POSTER: Modeling Provenance and Understanding Reproducibility for OpenRefine Data Cleaning Workflows

Timothy McPhillips Lan Li Nikolaus Parulian Bertram Ludäscher
School of Information Sciences, University of Illinois at Urbana-Champaign
{tmcphill,lanl2,nnp2,ludaesch}@illinois.edu

1 INTRODUCTION

Preparation of data sets for analysis is a critical component of research in many disciplines. Recording the steps taken to clean data sets is equally crucial if such research is to be transparent and results reproducible. OpenRefine is a tool for interactively cleaning data sets via a spreadsheet-like interface and for recording the sequence of operations carried out by the user along with the details of all changes made to a data set [VDW13]. OpenRefine uses these records to provide an undo/redo capability that enables a user to revisit the state of the data set at any point in the data cleaning process. OpenRefine additionally allows the user to export sequences of recorded operations as *recipes* that can be applied later to different data sets. However, because such exported recipes do not include edits made manually to individual cells, a recipe in general cannot represent an entire, end-to-end data preparation workflow.

Here we report early results from an investigation into how the operation history recorded by OpenRefine can be used to (1) facilitate reproduction of complete, real-world data cleaning workflows; and (2) support queries and visualizations of the provenance of cleaned data sets.

2 RESEARCH AIMS

The results described here represent initial steps in our efforts to:

- Understand and describe the native data and history model of OpenRefine using the concepts and terminologies of the reproducible-research and provenance communities.
- Discover what provenance queries can be supported by the OpenRefine data model and operation history. Demonstrate queries that reveal key aspects of the provenance of cleaned data sets.
- Extend the YesWorkflow process, data, and provenance models as needed to represent the operations, transformations, data structures, data flows, and data dependencies that characterize data cleaning workflows.
- Employ YesWorkflow to represent end-to-end workflows carried out using OpenRefine so that they can be visualized readily and queried prospectively.
- Identify provenance queries important for achieving research transparency that apparently *cannot* be satisfied using just the information recorded by OpenRefine. Develop means to augment the operation history with additional information needed to support these critical queries.
- Employ computational environments that can be reproduced reliably across multiple computer systems maintained by different researchers. Enable other members of the community independently to repeat our experiments and demonstrations, and to review and reproduce our results on their own computers.

3 RESEARCH TOOLS EMPLOYED

We carry out the research reported here using the following tools. We run OpenRefine version 3.1 in Java 8 runtime environments running on multiple platforms. For automating operations with OpenRefine we access the OpenRefine HTTP API by using and extending the Python client library XXX. We use the YesWorkflow toolkit for modeling data cleaning workflows and representing OpenRefine operation histories in queryable form. We employ XSB Prolog for expressing and performing Datalog-style graph and provenance queries, and GraphViz for rendering visualizations of query results. We use GitHub to share research artifacts between coauthors and with the research community. We depend on Ansible, Vagrant, and Docker for making research environments reproducible across coauthors' computers and for enabling other researchers to repeat our experiments on their own computers. Finally we share preconfigured environments for reproducing our results using resources provided by Whole Tale and MyBinder.

4 RESULTS

We demonstrate that under certain conditions complete data cleaning workflows carried out within OR can be repeated fully automatically in a different instance of OR. We shown that this requires using information that is recorded by OR for its undo/redo feature, but that is not exportable from OR via recipes or its native HTTP API. We show that key queries of the provenance of cleaned data sets, and of particular columns, rows, and cells in the final data set, can be satisfied using the information captured by OR for its undo/redo feature. Will illustrate the usefulness of YesWorkflow-style workflow diagrams for making data cleaning workflows transparent and easy to review, and for rendering portions of the overall workflow to represent the result of provenance queries.

5 FUTURE WORK

REFERENCES

[VDW13] R. Verborgh and M. De Wilde. *Using OpenRefine: The essential OpenRefine guide that takes you from data analysis and error fixing to linking your dataset to the Web.* Community experience distilled. Packt Publishing, Birmingham Mumbai, 2013. OCLC: ocn892971028.