The Design and Implementation of a Cleaning System Prototype

He Yang1, *, Weiwei Liu1, Xiaohui Wang1, He Liu1, Bin Yu2, Hongwei Zhou3

1Artificial Intelligence on Electric Power System State Grid Corporation Joint Laboratory, Global Energy Interconnection Research Institute co.Ltd, Beijing 102209, China
2Beijing Key Laboratory on Integration and Analysis of Large-scale Stream Data, College of Computer Science, North China University of Technology, Beijing 100144, China
3State Grid Jiangsu Economic Research Institute, Jiangsu 210000, China

*Corresponding author email: yanghe@geiri.sgcc.com.cn

Abstract. As we all know, data is one of the most valuable assets, however, raw data is often problematic, not conducive to the training of algorithm models. To cope with this, we can process the dirty data with cleaning systems [1] to obtain standard clean data for data statistics, data mining and other use. Instead of manually modifying data, writing SQLs or other cumbersome methods which are popular present ways to clean data, the article proposes an approach by making use of the Hadoop big data platform to support massive data and support the cleaning of multiple heterogeneous data sources. Moreover, our system prototype supports custom rules and algorithms, can export results to a specified database, greatly simplifying the workload of data cleaning personnel. Based on the system design and theoretical verification presented in this paper, the author implemented a big data cleaning tool based on big data platform. The typical data cleaning process shows that the data cleaning can be achieved and user operations can be simplified on the basis of the theory proposed in this paper.

1. Introduction

Data is a valuable asset. We need data to accurately locate, analyze, improve efficiency, and optimize operations. However, many data are not completely accurate or irregular. This affects the data mining and analysis, which may cause waste of resources and miscalculation.

There are four main methods for data cleaning (you can get related technologies from here [2]).

1) Manual implementation
   Manual inspection by data cleaning personnel takes time and effort, and is inefficient. When the amount of data is large, manual operations cannot meet data cleaning requirements.

2) Writing an Application or SQL Script
   The program or SQL that is written can only target specific rules for specific dirty data. It is inflexible. If you need to change the cleaning rules, you need to change the program or script. The workload is large.

3) Methods to solve problems in specific areas
   For example, address standardization and so on.
(4) Methods unrelated to specific areas
This mainly refers to removing duplicate records.

In order to implement a universal data cleaning tool, this paper designed a data cleaning tool based on big data platform, which can customize rules (There are similar methods [3]) and algorithms. The big data platform[4][5] solved the problem that method 1 couldn't handle large amount of data; rules automatically generated SQL, Corresponding to method 2; algorithm can be customized or built-in algorithm can be used, corresponding to method 3; and we specifically open up a type for cleaning duplicate records, corresponding method 4.

2. Design of the data cleaning tool
As shown above, this article has designed a universal data cleaning tool entirely by ourselves (There are similar researches [6] [7] [8] [9]). This chapter first introduces the framework design of the tool, and then introduces the rules, algorithms, and the removal of duplicate data.

2.1. Data cleaning tool frame design
Cleaning tools are based on big data platforms. The main modules are as follows.

1) Unified Login and Rights Management Module
The system is pre-configured with five types of roles: system administrator, app auditor, app operator, checking user, and system auditor. You can add or remove roles as needed.

2) Data source management module
Users can create, update, delete data source in this module, supporting txt, csv, excel and other types of files, as well as MySQL, oracle and other databases.

3) The task module
This module is the core of all functions. In this module, we set the cleaning task, select the cleaned data source, choose the target exporting database of the result table, configure the cleaning rules, algorithms or delete the duplicate data, then select the application rule, the column name to be handled and the target column of the rule or algorithm, and finally complete the creating, after created tasks can be run in the task schedule module, we can view the report of a successful job in the cleaning report module.

Figure 1. Data cleaning tool architecture
2.2. Rules design of data cleaning tool
You can select rule groups and create new rules in rules management module. The rules created are public rules. The rules are divided into three categories:

1) Precise match
   By associating the dimension table with the fact table, we replace the field in the fact table with the content in the dimension table. For example, in the power data, replace 1 with "overground indoor", 2 with "overground outdoor" and 4 with "underground indoor";

2) Fuzzy Matching
   Regular or ordinary characters can be replaced, we replace the contents of the column with a specified format, and this type of rules can be nested;

3) Expression
   Expressions are a more flexible form. You can specify the expressions supported by Hive. You can choose to add a new column or overwrite the value of an existing column.

2.3. Algorithm design of data cleaning tool
Algorithms and rules are different, you need to use algorithms or models which has been written to deal with data in the database or text-type files, so when processing we first export the table data from HDFS (Hive's table data is stored in HDFS) to the local file system, then in the local file system the table data is processed according to the set parameters. For some parameters, it can be directly set in the data cleaning tool. For some complicated parameters, it needs customized development. After the processing, the result file is put into the HDFS and then imported into Hive according to the delimiters. Finally, if needed you can choose to export the data in Hive to the target database.

2.4. Duplicate data removal design of data cleaning tool
We directly query the selected column of the table, reserve one of the duplicate selected columns, and remove redundant data.

3. Implementation of the data cleaning tool
Based on the design of a universal data cleaning tool, this chapter will introduce the framework implementation of the tool. In the instance system, the task has three types: processing by line, processing by file, and deleting duplicate data. These three task types correspond to the rules, algorithms, and deletion of duplicate data, all of them will be described later in this chapter.

3.1. Implementation of data cleaning tool framework
This section will describe in detail the implementation of the three modules of the data cleaning tool.

1) Unified login and rights management module
   The role authority is refined when implemented: system administrator has rights of user management, organization management, role management, security parameter configuration, app auditor has the rights to audit data source and task, app operator has rights of data source management, rule management, algorithm management, task management, task scheduling management, cleaning report management, checking user has rights of user status monitoring, login statistical analysis, access statistical analysis, system logs, app logs, system auditor has rights to audit user and role.
   The rights of a role can be reduced, and a newly created role’s right cannot cover other roles’ rights. That is, the rights of each role cannot exceed the built-in roles. For example, the app operator cannot get the right of data source audit. This can avoid excessive role rights.

2) Data source management module
   When it is implemented, after a new data source is created, the new data source can be automatically imported into Hive. The cleaning result can be automatically exported to the specified data source. The import and export process is generally performed through Sqoop [10]. The user only needs to configure the data source, the program will automatically generate Sqoop execution statements for import and export, with regard to import and export which cannot be completed using Sqoop, such as webligic JNDI
data source, our tool also supports import and export by handling the import and export by the program itself.

3) Task module
When it is implemented, the data source that has been newly created and got approval in the data source management is first selected in this module, and then the destination data source is selected (After the execution is completed, the result is automatically exported from the big data platform to the target data source. The exported table name is ‘dc_’ plus target table name, for example if we set the target table named ‘project’, the actual exported table will be ‘dc_project’), and later we select and configure rules or algorithms (you can also create new rules at the same time), if it is a rule we are required to select the column to be applied and input the column name of the result. If it is an algorithm we are needed to configure the relevant parameters of the algorithm, after the configuration is complete, click finish to create a new task. After the task configuration is completed and the task gets approval in the task management, the task scheduling management module can be used for timing or manual execution. The successful tasks’ cleaning report can be viewed in the cleaning report module. The report will calculate statistics on the cleaning result, including the task ID, task name, whether it uses algorithm or rule, rule or algorithm ID, the total number of records in the original table, the number of records be processed during the cleaning and the execution time.

3.2. Implementation of data cleaning tool rules
The three types of rules are implemented in different ways, precise matching works through the join with dimension table, fuzzy matching works through the function ‘regexp_replace’, which can be nested, our tool automatically nests multiple ‘regexp_replace’ functions, the expressions work flexibly through Hive expression to get results.

The rules belongs to the task type of processed by line, and the steps when creating a new line-by-line task are as follows;
1) Select the data source and the target data source, if the rule contains an exact match, select the data source to be joined;
2) Select the rule group, select the field processed by the rule and input the name of target field, and then we can add a single rule or add the corresponding rules in batches. At this time, the system will apply the selected rule to the selected fields in batches to generate a new column. The new column’s name is inputted target field name, if the column name is duplicated with the existing column name, the new column will eventually overwrite the old column and be stored in the result table. The detailed generation process is as follows:
   Generate an intermediate process table(rules result table), in which all the columns name will be added with the table name plus ‘_’ as the new column name, for example, the data field of the table ‘a’ is named ‘a_data’ in the intermediate process table to avoid repetition of original column names with target column names. The intermediate result table contains all the converted column names and target field columns. After the intermediate process table is generated, the final result table is generated. We will remove the table name plus ‘_’ to restore the original column name as the final result table column name. For example, in the above example, if there is no target column named ‘data’, the value of ‘a_data’ will be the value of the column ‘data’. If there is a target column named ‘data’, the final result table will not retain the value of ‘a_data’, and the value of the target column ‘data’ is directly used as the result of the column ‘data’ of the table.
3) Finally click ‘Finish creating task’ to create the task.

3.3. Implementation of data cleaning tool algorithms
Two algorithms (or being called as models) are built into the tool during implementation:
1) address normalization\[11\]
   The program was written in java, which can parse out address element categories for inputted Chinese addresses, complement, deduplicate, and correct unqualified addresses to achieve the purpose of address normalization. Data cleaning tools are also written in java. The machine already has a jdk environment.
After the data exported from the big data platform is submitted to the model, the as_id (address id), as province (address of the province to which the address belongs), and the as address (original address) are located in the data. The model will automatically generate the target file, and then import the file to the big data platform.

2) Data matrix fill model

In the power data, electricity meter records electricity consumption every 15 minutes, every 30 minutes, or every 60 minutes. Correspondingly, 96 points, 48 points, or 24 points of data are generated on a single day. However, these data may be empty at times, affecting the analysis of the data. There are three algorithms integrated in the python program, you need to preinstall the Python environment and the corresponding library on the server, the three algorithms are improved alternating least square (IALS), kalman smoother recovery machine (KARM), cluster-based best match scanning (CBMS). Choose one algorithm, specify the starting column number and ending column number of the 96-point data. This algorithm will interpolate values to fields whose value is null, by the way, IALS and CBMS can also evaluate the quality of non-null anomalies, discover their quality problems, and be used as models for data evaluation.

3.4. Implementation of data cleaning tool deduplication

When this function is performed, the selected column is directly ‘group by’ and a column of ‘count (1)’ is added to record the number of merged records. The intermediate result table is not generated, the target table is directly generated, and after execution the target table is exported from the Hive to the target database.

4. Application of the data cleaning tool

4.1. Application status

At present, the tool has passed the functional test of China Electric Power Research Institute and the third-class security test. It has been successfully launched at the headquarters of the State Grid Corporation of China and has been piloted in the Shandong Transportation Supervision Center, obtaining a pilot report.

4.2. Tool screenshot

Figure 2. Shows the view of app operators. Different roles would see different major modules.
Figure 2. View of app operators

5. Conclusion

5.1. Summary to the full text
This article introduced the application scenario of cleaning tools, proposed the problems of the existing solutions, and then proposed a data cleaning tool design method and implementation method, the new method is based on big data platform, supports custom rules, custom algorithms and delete repeated data, the realization of the prototype of the cleaning tool have achieved good response in the application of big data cleaning in the State Grid Corporation of China, indicating that it can meet the design requirements.

5.2. Future work
The implementation of the data cleaning tool proposed in this paper has been considered to be an effective method, but it inevitably has some areas that can continue to improve, specifically as follows:

1) Deduplication only supports the case where all the columns that need to be output are duplicated. In the future, we will try to implement a ‘group by’ with some fields, and the tool will select one of the other fields as the outputs. This implementation can use ‘collect set (non-group by field)[0]’ to meet the demand;

2) The integrated algorithms and models are not rich enough, and more algorithms and models will be added in the future;

3) The currently supported databases are MySQL and oracle. In the future, more data sources will be adapted.

Acknowledgments
The work was supported by State Grid Corporation of China Headquarters Project.
References

[1] Rahm E, Do H. H. Data Cleaning: Problems and Current Approaches [J]. IEEE Data Engineering Bulletin, 2000, 23(4): 3 - 13.

[2] Heiko Muller, Johann-Christoph Freytag. Problems, Methods, and Challenges in Comprehensive Data Cleansing [M]. Technical Report. HUB-IB-164, Humboldt-Universitat zu Berlin, Institute fur Informatik, 2003.13 ~ 23.

[3] Duncan K, Wells D. A rule based data cleansing [J]. Journal of Data Warehouse.

[4] Shvachko K, Kuang H, Radia S, et al. The hadoop distributed file system. Mass Storage Systems and Technologies (MSST). 2010 IEEE 26th Symposium on 2010, p. 1 - 10.

[5] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters. Communications of the ACM 2008. Vol. 51, p. 107 - 113.

[6] Quass, D.: A Framework for Research in Data Cleaning. Unpublished Manuscript. Brigham Young Univ., 1999.

[7] Raman, V.; Hellerstein, J.M.: Potter’s Wheel: An Interactive Framework for Data Cleaning. Working Paper, 1999.

[8] Lee, M.L.; Lu, H.; Ling, T.W.; Ko, Y.T.: Cleansing Data for Mining and Warehousing. Proc. 10th DEXA, 1999.

[9] Hernandez, M.A.; Stolfo, S.J.: Real-World Data is Dirty: Data Cleansing and the Merge/Purge Problem. Data Mining and Knowledge Discovery 2 (1): 9 - 37, 1998.

[10] Sqoop User Guide: http://sqoop.apache.org/docs/1.4.6/SqoopUserGuide.html.

[11] Borkar, V., Deshmukh, K., Sarawagi, S. Automatically extracting structure from free text addresses. IEEE Data Engineering Bulletin, 2000, 23 (4): 27 ~ 32.