DSL Name
DSL Report Card

Jingyuan Xu
School of EECS
Oregon State University

April 4, 2016

1 Introduction

A DSL language in data cleaning DSL, which can help programmers save their time in data preprocessing steps. Currently, There are some tools can be utilised for data cleaning. Such like Google Refine; also, some programming language can be used in this area, such like R and Python. However, there are some limitations when users use ready- made tools/languages.
Tools: For example in Google refine, users only have the opinions which already build-in the tool. If the wrong data type is unusual, the suggestions may hard to understand. Even users can follow the suggestions, and they also need to edit the error data manually. If this happens to a large dataset, users will lose the confidence with the data already cleansed, and worry about their analysis result.
Programming Language: R and Python have libraries for data analyzing. But first users need to make sure the packages are well written and provide methods for their host language. For example, R has dplyr package. But dplyr can not provide methods for R summary functions (e.g. mean(), or sum()) [1]. Also, users need to think about package encapsulation. If users want to add a feature to deal with a new wrong data type, they may face problems.

2 Users

The DSL can be used by programmers who knows Haskell and have a general idea in data cleaning. The primary usage for the DSL is data cleaning. If users not familiar with data cleaning, or they need visual control in the clean processing, they may need other tools to help them.

3 Outcomes

The DSL may help a programmer finish these steps in his data preprocessing:
He can use DSL to deal with the messy dataset with different rules (filter, rename, distinct). These various rules can random combine each other. For example, the programmer wants to remove the repeating data lines, and check the specifical data is in the reasonable range (e.g. age cant be a negative number), or

spelling errors, etc. Finally, he can output the cleaning result as a standard data file, or connect to a database, then show in the tables directly.

4 Use Cases / Scenarios

Incomplete data
1. check the data pair match or not
e.g. address and zip code match each other or not
2. null value exist:
e.g. the data show like NULL
3. typo data

Repeating data
1. Data is not in reasonable range
e.g. age is a negative number 2. Different naming conventions
e.g. zip code and postcode have the same meaning

How to fix:
1. Estimation: use statistic method to update the wrong data
2. Casewise deletion: delete the bad data record
3. Pairwise deletion: If a data pair happens issues, and it is not useful for data analysis, then remove the data pair.
4. Simple replacement policy: e.g. use zip code instead of Postal Code, postcode

References

[1] https://cran.rstudio.com/web/packages/dplyr/vignettes/introduction.html