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 [VDW13]. OpenRefine uses its operation history 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. Although OpenRefine records details about every change made to a data set, exported recipes do not include edits made manually to individual cells. Consequently, a recipe in cannot generally 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 for easy review.

2 AIMS

The work described here represents initial steps in our efforts to:

- Understand and describe the native data model and operation history 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 research team members. Enable members of the community to independently repeat our experiments and demonstrations, and to reproduce, review, and evaluate our results on their own computers.

3 TOOLS

We use the following tools. We run OpenRefine version 3.1 [OR18] in Java 8 environments on multiple platforms. We access the OpenRefine HTTP API by using and extending the OpenRefine Python Client Library [ML18]. 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 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 systems. Finally, we share preconfigured computing environments for reproducing our results using resources provided by the Whole Tale and MyBinder projects.

4 RESULTS

Code and other artifacts in the GitHub repository accompanying this poster demonstrate the progress we are making towards transparent and reproducible data cleaning workflows. Manually performed data cleaning workflows later can be repeated automatically in different instances of OpenRefine on the same or different data sets using information gathered by OpenRefine but not easily exported as recipes. Key queries of the provenance of cleaned data sets–and of particular columns, rows, cells, and values in a final data set–also can be satisfied using information captured by OpenRefine to support its builtin undo/redo feature. YesWorkflow-style workflow diagrams make full data cleaning workflows transparent and easy to review. Moreover, it is useful to represent the results of key provenance queries by rendering select portions of the overall workflow.

REFERENCES

[ML18] P. Makepeace and F. Lohmeier. OpenRefine Python Client Library. https://github.com/opencultureconsulting/openrefine-client, 2018.

[OR18] OpenRefine: A free, open source, powerful tool for working with messy data. http://openrefine.org/, 2018.

[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.