Scalable data cleansing based on qualitative attributes

Mika Huttunen
Helsinki University

ABSTRACT

Abstact yet to be written..

Keywords

data cleansing, big data, scalability

1. INTRODUCTION

Nowadays companies are gathering large amounts of data in different ways such as via user input, and all kinds of sensors. It's also becoming easier and easier for them to use data in all kinds of decision making, analytics, and for example in automating human-involved tasks. Problems arise when the used data is *dirty*, or invalid, and thus can lead into making incorrect decisions. These can sometimes cause serious issues - especially in health care and financing sides [7].

Humans often make data input errors via misspelling, and sensors may grab unwanted noise along the data they're designed to catch. In fact, over 25% of critical data in the world's top companies is flawed [8]. Not to mention that today, the variety of data is also large which leads into collecting data of different formats together. *Data cleansing* is the solution to the above-mentioned problem. For data controllers, understanding data cleansing, and the problems it tries to solve is thus naturally important.

Data cleansing is a research field that explores ways to improve *quality* of dirty data. If data isn't of high quality, it means that it has usually both *schema*, and *instance* level problems [9]. Another used definition for dirty data is that it doesn't meets its usage needs. A simple example of that would be such that a front developer of a service can't provide end-users a good UX because the server cannot deliver all kinds of useful data. This could happen when there are faults in the way the data is stored in the database.

Figure 1 shows how *single-source problems* with data quality can yet be divided into schema and instance level data problems. Single-source problems simply stand for such that can occur in data that is stored in a single site with possible

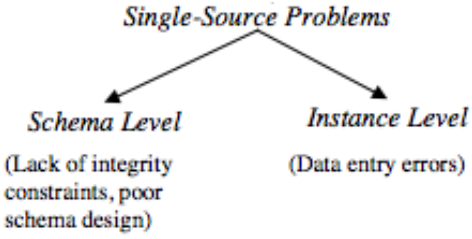

Figure 1: Single-source data quality problems [9]

data replication on the database side. The same problems can also arise in *multi-source* systems where the data may also be scattered across several sites.

Instance level problems in data arise as attributes of data *tuples* being out of their scope, or just wrong. These contain misspellings, and *outliers* by for example possible sensor errors, and processing errors before gathered data is actually stored. These problems can be detected via *quantitative techniques* related to data cleansing that focus on error detection and correction based on numerical attributes of data [6]. If for example some framework can reduce multi-dimensional data in two-dimensional space, it's fairly easy for an expert to detect outliers in that space, and thus possible errors in the data.

Schema level problems on the other hand arise as *violations* of *rules* such as *functional dependencies* (FD) [7] (p. 289 – 291). As an example, functional dependency

$$ZIP \rightarrow STATE \tag{1}$$

defines a constraint between data attributes ZIP and $STATE$ so that ZIP implies the $STATE$. In other words, if we had a dataset with two tuples having the same ZIP attribute, but different values of $STATE$, and we trust rule 1, at least one of them is erroneous.

Duplicate (or partically duplicate) data entries can also be considered both schema level, and instance level data quality problems [7] (p. 283). They usually appear as the same entries with some attributes having different values from each other, and thus should be classified as instance level problems. They can however be detected by similar rules that are used for other detecting schema level problems.

Consider us having a person register where each person has its name, phone number, address, and zip code defined. Now we could have tuples t_1 and t_2 with the same name, phone number, and address, but different zip codes. Tuples t_1 and t_2 could be detected being duplicate via a rule

which defines that each tuple is unique by a combination of (*name*, *phone number*, *address*). This would be possible for example via a rule such as *UDF* (user-defined function).

In traditional ways, the data errors are detected by algorithms that compare data tuples trying to find rule violations. Rules can either be pre-defined by an expert, or automated code, or such that the data cleansing framework itself generates them by processing the data. The found errors can afterwards be fixed either via totally automatic, or (partially) human guided procedures.

This seminar report focuses mostly on *qualitative* data cleansing frameworks that are highly scalable for *big data*. They detect data errors via data quality rules such FDs, *CFDs* (conditional functional dependencies), *DCs* (denial constraints) which are explained in the paper by Ilyas and Chu [7] (p. 287 - 301). I will leave further investigation of them up to the reader.

Some data cleansing frameworks like *BigDansing* also apply UDFs alongside the traditional rules (FDs, CFDs and DCs) [8]. UDFs extend traditional rules by allowing usage of more complex norms using procedural language. Using UDFs for error detection thus improves error detection accuracy compared to using just traditional data quality rules.

Some data cleansing frameworks on the other hand don't require cleansing the whole source data, but rather solve the problem of queries returning invalid results by other means. *SampleClean* for instance applies sampling for improving answer quality for queries while having a dirty database instance on the background [11].

The rest of the report is divided as follows: Section 2 introduces recent findings, and history related to qualitative big data cleansing. It also discusses, where we stand on the field at the moment. Section 3 discusses recent frameworks introduced in Section 2 in more detail. It does also analysis and comparison with them. Section 4 discusses open challenges in big data cleansing field, and Section 5 has my own ideas for future work. Section 6 concludes the report.

2. RELATED WORK

Data cleansing has been an interesting topic for decades, and there has been a large amount of research on the field in the recent years, Extensive summary from 2015 by Ilyas and Chu discusses recent techniques on cleansing relational data [7]. The summary introduces algorithms for finding different kinds of data quality rules based on a database instance, and its schema. Error detection is afterwards possible by looking for erroneous tuples with tuple-wise comparisons using one rule at a time.

For error repairing, Ilyas and Chu introduce a classification of different data repairing techniques [7] (p. $330-332$). Based on the classification, they also discuss recent frameworks that apply different techniques. Figure 2 represents data repairing techniques classification.

As we can see from the figure, we can have three different *repair targets*. We can either repair data, rules, or both. So far, we have only been discussing data repairing which is about repairing data based on a set of trusted data quality rules. This report will bypass techniques for the other two targets, but the latter repair targets are yet mentioned next.

Rules only repairing stands for trusting that data is valid, and a set of predefined data quality rules should be modified so that they match to the data. Similarly trusting both data and rules is an approach for modifying both data and rules in such way that they eventually match with each other.

We can also see from figure 2 that data repair target is yet being divided into two sub methods. There are methods for detecting and repairing data errors using one rule at a time, or doing the same thing *holisticly*. A holistic method stands for using multiple rules for detection and repairing at the same time, and an example application of the technique is *Holistic data cleaning* [5]. It can be shown, that the holistic technique improves the error detection accuracy compared to non-holistic methods [7] (p. 331).

The following part of the report is yet to be written out. But the plan was to talk about techniques related to doing data cleansing with big data, and afterwards recent frameworks, that apply these techniques. There are ones like BigDansing (distributed rule-based error detection and correcting), SampleClean (probability based fixed data sampling while having dirty database on the background), DeDoop / Dis-Dedup (frameworks specifically designed for duplicate tuples detection) [3], and HoloClean [10]. HoloClean is by far the most interesting of the above-mentioned frameworks. It offers holistic data repairing, and unifies a range of data repairing methods under a common framework.

Abedjan et al. show in their research paper from 2016 that even though frameworks such as BigDansing are highly scalable, and can be applied to cleansing all kinds of datasets, they can repair around 36% of all possible errors - even with well-optimized configuration parameters. If an expert is however used to cleanse data by applying several frameworks in the right order for specific data cleansing tasks, much better repairing accuracy is met [1].

This specific research probably inspired partially the same research team in designing HoloClean. According to the Rekatsinas et al., HoloClean achieves over twice the repairing accuracy of existing state-of-art methods [10].

3. STATE-OF-ART FRAMEWORKS

This section discusses the recent frameworks on big data cleansing field. I introduce BigDansing [8], SampleClean [11], Dis-Dedup [3], and HoloClean [10]. Although the frameworks are quite different, some deeper analysis between them can be found at the end of this section.

3.1 BigDansing

BigDansing is a highly scalable framework for data cleansing. It makes parallelised error detection possible by providing an abstraction for data quality rules (FDs, CFDs, DCs and UDFs), and having an internal solution for efficiently handling abstracted rules. BigDansing can be run on top of a distributed computing system like MapReduce.

Discussion related to this and rest of the frameworks is yet to be written..

4. OPEN CHALLENGES

We have seen that there are highly scalable, and accurate frameworks for cleansing relational and structured data. There is however a large variety of data that doesn't fit into those categories like JSON and text documents, images, audio files and so on. Data quality problems related to them still remain unexplored [2].

Along with open challenges for data variety, velocity also

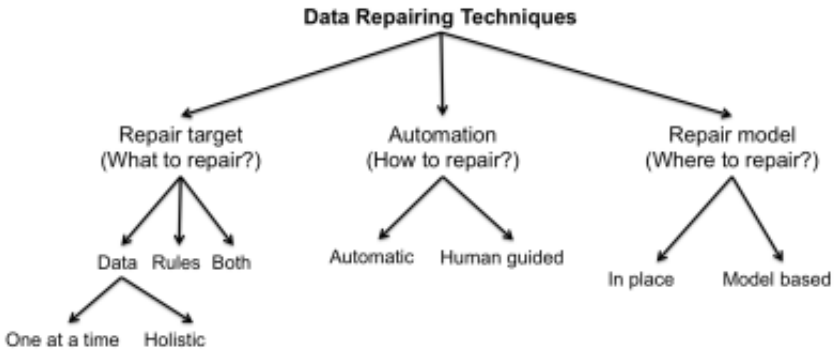

Figure 2: Data repairing techniques [4]

poses a problem in big data cleansing. Even though the running times of recent frameworks for actual cleansing of large datasets are reasonable, they don't fit for continuos repairing of streaming data. Data sampling can however be applied for always having valid data for querying.

5. OWN IDEAS

Chu and Ilyas state that data quality problems related to cleansing semi-structured or unstructured data yet remain unexplored [2]. I think that for the next steps in data cleansing, one should focus on this specific problem. Many used databases today are after all NoSQL, and store data in a way where rule-based error detection is not applicable.

6. CONCLUSION

Conclusion yet to be written..

7. REFERENCES

[1] Z. Abedjan, X. Chu, D. Deng, R. C. Fernandez, I. F. Ilyas, M. Ouzzani, P. Papotti, M. Stonebraker, and N. Tang. Detecting data errors: Where are we and what needs to be done? *Proc. VLDB Endow.*, 9(12):993–1004, Aug. 2016.

[2] X. Chu and I. F. Ilyas. Qualitative data cleaning. *Proc. VLDB Endow.*, 9(13):1605–1608, Sept. 2016.

[3] X. Chu, I. F. Ilyas, and P. Koutris. Distributed data deduplication. *Proc. VLDB Endow.*, 9(11):864–875, July 2016.

[4] X. Chu, I. F. Ilyas, S. Krishnan, and J. Wang. Data cleaning: Overview and emerging challenges. In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD '16, pages 2201–2206, New York, NY, USA, 2016. ACM.

[5] X. Chu, I. F. Ilyas, and P. Papotti. Holistic data cleaning: Putting violations into context. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 458–469, April 2013.

[6] J. M. Hellerstein. Quantitative data cleaning for large databases, 2008.

[7] I. F. Ilyas and X. Chu. Trends in cleaning relational data: Consistency and deduplication. *Found. Trends databases*, 5(4):281–393, Oct. 2015.

[8] 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 *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, pages 1215–1230, New York, NY, USA, 2015. ACM.

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

[10] T. Rekatsinas, X. Chu, I. F. Ilyas, and C. Ré. Holoclean: Holistic data repairs with probabilistic inference. *Proc. VLDB Endow.*, 10(11):1190–1201, Aug. 2017.

[11] J. Wang, S. Krishnan, M. J. Franklin, K. Goldberg, T. Kraska, and T. Milo. A sample-and-clean framework for fast and accurate query processing on dirty data. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, pages 469–480, New York, NY, USA, 2014. ACM.