, A. K. Elmagarmid, H. Elmeleegy, M. Ouzzani, and A. Qi. Behavior based record linkage. *Proceedings of the VLDB Endowment*, 3(1):439–448, 2010.

[28] M. Yakout, A. K. Elmagarmid, J. Neville, and M. Ouzzani. GDR: A system for guided data repair. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, 2010.

[29] M. Yakout, A. K. Elmagarmid, J. Neville, M. Ouzzani, and I. F. Ilyas. Guided data repair. *Proceedings of the VLDB Endowment*, 4(5), 2011.