Read Me - Data Analysis

Shawn Unger

May 2017

Contents

1 Introduction

This report will explains steps in which one would go about setting up for

1. Using the data analysis programs used in the folder
2. Running the program on a selected file
3. Retrieving the information the analysis provides

Before continuing with the first step make sure to:

- Have a access to the internet
- Have a computer with space to download a minimum of 10 GB of data and software.
- Have drop-box account and access to the two folders named:
    1. "Files-Shawn"
    2. "Shawn
    3. "Hathi_file_analysis"

    Note: For this manual, reference code used in the third file listed. Anything found in file 1 and 2 is there for reference on earlier versions, scrap work and prior analysis during the development stages

2 Getting Things Ready

The following are each software required for the analysis and where to download them. Each is broken down in to sections on where the software can be attained and how it should be set up.

2.1 Drop-box

Download

- Go to `https://www.dropbox.com/downloading` and download drop-box onto your computer

Set Up

1. Run the following Code
   install.packages("rmarkdown");
   install.packages("knitr");
   install.packages("evaluate");
   install.packages("formatR")

2.2 IPython

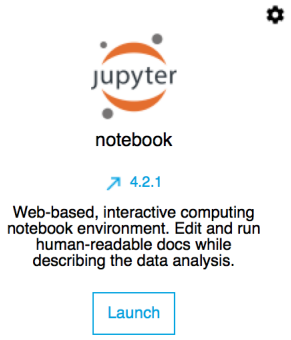
Download

- https://www.continuum.io/downloads

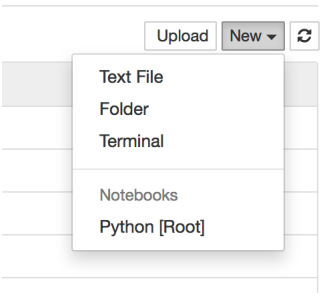
Set Up

1. Start the Navigator Panel by clicking on the app with the following icon:
   

2. Press on the "Launch" icon under Jupyter Notebook that looks as such:



3. At the top right hand side press on the "New" flip down menu icon and choose the "Pyhon[Root]" option which should look as the following:



4. In the notebook copy the following code:
   !pip install nltk
   !pip install re
   !pip install pprint
   !pip install string
   !pip install pandas
   !pip install matplotlib

5. Run this code pressing the icon that looks some what like a play button pointing at vertical line:

3 Running the Analysis

3.1 Steps for cleaning a file

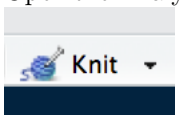
1. Choose file which will be scanned
2. Place file in folder called "Hathi_file_analysis"
3. Open ipython notebook by following steps 1 to 3 from Part 2.2
4. Go to your dropbox folder
5. Go to "Hathi_file_analysis"
6. Go to "Programs"
7. Run the file called "Cleaner_2017"
8. Run "Step 1" on and when prompted, enter the name of the file you wish to analyze
9. Under "Step 2" you will see the following:
   create_new_file(text, text_name,[a], [b], [c],[d])
   - for any field left blank, all types of the given field will be accounted for (nothing left out)
   - In field "a" list the text types which the clean algorithm will account for
   - In field "b" list the countries which the clean algorithm will account for
   - In field "c" list the languages which the clean algorithm will account for
   - In field "d" enter 1 to account for US government files, and 0 if to account for non US government files
10. Run step 2 for cleaning algorithm to begin

3.2 Steps to analyzing file

1. Clean file using steps listed in 3.1
2. Go to the folder containing the file level at which it which you would like to analyze:
   - Go to the "Files" folder for the entire cleaned file being analyzed
   - Go to the "Files" folder then to "Outputs" then "Correct_Both" for analysis of data that has the same correct results by Algorithm and Hathi file

- Go to the "Files" folder then to "Outputs" then "Correct_Hathi" for analysis of data that has the same correct results by Hathi
- Go to the "Files" folder then to "Outputs" then "Correct_Algo" for analysis of data that has the same correct results by Algorithm
- Go to the "Files" folder then to "Outputs" then "Incorrect" for analysis of data that has the same incorrect results by Algorithm and Hathi

3. Open the Analysis.Rmd and press Knit icon that looks like the icon bellow:



3.3 How the program functions

1. First "cleaner" note book takes in file name and notes to the master list (in the file folder) that the file was scanned

2. Each line in the file is then passed through "New File Maker" which analyzes the content and whether this is the data the user is looking to analyze based on their chosen configurations:

   - Language
   - Region
   - Years
   - Government Doc

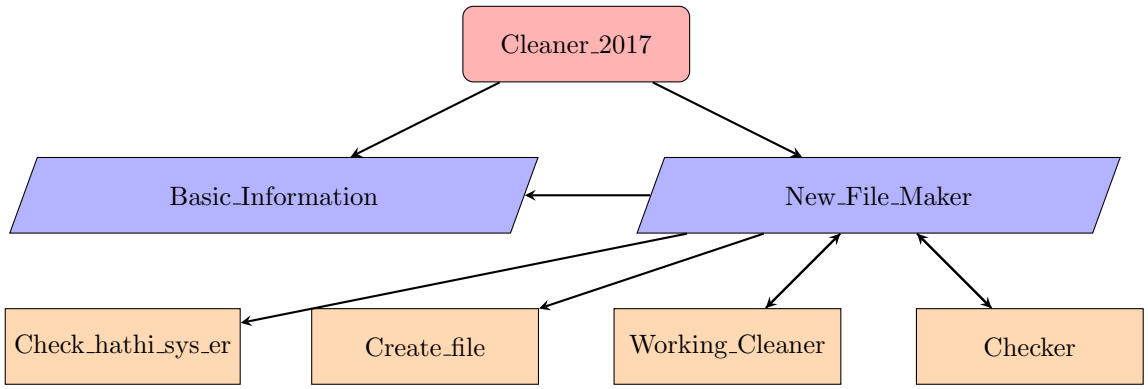
3. The lines that fit the setting are then cleaned and are given attributes based on description entry of the data

4. The new cleaned lines are then contrasted between the output of the algorithm and information listed by the hathi file, and are sorted based on whether the description of the data was:

   - interpreted correctly by hath file
   - interpreted correctly by algorithm
   - interpreted correctly by both
   - neither interpreted correctly

5. The data is separated in to the 4 groups that each line matches with (based on 1 of 4 categories form above)

4 Dependencies



5 Programs and their Code

5.1 Create_file

Running the file

- Asks for name of the file to process
- Creates a path to the file
- Create a new text file to be used for cleaned data in the "Files" folder
- Create a new text file to be used for left over data in the "Left_Overs" sub-folder with the "Files" folder

5.2 Cleaner_2017

The following describes the result of running each given step:

1. By running this step the program:
    - Runs "New_File_Maker" code and the functions within it
    - Imports needed packages
    - Allows user to choose which file they would like to analyze
2. Runs cleaning Algorithm and adds the analyzed file to master file

5.3 Working_cleaner

This file runs an elaborate cleaning function called "cleaner" which is depended on over half a dozen other functions to run. Bellow is the description of the function, for more information please see the code:

1. **cleaner(line)**
   Takes in the line and uses the description within it in order to out out the break down of the description that are independent of publishing dates, the earliest and latest years which they were published, whether there are multiple or single date present in the description, the description about the document itself given by hathi (without the dates), and what the algo found based on the 14 types of checks listed under "data_cross_check" (see bellow)

5.4 Checker

The following are the functions within this file and what they do:

1. **language_checker(line, language)** :
   Check if the second last item in the data given in line is the same as the one given by language, if true return the line, otherwise return empty string

2. **country_checker(line, country)** :
   check if the third last item in the data given in line is the same as the one given by country, if true return the True, else return False

3. **data_cross_check(line)** :
   Check if the date proposed by the hathi file in line are the same or different from one found by the algo, which is the third last item in line, and then gives it the appropriate label based on whether the findings are correct or not. The function outputs a 14 element row vector, which is zero everywhere except being equal to one if the corresponding rows description matches the line. Below is the description that goes along with each row:

   - 1 = algo found single date and non match with hathi
   - 2 = algo found multiple date and non match with hathi
   - 3 = algo found single date and match with hathi
   - 4 = algo found multiple date and final date is match with hathi
   - 5 = algo found multiple date and start date is match with hathi
   - 6 = algo found multiple date and hathi date is between start and finish date
   - 7 = no date found by algo and hathi lists date
   - 8 = no dates found by both algo and hathi
   - 9 = algo found single data and hathi did not list any
   - 10 = algo found multiple dates and hathi did not list any date
   - 11 = hathi lists a "9999" date and algo did not find a date
   - 12 = hathi lists a "9999" date and algo found single date
   - 13 = hathi lists a "9999" date and algo found multiple dates
   - 14 = line is too short and thus was not evaluated

5.5 Check_hathi_sys_er

The following are the functions within this file and what they do:

1. **create_LO_files(text_name)** :
   This program creates left over files that will be used to analyze the data based on different characteristics over time

2. **clean_left_over(text_name))** :
   The Algorithm here checks between the output of the algorithm and that provided by hathi file, and then outputs what it finds in to the respective left over file based on its bin

5.6 Sorter

The following is the function within this file:

1. **sort_and_append(line, number)**:
   Given a line, and whether it is the same correct result as the hahti file, correct result but hathi has wrong results, vice verso , or both hathi and algo have incorrect result, add to the folder and cumulative data set of this type

6 Outputs

Once the first layer of analysis is run on a specific file the following are the outputs:

1. The file's name and the attributes used for the analysis will be added to the "Master" list found in the "Files" folder
2. A file called "Just_Cleaned" is created in the Files folder, which is
    - the original file with new data appended as 5 most right rows on the data lines that was selected to be analyzed and cleaned
    - refreshed (wiped then rewritten) each time the program is run
    - leveraged for the next few outputs
3. "Just_Cleaned" and "Data" file updated in the 4 folders inside the "outputs" folder where the former file is a cleaned version of the data that corresponds to the title of the sub folder it is in and the latter file is the running sum of such data.
4. Running the analysis file provides a analysis report of the data in both the Just cleaned and Data files, with a summary report of the data for each as well.