INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

Big Data Algorithms

TP5: Twitter Real-Time Sentiment Analysis

We will implement a very basic application that analyses in real-time the **sentiment of users' tweets on different topics**. Given a set of keywords, we will collect the tweets containing those keywords and then decide whether the feeling of those tweets is positive or negative. We will use the **Twitter Streaming API** to access tweets in real-time and the **Apache Flink Twitter Connector** to inject and process them in Flink.

Exercise 1. Setting up the tweet access

Follow the steps here to setup the Flink Twitter Connector:

https://ci.apache.org/projects/flink/flink-docs-stable/dev/connectors/twitter.html

Note that, in order to access the Twitter Streaming API, you need a Twitter developer account with a verified mobile phone number. You can apply for a Developer Account here:

https://developer.twitter.com/en/apply/user

You will have to provide detailed information about how you will use Twitter's APIs. For the description you can indicate "*Student project working on Big Data algorithms with Apache Flink – sentiment analysis using the Twitter Connector on live tweets*". You will not publish any comments, tweets, likes or retweets. As for the results of the analysis: "*the results will not be publicly available, will only be displayed locally on the terminals used for the project, in a command terminal*".

Once your account approved, connect to the Twitter Development Platform (https://developer.twitter.com) and create a new application (Account > Apps > Create an App), in order to get the credentials (Keys and Access Tokens) needed to access the tweets in real-time.

Exercise 2. Running a first example

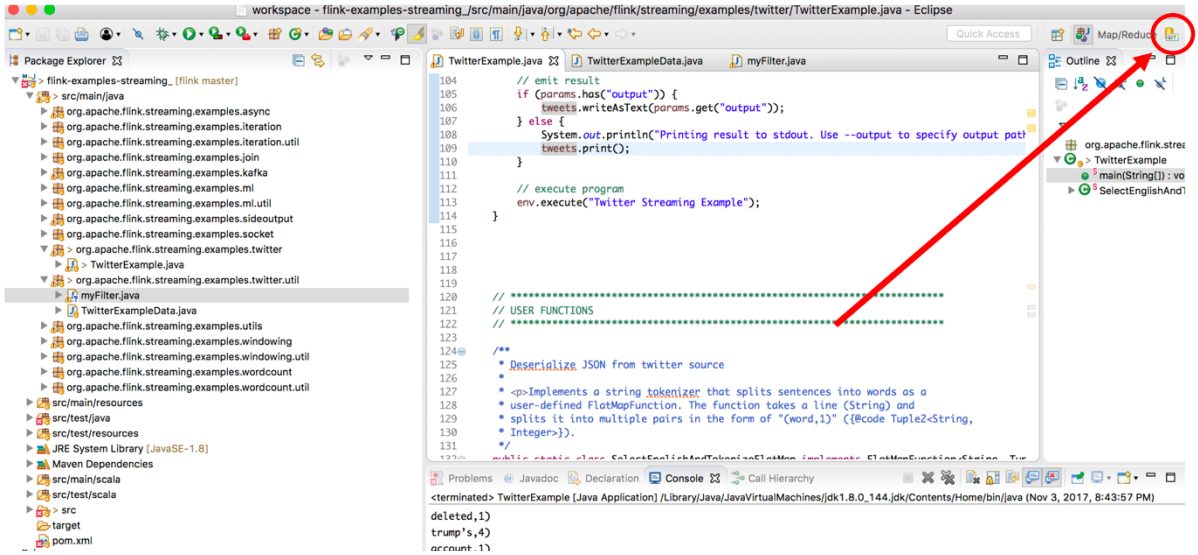
Flink comes with a simple example implementing a **word count** on a stream of tweets: `TwitterExample.java`, located in the streaming examples folder. You have two options to execute this example and do the rest of the TP:

Using the provided VM (where everything is already installed):

https://filesender.renater.fr/?s=download&token=87a45a37-51c7-a114-6e2e-961c3ba8cfe8

Using your local machine:

In order to run this example in Eclipse, first clone the Flink Git repository to your local machine. Switch to the Git view in Eclipse:



Then, in the Git repositories tab (on the left) paste the URL of the repository to clone, available here:

https://github.com/apache/flink.git

In the command line build the Maven project:

```
cd flink-examples-streaming
mvn clean package
```

Next, go back to the traditional Java view in Eclipse and import a new Maven project in your workingplace: import the whole `flink-examples-streaming` folder. Add the following line in your `pom.xml`:

```
<dependency>
        <groupId>commons-logging</groupId>
        <artifactId>commons-logging</artifactId>
        <version>1.1.1</version>
</dependency>
```

Open the `TwitterExample`, fill the execution arguments (Run > Run Configuratios > Arguments) with your data (the keys and tokens from the Twitter Development Platform) and execute the program. Modify it to display the whole tweets instead of the word counts.

Exercise 3. Collecting relevant tweets

By default, `TwitterSource` uses an endpoint (`StatusesSampleEndpoint`) that returns a **random sample of tweets**.

If you want to collect **specific tweets** (containing custom keywords, from specific users, or geo-locations), you have to change the endpoint to one that implements a custom interface:
`TwitterSource.EndpointInitializer`

Write a filter class implementing `EndpointInitializer` where you define a new `StreamingEndpoint` as a `StatusesFilterEndpoint`.

https://github.com/twitter/hbc/blob/master/hbc-core/src/main/java/com/twitter/hbc/core/endpoint/StatusesFilterEndpoint.java

The `StreamingEndpoint` class has a specific method `tractTerms` which takes as an argument a list of keywords used for filtering.

Then, in the `TwitterExample`, set this filter as endpoint (using the `setCustomEndpointInitializer` method of the `TwitterSource` class).

Exercise 4. (Basic) sentiment analysis

Tweets are collected continuously as a stream. For this exercise, collect a **limited set of tweets** (say, 100) and then extract all the words. Using the two dictionaries provided in the TP (negative and positive words) give a score to each word:

- +1 for words in the positive dictionary
- -1 for words in the negative dictionary
- 0 for words not present in any of the two dictionaries.

The final score for your set of tweets will give the (broad) sentiment of the tweets on the specific topic.

Exercise 5 (optional). (More accurate) sentiment analysis

The two previous dictionaries are very simple – the choice is binary (+/-), they don't deal with ambiguity etc. Search on the Internet for more fine grained dictionaries. For instance, instead of +/-, giving a score to each word according to its sentiment. Use this more accurate dictionary to compute a new score for the same set of tweets and compare it to the previous approach.