ACTIVITY HADOOP - MAPREDUCE

**KIN TAT, TAN**

Sumário

[GitHub 3](#_Toc484875388)

[Introduction 3](#_Toc484875389)

[Activity 1 – Trending Tops 3](#_Toc484875390)

[Activity 2 – Text cleanup using chain mapper 4](#_Toc484875391)

[Activity 3 – The Top-N pattern 4](#_Toc484875392)

[Activity 4 - Sentiment of hashtags 5](#_Toc484875393)

[Activity 5 - Join all previous mapreduce Job in a single application 6](#_Toc484875394)

# GitHub

<https://github.com/tankintat/bigdataudl/tree/master/Hadoop%20-%20MapReduce%20-%20Act>

# Introduction

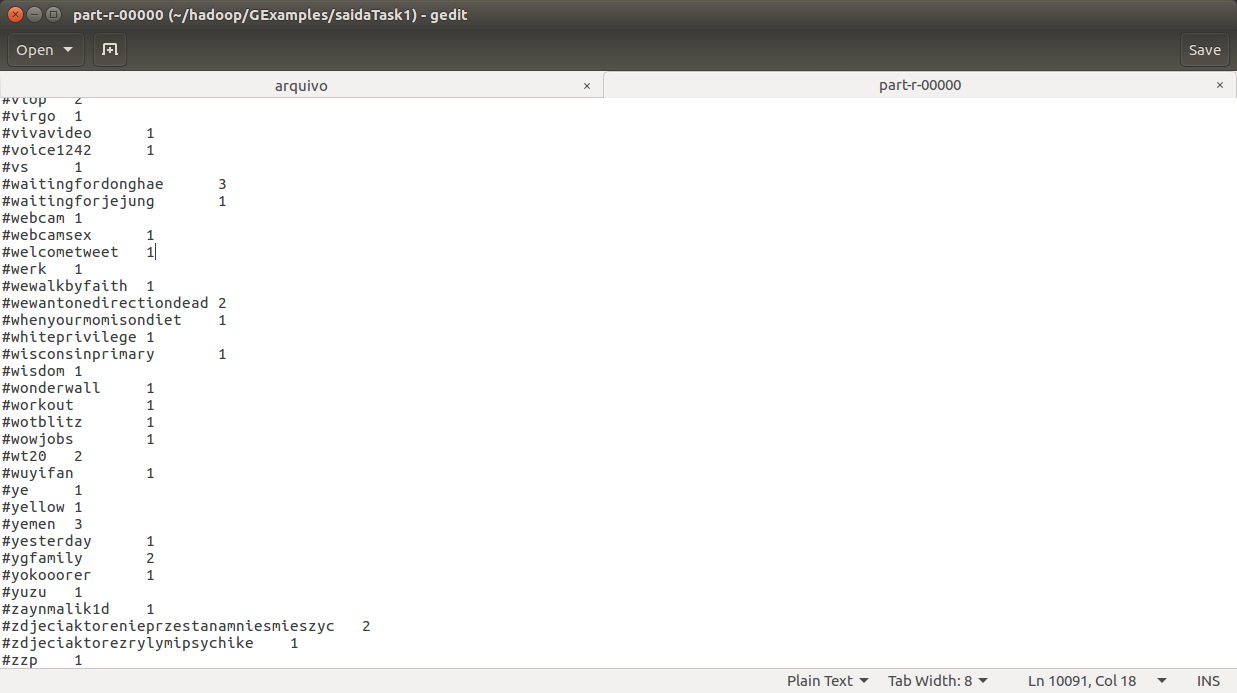
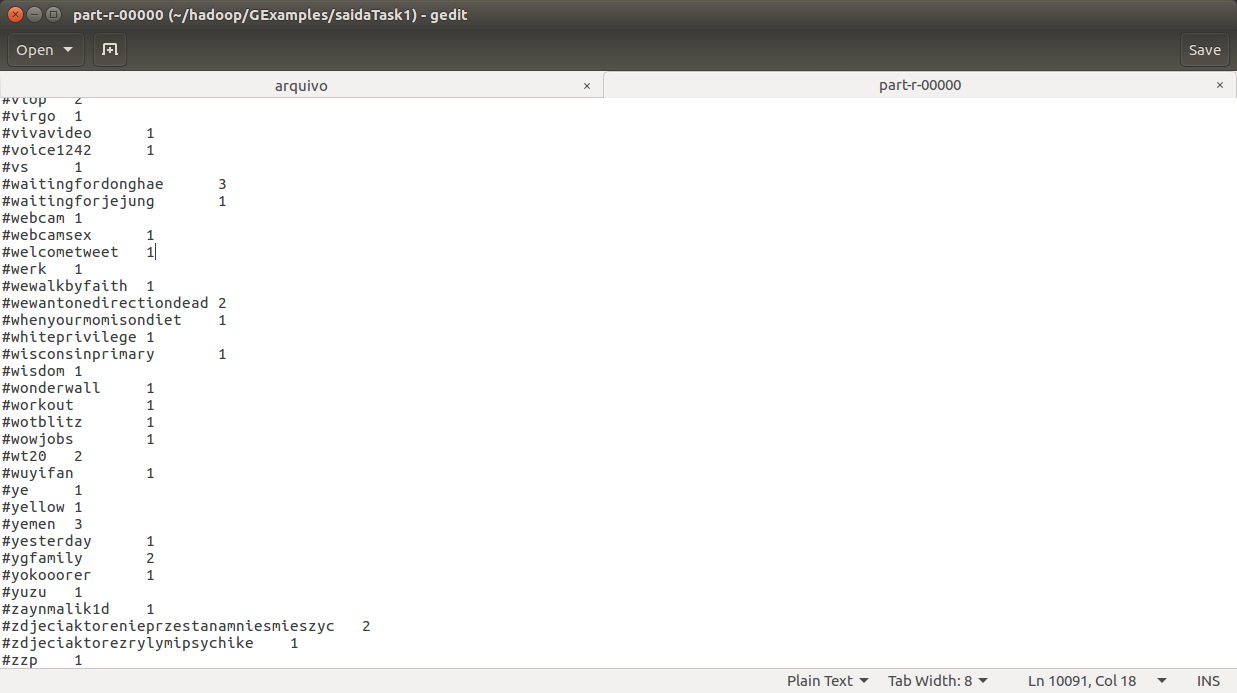
I created the artefacts in the same package, that means makes more easily to put debugger and test in the HDFS. In this document, I will show the results getting out in the debugger output and HDFS execution. Some input archives I remove some data because my computer brake in some processes.

# Activity 1 – Trending Tops

The objective about this activity was extract all the existence hashtags in a input text file. To a Twitter is a most important to know how much comments are talking about during the day. So the implementation was about the classical architecture MapReduce and the example count words we got the solution.

**hadoop jar Artefactos/ActMapReduce.jar eps.examples.mapreduce.Task1 /user/tan/Act1Input1/ /user/tan/Act1Output1**

Result the activity 1



# Activity 2 – Text cleanup using chain mapper

The objective this activity was using a json file to filter the properties in order to get information. Therefore was using the chain mapper and reduce because the chain implementation is possible to makes a sequence process mapper and reduce, so are more easily to control each step by the extraction.

First Mapper Step: Separate the properties in the json input file. The objective its to ignore the useless properties and use just useful properties.

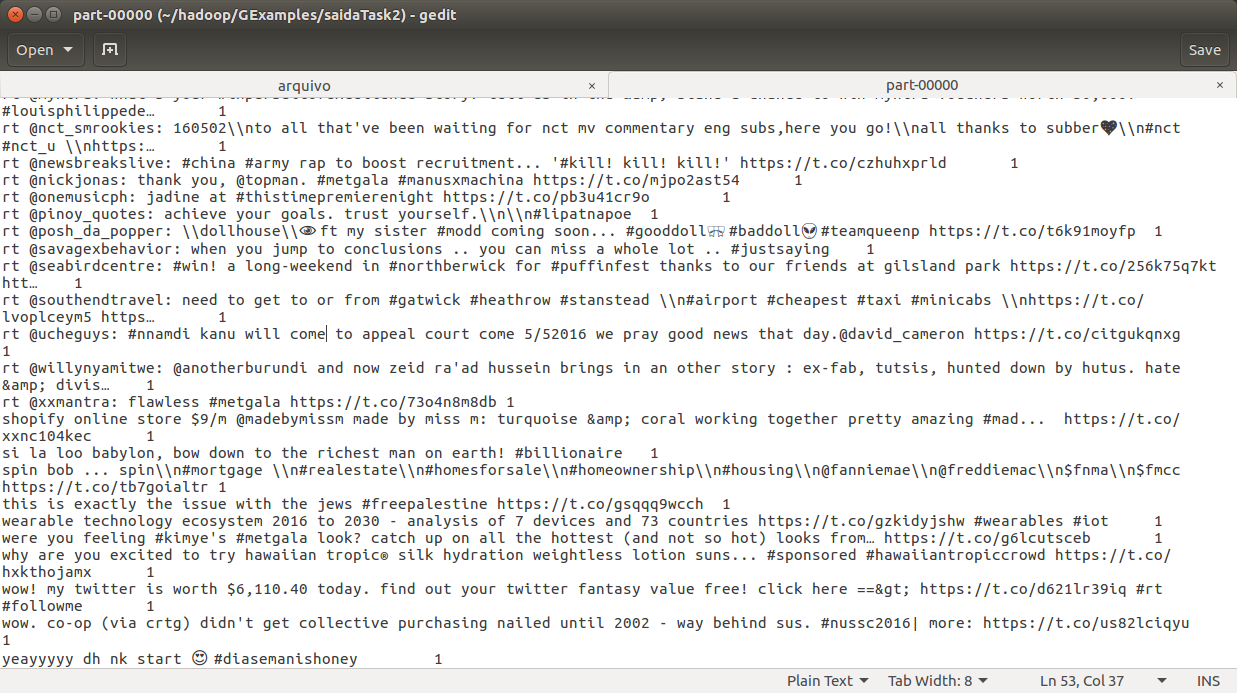
Second Mapper Step: Get just extract the defined language twitted.

Third Mapper Step: Assuring has a Hashtag and Text in the tweet.

Fourth Mapper Step: Text-lower case

First Reduce: To count the similar hashtags

**hadoop jar Artefactos/ActMapReduce.jar eps.examples.mapreduce.Task2 /user/tan/Act2Input1/ /user/tan/Act2Output1**

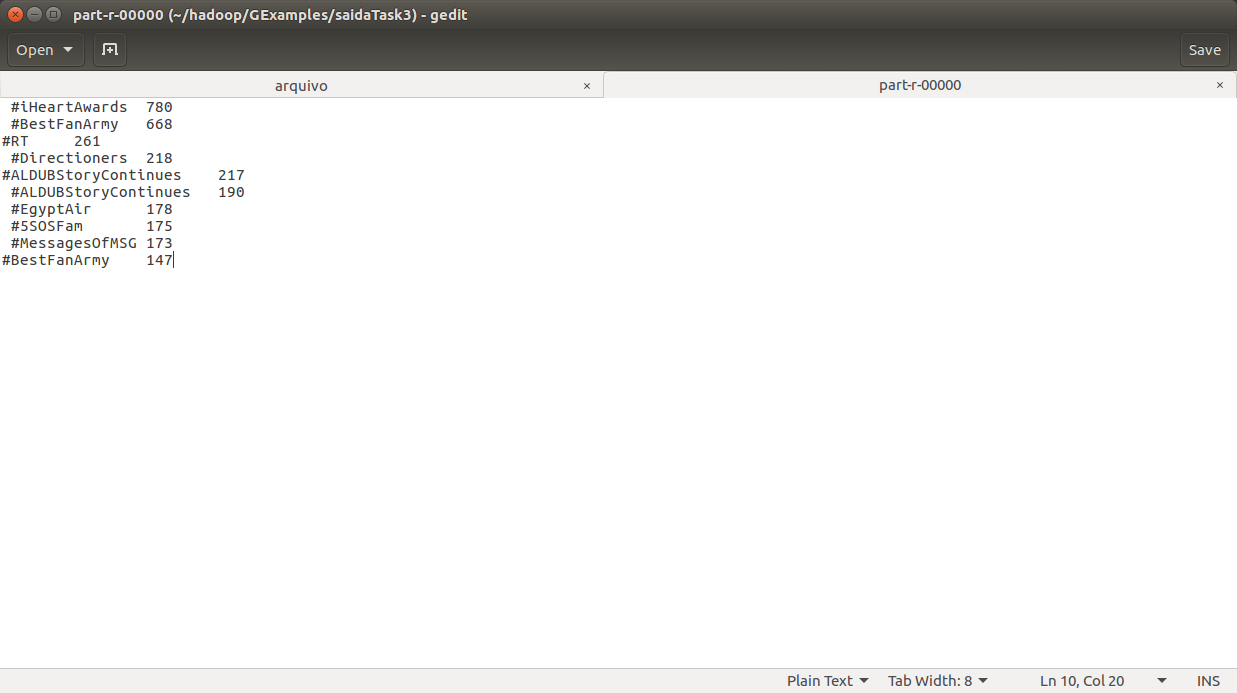


# Activity 3 – The Top-N pattern

The objective this activity was using all methods about the Mapper and Reduce, so we can use the setup and cleanup in these classes to order to get more fast and clear code.

The idea its get the results getting in the exercise one to input this activity, so then each mapper get a part the document and appoint the more appears with TreeMap help and passing one reduce to couple all mappers results and put the Top N Hashtags. The solution was convert the input and output type to Text. I loose a lot of time in a error that no makes sense, compatibility input and output. So I just use the [job.setInputFormatClass(KeyValueTextInput.Format.class)] to convert input to Text.

**hadoop jar Artefactos/ActMapReduce.jar eps.examples.mapreduce.Task3 /user/tan/Act1Output1/part\* /user/tan/Act3Output1/ 15**

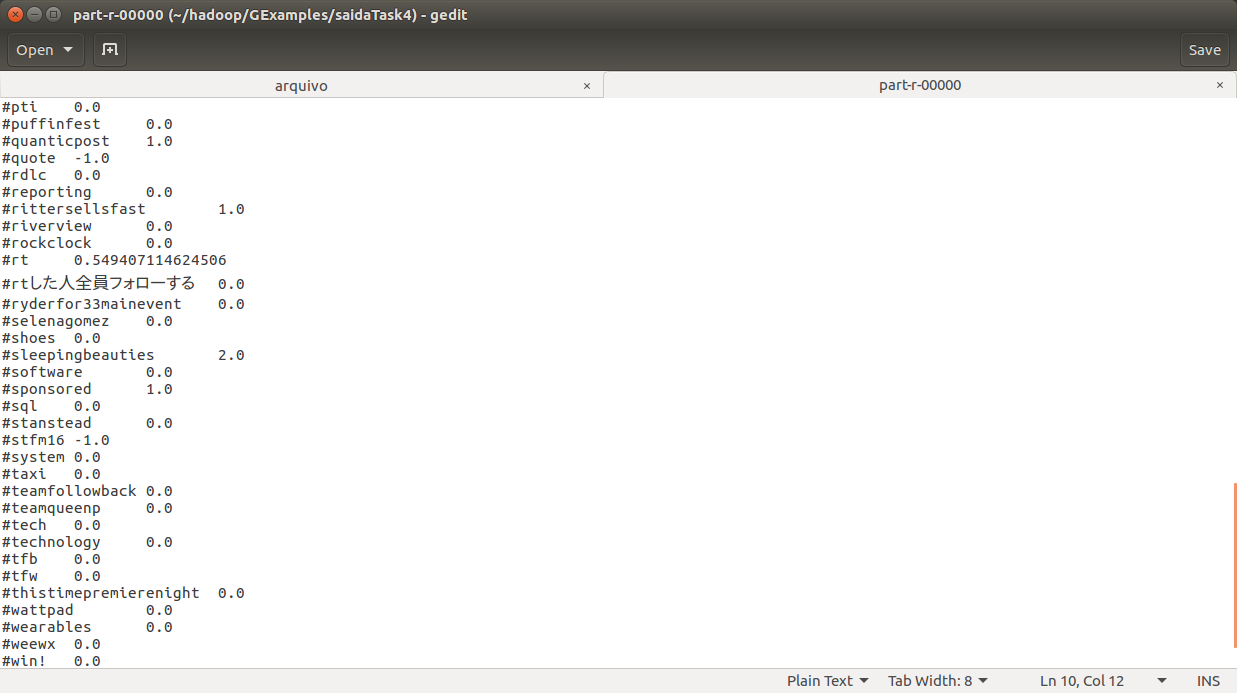


# Activity 4 - Sentiment of hashtags

The objective this activity was in each tweet get the quantity of the positive and negative in the sentence, to know if the hashtags its positive comments ou negative.

Use the similar word count, but use more two inputs documents positive and negative list, to find in each sentence how many words appears. And finally use the Weighted arithmetic to find the results.

**hadoop jar Artefactos/ActMapReduce.jar eps.examples.mapreduce.Task4 /user/tan/Act2Output1/ /user/tan/Act4Output1/ /user/tan/positive-words.txt /user/tan/negative-words.txt**



# Activity 5 - Join all previous mapreduce Job in a single application

The objective the last activity its join all the previous exercises in a one application. But the problem was use the ChainMapper concept, cause each exercise was makes a different way and I had with compatibility problem.