MAP670G - Data Stream (2023 - 2024) - Lab Subject

Twitter Streaming Use case

In this TP, we experiment Twitter Streaming API using python and develop a streaming application that use Kafka as the caching framework.

Sign-up for the twitter Developer Platform on the following link and note down the Bearer Token.
https://developer.twitter.com/en/docs/developer-portal/overview

Note: You need to verify your mobile number on twitter to sign up for the developer account.

You can use Python Tweepy library for streaming tweets using python. More information on using tweepy library can be found at this link:
https://dev.to/twitterdev/a-comprehensive-guide-for-using-the-twitter-api-v2-using-tweepy-in-python-15d9

Simple example on streaming tweets based on a key-word.

```
import tweepy

client = tweepy.Client(bearer_token='YOUR_BEARER_TOKEN_HERE')

# Replace with your own search query
query = 'covid -is:retweet'

# Replace the limit=1000 with the maximum number of Tweets you want
for tweet in tweepy.Paginator(client.search_recent_tweets, query=query,
tweet_fields=[ 'created_at', 'lang', 'possibly_sensitive'], max_results=100).flatten(limit=1000):
    print(tweet, tweet['lang'], tweet['possibly_sensitive'])
    print('\n')
```

More information on tweet object fields can be found here:
https://developer.twitter.com/en/docs/twitter-api/data-dictionary/object-model/tweet

Practice Exercises

The objective of this TP is to stream the tweets and analyze the sentiment of tweets by analyzing the words used in the tweets.

[the scripts/code can be written in one of the following languages: Java, Scala, Python]

1. Write a script: **ingest-tweets.py** that stream tweets based on a keyword (along with needed supporting fields like created_at, lang etc.) from twitter api using tweepy library and ingest them into a kafka topic say *"raw-tweets".* The keyword for streaming tweets can be anything say, Covid, Brexit, China, Russia, Ukraine etc. The data ingested into the *raw-tweets* topic can be in json format.

2. Write a script: **filter-tweets.py** that listens to the Kafka topic *"raw-tweets"* that writes to the topic *"en-tweets"* if the tweet language is English and writes to *"fr-tweets"* topic if the tweet language is French.

3. Write a script: **sentiment-tweets.py** that listens to two Kafka topics "*en-tweets*" and "*fr-tweets"* and classify the sentiment of the tweets whether the sentiment is positive or negative.
- If the tweet sentiment is positive, write it to the topic *"positive-tweets".*
- If the tweet sentiment is negative, write it to the topic *"negative-tweets".*

4. Write a script: **archive-data.py** that archives all topics data (raw-tweets, en-tweets, fr-tweets, positive-tweets, negative-tweets) in a text file.

5. Write a script: **monitor-kafka.py** that monitors all the Kafka topics and prints the status of each Kafka topic in the console in real-time for monitoring purposes. The status information should be in the format:
Topic-name, Partition-id, offset-id, timestamp.

WARNING! Plagiarism will not be tolerated and everyone who are involved will be strictly penalised.

You need to upload a lab-scripts.zip file on the moodle containing all the scripts (without archive text files).
Number of students per group 3. **Deadline : Decembre 25th, 2023.**