CS 513 Theory & Practice of Data Cleaning

Final Project Report: What's On The Menu?

Artsiom Strok  
NetID: astrok2   
[astrok2@illinois.edu](mailto:astrok2@illinois.edu)  
Ramin Melikov  
NetID: melikov2  
[melikov2@illinois.edu](mailto:melikov2@illinois.edu)Jonah Willis  
NetID: jonahww2  
[jonahww2@illinois.edu](mailto:jonahww2@illinois.edu)

*Abstract*—This electronic document is a “live” template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. *\*CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract*.

Keywords—NYPL, menus, data cleansing, OpenRefine, YesWorkflow, provenance.

# Overview and initial assessment of the dataset

The New York Public Library is digitizing and transcribing its collection of historical menus. The collection includes about 45,000 menus from the 1840s to the present, and the goal of the digitization project is to transcribe each page of each menu, creating an enormous database of dishes, prices, locations, and so on. As of early July, 2020, the transcribed database contains 1,334,431 dishes from 17,545 menus.

This dataset is split into four files to minimize the amount of redundant information contained in each (and thus, the size of each file). The four data files are *Menu, MenuPage, MenuItem*, and *Dish*. These four files are described briefly here, and in detail in their individual file descriptions below.

## Menu

The core element of the dataset. Each Menu has a unique identifier and associated data, including data on the venue and/or event that the menu was created for; the location that the menu was used; the currency in use on the menu; and various other fields.

* id
* name
* sponsor
* event
* venue
* place
* physical\_description
* occasion
* notes
* call\_number
* keywords
* language
* date
* location
* location\_type
* currency
* currency\_symbol
* status
* page\_count
* dish\_count

Each menu is associated with some number of MenuPage values.

## MenuPage

Each MenuPage refers to the Menu it comes from, via the menu\_id variable (corresponding to \_Menu\_id). Each MenuPage also has a unique identifier of its own. Associated MenuPage data includes the page number of this MenuPage, an identifier for the scanned image of the page, and the dimensions of the page.

* id
* menu\_id
* page\_number
* image\_id
* full\_height
* full\_width
* uuid

Each MenuPage is associated with some number of MenuItem values.

## MenuItem

Each MenuItem refers to both the MenuPage it is found on -- via the menupageid variable -- and the Dish that it represents -- via the dish\_id variable. Each MenuItem also has a unique identifier of its own. Other associated data includes the price of the item and the dates when the item was created or modified in the database.

* id
* menu\_page\_id
* price
* high\_price
* dish\_id
* created\_at
* updated\_at
* xpos
* ypos

## Dish

A Dish is a broad category that covers some number of MenuItems. Each dish has a unique id, to which it is referred by its affiliated MenuItems. Each dish also has a name, a description, a number of menus it appears on, and both date and price ranges.

* id
* name
* description
* menus\_appeared
* times\_appeared
* first\_appeared
* last\_appeared
* lowest\_price
* highest\_price

<http://menus.nypl.org/about>

<https://www.kaggle.com/nypl/whats-on-the-menu>

You should describe the structure 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. In addition, you should answer the following questions: Are there use cases for which the dataset is already clean enough? Are there use cases for which the dataset will not be clean 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.

# 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 the use case. Document the process and result of this phase, both in narrative form along with supplementary 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 narrative form.

# Data cleaning with other tools

If you find that certain data cleaning steps are not well suited for OpenRefine (e.g. due to scalability or other issues), consider using an alternative, more suitable solution, e.g., Python, R, or other tools such as Trifacta Data Wrangler, Tableau, etc. Document your choice and answer the same questions as in Step 2.

# Developing a relational schema

Develop a relational 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! You can also use other query languages such as Datalog to profile the dataset and check the ICs, but you should not use a procedural language such as Python, R, etc.

A screenshot of a computer

Description automatically generated

# Creating a workflow model

Create a workflow model of your overall data cleaning workflow: What are the key inputs and outputs of your workflow? What are the dependencies? Note: Here you may want to model the various steps you have executed with OpenRefine as parts of the workflow. This way, the workflow model more clearly describes what actually happened to what parts of the data. Create a visual representation of your overall workflow using YesWorkflow or other diagramming tools. Supplementary material to help with YesWorkflow will be posted on Piazza. Also create a visual representation of your OpenRefine workflow using OR2YWTool (https://pypi.org/project/or2ywtool) or other appropriate tools. The OR2YWTool provides an auto-parsing method from Openrefine Operation History JSON file to YesWorkflow model (developed by Lan Li and Nikolaus Nova Parulian). Please include both overall workflow and OpenRefine workflow in your project report.

# Developing provenance

Develop provenance queries (in Datalog / DLV) that show on which inputs and intermediate data and steps the outputs of your workflow depend (cf. Provenance Assignment).

# Contribution

Develop provenance queries (in Datalog / DLV) that show on which inputs and intermediate data and steps the outputs of your workflow depend (cf. Provenance Assignment).