COMP9313 21T3 Project 1 (12 marks)

Problem statement:

Detecting popular and trending topics from the news articles is an important task for public opinion monitoring. In this project, your task is to perform text data analysis over a dataset of Australian news from ABC (Australian Broadcasting Corporation) using **MRJob**. The problem is to compute the weights of each term regarding each year in the news articles dataset.

Input files:

The dataset you are going to use contains data of news headlines published over several years. In this text file, each line is a headline of a news article, in format of "date,term1 term2 ... ... ". The date and texts are separated by a comma, and the terms are separated by the space character. A sample file is like below (note that the stop words like “to”, “the”, and “in” have already been removed from the dataset):

```
20191124,woman stabbed adelaide shopping centre
20191204,economy continue teetering edge recession
20200401,coronanomics learnt coronavirus economy
20200401,coronavirus home test kits selling chinese community
20201015,coronavirus pacific economy foriegn aid china
20201016,china builds pig apartment blocks guard swine flu
20211216,economy starts bounce unemployment
20211224,online shopping rise due coronavirus
20211229,china close encounters elon musks
```

This small sample file can be downloaded at:

https://webcms3.cse.unsw.edu.au/COMP9313/22T2/resources/76308

Term weights computation:

To compute the weight for a term regarding a year, please use the TF/IDF model. Specifically, the TF and IDF can be computed as:

TF(term t, year y) = the frequency of t in y

IDF(term t, dataset D) = $\log_{10}$ (the number of years in D/the number of years having t)

Finally, the term weight of term t regarding the year y is computed as:

Weight(term t, year y, dataset D) = TF(term t, year y)* IDF(term t, dataset D)

Please import math and use math.log10() to compute the term weights.

Output format:

If there are N terms in the dataset, you should output exactly N lines in your final output file on HDFS, and these lines are sorted by terms in alphabetical order. In each line, you need to output a list of <year, weight> pairs, and these pairs are sorted by year in ascending order. Specifically, the format of each line is like: “term**\t**

$Year_1$,$Weight_1$;$Year_2$,$Weight_2$;… …;$Year_k$,$Weight_k$". For example, given the above data set, the first few lines of the output should be (there is no need to remove the quotation marks which are generated by MRJob):

```
"adelaide"  "2019,0.47712125471966244"
"aid" "2020,0.47712125471966244"
"apartment" "2020,0.47712125471966244"
"blocks"    "2020,0.47712125471966244"
"bounce"    "2021,0.47712125471966244"
"builds"    "2020,0.47712125471966244"
"centre"    "2019,0.47712125471966244"
"china"     "2020,0.3521825181113625;2021,0.17609125905568124"
```

The entire output could be checked at:

https://webcms3.cse.unsw.edu.au/COMP9313/22T2/resources/77904

Code format:

Please name your python file as "project1.py" and compress it in a package named "zID_proj1.zip" (e.g. z5123456_proj1.zip).

Command of running your code:

To reduce the difficulty of the project, you are allowed to pass the total number of years to your job. We will also use more than 1 reducer to test your code. Assuming there are 20 years, and we use 2 reducers, we will use the following command to run your code:

```
$ python3 project1.py -r hadoop hdfs_input -o hdfs_output --jobconf
myjob.settings.years=20 --jobconf mapreduce.job.reduces=2
```

In this command, `hdfs_input` is the input file in HDFS, and `hdfs_output` is the output folder in HDFS. You can access the total number of years in your program like "N = jobconf_from_env('myjob.settings.years')" (use "from mrjob.compat import jobconf_from_env" in your code).

Please ensure that the code you submit can be compiled. Any solution that has compilation errors will receive no more than 4 points.

Marking Criteria:

Your source code will be inspected and marked based on readability and ease of understanding. The documentation (comments of the codes) in your source code is also important. Below is an indicative marking scheme:

Result correctness: 6
Algorithm design (the use of design patterns learned to reduce memory consumption and to improve efficiency): 5
Code structure, Readability, and Documentation: 1

Submission:

Deadline: Sunday 3rd July 11:59:59 PM

You can submit through Moodle:

If you submit your assignment more than once, the last submission will replace the previous one. To prove successful submission, please take a screenshot as assignment submission instructions show and keep it by yourself. If you have any problems in submissions, please email to yufan.sheng@unsw.edu.au.

Late submission penalty

5% reduction of your marks for up to 5 days

Plagiarism:

The work you submit must be your own work. Submission of work partially or completely derived from any other person or jointly written with any other person is not permitted. The penalties for such an offence may include negative marks, automatic failure of the course and possibly other academic discipline. Assignment submissions will be examined manually.

Relevant scholarship authorities will be informed if students holding scholarships are involved in an incident of plagiarism or other misconduct.

Do not provide or show your assignment work to any other person - apart from the teaching staff of this subject. If you knowingly provide or show your assignment work to another person for any reason, and work derived from it is submitted you may be penalized, even if the work was submitted without your knowledge or consent.