

COMP9313 2022T2 Project 3 (22 marks)

Finding Similar News Article Headlines Using Spark

In this problem, we are still going to use the dataset of Australian news from ABC. Similar news may appear in different years. Your task is to find all similar news article headline pairs across different years.

Background: Set similarity self-join

Given a collection of records R , a similarity function $\text{sim}(\cdot, \cdot)$, and a threshold τ , the set similarity self-join on R , is to find all record pairs r and s from R , such that $\text{sim}(r, s) \geq \tau$. In this project, you are required to use the Jaccard similarity function to compute $\text{sim}(r, s)$. Given the following example, and set $\tau=0.5$,

id	record
0	1 4 5 6
1	2 3 6
2	4 5 6
3	1 4 6
4	2 5 6
5	3 5

The result pairs are <https://powcoder.com>

pair	similarity
(0,2)	0.75
(0,3)	0.75
(1,4)	0.5
(2,3)	0.5
(2,4)	0.5

Input files:

In the 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 (note that the stop words have been removed already). A sample file is like below:

```
20191124,woman stabbed adelaide shopping centre
20191204,economy continue teetering edge recession
20200401,corononomics 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 sample file “tiny-data.txt” can be downloaded at:

<https://webcms3.cse.unsw.edu.au/COMP9313/22T2/resources/76321>.

Note that it is possible that one term appears multiple times in a headline. You need to convert the headline to a set first to compute the Jaccard similarity.

Output:

The output file contains all the similar headlines together with their similarities. In each pair, the headlines must be from different years. Please use the index of the headline in the file as its ID (starting from 0) and use the IDs to represent the headline pairs. Each line is in format of “(Id₁,Id₂)\tSimilarity” (Id₁<Id₂, and there should have no duplicate pairs in the result). The similarities are of double precision. The pairs are sorted in ascending order (by the first and then the second).

Given the example input data with threshold 0.1, the output file is like:

```
(0,7)\t0.1111111111111111
(1,4)\t0.1
(1,6)\t0.125
(4,6)\t0.1111111111111111
```

Code format:

Name the scala file as “SimilarNews.scala”, the object as “SimilarNews”, and the package as “comp9313.proj3”. Your program should take three parameters: the input file, the output folder, and the similarity threshold. Briefly describe your optimization techniques in a file “Optimization.pdf”.

Run in Google Dataproc - Cluster configuration:

Create a bucket with name “comp9313-<YOUR_STUDENTID>” in Dataproc. Create a folder “project3” in this bucket for holding the input files.

This project aims to let you see the power of distributed computation. Your code should scale well with the number of nodes used in a cluster. You are required to create three clusters in Dataproc to run the same job:

- Cluster1 - 1 master node and 2 worker nodes;
- Cluster2 - 1 master node and 4 worker nodes;
- Cluster3 - 1 master node and 6 worker nodes.

For both master and worker nodes, select n1-standard-2 (2 vCPU, 7.5GB memory).

Unzip and upload the following data set to your bucket and set τ to 0.8 to run your program: <https://webcms3.cse.unsw.edu.au/COMP9313/22T2/resources/78447>.

Record the runtime on each cluster and draw a figure where the x-axis is the number of nodes you used and the y-axis is the time of getting the result, and store this figure in a file “Runtime.jpg”. Please also take a screenshot of running your program on Dataproc in each cluster as a proof of the runtime. Compress the three screenshots into a zip file “Screenshots.zip”.

Create a project and test everything in your local computer, and finally do it in Google Dataproc.

Marking Criteria:

Your source code will be inspected and marked based on readability and ease of understanding. The efficiency and scalability of this project is very important and will be evaluated as well. Below is an indicative marking scheme:

Result correctness: 12
Efficiency and Scalability: 8
Code structure, Readability, and Documentation: 2

Assignment Project Exam Help

Submission:

Deadline: Sun 14th Aug 11:59:59 PM

You can submit through Moodle. You need to submit four files: SimilarNews.scala, Optimization.pdf, Runtime.jpg, and Screenshots.zip. Please compress everything in a package named “zID_proj3.zip” (e.g. z5123456_proj3.zip).

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.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder