EECS4415 Big Data Systems

Fall 2018

Assignment 3 (10%): Streaming Text Analytics using Python

Due Date: 11:59 pm on Friday, Nov 23, 2018

Objective

In this assignment, you will be designing and implementing streaming applications for performing basic analytics on textual data. The data stream to be analyzed will be coming from the Twitter stream. Streaming applications may seem complex but understanding how they operate is critical for a data scientist. To allow you to explore a more complex implementation in a short period of time, you are allowed to develop code based on already existing online code snippets and libraries with proper attribution. In the scope of this assignment, you will learn:

- How to capture real-time data
- How to setup a stream processing pipeline
- How to process and get basic insights
- How to store the final processing results to a file

Important Notes:

- You must use the *submit* command to electronically submit your solution by the due date.
- All programs are to be written using Python 3.
- Your programs should be tested on the *docker image that we provided* before being submitted.
- To get full marks, your code must be well-documented.

What to Submit

When you have completed the assignment, move or copy your python scripts and outputs in a directory (e.g., assignment3), and use the following command to electronically submit your files:

```
% submit 4415 a3 <all your python scripts> <sample output in *.txt format> readme.txt team.txt
```

The `team.txt` file includes information about the team members (*first name, last name, student ID, login, yorku email*). The `readme.txt` file includes step-by-step instructions on how to run each application. You should also submit the python scripts (any *.py script) with representative names. In addition, you should submit any output file (*.txt) that example runs of your application produce. You can submit files individually after you complete each part of the assignment– simply execute the *submit* command and give the filename that you wish to submit. You may check your submission's status by:

```
% submit -l 4415 a3
```

A. Identifying Trends in Twitter (60%)

Twitter is one of the main online social networks where users post and interact with messages known as "tweets". Tweets allow for instant, short, and frequent communication and they have been proved an effective way to communicate news and other timely information. Therefore, a practical use for Twitter's functionality is to be used for identifying trends in real-time. Identifying trends is important for several industries and services, including marketing, customer service, and crisis response.

Your task is to design and implement a Twitter streaming application that tracks specific hashtags and reports their popularity (# occurrences) in real-time. In particular, you need to:

- Identify 5 related #hashtags (e.g., political parties, companies, product brands, stocks, etc.)
- Collect tweets mentioning any of the 5 #hashtags in real-time
- Compute the number of occurrences of each of the mentioned hashtags
- Plot the results of your analysis in real-time. Alternatively, you can decide to store the results in a file, post-process them as a batch (offline) and create a plot based on the post-process analysis. The results are based on the time window that your application is running (from the time it begins, until it is killed or interrupted/stopped).

For the needs of your assignment, you will need to stitch together a number of technologies that can enable the analysis to be performed, including:

A Twitter client: This is an application that connects to the Twitter service and obtains tweets as they become available. It requires to create your own credentials to access the Twitter APIs. See Appendix A.

Apache Spark Streaming: This is an apache spark streaming application that connects to your twitter client, receives the tweets as a stream, performs real-time processing of the incoming tweets, extracts useful information, and computes the quantities of interest (i.e., number of occurrences of a #hashtag).

Real-time reporting: This is a visualization component that reports through a plot the results computed by the apache spark streaming application in real-time. This can be implemented using AJAX (asynchronous HTTP calls); see the resources of the Appendix B for examples. Alternatively, results can be stored in a file in real-time, post-processed as a batch (offline), and presented as a plot.

Notes:

- Several implementation approaches exist for this application. You are encouraged to make assumptions, make decisions and follow the technical path that you feel is more appropriate. You will have a chance to explain your approach during the marking session
- Appendix A provides instructions on how to setup a Twitter application
- Appendix B provides several online resources related to the assignment

B. Real-time Sentiment Analysis of Twitter Topics (40%)

In the field of computational linguistics and natural language processing, *sentiment analysis* aims to determine the attitude of a speaker, writer, or other subject with respect to some topic or the overall contextual polarity or emotional reaction to a document, interaction, or event. A basic task in sentiment analysis is classifying the **polarity** of a given text at the document, sentence, or feature/aspect level—whether the expressed opinion in a document, a sentence or an entity feature/aspect is *positive*, *negative*, or *neutral*. For example, consider the following three text inputs:

"I love ice cream a lot"

"I dislike ice cream a lot"

"ice cream is made from milk"

One would expect that the polarity of the first is (rather) *positive*, of the second is (rather) *negative* and of the third is (rather) *neutral*. The word "rather" is used here to express subjectivity, since humans not always agree about the polarity of a sentence. We rely on an out-of-the-shelf library to perform sentiment analysis. The analysis will be on the level of a document, where the document is a tweet (i.e., all the words in a single tweet).

Your task is to design and implement a Twitter streaming application that performs sentiment analysis of tweets related to competitive topics and provides a real-time monitoring of the polarity.

- Identify 5 competitive topics (e.g., political parties, companies, product brands, stocks, etc.)
- Manually select a set of 10 hashtags that better describe each of the topic identified above
- Collect tweets related to the 5 topics in real-time and perform sentiment analysis for each topic
- Plot the results of your analysis in real-time. Alternatively, you can decide to store results in a file, post-process them and create a plot based on the post-process analysis

Notes:

- The implementation approach for the streaming application should be similar to the one followed in Part A
- For the sentiment analysis you should employ Python's Natural Language Toolkit (NLTK) library
- Appendix A provides instructions on how to setup a Twitter application
- Appendix B provides several online resources related to the assignment

Appendix A – Setting up a Twitter Application & Installing Tweepy

To start collecting tweets, you need to set up a Twitter application and get credentials that allow you to pull tweets out of the twitter streaming API. Then, you need to develop a Twitter client that connects to Twitter and acquires Twitter data. You can do that using Tweepy.

Create a Twitter Application and Obtain OAuth Access Keys

Briefly, you need to:

- Create a Twitter developer account: https://developer.twitter.com/en/apply-for-access
- Create a New Application
- Fill in your Application Details
    - *Name*: Your app name. It needs to be a unique name across all twitter applications
    - *Description*: A short description for your app
    - *Website*: The website address where the app will be hosted. Use a placeholder for now
    - *Callback URL*: Ignore this field
- Create Your Access Token
- Choose what Access Type You Need (choose 'Read only')
- Make a note of your OAuth Settings

Once you've done this, you will have the following OAuth settings.

- Consumer Key
- Consumer Secret
- OAuth Access Token
- OAuth Access Token Secret

You should keep these secret, since anyone with the keys, could effectively access your Twitter account.

Detailed information is provided here: http://docs.inboundnow.com/guide/create-twitter-application/

Install Tweepy

Tweepy is a python library for accessing the Twitter API. You can install Tweepy using pip:

```
$pip install tweepy
```

You may also use Git to clone the repository directly from Github and install it manually:

```
$git clone https://github.com/tweepy/tweepy.git
$cd tweepy
$python setup.py install
```

The next step is to use Tweepy to create a Twitter application that uses your Twitter credentials.
More information: https://github.com/tweepy/tweepy

Appendix B – Useful Online Resources and Tutorials

Spark Streaming Programming Guide
http://spark.apache.org/docs/latest/streaming-programming-guide.html

Python Streaming Examples
https://github.com/apache/spark/tree/master/examples/src/main/python/streaming

An easy-to-use Python library for accessing the Twitter API
http://www.tweepy.org/

Apache Spark General Tutorial
https://www.toptal.com/spark/introduction-to-apache-spark

Apache Spark Streaming Tutorial: Identifying Trending Twitter Hashtags
https://www.toptal.com/apache/apache-spark-streaming-twitter

Twitter Trends Analysis using Spark Streaming
http://www.awesomestats.in/spark-twitter-stream/

Apache Spark Streaming with Twitter (and Python)
https://www.linkedin.com/pulse/apache-spark-streaming-twitter-python-laurent-weichberger/

Twitter-Sentiment-Analysis-Using-Spark-Streaming-And-Kafka
https://github.com/sridharswamy/Twitter-Sentiment-Analysis-Using-Spark-Streaming-And-Kafka

Twitter Sentiment Analysis using Python
https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/

Sentiment Analysis on Reddit News Headlines with Python's Natural Language Toolkit (NLTK)
https://www.learndatasci.com/tutorials/sentiment-analysis-reddit-headlines-pythons-nltk/