TDT4305 Big Data, Spring 2019

Project phase 2
Data Analysis with Spark

Due date:

4 April 2019 23:59

The project of the course TDT4305 consists of two phases. This document describes the second phase which focuses on creating a data analysis application on a large dataset. As in the previous phase, you will work with the Apache Spark framework[1] and you can choose freely which of the Spark-compatible languages you prefer: Python, Scala or Java. You can work in groups of max. two people, providing both names in your report and both should deliver the project on Blackboard.

1 Objective

Twitter[2] is a social media platform where users can post and interact with messages, known as tweets. In this task you will work with a preprocessed dataset downloaded from the publicly available Twitter Api. The aim of this phase is to create a Spark application that will find the best matching users to a given user in terms of number of words used in their tweets.

1.1 Data

A link to the dataset is available on Blackboard, under the "Prosjekt\Project" tab, and it contains the following:

tweets.tsv

It is represented in TSV format and contains the following columns:

1. **user**
2. **tweet_text**

[1] http://spark.apache.org/

[2] https://twitter.com/

2 Important notes before you start

- Try to use appropriate Spark functions with respect to the tasks you are trying to accomplish. Use of Spark and distributed execution concepts in your design will be taken into consideration, so use Spark as much as possible in your computations.
- Try not to use external libraries. The tasks in the project can be implemented with core Python/Scala/Java.
- Your design should be scalable, so **don't** use sets, lists, and other types of Python/Scala/Java collections.
- To output results into a file, use saveAsTextFile function, and format your output according to the task. The expected output format is given.
- To avoid multiple files when exporting results into a file, decrease number of data partitions using coalesce(1) or repartition(1) functions. In your report, write down which one you used and why.
- The file is preprocessed and all text is in lower case, so you don't need to care about case-sensitive comparison.
- Words in the tweet texts are separated by the space character ' '.

3 Similar Users

In this second phase of the project, you are expected to find similar users to a given queried user depending their tweet words. The intuition behind that is that users with common interests will tweet about similar topics, and use the same words. You are expected to implement a User Recommendation application in Spark to find similar users.

Your program will take four parameters:

1. the queried user
2. the k number of recommened users
3. the input file that contains the tweets
4. the output file to write the results

For the given user, your Spark program is expected to calculate, using only Spark functions, the similarity to all other users and select the k users with the maximum similarity score. For users with the same similarity score, sorting should be alphabeticaly. The similarity score for two users can be calculated according to their common words, using the equation:

$$sim(x, y) = \sum_{w \in x \cup y} min\{freq(w, x), freq(w, y)\} \tag{1}$$

The set $w \in x \cup y$ is all the words that the two users have in their tweets. The $freq(w, x)$ is the frequency of word w in user's x tweets. The results should be in the form of <recommended user>tab<number of common words>. Examples are shown bellow.

3.1 Example

Assume the following dataset:

```
mary   the apple is red
john   the apple is green
paul   green apple is nice apple
john   i like apple
dave   the pears are nice
mary   i don’t like apple
paul   i like apple
paul   apple is nice
```

In this example we want the top-2 users:

Input user:	‘mary’	‘john’	‘dave’	‘paul’
Expected Results:	john 6	mary 6	john 1	john 6
	paul 5	paul 6	mary 1	mary 5

4 Program

Create a standalone program called **recommend** that accepts four parameters with the following flags:

- -user <string user name>
- -k <number of users>
- -file <full path of the tweets file>
- -output <full path to the output file>

Example:
spark-submit recommend.py -user ‘tattoosbytaz’ -k ‘10’ -file ‘./dataset/tweets.tsv’ -output ‘./results/out1.tsv’

The queried user will be in string. The input file will be the tweets.tsv as described before. The k results in the form described before, should be written in the output file using the saveAsTextFile() method.

4.1 Delivery

You should deliver to Blackboard a ZIP file containing:

- a short report (PDF)
- source codes of your script/program with the described names (.py/.scala/.java)
- for scala and java coding, include also a *RUNNABLE* jar.

Do not include the datasets in the ZIP file.

In your report, briefly describe how your programm works and which Spark functions you used in your solution and why. You can include examples and/or code parts to help you describe better. If you use scala or java, write also the compilation and spark submit parameters. As mentioned before, you can work in groups of max. two people, providing both names in your report and both should deliver the project on Blackboard.

There will be slots for presentation (and questioning) about the project on 1-12 April.