Query-Driven Data Cleaning for Exploratory Queries

Stella Giannakopoulou
EPFL
stella.giannakopoulou@epfl.ch

ABSTRACT

Cleaning dirty data is a costly process which is subjective to the user who manipulates the data. Data scientists spend most of their time detecting and repairing errors before analyzing their data. At the same time, the existing optimized solutions for data cleaning require a significant amount of time to execute and are usually applied as a separate offline process, decoupled from data analysis [2]. Having data cleaning as an offline process might result in paying the cost of cleaning parts of the dataset which are never used by the analysis process, while also requiring users to have a priori knowledge of the types of errors in the data [1].

We present an approach that masks the data cleaning cost by cleaning only the part of the dataset that gets accessed by exploratory queries. We integrate query processing with repairing violations of functional dependencies. In order to ensure clean query answers, we apply query relaxation techniques which enhance the query answer with the appropriate information from the dataset that is required for the detection and repairing of functional dependency violations. After the execution of each query we isolate the changes made in the dataset and we apply the delta to the original dataset. Thus, we incrementally obtain a clean instance of the dataset by adding an overhead to each query.

Our approach for detecting and repairing violations of the data accessed by a query requires two levels of operations. At a higher level we distinguish between the type of query; in this work we consider Select Project (SP) and Join queries. Then, at the lower level we address the optimization requirements of the queries.

1. LOGICAL LEVEL OPERATIONS

In order to ensure clean answers for SP and Join queries, we apply query relaxation techniques which augment the result set with the entities of the dataset which are in the context of the query. In the case of SP queries given a dataset with a functional dependency FD:$X \rightarrow Y$, and a query which accesses an attribute that participates in the functional dependency, then the system triggers the violation detection procedure on the answer of the query. Then, for the erroneous cells of the query answer which belong to X, it augments the query result with all the values of the whole dataset which have a matching value for the attributes in Y. Similarly for the erroneous cells which belong to Y. A similar idea also holds for datasets which violate multiple functional dependencies.

In the case of Join queries we might have two dirty datasets where each of them violates a set of functional dependencies, while at the same time there might exists a set of functional dependencies that the join result must satisfy. Providing consistent answers in the case of join queries is more challenging than the case of SP queries, because any repair that changes a cell in one of the datasets might affect the join result, and similarly, a repair that changes the join result must be propagated to the original tables.

© 2018 ACM. ISBN 978-1-4503-2138-9…$15.00
DOI: 10.1145/1235

2. PHYSICAL LEVEL OPERATIONS

Physical level operations involve the low-level operations required to integrate query processing with data cleaning. More specifically, after performing the violation detection over the relaxed answer set, we transform the answer of the query into a probabilistic answer [3] which contains all the possible values that each of the erroneous cells can take. Then, using this probabilistic information, the system updates the dataset with the probabilistic information obtained through the query. Therefore, at the physical level we focus on operations that deal with the problem of query processing and data cleaning over probabilistic data in order to efficiently obtain a clean version of the dataset.

3. REFERENCES

[1] Hotham Altwaijry, Sharad Mehrotra, and Dmitri V. Kalashnikov. QuERy: A Framework for Integrating Entity Resolution with Query Processing. *PVLDB*, 9(3), 2015.

[2] Stella Giannakopoulou, Manos Karpathiotakis, Benjamin Gaidioz, and Anastasia Ailamaki. Cleanm: An optimizable query language for unified scale-out data cleaning. *Proc. VLDB Endow.*, 10(11):1466–1477, August 2017.

[3] Dan Suciu, Dan Olteanu, R. Christopher, and Christoph Koch. *Probabilistic Databases*. Morgan & Claypool Publishers, 1st edition, 2011.