IJDC | *Brief Report*

On the Reusability of Data Cleaning Workflows

Lan Li
School of Information Sciences
University of Illinois, Urbana-Champaign

Bertram Ludäscher
School of Information Sciences &
National Center for Supercomputing Applications (NCSA)
University of Illinois, Urbana-Champaign

Abstract

The goal of data cleaning is to make data *fit for purpose*, i.e., to improve data quality, through updates and data transformations, such that downstream analyses can be conducted and lead to trustworthy results. A transparent and reusable data cleaning workflow can save time and effort through automation, and make subsequent data cleaning on new data less error-prone. However, *reusability* of data cleaning workflows has received little to no attention in the research community. We identify some challenges and opportunities for reusing data cleaning workflows. We present a high-level conceptual model to clarify what we mean by reusability and propose ways to improve reusability along different dimensions. We use the opportunity of presenting at IDCC to invite the community to share their uses cases, experiences, and desiderata for the reuse of data cleaning workflows and recipes in order to foster new collaborations and guide future work.

Keywords: data cleaning · workflow automation · reusable workflows

Submitted 15 March 2022~ *Accepted* 28 April 2022

Correspondence should be addressed to Lan Li and Bertram Ludäscher. Email: {lanl2,ludaesch}@illinois.edu

This paper was presented at International Digital Curation Conference IDCC22, online, 13-16 June, 2022

The *International Digital Curation Conference* is an international journal committed to scholarly excellence and dedicated to the advancement of digital curation across a wide range of sectors. The IJDC is published by the University of Edinburgh on behalf of the Digital Curation Centre. ISSN: 1746-8256. URL: http://www.ijdc.net/

https://dx.doi.org/10.2218/ijdc.v17i1.828
DOI: 10.2218/ijdc.v17i1.828

Introduction

The goal of data cleaning is to make data *fit for purpose*, i.e., to improve data quality, through updates and data transformations, such that downstream analyses can be conducted and lead to trustworthy results. A transparent and reusable data cleaning workflow can save time and effort through automation, and make subsequent data cleaning efforts on new data less error-prone (Li et al., 2019). However, *reusability* of data cleaning workflows has received little to no attention in the research community. In the following, we identify some challenges and opportunities for reusing data cleaning workflows. We present a conceptual model to clarify what we mean by reusability and propose ways to improve reusability along different dimensions. Finally, we solicit input from the community to test and validate our conceptual model and prioritize future work and tool development.

What does it mean to reuse a data cleaning workflow?

Consider a data curator or researcher who cleans a "dirty" dataset D, obtaining a new dataset D' with improved data quality (Figure 1). Let us further assume that the workflow W that the user has executed (denoted $D \overset{W}{\rightsquigarrow} D'$) has been captured in the form of a (potentially reusable) *recipe* R, i.e., R contains retrospective and/or prospective *provenance* information that describes how D' was obtained from D while executing W. It then makes sense to say that applying R to D yields D', or $D' = R(D)$ for short.

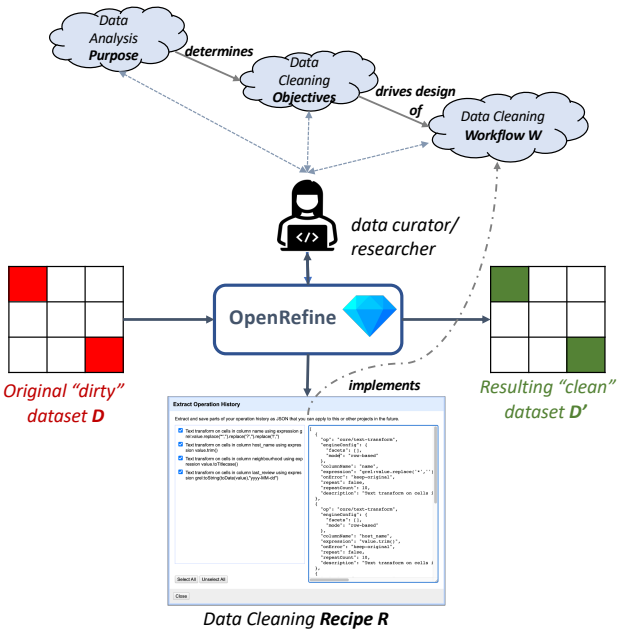

Figure 1. The researcher's *analysis purpose* determines the *data cleaning objectives* to transform the "dirty" dataset D into a "clean" dataset D' that is *fit-for-purpose*. The researcher develops a plan, the data cleaning *workflow* W, which is then executed, yielding D'. A data cleaning tool (here: OpenRefine) may capture a (potentially reusable) *recipe* R as a "by-product" of executing W. The recipe R may be reusable on a new dataset E.

A popular data cleaning tool for which the above assumption[1] is true is OpenRefine (OR, 2021). The recipe R can be obtained by exporting the *operation history* of a previously executed data cleaning workflow W. In the case of OpenRefine, additional provenance information can be harvested from internal *project files* and then used for further analysis of W or to enrich R with *hybrid* provenance information, i.e., combining retrospective and prospective provenance elements (Parulian et al., 2021b).

Definition 1 (Recipe Reuse) Let R ($=R_{D,W}$) be the recipe for the data cleaning workflow W that was used when cleaning dataset D, i.e., with $D \overset{W}{\rightsquigarrow} D'$. We say that recipe R is being *reused* whenever we apply it to a different dataset $E \neq D$, denoted $E' = R(E)$.

This definition is rather straightforward: Reusing a recipe simply means applying it to a new dataset. What could possibly go wrong? A lot, as it turns out.

Challenges when trying to reuse a data cleaning recipe

Let R be the recipe that was created when cleaning D (via some workflow W) to obtain D', and let $E \neq D$ be another dataset. The following are some of the many challenges that may prevent R from being reusable for E:

1. R may not be *safe* for E. For example, if D has a *numeric* type in some column C, but in E that same column has type *string*, then applying arithmetic operations on C is allowed for D, but not for E, resulting in a *type error*. Therefore, the part of R that applies arithmetic operations cannot be reused (directly) for E.
2. R may not be *meaningful* on E for various reasons. For example,
   (a) if schema(E) is very different (or even disjoint) from schema(D), then the operations in R cannot be applied meaningfully to E, since recipe operations are typically defined at the column level, and
   (b) even in the best case, i.e., where schema(E) = schema(D), there may be further problems, e.g.,
      i. if the *semantics* of data in E is different from the one in D, or
      ii. if the *purpose* for analyzing E is different from the one for D.

Example 1 Consider a dataset D about Airbnb listings in a certain region or city[2]. The dataset includes information about hosts, the address and geographic location of rentals, room types, minimum number of stays, price, etc. Let D use a decimal representation for coordinates, e.g., a place might have attributes lat = 55.946944 and long = −3.201667. If R contains numerical operations on these columns, then these can *not* be reused "as is" on a dataset E which represents coordinates in degrees, minutes, and seconds (here, e.g., lat = 55° 56′ 49″ N and long = 3° 12′ 6″ W), even if the schemas are otherwise the same. This is an example for challenge (1) above, since a part of R is *not type safe* for E.

[1] i.e., that an interactive data cleaning workflow W can be captured in the form of a recipe R
[2] e.g., see https://www.kaggle.com/datasets/jinbonnie/chicago-airbnb-open-data

Example 2 A trivial example of challenge (2a) is when D and E have very different schemas and/or semantics. If D is the Airbnb dataset from above and E is a dataset about historical restaurant menus (New York Public Library, 2020), nobody would reasonably expect R to be reusable for E (but individual, generic data cleaning steps might be reusable, e.g., a transformation converting various date formats into ISO-standard form: see below).

Example 3 Now assume that $\mathsf{schema}(E) = \mathsf{schema}(D)$. A more interesting example for challenge (2b) is when the analysis purposes of D and E are different. For example, the purpose of analyzing D may have been to count the available listings *per neighborhood*, so the data cleaning objective was to *standardize the names* in the neighborhood column. In contrast, the purpose for E may be to count the available listings *within a certain radius* from a geographic location, given via lat-long coordinates, so the data cleaning objective for E would be to check and convert (if necessary) the lat-long columns. These different purposes give rise to different data cleaning objectives and thus to different workflows and recipes. In particular, the original R will not be reusable to check and convert coordinates since those columns were not even touched by R in our example.

A simple conceptual model for recipe reuse

The following is a brief description of a simple conceptual model for recipe reuse (cf. Figure 1):

- A researcher or data curator has a *data analysis purpose* P in mind (cf. Example 3).
- Often, we can associate with P one or more questions (or *queries*) Q that the researcher wants to answer using the given dataset D, e.g.,
  - "*How many rentals in this price range are available for this zip code?*"
- From the analysis purpose P (and associated questions/queries Q) we can derive a set of *data cleaning objectives* O: What statements should be true for the cleaned D'?
- In order to achieve these objectives, the user will develop and then execute a *data cleaning workflow* W to obtain the clean(er) dataset D' using a suitable tool such as OpenRefine.
- The tool (or appropriate extensions/companion tools) should allow the recording of *provenance* information, which can be used to derive a *recipe* R (= $R_{D,W}$) that may be reused on different datasets $E \neq D$ in the future.
- Before applying R to E, we need to make sure that it is (type) *safe* and (semantically) *meaningful* to do so. This may require some analysis and comparison of the *schemas* of the original dataset D and the new dataset E for which R is to be reused.
- In some cases, R might be reusable "as is", i.e., directly, without any change to R.
- In many cases, however, we will need to adapt R or decompose it into smaller *modules* (i.e., *subworkflows*) or even *individual operations*, to achieve some level of reusability.

With these conceptual elements in place, we can now refine our notion of reusability:

Definition 2 (Reusability of R ecipes) We say that R (=$R_{D,W}$) is *directly reusable* for a new dataset E, if schema(E) = schema(D) and purpose(E) = purpose(D). Otherwise, we say that R is *possibly reusable with modifications*, i.e., if there are schema changes or changes in the purpose of E relative to the original D that was used when capturing R.

In case of the latter, the problem is now to obtain a modified version R' (or a set of modified subworkflows of R) that can be reused for cleaning E.

Improving the reusability of data cleaning workflows

There is no shortage of technical challenges when trying to reuse a data cleaning workflow W, in the form of an executable recipe R, on a new dataset E. Below we sketch some initial ideas and approaches towards improving the reusability of recipes.

Exploiting the modular structure of recipes

In OpenRefine the individual operations of a recipe E can be analyzed with respect to their column input/output signatures, i.e., an operation can be modeled as a function $f : X_1, \ldots, X_n \rightarrow Y_1, \ldots, Y_k$ that reads values from n input columns $X_1, \ldots, X_n$ and that updates values in k output columns $Y_1, \ldots, Y_k$. Often $n = k = 1$, and $X_1 = Y_1$, i.e., many OpenRefine operations read a single input column X_1 and update the values in that same column (hence the output column $Y_1 = X_1$): e.g., trimwhitespace() is such an operation. By analyzing such dataflow dependencies between operations, the modular structure of a recipe can be revealed (Li et al., 2021; Parulian et al., 2021a). The reusability improvement opportunity then results from the fact that while a recipe R may not be reusable as a whole, some subworkflows may be reusable. We call such reusable subworkflows, i.e., which may be reused in other recipes, data cleaning *modules*. The reusability of modules can be further improved, e.g., by taking *schema mappings* into account, i.e., if a module M_D was part of a recipe $R_{D,W}$, it may be necessary to change it into M_E to take into account the different column names used in E. This assumes that schema matching information (from schema(E) to schema(D)) is available or can be inferred, i.e., we can determine how columns in the new dataset E correspond to the original columns in D.

Generalizing data cleaning operations

Consider two operations o_1 : US_dates $\rightarrow$ ISO_dates and o_2 : EU_dates $\rightarrow$ ISO_dates that match dates of the form MM/DD/YYYY and DD.MM.YYYY, respectively and convert them to ISO-standard form YYYY–MM–DD. If we combine these two operations into a single operation o_3 : US_dates $\cup$ EU_dates $\rightarrow$ ISO_dates, then the domain of o_3 includes the domains of both o_1 and o_2 as subdomains or special cases. Therefore, we can say that o_3 *generalizes* both o_1 and o_2, and we can use this generalization partial order as a proxy for a reusability partial order: $o_1 \preceq o_3$ and $o_2 \preceq o_3$, i.e., o_3 is *more reusable than* o_1 and o_2.

Conclusions

Given the high cost and error-prone nature of data cleaning workflows, it seems desirable to identify reusable parts (*modules*) of data cleaning recipes. We have sketched some of the challenges and opportunities for recipe reuse and now invite the community to share their uses cases, experiences, and desiderata for the reuse of data cleaning workflows and recipes in order to foster new collaborations and to guide future work.

References

Li, L., Ludäscher, B., and Zhang, Q. (2019). Towards more transparent, reproducible, and reusable data cleaning with OpenRefine. *iConference 2019 Proceedings*. http://hdl.handle.net/2142/103330.

Li, L., Parulian, N. N., and Ludäscher, B. (2021). Automatic Module Detection in Data Cleaning Workflows: Enabling Transparency and Recipe Reuse. In *16th International Digital Curation Conference (IDCC)*. https://doi.org/10.2218/ijdc.v16i1.771.

New York Public Library (2020). Whats on the menu? http://menus.nypl.org/data.

OR (2021). OpenRefine: A free, open source, power tool for working with messy data. github.com/OpenRefine.

Parulian, N. N., Li, L., and Ludäscher, B. (2021a). or2yw: Modeling and Visualizing OpenRefine Histories as YesWorkflow Diagrams. In *iConference 2021 Proceedings*. http://hdl.handle.net/2142/109699.

Parulian, N. N., McPhillips, T. M., and Ludäscher, B. (2021b). A model and system for querying provenance from data cleaning workflows. In *Provenance and Annotation of Data and Processes (IPAW)*, volume 12839 of *LNCS*, pages 183–197. Springer. https://doi.org/10.1007/978-3-030-80960-7_11.