Rivet User Guide

Author(s): James Baggs, Yuxu Yang

The instructions below require a valid installation of Rosie, Rivet, Python, and Bash/Unix shell. If you are unsure whether or not you have these properly installed, please see our installation guide.

File Pattern Creation

In order to use the Rivet system to create a file pattern of the sample data, you will need to run the Rivet.py system with the -s (or --sampleSize=) option which specifies the percentage of lines to be sampled from the original data file. To automatically generate an RPL file, the option -p (or --prunePercentage=) is used to inform Rivet to prune all patterns that matches a smaller percentage of the sample data than the specified percentage. If a user dos not provide a prune percentage, a user interface will be displayed for a user to select desired patterns manually to form the output file pattern. At this time, a user will also be able to name the pattern, and the RPL file name, as opposed to the defaults of “auto” and “auto.rpl”, respectively.

Syntax: `$python Rivet.py <options> <dataFile>`

`<dataFile>` - The datafile that need to be analyzed (e.g. NewDataSet.csv)

Options:	Description
-s <x> or --sampleSize=<x>	Line percentage user wish to sample from original data
-p <x> or --prunePercentage=<x>	Prune percentage for Rivet to automatically select patterns that have a matching percentage above this value

Example commands:

`$python Rivet.py -s 10 NewDataSet.csv`

- Runs the Rivet system on a 10% of randomly selected lines from the NewDataSet.csv dataset. Since no prune percent is specified, the user will be prompted via command line to choose which patterns to include in the output RPL. The user will also be prompted to name the new pattern and file.

`$python Rivet.py -s 10 -p 1 NewDataSet.csv`

- Runs the Rivet system on a 10% of randomly selected lines from the NewDataSet.csv dataset. All patterns which mach more than 1% of the sampled data will be automatically select by Rivet to generate a new pattern named auto and then be put into a new file name auto.rpl

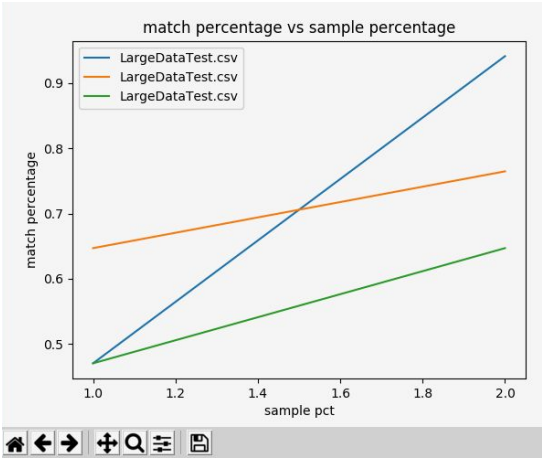
Testing

- Black Box Test

The following Black Box Testing tools provide users with a way to assess the quality of RPL files in terms of their match percentage and their data parsing speed.

- PruneSampleAnalysis.py - Directory-based file paramater analysis

  PruneSampleAnalysis is a tool which allows users to determine what effect sample sizes have on match percentages.

  - Place any files you wish to analyze in the folder "2017FallTeam11/resource/test"
  - From the project's BlackBoxTest directory, run the command:
    - `python PruneSampleAnalysis.py ../resource/test`
  - A graph such as the one below should be displayed, showing how the sample size of files in a directory can affect its match percent.
  - 

- ExtractionTime.py - Data Extraction Timing tool: RPL Enhanced vs Brute Force

  This tool runs Rosie's Brute Force Extraction and RPL-Enhanced extraction on a specified dataset to compare the differences in speed

  - Usage: `>python ExtractionTime.py <options> <datafile> <RPLfile> <pattern>`
  - Valid options include `-t <x>` and `--timingRuns=<x>` to specify the number of times that each method should be run. The default is 5.
  - Expected output should run each type extraction method the specified number of time and output the average time for each type to the console. An example is in the figure below:

```
*Begin File Pattern Enhanced Timing Runs
auto.rpl
Data Successfully Extracted. Saved to /result/customizedResult.json
Match Rate : 47.06%
trial #0: 5.67 seconds
average RPL time: 5.67
average brute force time: 29.08
```

- FinalDataPattern.py

  FinalDataPattern provides users with a way to test the match percentage of a specified RPL file and pattern against a particular data file

  - Usage: `>python FinalDataPattern.py <datafile> <RPLfile> <pattern>`
  - Example: `>python FinalDataPattern.py NewDataSet.csv auto.rpl auto`
  - The output prints the pattern's match percentage to the console:
    - `Match Rate: 100%`

- Unit Test
  - Open PyCharm(it must be Professional version)
  - In Package Explorer: Right click test folder under 2017FallTeam11
  - Click "Run "Unittests" in test with Coverage"
  - Wait for report
  - If prompt: *"Do you want to display coverage data for 'Unittests in test Coverage Results'?".*
    - Select "Add to Active Suites"
  - Report should be similar as image below