Requirement Engineering For Databases And Data Cleaning

Ishank Jain
Department of Computer Science
University of Waterloo
Waterloo, Canada
ishank.jain@uwaterloo.ca

Marian Boktor
Department of Computer Science
University of Waterloo
Waterloo, Canada
mkayboktor@uwaterloo.ca

***Abstract*—Data cleaning plays an essential role in ensuring data quality for enterprise applications. Extensive research has been done in this area, and many data cleaning algorithms have been translated into tools to detect and to possibly repair certain classes of errors such as outliers, duplicates, missing values, and violations of integrity constraints such as Fuctional Dependencies (FDs), Conditional Functional Dependencies (CFDs), and Denial Constraints. In this paper, we begin with the introduction that explains the basics requirements, advantages, types of error, and types of data cleaning tools. Section II introduces databases and data quality criteria, followed by an overview of data cleaning task in both research and industry. In section III, we investigate four data cleaning tools which uses different machine learning algorithms to detect and repair dirty data along with evaluation. We end with conclusions and future work.**

***Index Terms*—ActiveClean, BoostClean,Databases, Data Cleaning,Data Tamer, ETL, HoloClean, Machine Leaning.**

I. Introduction

With the increasing importance of data driven approaches in industry, data cleaning researches has turned into an urgent and essential part. Since, the use of dirty and unclean data to produce results can lead to false inference and misled choices, it is critical to clean the data before it is utilized. Subsequently, it is important to contemplate the viability of the accessible data repairing and error detection tools used in tackling real data cleaning issues. In this paper we will see four tools, in particular: ActiveClean, BoostClean, HoloClean, and Data Tamer.

To begin, we will define what it means for the data to be clean. Clean data needs to pass a set of quality criteria that we define below:

1) Consistency: Two data items (tuples/records) in the dataset should not contradict each other.
2) Uniformity: Dataset integrated from different sources should follow same measures, for instance schema description.
3) Validity: Data should adhere to quality rules such as FDs, CFDs and denial constraints.

A. Present methods

In this section, we discuss various data cleaning methods that are available today and classify them into four categories:

1) Pattern detection and repairing tools such as Trifacta and Katara, discover patterns in the data. These patterns are either semantic or syntactic patterns, and these are used to detect errors (cells that do not conform with the patterns).
2) Violation Rule-based detection algorithm,in this class the rules can range from a not null or a distinct constraint to multi-attribute based rules such as denial constrains and functional dependencies (FDs) to user-defned functions. Within this set of rules, a user is able to specify a collection of rules that clean data will obey.
3) Qualitative error detection algorithms expose outliers, and glitches in the data [5].
4) De-duplication and Record merging algorithms for detecting duplicate data in the daataset such as the Tamer. These tools perform entity consolidation when multiple items have data for the same entity [5].

B. Challenges

There are various challenges associated with preforming data cleaning. They are:

1) Lack of real world data: most the tools are examined on synthetically built data, this allows us to test the performance capabilities of the tool, but it doesn't make clear the tool's real world applicability. Moreover, its hard to judge the effectiveness of the tool.
2) Specific error based tools: Real-world data generally have various types of errors. For example, an error might be part of inconsistent duplicate tuple and violate an quality constraint simultaneously, which may lead to error not identified by the tools.
3) Cost of Human involvement: Almost all applicable tools involve humans at some stage, for instance, to verify detected errors, to specify cleaning rules such as FDs and CFDs, or to provide feedback that can be to tune the parameters of machine learning algorithm.

Fig. 1. Data Cleaning Cycle

4) Overfitting: If a machine learning model over-fits the training data, this can be avoided by adding penalty term. This is necessary to be able to integrate future data sources.

These are the various challenges which commonly arise when evaluating machine learning tools for data cleaning.

C. Required characteristics

When we evaluate machine learning tools for data cleaning there are certain features which one should look for:

1) Systems needs to have fully automated algorithm with minimum involvement from a person to keep the human cost in check.
2) The tools should be able to integrate new data sources as they are added to be able to avoid re-training cost and model selection cost.
3) It is preferred for a tool to provide a scripting language or a interacting interface for programmers to interact with the model and allow them to add custom functions.
4) The tool should be able to correctly identify dirt data. This is necessary to have high recall for identifying dirty records so that human intervention can be avoided. (A record is a group of related data held within the same structure)

D. Basic Requirements

There are certain requirements when building a data cleaning tool:

1) Training data is needed to train the machine learning model. The training data contains dirty data which can be detected by integrity rules or outlier detection algorithms.

2) Clean data is provided by the set of expert analysts which is used to generate performance results. Based on the results the machine learning parameters can be altered to improve results.
3) Test data, once the machine learning model is trained it can be used to clean the test data which is a real-world dataset.
4) Integrity violation rules such as Denial constraints and Functional dependencies are needed to detect dirty or erroneous data in the dataset. For instance, Holoclean uses Denial constraints to detect dirty data and label them as "Don't know" cells (D_k).
5) Machine learning algorithm to train the model to data cleaning may include:
   - Clustering algorithm such as correlation clustering algorithm to detect outliers and dirty cells. For instance ActiveClean, Tamer.
   - Neural network based algorithm which is trained on a feature graph model to generate potential domain, for instance, HoloClean.
   - Classification and boosting algorithm (SVM, Nave Bayes etc.) to assign the correct class labels from the domain based on a loss minimization function, for instance, BoostClean.

E. Advantages of data cleaning

There are several benefits of cleaning the data before utilizing its anaytics:

1) Improved decision making: Since dirty data can lead to misinformed decisions, data cleaning improves the results by supporting better analytics.
2) Streamlines Business Practices: Eradicating duplicate data from the database can help business enterprises to streamline business practices and save a lot of money [9].
3) Improve productivity by eliminating delays due to interruptions caused by inter-departmental communication, thus leading to cohesive work-flow.

F. Types of Errors

An error can be defined to be a deviation from its true value. More formally, given a data set, a data error is an atomic value (or a cell) that is different from its given ground truth [**?**]. Refer fig 2.

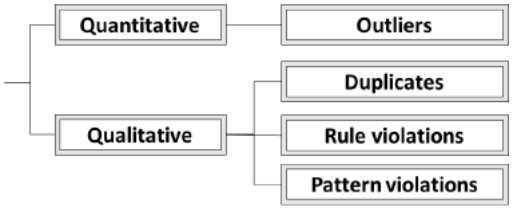

Fig. 2. Error types

1) Outliers error refers to data values that differ from the general distribution of values in a attribute in the dataset, for instance, a person over age of 50 in a dataset where most people are aged 20-35.
2) Duplicates error refers to distinct records that point to the same record in the dataset. If attribute values for the duplicates do not match, it could indicate an error.
3) Rule violations refer to values that violate integrity constraints, such as Not Null constraints and functional dependencies, and Denial constraints.
4) Pattern violations refers to the values that violate semantic and syntactic constraints, for instance, data type, and misspelled words, for instance, New York can be misspelled as New Ark or New York etc.

Few errors may fit into more than one category of error, so this list only presents the general error types which are found in the dataset. Further, the list may not include all kinds of error, but presents the types of error which are most commonly found in a dataset.

Since data cleaning is mostly performed on data stored in databases, we chose to introduce some specifications of quality database and software along with their positive impact on both individuals and businesses. Section II provides an overview of that.

II. Databases

A database is generally known as an electronically organized assemblage of data, which in turn ease the process of data access, manipulation and update. Its primary goal is to help individuals and organizations to store, manage and retrieve information.

A. Specifications of a Quality Database

The first and foremost aim of having a database of high quality is to aid in reinforcing and guaranteeing the accuracy and integrity of data. Another primary target is to separate data into subject-based tables to decrease redundant information, and supply access with the information necessary for combining and organizing the data in tables as required. Moreover, accommodating data processing and reporting demands as well as information retrieval and user application have never been easier to structure and sustain, not to forget, data manipulation and maintenance while leaving other fields or tables unaffected.

Added to the above, not only should the data be normalized to curb redundancy and restructure transaction statures and enforce referencing integrity, but semantically equivocal data should also be represented the same even though it might be harnessed through multitudinous sources. Apart from that, the database should be agile and can encompass a range of need for any organization. On top of the need for adopting a singular view and access to precise inconsistent information, quality design must be followed. This means that data should be stored in a single logical unit replacing various files in order to: preserve data integrity and security; raise the overall performance of the database.

B. Useful Database Software Criteria

To begin with, the database software should be user-friendly and convenient as far as possible. There should be no complications and minimal obstacles if any when accessing and interfacing with the database software. To clarify, the database software should be as easy to use and as available to all types of users as can be. Furthermore, the software should not only be an option for personal computer and laptop users. However, perhaps it might be more recommended that the software be available to any and all users of portable devices, namely, phone and tablet users. Limiting it to a specific base of users might be reductive and might even turn out to be counterproductive.

More importantly, one major significance which ought not to be ignored is how supple and modifiable the database software should be. It should not be strict to a specific and non-dynamic range of options. On the other hand, it should be adjustable enough to meet the needs of any user or organization. That is not to be confused with consistency for companies. The database software is preferred to be specifically designed to rise up to and cover the demands of each individual company, depending on the separate and respective needs of each.

That is not everything concerning flexibility. Updates in databases are advisable to be split-second quick. That is to mean when one modification is carried out at a certain database, it should appear to all users who can access this specific database immediately, simultaneously and concurrently offering real-time insights. Since accessibility is key, no good database software should be available only offline. For this purpose, a genuinely usable software should have cloud access. To elaborate, data should not only be available strictly on a cloud-based system, but the software should offer data storage capacity on top of online access.

Above everything else, it is wise that the software should be ready to provide a wide range and a considerable variety of formats in lieu of being strictly limited to one format or the other, no matter how pervasive or prevalent that singular format might be, since following a singular format would probably limit the amount and diversity of users. Last but not least, certain regulations are ought to be sat to weed out any inconsistencies and abnormalities, as in line with standard data constraints. This is to maintain and upkeep the integrity and genuine transparency of data fields.

C. Quality Database Benefits

1) Centralized systems: One great and common challenge for all developing businesses is tracing the growing amount of data. Quality database system can make keeping central control over business-sensitive data so much easier, safer and more secure whenever arises the need in order to increase your opportunities to succeed [10].

2) Profounder reliance on analytical systems: It is rather staggering how often companies use data which they can neither trust nor guarantee only to yield results that are at best dubious and at worst actively destructive. An estimate study researches of about 450 reveal that around 60 percent of senior executives are tentative about the refinement of the data quality of their organizations. Despite that, even erroneous data is considered a superior alternative to complete lack of data upon meeting the lurking presence of deadlines. The instruments for data quality can guarantee only worthy and reliable data is used for decision making processes, which therefore raises confidence in and systematic reliance on those analytical systems.

3) Better management of human resource (HR) matters: For all office managers, there are multitudinous benefits of managing HR affairs for database systems. To begin with, the single first and foremost primary advantage of using an HR database for staff management record-keeping is simply that it can save time as well as money. A second and equally important benefit is that it can not only expedite but also the overwhelming major HR functions; the most leading and noticeable instances being automating routine jobs and speeding up data processing like staff hours, leave, benefits, payroll among other things. Evidently, such perks can enable you to have more free time to focus on the growth of your business.

4) Managing customer data and relationships: For all customer-based businesses, good customer relationship management (CRM) database is key for all accounts of growth contingencies. Completely whole CRM databases are usually resilient enough to not only store but also work through everything raging from customer contact details through interaction history and accounts all the way to freshly newfound prospects and even leads and business perspectives. On top of all that, and quite surprisingly, a mentionable number of CRM systems can even make it easier to operate and trace marketing campaigns; for instance, email newsletters. To exemplify, one common problem is customers who call their IT service providers and being forced to recite model number OS version and patch levels for their PC. The more obvious and more preferable solution would be if the provider had record knowledge of the versions of everything that the customer has installed on their system when the call was made. In short, and in other words, IT customer service is a heavily data-comprehensive process which significantly benefits from cutting edge verified information. That is to mean, that ultimately in the vast and fast-paced world of IT Service Management (ITSM) information breeds customer satisfaction very conducively.

5) Efficient inventory tracking: Proper balancing out of the inventory can often feel like a chore. Redundancies, sitting on a shelf risking waste, or deficiencies can be common jeopardizing customer satisfaction and potentially hurting company reputation. Also, human errors abound when keeping manual track of your inventory. Such errors include miscount data entry errors and misplacing sheets or notes. By utilizing and implementing an inventory tracking database, especially accompanied with electronic data interchange and barcode scanning, the previously mentioned risks can be evaded and lost sales can be brought down to a bearable minimum while simultaneously optimizing chances for growth. Moreover,

manually reconciling data is a black-hole of time-consuming valuable resources since manual reconciliation only ranges at a remarkably limited capacity. The rule of diminishing returns thwarts progress as data sources and linked error rates rise which would insinuate that rules-based automation can help contain reconciliation in terms of time and effort.

6) Planning for growth: Most business databases have some form of reporting capabilities from analyzing input data and productivity tracking, to anticipating future trends and customers' needs. If a company is planning a strategy for growth, a sturdy database system can be a business' most prized asset.

7) Increased revenues and reduced costs: When a business is able to make decisions on a foundation of high quality, validated data, positive top-line outcomes are a likely result. Unreliable data results in less confident decisions that can often lead to missteps and rework that dont deliver increased revenues. However, and in a surprisingly similar fashion, if data quality enables an organization to complete a project correctly the first time around, it also enables the organization to operate more efficiently and complete more projects. Project delays due to course corrections burn through budgets and slow business growth.

III. Data cleaning tools

In this paper we will discuss four different types of machine learning approaches to data cleaning while discussing the tools. ActiveClean and Tamer use clustering techniques to identify the outliers, HoloClean uses two-layer neural network model to train the model to repair the dirty data in the dataset, and finall BoostClean uses classification and boosting algorithms to train a model that minimizes the cost funtion using stochastic gradient descent.Further, we will discuss each tool in detail and describe the machine learning techniques used by each tool. Based on our estimations we will first present the evaluation as reported in the academic papers and then our F_{60} score estimantion.

A. Importance of data cleaning

If the data is clean, everyone will be more efficient at what they do, as they will be able to extract the right information they need in order to complete their task. That is to say that, data cleansing is an essential need for both individuals and parties (or businesses). As for businesses, it is always important to know your customers and target market better through having the most recent, accurate information about them. This definitely helps in getting the best out of efforts made to market businesses products and services.

More importantly, data cleansing greatly helps in increasing the overall productivity as a result of data quality improvement. Quality data is achieved through cleaning when all outdated and erroneous information is removed or repaired. Consequently, a huge amount of time and effort is saved for the employees by dealing with less, yet more accurate, set of files and documents. Not to mention, the impact on operational and maintenance cost reduction when data quality is enhanced. Moreover, continuously having errors in companys work can lead to harm its reputation and thus its profit. On the contrary, business enterprises can exceptionally promote their revenue by improving their response rates and reaching the customers conveniently and quickly.

Unfortunately, however, a considerable multitude of conglomerates fall short of setting data quality management as a top priority. What is even more astonishing and worthy of note is that an astoundingly shocking number of them do not keep up-to-date records of how frequently quality control checks are carried on concerning their customer data. High-quality data and precise sense of information are paramount to the decision making process. Clean data is conducive to superior and overall greater analytics on top of coherent and holistic business intelligence which in turn is able to make superior decision making and implementation much easier. Ultimately, a more astute sense of data for any company can build towards finer decision making which therefore will add up to the success of any enterprise eventually. One glaring instance and shining evidence of this is that any manufacturing company can encounter serious obstacles if configuring robots and other production machines were based on poor processing date. Apart from that, a given online business can be at the risk of being penalized from the government simply by not rising up to the expected privacy regulations for its customers [11].

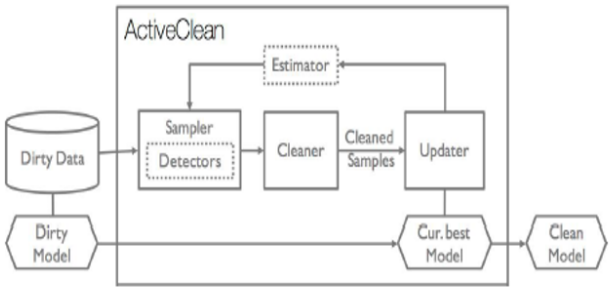

Fig. 3. ActiveClean Architecture

B. ActiveClean

ActiveClean is a progressive framework for training Machine Learning models for data cleaning [1] [3]. The system cleans small batches of data and updates the model iteratively. The model includes various optimization techniques, for instance, importance weighting and dirty data detection. ActiveClean framework prioritizes data cleaning by identifying records, which if cleaned, are likely to change the model's predictions. The framework applies to a large class of models which optimizes loss minimization problems which can be solved by gradient descent method. This captures Neural Networks (NN), Linear Regression, SVMs, and some other types such as Latent Dirichlet allocation (LDA).

The ActiveClean Framework assumes that there is a featurizer F (which acts as a black box) that maps every record in a dataset r to a feature vector x and label y. The framework focuses on a class of predictive analytics problems, ones that can be expressed as the minimization of loss functions, which will be trained on the output of applying F to the records in dataset. The problem is to find a vector of model parameters θ by minimizing a loss function over all labeled training examples $f(x_i; y_i)_{i=1}^{N}$:

$$\theta^* = argmin_\theta \sum_{i=0}^{n} l(x_i, y_i; \theta) + r(\theta)$$

here, r(θ) penalizes the feature weights in θ to avoid overfitting based on whether the value is high or low.

1) Required Input: Input to ActiveClean are four user-defined parameters(UDF). the first and second paramenters are used to train the model, and are always available to implement directly. The third and fourth parameters have available default values, but can be tuned manually.

- Model: In ActiveClean system the user chooses a predictive machine learning model (such as linear regression, SVM, LDA) for a featurizer F, and a loss optimization problem L that will map the records to its feature vector x and label y.
- Gradient function: The gradient function returns the gradient of the loss in every batch iteratively.
- Stopping Criteria: Data are cleaned in batches of size b and the user can alter it (default size 50). Once the batch size drops below b the system stops in next iteration.
- Cleaning Function: A cleaning operation C() can be applied to a record r in the dataset(or a set of records) to recover the clean record i.e. C(r).

The system begins by training the model L on the dirty dataset to find an initial model $\theta^{(d)}$ that the system will iteratively improve. The Sampler selects a sample of size b records from the dataset and passes the sample to the Cleaner, which executes C() on the whole sample and outputs their cleaned versions. The Updater uses the cleaned sample to update the weights of the model, thus moving the model closer to the true cleaned model. Finally, the system terminates due to a stopping condition.

ActiveClean also includes numerous optimizations such as: using the information from the model to inform data cleaning on samples, dirty data detection to avoid sampling clean data, and batching updates.

ActiveClean is evaluated on five real-world datasets UCI Adult, UCI EEG, MNIST, IMDB, and Dollars For Docs with both real and synthetic errors. The results show that our proposed optimizations can improve model accuracy by up-to 2.5x for the same amount of data cleaned [1].

C. BoostClean

BoostClean is a framework that automates the process of repairing and detecting a class of data errors called domain value violations that occur when an attribute value is outside of its domain value. BoostClean automatically selects an ensemble of error detection and repair combinations using statistical boosting. BoostClean selects this ensemble from an extensible library that is pre-populated general detection functions, including a novel detector based on the Word2Vec deep learning model, which detects errors across a diverse set of domains [4].

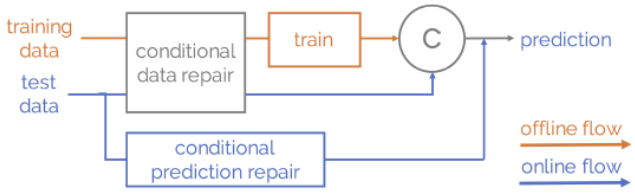

Fig. 4. WorkFlows in BoostClean

The input to BoostClean framework is relational table (Database), a library of detector functions D that generate (possibly incorrect) predicates that match candidate dirty records, a library of repair functions F that transform or delete a record, and a user-specified classifier training procedure train().

1) Detection Generator: The BoostClean has a library that transforms feature extraction functions into detector generators automatically called IsoDetect. IsoDetect uses a approach that automatically performs the latter threshold tuning task so that developers can focus on feature engineering.

The BoostClean implements outlier detection algorithms called isolation forest. The isolation forest creates a large set of isolation trees and classifies records with short average path length as outliers. Generally,the Isolation Forest assumes that the outliers are more easily separable from the rest of the dataset than non-outliers. The algorithm grows a forest of isolation trees, where each tree is randomly grownit selects a random attribute and a random threshold value until a leaf node contains a single record. The length of the path to the leaf node is a measure for the outlierness of the record a shorter path more strongly suggests that the record is an outlier. Isolation Forests have a linear time complexity and very small memory requirements.

Hyper-parameters : The Isolation Forest used in the system has a few hyperparameters that need to be tuned, namely, the maximum branch length and a threshold that determines outlier v.s. not outlier.

False Positive and False Negatives : a predicate learned from data will have false positives and false negatives. If a predicate is too uncertain and applies the repair to many spurious records, then it will reduce classification accuracy. By boosting, we can protect the system against uncertain detectors.

2) Repair Functions: The repair function takes a record as $f_i \in F$ input and modifies the records attributes. BoostClean has two types of repairs: data repairs are applied to the training

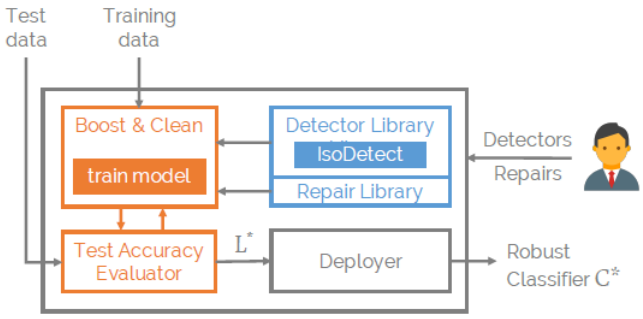

Fig. 5. BoostClean Architecture

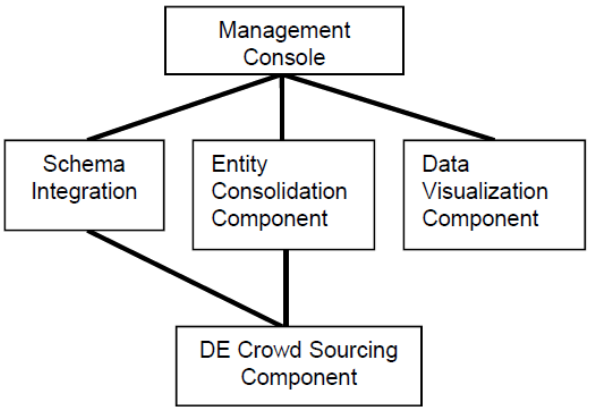

Fig. 6. Data Tamer Architecture

data prior to running the training procedure, while prediction repairs modify the label of the records after the classifier makes a prediction.

- Data repairs modify the values of a training record in response to a detected error (due to a predicate). The repair functions can modify the label, records, attributes, or simply delete the record from the training dataset.
- Prediction repairs, on the other hand, take as input the non-transformed record along with the classifier prediction, and replaces the prediction with a default value.

3) Conditional Repair: BoostClean applies repair functions to specific sets of records through the use of conditional repairs. A conditional repair $l_k = (p_k; f_k)$ is a tuple where $p_k = d_i(X_train; Y_train)$ is the output of a detector generator and $f_k \in F$ is a repair function. A conditional repair is compiled into generation procedure that returns a repair function; the repair function takes as input a possibly cleaned record r, along with its original uncleaned version r_orig.

BoostClean evaluation on a collection of 12 datasets from Kaggle, the UCI repository, realworld data analyses, and production datasets that show that BoostClean can increase absolute prediction accuracy by up to 9% over the best non-ensembled alternatives [4].

D. Data Tamer

Data Tamer is a data curation system. The Data Tamer takes as input a sequence of data sources to add to a composite that is being constructed over time [6]. A new source is subjected to machine learning algorithms to perform attribute identification, grouping of attributes into tables, transformation of incoming data and de-duplication. There are four sub-systems in Tamer:

1) Schema integration: : It takes an attribute, A_i from a data source and compare it to the collection of other attributes pairwise. Data Tamer can perform String comparison using trigram cosine similarity, tokenize and measure TF-IDF cosine similarity between columns or measure minimum description length (MDL) that uses a measure similar to Jaccard similarity to compare two attributes.

2) Entity Consolidation: : It looks for the records that are similar enough to be labelled as duplicates. The de-duplication process is divided into multiple tasks as: Bootstrapping the Training Process To learn deduplication rules from a training set of known duplicates and non-duplicates assuming that duplicate records will have at least one attribute with similar values. Categorization of Records Data Tamer does categorization in two steps. First, obtain a set of representative features that characterize each category. The features are obtained by clustering a sample of the records from the available sources. For the clustering of samples Data Tamer uses centroid-based algorithm such as k-means++ for this task. Second, assigning each record to the closest category (w.r.t. to distance function such as cosine similarity).

Finally, Data Tamer uses Naive Bayes classifier to obtain the probability that a record pair is a duplicate given the similarities between their attributes. The duplicate records are clustered together to output a consolidated dataset.

Data Tamer when tested against a web aggregator on 50 manually labelled sources. The system achieved 100% precison and recall of 98.9% and F_60 score of 0.98 as comapred to 97& precision, 4% recall and F_{60} of 0.04.

E. HoloClean

HoloClean is a "framework for holistic data repairing driven by probabilistic inference" [2]. HoloClean brings together qualitative data repairing, which relies on integrity constraints such as denial constraints, with quantitative data repairing methods, which leverages statistical properties of the input data such as the frequency of a value in a column. HoloClean takes as input a inconsistent dataset and then automatically generates a probabilistic program that performs data repairing. The HoloClean framework has three components:

1) Error Detection: : Goal is to detect cells in dataset D with potentially inaccurate values. This process separates D into noisy and clean cells, denoted D_n and D_c, respectively. HoloClean uses error detection methods, such as denial constraints to detect erroneous cells and methods that rely on external and labeled data called clean data. Denial constraints includes several types of integrity constraints such as functional dependencies, conditional functional dependencies, and metric functional dependencies [9].

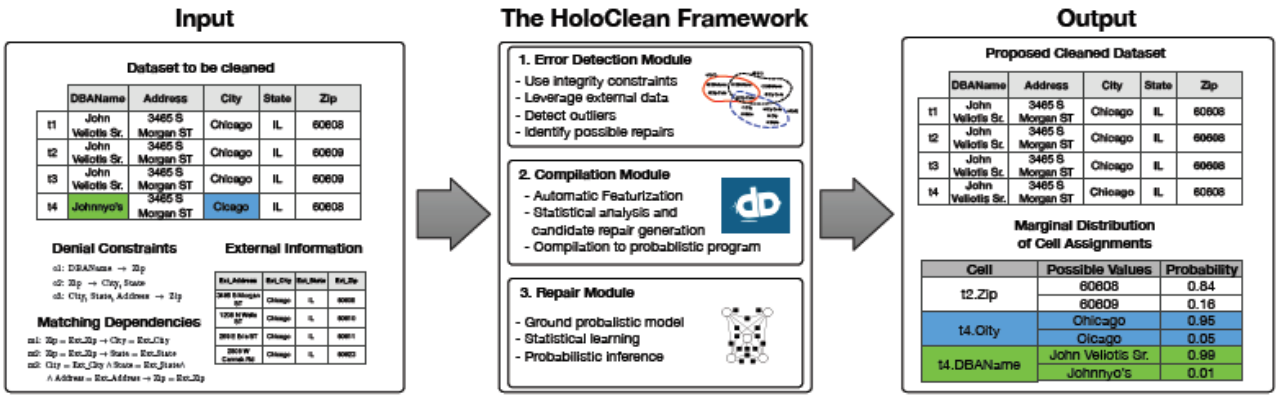

Fig. 7. HoloClean Architecture Overview

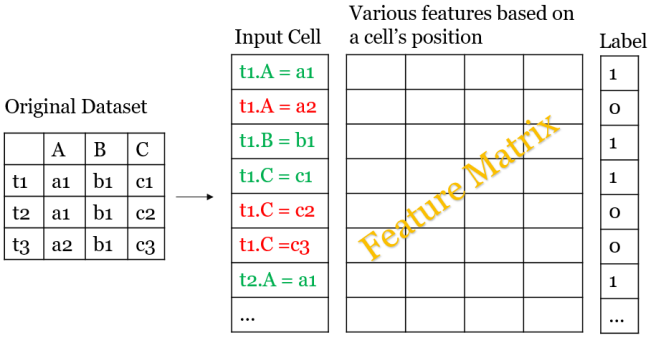

Fig. 8. Original HoloClean Model Workflow

2) Compilation: :HoloClean follows probabilistic semantics to express the uncertainty over the value of noisy cells. HoloClean relies on factor graphs to represent the probability distribution over variables T_c. A factor graph is a hypergraph (T; F; θ) which has a set of nodes that correspond to random variables and F is a set of hyperedges T. Each hyperedge $\phi \in F$; where $\phi \subset T$, is referred to as a factor.

3) Data Repairing: : To repair data D, the system runs statistical learning and inference over the joint distribution of variables and feed into two layer neural-network model to compute the marginal probability for all values $d \in dom(c)$. Variables that correspond to clean cells in D_c are treated as labeled examples to learn the parameters of the model.

Each repair by HoloClean is associated with a calibrated marginal probability. For example, if the proposed repair for a record in the initial dataset has a probability of 0.6 it means that HoloClean is 60% confident about the repair.

The limitations for most of the present machine learning data cleaning techniques are the current requirements of having clean data labels to train the model. It may be difficult to acquire sufficient test labels and may incur extra cost of expert involement to prepare clean test labels.

Finally, there is a need to add support for control and monitoring of data in real-time, since the tool is expected to cleanse the data before it is saved onto a database, or used for triggering action linked to the status of data. Also, the tools should enhance the capabilities from syntactical (structure of the data) to semantic (meaning of data, metadata) cleaning.

HoloClean is able to find the data repairs with an average precision of 90% and an average recall of above 76% across a diverse array of datasets exhibiting different types of errors [2]. The estimated F_60 score framework is for holoclean model is 0.76.

F. HoloClean Course Project

Our project leveraged the domain knowledge of the dataset to sample training data in an effective manner from the Clean Data. In our project we intended to reduce the number of training points considered to train the neural network model which is used to clean d_k cells, this helped us to bring down the memory utilized during the whole process and at the same time also reduce the runtime for the process. For our project we also looked into locality-sensitive hashing, edit distance based clustering and clustering based on column dependencies.

We were motivated to use our knowledge of schema (ref fig.9, fig.11), dimension table (holds attributes that dont vary much) to reduce the numbers of training points considered for training. Moreover, in the HoloClean, we learn that a lot of the time in the cleaning process is spent on generating feature graph and training the model. We wanted to reduce the number of training points used in training the model using sampling technique and also reduce the time taken in cleaning. There were challenges that we faced: Selecting the samples that are representative of the whole dataset, and the cost for selecting a representative sample can sometimes be more than just training the samples.

The present HoloClean model used SQL query that helps the system to decide the cells which are considered for training the model. It filters out the d_k cells from training set, which could not be weak labelled. We updated this SQL query to filter out dimension table duplicates cells from going forward for training, therefore, reducing the number of redundant points in

Dimensions Measures

c_name	c_city	c_contact	p_name	p_company	p_category	s_name	s_city	s_country	quantity	amount
Adam	NE	226-679	Dove	UL	Soap	Lidl	NE	UK	2	40
									6	100
John	NE	485-234	Omo	UL	Wash	Aldi	LO	UK	10	150
									17	300
Mark	BR	824-746	Axe	UL	Deo	Aldi	LO	UK	12	32
Mark	BR	824-746	Dove	UL	Soap	Boots	BR	UK	1	7

Fig. 9. Dimensional Model

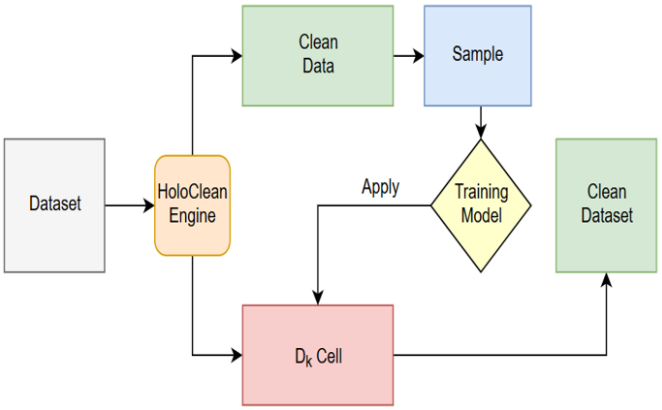

Fig. 10. Updated HoloClean Model Workflow Showing Sampling module

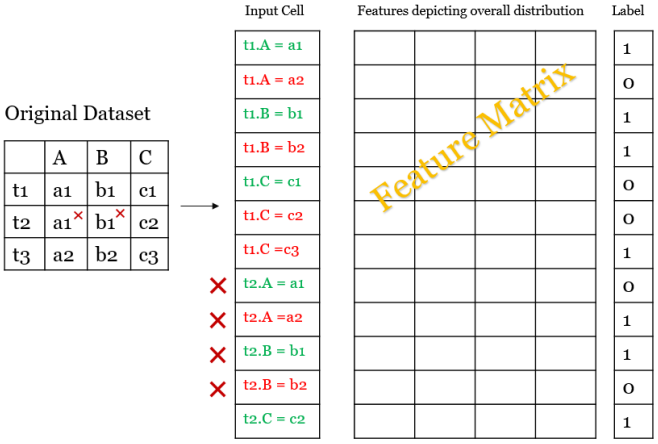

Fig. 11. Updated HoloClean Model Workflow Eliminating Duplicates

the training set. As a result of reduction in number of points considered for training, we observe a drop in the execution time of cleaning process, increase in correct repairs increased with sampling technique, thus increasing the recall.

IV. Application of the Tools

The data cleaning tools can be used in various kinds of datasets.

1) Web data aggregation: aggregator integrates about several thousand URLs, gathering data on events and site data. THe data needs to be semantically and cohesively integration of facts.
2) Health Services Application: the medical records of patient from different branches of the hospital/clinics may need to be put together or even the data collected from medical devices needs to be shared. This requires elimination of duplicate records, correct schema model, and semantically correct data. For instance, the state for City 1 in Arizona has to be correctly identified as AZ when merging data else it should be reported as error.
3) Police records: similar to the hospital records the criminal data is integrated and shared between different regions. This data needs to maintain same schema and has to be semantically correct.
4) Lab experiments: Most of the scientists are using different techniques and collecting experiment-specific data such as concentration and density. Unfortunately, there are no standards for attribute names, no standards for measurement units, and not even a standard for the

language for text (English, French..). There is a need to aggregate data so that it is understood by everyone.

V. Conclusion and future work

In this paper we have discussed the specifications on the databases. we have discussed what the clean data should look like, advantages of having clean data, benefits of a qualtity databases, and the reasons and requirements of data cleaning tools. Further, we took a look at four different machine learning tools namely ActiveClean, BoostClean, Data Tamer, and HoloClean that can be used to address various problem in data cleaning such as de-duplication, outlier detection, rule-based integrity violations etc. We observe that there is no single all purpose tool for the different data sets and all types of errors. The current machine learning models aim to achieve high recall and try to improve precision by allowing human to tune the parameters or to manually suggest the clean data values and retraining the model.

Lastly, We have discussed our approach we used to improve memory utilization and time improvement in HoloClean framework. The approach utilizes the data warehousing technique of splitting the dataset into dimension and fact table; then clustering the duplicates together to reduce the number of training points for the neural network model.

In future, we want to work towards an architecture that uses various data cleaning tools in a pipeline to build a singular solution for data cleaning needs in the industry.

Acknowledgment

We would like to thank Dr. Daniel M. Berry for his continuous support and feedback throughout the project.

References

[1] Krishnan, S., Franklin, M.J., Goldberg, K., Wang, J. and Wu, E., 2016, June. Activeclean: An interactive data cleaning framework for modern machine learning. In Proceedings of the 2016 International Conference on Management of Data (pp. 2117-2120). ACM.

[2] Rekatsinas, T., Chu, X., Ilyas, I.F. and R, C., 2017. Holoclean: Holistic data repairs with probabilistic inference. Proceedings of the VLDB Endowment , 10 (11), pp.1190-1201.

[3] Krishnan, S., Wang, J., Wu, E., Franklin, M.J. and Goldberg, K., 2016. ActiveClean: interactive data cleaning for statistical modeling. Proceedings of the VLDB Endowment, 9(12), pp.948-959.

[4] Krishnan, S., Franklin, M.J., Goldberg, K. and Wu, E., 2017. Boostclean: Automated error detection and repair for machine learning. arXiv preprint arXiv:1711.01299.

[5] Abedjan, Z., Chu, X., Deng, D., Fernandez, R.C., Ilyas, I.F., Ouzzani, M., Papotti, P., Stonebraker, M. and Tang, N., 2016. Detecting data errors: Where are we and what needs to be done?. Proceedings of the VLDB Endowment, 9(12), pp.993-1004.

[6] Stonebraker, M., Bruckner, D., Ilyas, I.F., Beskales, G., Cherniack, M., Zdonik, S.B., Pagan, A. and Xu, S., 2013, January. Data Curation at Scale: The Data Tamer System. In CIDR.

[7] Guyon, I., Matic, N. and Vapnik, V., 1996. Discovering Informative Patterns and Data Cleaning.

[8] Data cleansing, Wikipedia, 01-Aug-2019. [Online]. Available: https://en.wikipedia.org/wiki/Datacleansing. [Accessed: 01-Aug-2019].

[9] 5 Advantages of Data Cleansing, Invensis Technologies, 25-Mar-2019. [Online]. Available:https://www.invensis.net/blog/data-processing/5-adva tages-of-data-cleansing/. [Accessed: 02-Aug-2019].

[10] H. Moreno, The Importance Of Data Quality – Good, Bad Or Ugly, Forbes, 05-Jun-2017. [Online]. Available: https://www.forbes.com/sites/forbesinsights/2017/06/05/the-importance-of-data-quality-good-bad-or-ugly/32a4e1510c4d. [Accessed: 01-Aug-2019].

[11] S. Krishnan, D. Haas, M. J. Franklin, and E. Wu, Towards reliable interactive data cleaning, Proceedings of the Workshop on Human-In-the-Loop Data Analytics - HILDA 16, 2016.

[12] Koronios, Andreas Peter Chanana, Vivek 2007, Examining data cleansing software tools for engineering asset management, USA IGI Publishing