CS-513/IS-532 Theory & Practice of Data Cleaning
Dr. Bertram Ludäscher

Final Project: End-to-End Data Cleaning Workflow

The goal of the final project is to use various tools and techniques covered in this course together in a small end-to-end data cleaning workflow.

Forming Groups. You are encouraged to form teams of 2 or 3 students, but the project can also be conducted individually. You may want to use one of the class communication forums (Moodle, Piazza, Slack) to find teammates. For the options below, **individuals** would typically choose option (a) (to leverage prior work), while **groups** of 2 or 3 students should make use of option (b) (or option (c), when applicable). Finally, **groups of 3** should also try and tackle at least one of the optional parts of the project.

Start by reviewing the web sites (a) and (b) for which datasets to be cleaned exist, or think about a dataset of your own choosing (c):

- (a) US Farmers Markets
  ⇒ `https://www.ams.usda.gov/local-food-directories/farmersmarkets`
- (b) New York Public Library's crowd-sourced historical menus
  ⇒ `http://menus.nypl.org/`, or
- (c) Bring a dataset of your own that you'd like to work with (requires approval: see below)

Reference versions for (a) and (b) will be linked from the course pages. If you choose your own dataset via option (c), use a dataset that is either publically available (preferred) or one that you are allowed to share with your teammates and the instructor and TAs. In that case, email the instructor (`ludaesch@illinois.edu`, Subject: "Final Project Dataset") and the TAs, describing key information about the dataset: URL (if available), schema and contents, size, and expected data cleaning challenges. The instructor or TAs will then approve your project option (c) as appropriate.

The recommended overall data cleaning workflow for the project should include the following parts:

1. **Overview and Initial Assessment** of the dataset. Here you should describe the *structure* (i.e., schema) and *content* of the dataset and *quality issues* that are apparent from an initial inspection. You should also describe a (hypothetical or real) *use case* of the dataset and derive from it some *data cleaning goals* that can achieve the desired *fitness for use* of this dataset. Also answer these questions: Are there use cases for which the dataset is *already* "clean enough"? Are there related use cases for which the dataset will *never* be good enough? You can speculate a bit here – but the rest of the project should focus on a "middle of the road" use case that requires a practically feasible amount of data cleaning. All of these points and answers should be written up in *narrative* form.

2. **Data Cleaning with OpenRefine**. In this first hands-on part of the project, you should use OpenRefine to clean the chosen dataset—either (a) or (b) or your own (c)—as much as needed for your chose use case or data cleaning goals. Document the result of this phase, both in *narrative* form and with *supplemental* information (e.g., which columns were cleaned and what changes were made?). Can you quantify the results of your efforts? Also provide *provenance* information from OpenRefine. Pay close attention to what OpenRefine includes and does *not* include in its *Operation History*! If important information is missing in the latter, provide that information in other ways, e.g., explaining things in the narrative.

3. (***Optional***) **OpenRefine Alternative**. If you find that certain steps are not well suited for OpenRefine (e.g., due to scalability or other issues), consider applying an alternative solution, e.g., using Python, R, or another tool such as Trifacta Data Wrangler, Tableau, etc. Document your choice and provide the corresponding artifacts as in Part 2 for OpenRefine (i.e., *narrative* and *supplemental* files).

4. **Develop a Relational Database Schema** for your dataset. What logical *integrity constraints* (ICs) can you identify? Load the data into a SQLite database with your target schema. Use *SQL queries* to *profile* the dataset and to *check* the ICs that you have identified! Think of the queries as *denials* that retrieve those rows that can be used as "witnesses" of the IC violations.

5. **Create a Workflow Model** of your data cleaning workflow: What are the key inputs and outputs of your overall workflow? What are the dependencies? Note: You may want to model the various steps you have executed with OpenRefine as parts of a *subworkflow*. This way, the YesWorkflow (YW) model more clearly describes what actually happened to what parts of the data. Create a visual version of your workflow using the YW tool. (If you are conducting the project as an *individual*, i.e., not in a group, and run into major problems with YW, you can use another graphing tool to render the workflow.)

6. (***Optional***) **Provenance Queries**. Develop provenance queries (in Datalog / DLV) that show on which inputs and intermediate data and steps the outputs of your workflow depend. Such a query is similar to a query on a family tree that returns all ancestors of a given person (cf. provenance lectures, assignments).

What to Submit: Specific Deliverables

The above parts often require you to write a *narrative* that describes and explains what you have done and why. Be as precise and to the point as you can! Use screenshots as needed to clarify a point, explanation, or observation. The combined narrative for all the executed parts forms the **Project Report**. You can create it with any word processor (Word, Libre/OpenOffice, LaTeX, etc.), but please submit only a **single PDF** file (for easier handling, grading, and annotation). There is no fixed minimal length for the project report, but for each executed part above, you probably will need at least 0.5–1 page for the narrative. So if you have executed say 5 of the 6 parts (the mandatory four, plus one optional), your report probably has at least 5 pages of narrative text (*not* counting screenshots – *with* screenshots your report can quickly grow to 10–20 pages).

Supplemental Files. In addition to the narratives (which are submitted as a single PDF report), several of the above parts may require the submission of *supplemental* files:

1. No supplemental files are needed.

2. As supplemental information, include (1) a copy of the OpenRefine *Operation History* (copy-paste it into a file `OR-history.json`) and (2) *export* your OpenRefine project and submit the zipped project file.

3. If you use an OpenRefine alternative, answer the analogous questions as for OpenRefine (in the narrative) and submit analogous history files or other provenance information (as available for that tool).

4. In the narrative, explain (and show) how you go the data into SQLite (e.g., directly from the SQLite prompt, via a script, or a GUI), and explain the ICs and queries. As supplemental information submit SQL files (and load scripts, when available).

5. When using YesWorkflow, submit (1) the file that has the annotations (e.g., `myworkflow.txt`), (2) the Graphviz/DOT file generated (e.g., `myworkflow.gv`), and (3) a screenshot of the rendered workflow (this screenshot/figure should be directly included in the narrative part).

6. If you develop the optional provenance queries using Datalog/DLV, then submit those rule files (`myrules.dlv`) and an *execution log*, i.e., a text file `myrunlog.txt` that shows the results of running your rules in DLV (you can also include a screenshot in the narrative).

⇒ Put the project report and all the supplemental files in a **single folder** named *Your-Lastname-Projectname*; **zip** this folder; and submit a **single zipped file** with everything in it. ⇐