COMP9313 23T2 Project 2 (16 marks)

Problem statement:

In this problem, we are still going to use the dataset of Australian news from ABC. Your task is to find out the top-*k* most frequent co-occurring term pairs in each year. The co-occurrence of (w, u) is defined as: u and w appear in the same article headline (i.e., (w, u) and (u, w) are treated equally).

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:

```
20030219,council chief executive fails to secure position
20030219,council welcomes ambulance levy decision
20030219,council welcomes insurance breakthrough
20030219,fed opp to re introduce national insurance
20040501,cowboys survive eels comeback
20040501,cowboys withstand eels fightback
20040502,castro vows cuban socialism to survive bush
20200401,coronanomics things learnt about how coronavirus economy
20200401,coronavirus at home test kits selling in the chinese community
20200401,coronavirus campbell remess streams bear making classes
20201015,coronavirus pacific economy foriegn aid china
20201016,china builds pig apartment blocks to guard against swine flu
```

This small sample file can be downloaded at:

https://webcms3.cse.unsw.edu.au/COMP9313/23T2/resources/88352

Output format:

You need to ignore the stop words such as “to”, “the”, “in”, etc. (refer to the broadcast variable on how to do this efficiently). A stop word list is stored in this file:

https://webcms3.cse.unsw.edu.au/COMP9313/23T2/resources/88354

Please get the terms from the dataset as below:

- Split the headline by the space character to obtain terms.
- Ignore the stop words such as “to”, “the”, “in”, etc.
- Ignore terms starting with non-alphabetical characters, i.e., only consider terms starting with “a” to “z”.

Your Spark program should generate a list of (***k*** * *total years*) results, each of which is in format of “Year\tTerm$_1$,Term$_2$:Count” (the two terms are sorted in alphabetical

order and separated by ","). The results should be first ranked by the year in ascending order, and then by the co-occurrence count of a pair in descending order, and finally by the term pair in alphabetical order.

Given $k = 1$ and the sample dataset, the output is like:

```
2003\tcouncil,welcomes:2
2004\tcowboys,eels:2
2020\tcoronavirus,economy:2
```

Code format:

Please name your two python files as "project2_rdd.py" and "project2_df.py" for using RDD and DataFrame APIs, respectively. Compress it in a package named "zID_proj2.zip" (e.g. z5123456_proj2.zip).

Command of running your code:

We will use the following command to run your code:

```
$ spark-submit project2_rdd.py input output stopwords k
```

In this command, `input` is the input file, `output` is the output folder, `stopwords` is the stop words file, and `k` is the number of pairs returned for each year.

Notes:

- You can read the files from either HDFS or the local file system. Using the local files is more convenient, but you need to use the prefix "file:///...". Spark uses HDFS by default if the path does not have a prefix.
- Please do not use numpy or pandas, since we aim to assess your understanding of the RDD/DataFrame APIs.
- You can use coalesce(1) to merge the data into a single partition and then save the data to disk.
- In the DataFrame solution, please do not use the spark.sql() function to pass the SQL statement to Spark directly.
- It does not matter if you have a new line at the end of the output file or not. It will not affect the correctness of your solution.

Marking Criteria:

Your source code will be checked and marked based on readability and ease of understanding. Each solution has 8 marks. Please ensure that the code you submit can be compiled. Below is an indicative marking scheme (for each):

Submission can be compiled and run on Spark: 3
Submission can obtain correct results: 3 • Correct term pairs • Correct counts • Correct order • Correct format • Correctly passing self-defined functions to Spark • Correctly using Spark APIs (RDD/DataFrame solution only RDD/DataFrame APIs allowed)
Efficiency of top-k computation: 1
Efficient stop words removal: 0.5
Code format and structure, Readability, and Documentation: 0.5

Submission:

Deadline: Sunday 16th Jul 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 siqing.li@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.