Automatic Field Identification for Tabular Data without Headers

Venkat Giridhar Mareedu[1] Keerthi Kant Raghunathan[2]
[1]MVSR Engineering College [2]Osmania University India

Abstract— finding useful patterns from large sets of data has brought new possibilities and insights which are unnoticeable to the human eye. But this is all possible only when we know what data it is i.e., when we have information about the metadata of the database under consideration. This paper aims to automatically identify table headers and their datatypes without any human intervention. Upon identification, this paper also implements a strategy to automatically import the data into a big data tool, SAS, to conduct further data analytics. These strategies are perfectly consistent even with a table with headers as it saves the time of the user who would otherwise manually map the fields.

Key words: Big Data, Data Mining, SAS, Artificial Intelligence, Clustering

I. INTRODUCTION

With the fast evolving world, the amount of data that is generated every day is increasing exponentially. This has generated an urge to process data in real-time. Real-time data processing enables for faster analysis and responses which could have sizeable impact on the operation in hand. Methods and various strategies have been proposed to automate the task of importing the data and performing the requisite analysis. Though largely successful, there are issues which have prevented the system from being a full automated one.

Any hindrance or presence of abnormalities in handling large chunks of data can create potential bottle necks. One such problem largely persistent is that of the unavailability of metadata information of a table. Metadata of a data is the information about the data. The no. of fields, the information that those fields contain, the types of values the fields contain are some of the important information that a metadata of a table contains. Generally when such data is not available, human intervention is required to map the fields and import the data so that the analysis can proceed further. With the use of several data mining techniques, this paper aims to automate this step of manually mapping the fields and importing the data.

Initially, the data to be imported is sampled in a random fashion and a subset is created. This subset is then processed each column at a time by finding the nearest column to it from a pre-defined dataset which has properly defined clusters wherein a cluster corresponds to a respective column. Ideally, by the end of the process, each column to be mapped would be a part of a single cluster. If not, the process is then repeated by picking another larger subset from the data to be imported. This paper mainly focuses on identification of columns from a student database. Note that the process discussed is application specific and can be adapted accordingly as the dataset to be imported has to match at least a minimum number of columns in the pre-defined clustered dataset.

II. METHODOLOGY

A. Data Mining

Let us refer the dataset for which the headers have to be identified as Dataset A and the dataset which is used train so as to form pre-defined clusters as Dataset B.

Initially, before the dataset A is input, we create clusters of clearly defined boundaries using Dataset B for which we have the metadata and are able to form clusters accordingly. Crucial information such as the type of the fields and the maximum length of the fields is captured so as to help in creating the schema for datasets which are to be input later. There are many techniques which can be used to be form the requisite clusters. But the results drawn from the following two algorithms have been significantly better than the rest for the student dataset used:

- k-Means Algorithm
- DBScan Algorithm

B. k-Means Algorithm

It is an iterative algorithm in which objects (a column value of each record) are moved among a set of clusters until the desired set is achieved. The algorithm is built on the concept of user specified input parameter (k) which is the number of clusters to be formed. This outperforms DBScan with its high precision due to fact that the number of clusters are specified beforehand which is not the case in DBScan algorithm. As we have the metadata of Dataset B, this algorithm is more suitable for our scenario. [1]

C. DBScan Algorithm

Density-based clustering locates regions of high density that are separated from one another by regions of low density i.e., it creates the clusters with a minimum size and density. This contrary to k-means, handles the outlier problem by ensuring that an outlier will not create a cluster. But as the number of columns/clusters are not specified, the algorithm fails to classify clusters with fine margins. [1]

D. Classification Techniques

Once the clusters are formed using the dataset B, the dataset A can be input to the model to determine headers and other Meta information regarding the data. There are several classifying techniques which can be applied to determine which column corresponds to which but the combination of multiple classifying techniques gives more assured results but at the cost of run time. The following combination of classification techniques works best for the student database:

1) K-Nearest Neighbors Classifier

Finding out the k nearest neighbors using simple Euclidean distance formula. The mean of all the clustered tuples is calculated and is compared against the individual tuple values to determine which column falls under which cluster. For character fields, attribute values can be used to determine the closeness among the fields under comparison. [3]

2) Classification by Backpropagation

This technique alone can be used without actually creating a clustered model in the first place. Backpropagation is a neural network learning algorithm. By training the network with sample dataset B, we can create a model to predict future mappings. The results of this can be compared with the results of the K-nearest neighbor classifier to improve the overall results of the system. If the root mean square error of the backpropagation process is negligible then the results from it can be used over the results from K-nearest neighbors classifier. [4]

E. Reading in of the Data

After we have identified the right table headers and data types, we can further enhance the system by performing an automatic reading in of the data for conducting further analysis. For the sake of demonstration we have used the tools MS Excel and SAS but there are plenty of other alternatives. The whole process described so far was easily achievable using Java due to the large number of predefined packages that were already available.

1) Connecting Java to Excel

Apache POI project [2] have a JAR file developed which can be used to create excel workbooks, create sheets and populate data in cells. But as we are concentrating on a very specific dataset that is of student details, we can anticipate the mandatory fields and let our program map the fields as required by opening a template excel sheet. Mapping of the fields is a straight forward task, a label is associated with each identified field which is used to populate the excel file with the corresponding column number in the actual raw data.

Information on the File Being Read-In

Filename - captured from folder: Batch_Samples.txt

File type Identified: Tab delimited

Details of the file layout in the table below for reference :

Variable Number	Field Name or Description of the field	Field Type Assigned	Field Length Assigned	Column Position or Order in Extract
1	ADMISSION_ID	Numeric	8	1
2	STUDENT_ID	Numeric	8	2
3	STUDENT_NAME	Character	25	3
4	FATHER_NAME	Character	21	4
5	GENDER	Character	1	5
6	INCOME_CATEGORY	Numeric	8	6
7	FEES	Numeric	8	7
8	ADDRESS	Character	81	8
9	DISTRICT	Character	14	9
10	PIN_CODE	Numeric	8	10
11	MOBILE_NUMBER	Numeric	8	11
12	ALT_NUMBER	Numeric	8	12
13	EMAIL_ID	Character	30	13

Fig. 1: Template Excel Sheet

Processing - Data Read In

Would you like to read in the Student data?	Yes
Would you like to read in the Department data?	No
Would you like to read in the Employee data?	No
Run the manual customizable program?	No

Import the Desired Data

Fig. 2: Macro enabled buttons to invoke shell commands to run SAS program

2) Executing SAS Programs from Excel

Once the fields are mapped automatically and filled in the excel sheet. Commands can be executed to run a SAS program using shell scripts. Code can also be tweaked to notify the user about the completion of the program. The user can then check the program log for verifying the results.

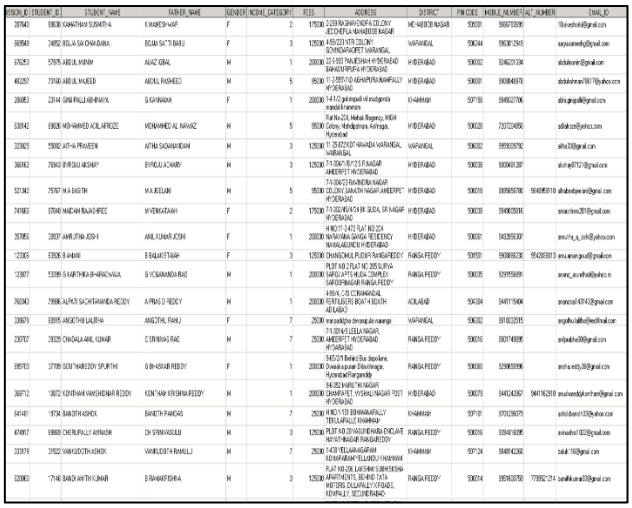

Fig. 3: Sample SAS dataset after read in

```
43      input
44                  ADMISSION_ID
45                  STUDENT_ID
46                  STUDENT_NAME $
47                  FATHER_NAME $
48                  GENDER $
49                  INCOME_CATEGORY
50                  FEES
51                  ADDRESS $
52                  DISTRICT $
53                  PIN_CODE
54                  MOBILE_NUMBER
55                  ALT_NUMBER
56                  EMAIL_ID $
57      ;
58      if _ERROR_ then call symputx('_EFIERR_',1);  /* set ERROR detection macro variable */
59      run;

NOTE: The infile '\\hdcaersnas\aers\ADV\edagaa_gaa0388501\EFT_UP\test\Keerthi
      Kant\rawdata\Batch_Samples.txt' is:

      Filename=\\hdcaersnas\aers\ADV\edagaa_gaa0388501\EFT_UP\test\Keerthi
      Kant\rawdata\Batch_Samples.txt,
      RECFM=V,LRECL=32767,File Size (bytes)=56815,
      Last Modified=08Oct2017:07:22:58,
      Create Time=08Oct2017:07:23:01

NOTE: 350 records were read from the infile
      '\\hdcaersnas\aers\ADV\edagaa_gaa0388501\EFT_UP\test\Keerthi
      Kant\rawdata\Batch_Samples.txt'.
      The minimum record length was 102.
      The maximum record length was 236.
```

Fig. 4: Log snippet confirming the results.

F. Requirements

- Computer
- Two datasets which have similar metadata characteristics but largely different data

III. PROPOSED SYSTEM

The designed system allows the user to identify header information automatically for tabular data for which there is no such information available. Upon identification, the system can automatically read in the data into a big data tool so as to facilitate further analysis. More functionalities are proposed to be added to the system such as:

- Expanding beyond the used technologies i.e. using other big data tools
- Enabling automatic report generation by anticipating user needs
- Schedule the read in of data
- Read in of unidentified columns so that the user can map later manually

IV. CONCLUSION

On implementing the system, the amount of burden on the user to manually map the fields reduces and the process can be automated. Redundant tasks of manually mapping the data which comes on real time basis needs to be automated so as to avoid potential bottleneck. The scope of this paper is to use existing technologies and further enhance them to provide a system which can deliver high accuracy rates.

REFERENCES

[1] Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems) by Jiawei Han, Micheline Kamber, Jian Pei Professor
[2] http://poi.apache.org/spreadsheet/quick-guide.html#CreateCells
[3] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
[4] Y. LeCun, L. Bottou, G. Orr, and K. Muller, "Efficient BackProp," in Neural Networks: Tricks of the trade, (G. Orr and Muller K., eds.), 1998.

