CS 6350: Big Data Analytics and Management

Spring 2018

HW#3

Spark Streaming and Recommendation System

Deadline: 11:59 pm April 13, 2018

This homework consists of 2 parts. The first part is related to spark streaming and visualization and second part is related to recommendation system.

Part A: (Some information of Part A is taken from Spark website)

Spark Streaming Background

This question deals with real-time streaming data arriving from twitter streams.

1.1 Spark Streaming

Spark Streaming is an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams. Data can be ingested from many sources like Kafka, Flume, Kinesis, or TCP sockets, and can be processed using complex algorithms expressed with high-level functions like map, reduce, join and window. Finally, processed data can be pushed out to filesystems, databases, and live dashboards. In fact, you can apply Spark's machine learning and graph processing algorithms on data streams.

Internally, it works as follows. Spark Streaming receives live input data streams and divides the data into batches, which are then processed by the Spark engine to generate the final stream of results in batches.

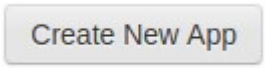
Spark Streaming provides a high-level abstraction called *discretized stream* or *DStream*, which represents a continuous stream of data. DStreams can be created either from input data streams from sources such as Kafka, Flume, and Kinesis, or by applying high-level operations on other DStreams. Internally, a DStream is represented as a sequence of RDDs.

Please review this site for details.

1.2 Source

In this assignment, we will use twitter as our stream source. To access twitter stream, please see the following instructions:

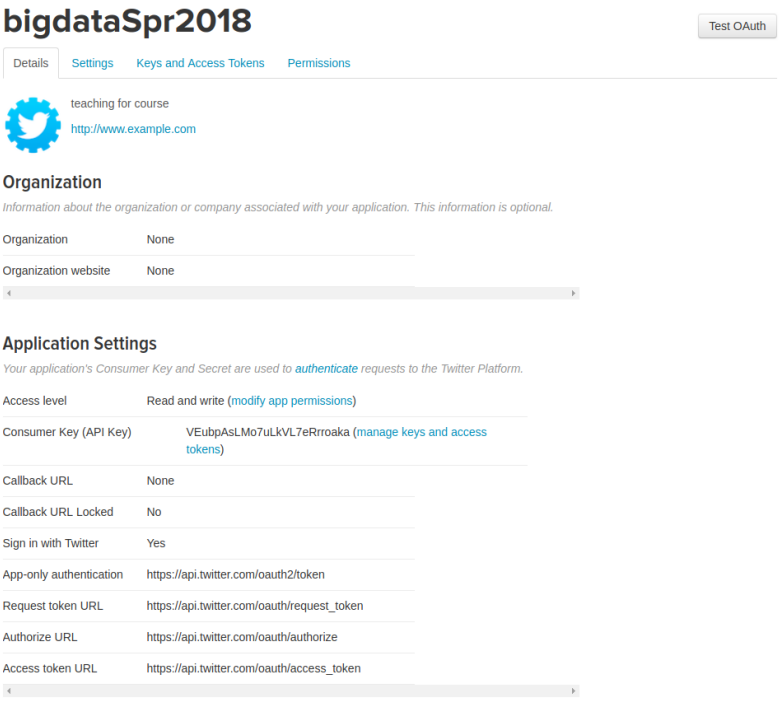
1) First create your own account at https://apps.twitter.com/

2) Click on 

to create your own twitter app

3) Filling out the form and submit

4) Once you have created your twitter app, you will observe a form as follows:



5) Click on Keys and Access Tokens, generate your own

Consumer Key

Consumer Secret

Access Token

Access Token Secret

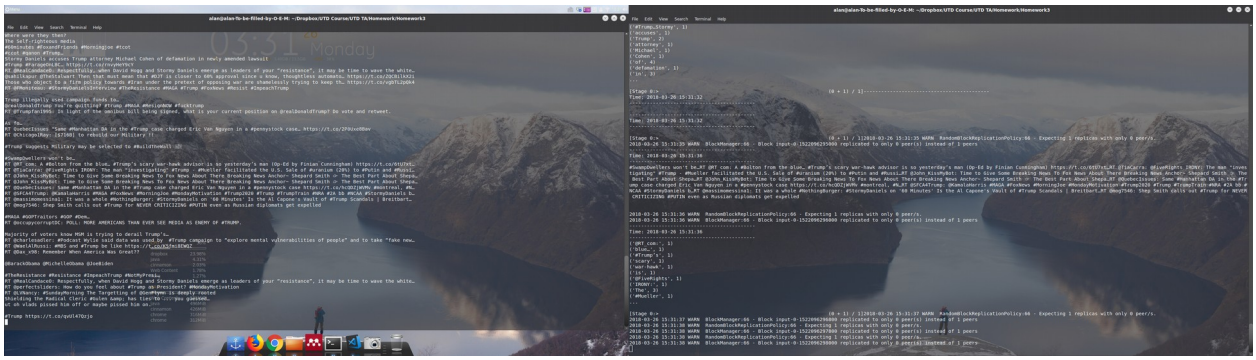
6) Crawling and Create a Stream

Crawling – refer to stream.py for details

Create a Stream – refer to spark.py for details

This setup code related to stream.py and spark.py can be found on elearning website.

Once you run these code (run stream.py first and next run spark.py), you are going to observe such a window



1.3 Task.

You are required to implement the following framework using Apache Spark Streaming, ElasticSearch and Kibana. The framework performs sentiment analysis of particular hashtags in twitter data in real time. For example, we want to do the sentiment analysis for all the tweets for #guncontrolnow and show their (e.g.,positive, neutral, negative) statistics in Kibana.

For this, first you will get the tweets via scrapper (stream.py). Next, you will write a sentiment analysis program to predict sentiment of the tweet message. Finally, you will visualize your findings using ElasticSearch/Kibana.

Scrapper -> Sentiment Analyzer/Common topic finder -> Visualizer (ElasticSearch/Kibana)

A initial setting up for python is provided. For scala, please refer to the lecture note.

1) Scrapper

We provide a sample scrapper (stream.py). However, you need to extend the code to support the following functionality.

The scrapper collects tweets and pre-process them for analytics. It is a standalone program written in Java/Python and should perform the following:

1. Collect tweets in real-time with particular hashtag. For example, we will collect all tweets with #guncontrolnow.
2. After getting tweets, we filter them by removing emoji symbols and special characters and discard any noisy tweet that do not belong to #guncontrolnow. Note that the returned tweet contains both the meta data (e.g., location) and text contents. You will have to keep at least the text content and the location meta data.
3. After filtering, you need to convert the location meta data of each tweet back to its geolocation info by calling google geo API and send the text and geolocation info to spark streaming.
4. Your scrapper program should run infinitely and should take hash tags as input parameters while running.

2) Sentiment Analyzer

Sentiment Analyzer determines whether a piece of tweet is positive, neutral or negative. For example,

> "President Donald Trump approaches his first big test this week from a position of unusual weakness." - has positive sentiment.
>
> "Trump has the lowest standing in public opinion of any new president in modern history." - has neutral sentiment.
>
> "Trump has displayed little interest in the policy itself, casting it as a thankless chore to be done before getting to tax-cut legislation he values more." - has negative sentiment.

You can use any third-party sentiment analyzer like Stanford CoreNLP or NLTK for sentiment analyzing.

In summary, for each hashtag, you perform sentiment analysis using sentiment analysis tools discussed above and output sentiment and geolocation of each tweet to some external bases (either save in a json file or send them to kibana for visualization).

3) Visualizer

Install ElasticSearch and Kibana. Create an index for visualization. Create a data table to show the sentiment of each tweet, i.e., "sentiment | tweet". Then, create a number of heat maps to show the geolocation distribution of tweets. More specifically, first heat map will show the geolocation distribution of all tweets, regardless of sentiment related to #guncontrolnow. Second and Third heat map will show geolocation distributions of positive tweets and negative tweets, respectively. When you send data from spark to ElasticSearch, you need to add a time stamp. In the dashboard, set the refresh time to 2 min as an example.

ElasticSearch and Kibana Tutorial:

https://www.dropbox.com/referrer_cleansing_redirect?hmac=ujzl8AlTA7YDWtFrtf6ejUdxegZomWxvghXeRhuXzr0%3D&url=https%3A%2F%2Fwww.oreilly.com%2Flearning%2Fa-guide-to-elasticsearch-5-and-the-elkelastic-stack

https://www.dropbox.com/referrer_cleansing_redirect?hmac=wZuGKR8yMuP89FSy6cl%2FCu4Nb%2BbTMMSYlCO6Lv2UjVA%3D&url=https%3A%2F%2Fwww.digitalocean.com%2Fcommunity%2Ftutorials%2Fhow-to-use-kibana-dashboards-and-visualizations

You should install all of these in your machine. CS6360 does not give you permission to install ElasticSearch, Kibana, etc.

Part B: Recommendation System

Use Collaborative filtering find the accuracy of ALS model accuracy. Use ratings.dat file. It contains

User id :: movie id :: ratings :: timestamp.

Your program should report the accuracy of the model.

For details follow the link: https://spark.apache.org/docs/latest/mllib-collaborative-filtering.html

Please use 60% of the data for training and 40% for testing and report the MSE of the model.