Universität Hamburg
Department Informatik
Wissenschaftliches Rechnen
Dr. J. Kunkel, H. Hatzel

Exercise Sheet 5
Big Data Analytics
WiSe 2016/2017
Due Date: 25.11.2015, 390 P

Please document your code, without sufficent documentation you won't recive any points.

1 Exploration of Wikipedia Data (R) (120 P)

Explore the dataset `/home/bigdata/5/wikipedia-clean.csv` with our cleaned Wikipedia data. Since the file is very big, you should develop and test your approach first on the subsets provided:
`/home/bigdata/5/wikipedia-clean-10MiB.csv` and `/home/bigdata/5/wikipedia-clean-100MiB.csv`
Write the script that it takes the input file as argument. Finally, run it on the full data set.

1.1 Guiding questions

1. Is there a correlation of article length with the length of sentences within the article?
2. How do the lengths of different categories compare?
3. Which categories have the shortest and longest articles?
4. Identify a few specific articles which, lengthwise, are outliers in their category.

Submission:

`1-wiki-explore.pdf`	Your R notebook with your analysis, code and plots.

2 IMDb Data Cleaning (Python) (120 P)

In this exercise, we will import and process semi-structured data on movies. We will use this data for tasks to come. The learning goals of this exercise is to make yourself familiar with object oriented programming in Python, while getting more familiar with data cleaning.
We will collection quotes from the IMDb as available from their dump. The data you need can be found in `/home/bigdata/5/imdb-quotes.txt`.
Approach the task as follows:

- Write an object oriented Python program for reading IMDb data
- Collect quotes
- Find a structured representation for quotes
- Write them to a CSV file in a structured manner

Measure the your programs runtime and evaluate if big data products should be used for this task. Consider the 5-Vs (challenges for big data).

2.1 Dataset: IMDb

The dataset contains famous quotes from movies. A new quote is started with the #, then followed by the movie name and year, then the episode title is listed. Multiple quotes of the same movie are separated with an empty line. The speakers of the lines should be separated from the content.

A few examples:

```
# "5 Second Movies" (2007) {Star Wars: Episode V (#1.22)}
Darth Vader: I am your father.
Luke Skywalker: [screams] NO!
Yoda: Yes.
Chewbacca: [laughs]

# The Gamers (2002) (V)
Rogar, the Barbarian: Speak, or I shall smite thee with my mighty
  blade!

Rogar, The Barbarian: Am I still unconscious?

Nimble the Thief: [shocked] The Shadow?
Ambrose: [scared] The Shadow?
Newmoon the Elf: [resolved] The Shadow.

The Gamemaster: You’re going to backstab him with a ballista?
Nimble the Thief: Uh huh
The Gamemaster: With a ..... siege weapon?
Nimble the Thief: Uh huh
```

Submission:

`2-imdb-clean.py` Your program which writes movies and their quotes to a CSV file.

`2-answers.txt` Your answers to the questions posed in the task above.

3 Implement a JOIN using MapReduce (Python) (150 P)

Implement a join operation to connect coordinates the CSV file generated from `/home/bigdata/5/netcdf.csv` with a second CSV file. Link the coordinates with geo-information from `/home/bigdata/5/wiki-coordinates.csv`. This way we can tell the weather conditions for any given place from our Wikipedia data. Output your data as a CSV with the following header:

```
place,coordinates,temperature,precipitation
```

Where place is the title of the wikipedia page.
For simpler debugging you can launch your program from the shell:

```
cat file1.csv file2.csv | ./map.py | sort | ./reduce.py
```

When running with Hadoop, use the command from the last exercise sheet with a second input directory as follows:

```
yarn jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar \
  -Dmapred.reduce.tasks=1 -Dmapred.map.tasks=11 -mapper $PWD/my-map.py -reducer $PWD/my-reduce.py \
  -input <input-1> -input <input-2> -output <output-directory>
```

3.1 Hints

An introductions to joins using map reduce can be found here:
`http://codingjunkie.net/mapreduce-reduce-joins/`
Note that the coordinates should be joined in a fuzzy way as the CSV file for the weather is not arbitrarily accurate. You are free to choose the exact method for joining them.

Submission:

`3-join.py` Your map reduce program to join data from `netcdf.csv` and `wiki-coordinates.csv`