REDS: Estimating the Performance of Error Detection Strategies Based on Dirtiness Profiles

Mohammad Mahdavi
Technische Universität Berlin
mahdavilahijani@tu-berlin.de

Ziawasch Abedjan
Technische Universität Berlin
abedjan@tu-berlin.de

Motivation

- Error detection is the task of finding wrong values
  - E.g., the red values in the table
- There are different error detection strategies
  - A rule violation detector with $Kingdom \rightarrow Lord$ [1]
  - A pattern violation detector with *not-null* [2]
  - An outlier detector [3]
  - …
- Not all the strategies are always accurate [4]

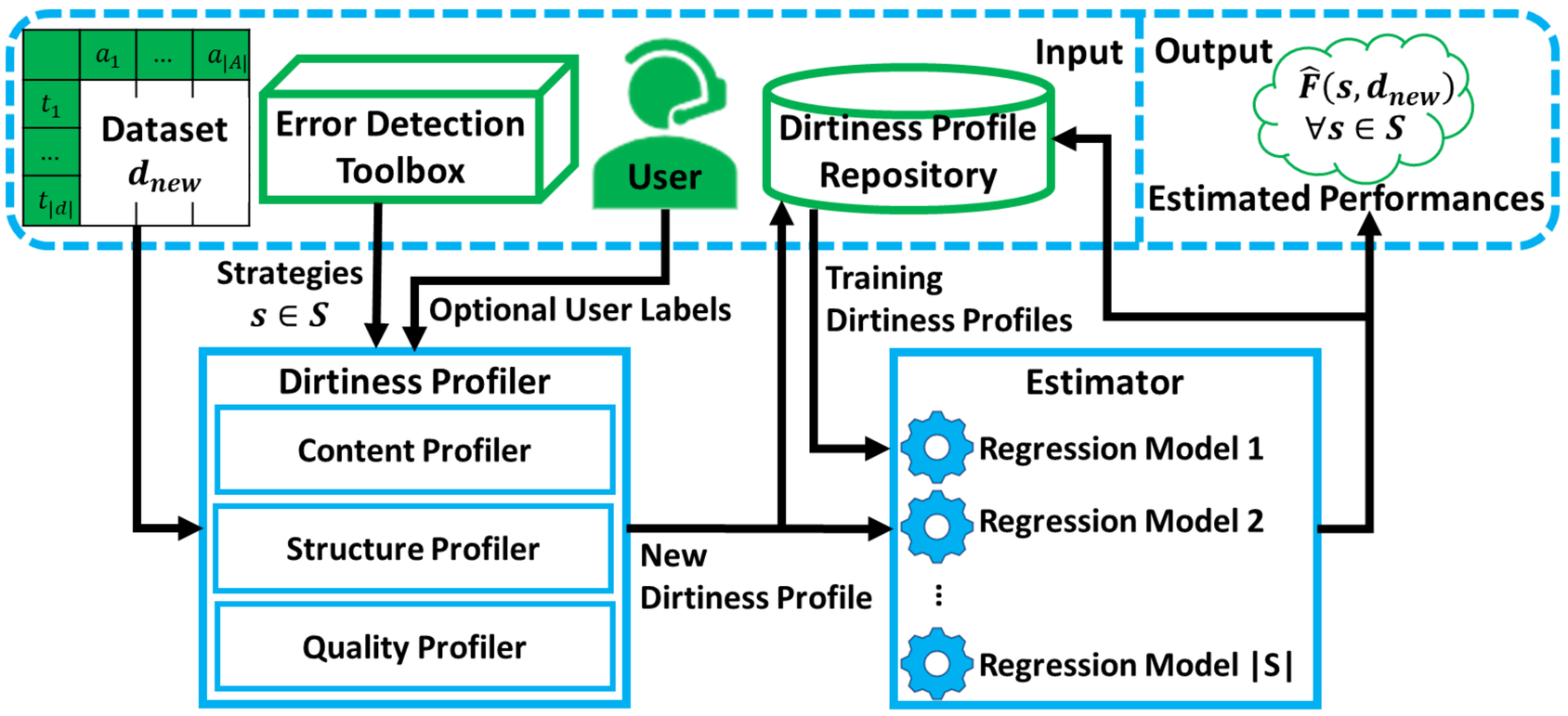
ID	Lord	Kingdom
1	Aragorn	Gondor
2	Gandalf	Ø
3	Saruman	Ø
4	Théoden	Shire

Research Question

- Given a dataset and a set of error detection strategies, how can we estimate the performance of strategies without involving the user to evaluate them?
  - How can we automatically represent the dirtiness of datasets?
  - How can we identify the dirtiness similarity of datasets?
  - How can we leverage the dirtiness similarity of datasets to estimate the performance of strategies on a new dataset?

REDS estimates the performance of error detection strategies without any user labels via representing datasets by their dirtiness profile.

The Workflow of REDS



Dirtiness Profile

- Content features
  - Represent data domain
  - E.g., top keywords
- Structure features
  - Represent data type distribution
  - E.g., the fraction of numerical data values
- Quality features
  - Represent error type distribution
  - E.g., the normalized output size of an outlier detection strategy

Experimental Overview

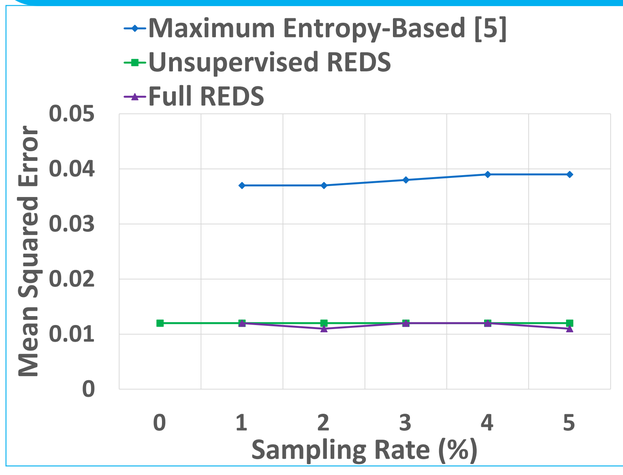
11 Datasets
Hospital, Flights, Rayyan, IT, Beers, Salaries, Address, Movies, Restaurants, Soccer, Tax

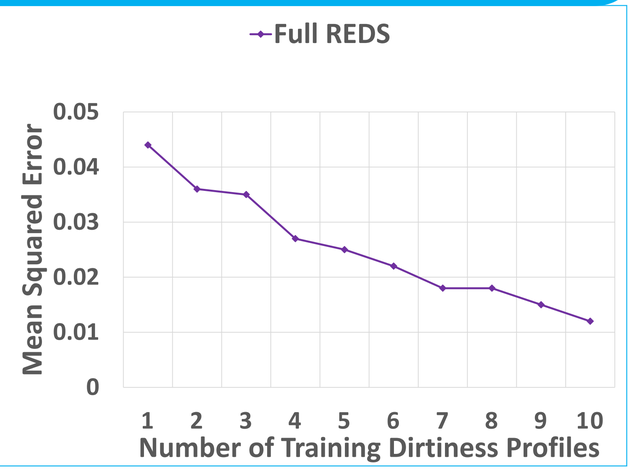
Baseline
Maximum Entropy-Based Approach [5]

Evaluation Measure
Mean Squared Error

4 Experiments
Effectiveness
Features
Regression Model
Repository Size

Experimental Results





References

[1] Michele Dallachiesa et al. 2013. NADEEF: A commodity data cleaning system. SIGMOD, 541–552.
[2] Ruben Verborgh et al. 2013. Using OpenRefine. Packt Publishing Ltd.
[3] Clement Pit-Claudel et al. 2016. Outlier detection in heterogeneous datasets using automatic tuple expansion. Technical Report. CSAIL, MIT.
[4] Mohammad Mahdavi et al. 2019. Raha: A configuration-free error detection system. SIGMOD, 865–882.
[5] Ziawasch Abedjan et al. 2016. Detecting data errors: Where are we and what needs to be done? PVLDB 9, 12, 993–1004.

Source Code

Our prototype is available online: https://github.com/bigdama/reds

Acknowledgement

This project has been supported by the German Research Foundation (DFG) under grant agreement 387872445.

Deutsche Forschungsgemeinschaft