Collecting Real-time Tweets through Twitter's Firehose

Robert P. Schumaker*

Professor - Computer Science Dept., University of Texas at Tyler

Director - Data Analytics Lab, University of Texas at Tyler

Consultant - ORS Research Design and Data Analysis Lab

January 27, 2022

Contents

*rob.schumaker@gmail.com

1 Process Overview

This document will demonstrate how to set up a real-time Twitter collection environment and the configuration/usage of TweetScrape. Completion of this tutorial will require a Twitter developer account, application registration with Twitter, a database server and an acquisition machine for TweetScrape. Depending on the options selected, a typical research project could incur zero cost excluding local computer hardware.

TweetScrape **cannot** collect all tweets. In fact, no program can collect all tweets. GNIP, the tweet collection service of Twitter, captures only 61% of all tweets (Maddock et al., 2015). TweetScrape and programs like it are limited by the performance of your computer processor, network bandwidth and Twitter's rate-limiting. If your computer is busy, your network is too slow or you are trying to collect too many tweets in a certain period of time, data will be lost. To maximize your collection ability, it is recommended to have a dedicated acquisition machine with a stable network connection that only runs TweetScrape. A separate database machine and research environment is encouraged.

The TweetScrape program provided **does not** collect historical tweets. The program is made for real-time data collection from the Twitter Firehose and cannot collect tweets outside of that environment. Historical tweets can be collected from the Twitter API (extremely lossy, approximately 1% of tweets can be collected), other researchers, purchased from Twitter (GNIP) or purchased from a third-party service provider (George Washington University Libraries, 2017).

An overview of the process architecture used in this document is shown in Figure 1.

From this figure, a dedicated TweetScrape computer is used to filter and collect specified hashtags from the real-time tweet stream of the Twitter Firehose. Collected tweets are then lightly processed and sent in a batch to a separate database machine. From there any number of clients can connect to the database and perform research. In the least ideal environment all three tasks could be performed by one computer, however, recognize that the potential sample size of tweets collected **will be** affected.

The rest of this document will focus on creating a Twitter Developer Account,

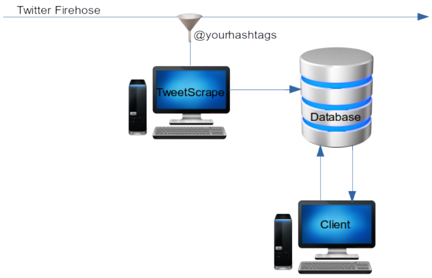

Figure 1: TweetScrape System Overview

registering your Twitter application, setting up a database environment, establishing an acquisition machine, running the TweetScrape program and understanding the data.

2 Twitter Overview

Twitter is a micro-blogging social networking service. It was created in 2006 and by 2018 had 500 million tweets daily. In 2020 Twitter reported 192 million active daily users including companies, brands, influential individuals (these three account types require "Off Twitter Notability"), government officials, nonprofit organizations, media outlets/journalists, entertainment and sports organization/figures, academics, (all require verification) as well as individual accounts.
Twitter initially limited posts to 140 characters, however, in 2017 the limit was increased to 280 and included short video.
Tweets can be directed towards individuals or organizations in one of two ways; direct message at's (i.e., '@') or hashtags (i.e., '#') respectively. Each tweet will contain specific information such as the unique identifier of the tweet, username of the Tweeter, timestamp of the tweet, tweet content, whether it is a retweet and geographic coordinates (if available).

2.1 Firehose

When a user sends a tweet, it first interacts with the Twitter servers to append tweet-specific information before going out on the Firehose. The Firehose sends out the tweet once. To access the Firehose a user will need a software-based listener that only collects those tweets matching certain criteria like a list of hashtags. When a tweet matches the criteria, the acquisition machine will then collect the tweets until the session ends or the network connection is disrupted.

2.2 Developer Account Access

Because the Firehose is a special environment, Twitter requires users to be approved for developer access. This means that you will be responsible for an application and following the terms of service. Twitter is quite strict on adhering to their terms of service and lengthy account disruptions or termination can occur if you are suspected to be in violation. They do not require you to write your own application, nor do they test your programming ability. Quite literally anyone can apply for a developer account.

2.3 OAuth

OAuth is the Open Authorization standard used by many tech companies to allow third-party applications to share account access safely and securely. The TweetScrape application will require OAuth credentials to access the Twitter Firehose. When you set up an application through the Twitter Developer portal, you will be given OAuth credentials. **Copy them and keep them safe.** Whoever has the credentials has your account access! They can post spam, cause trouble and have your account suspended. Keep the credentials safe. In particular you will have four pieces of information; consumer key, consumer secret, access token and access token key. The consumer key and secret can be thought of as similar to a username/password combination. It is used to authenticate the user. The access token and secret can be thought of as an application-level username/password combination. If a developer were to have multiple applications, the consumer key and secret would remain constant and the access token and secret would differ.

3 Tutorial - Collecting Real-time Tweets through Twitter's Firehose

3.1 Twitter Developer Account Setup

Collecting tweets from Twitter is free, however, it is not anonymous. Twitter will need to know some information about you and your application. Depending upon your account and application choices, approval from Twitter can be anywhere from instant to months. A word of caution, Twitter relies almost exclusively on bots. If you run into an account-related problem, you will likely never talk with a human so do whatever the bot tells you even if it conflicts with their terms of service (been there, done that).
The first step is to create a Twitter Developer account. This will require you to have a regular Twitter user account.

1. If you do not have a Twitter user account, go to **https://twitter.com** and create an account

2. Go to **https://developer.twitter.com**. Apply for a developer account and log in using your Twitter user account credentials. Be aware that account verification could take anywhere from minutes to months. Plan accordingly

3.2 Twitter Developer Application Setup

On the Twitter Developer end you will need to create a *project* and an *application*. Applications are the workhorse of projects. Projects also have differing access levels; essential, elevated and academic research. *Essential Access* allows for an application (1 max) to collect 500,000 tweets per month and has virtually no waiting period. This access level is useful for anticipated small research demands or those that need fast access. *Elevated Access* allows for applications (3 max) to collect 2,000,000 tweets per month. There could be a waiting period for this request. This access level is useful for those that need slightly more tweets but don't qualify for academic access. *Academic Research Access* allows for an application (1 max) to collect 10,000,000 tweets per month. There is a waiting period for this request (days/weeks). This access level is the highest published level, although specific custom requests can be made.

1. Log into your Twitter Developer Account (https://developer.twitter.com)

2. Click **Dashboard** on the left if the Dashboard isn't already visible
3. Click **New Project**. Fill in the *Project Name*, *Project Description* and *Use Case* boxes
4. You will also be asked *Project Access Level.* Please choose the one most appropriate to your needs (see the intro paragraph to this section for access level differences)
5. Once the *Project* has been approved, click **Add App**
6. Click **Create new**
7. Provide an *App name* for you to identify it. Click **Next**
8. Copy the *API Key* and *API Key Secret* to a safe place
9. Click **App settings**
10. At the top center, click **Keys and tokens**
11. Near *Access Token and Secret*, click **Generate**
12. Copy the *Access Token* and *Access Token Secret* to a safe place

3.3 mySQL Database Setup

In order to store tweet data, you will need a mySQL database. This can be either held on a local machine (preferably different from your TweetScrape machine) or by a Cloud provider. This database machine must be always on and connected to the Internet 24/7. A local machine can cost less, provided you have an extra computer, and be easier to setup/configure. A cloud machine can be more robust and easier to share data access with colleagues. Instructions for both local machine and cloud provider setup are provided below. You only need to pick one.

3.3.1 Local Machine

These instructions assume you already have an operating system installed.

1. Go to **https://dev.mysql.com/downloads** and download the mySQL Community Server

2. Install mySQL Community Server – choose Full setup (for a minimal install choose mySQL Server and mySQL Workbench)

3. For configuration questions, choose the defaults and select a mySQL admin/root password

4. Go to the tutorial section, **Database Configuration using mySQL Workbench**, to continue setup

3.3.2 Amazon AWS

Almost any cloud provider with database instances would be suitable (Amazon AWS, Microsoft Azure, Google Cloud, etc). Keep in mind that cloud-based resources are not free, but aren't expensive either. Cloud providers may also have trial periods or discounted rates.

In the previous talk, *Setting up an Amazon Web Services (AWS) and RStudio Data Connective Environment*, sections 3.1-3.3 will walk you through the setup process for a mySQL instance in AWS. Please see the ORS Data Science Resources webpage at *https://uttyler.edu/research/ors-research-design-data-analysis-lab/resources/data-science* or reach out to the author for the tutorial.

3.3.3 Database Configuration using mySQL Workbench

1. Open mySQL Workbench

2. Click the plus sign in a circle to create a connection to the database server

3. For **Connection Name** pick a name for your connection. The name doesn't really matter, it's to help you find it

4. For **Hostname** on a local machine, enter 127.0.0.1

5. For **Hostname** on an Amazon RDS instance, enter the RDS endpoint (e.g., oceanplatform0.cdb7tnix15tn.us-west-2.rds.amazonaws.com)

6. For **Username**, enter *root*

7. Click **Test Connection** and enter the mySQL admin/root password

8. If you successfully made the mySQL connection, click Ok to exit the **Setup New Connection** dialog box

9. If you were unsuccessful making the mySQL connection, find a file named *mysqld.cnf*, open it for editing, put a # symbol in front of *bind-address = 127.0.0.1*, and either restart the mysql process or reboot the computer

10. Click your new connection under **mySQL Connections**

11. Click **File**, **Open SQL Script** and open *TweetScrape_ dbsetup.sql*

12. Modify line 19 onwards for your chosen hashtags. To reduce the chance of Twitter rate limiting, keep it less than 30-50 hashtags for chatty hashtags

13. Modify near line 6836, select a username/password for TweetScrape

14. If you change the username, be sure to rename 'tweetscrape' to your new username in the five lines near line 6836

15. Click the lightning bolt to execute the script. If everything goes well you should have all green checkmark circles in the **Action Output** window

3.4 Setting up the Acquisition Machine

The acquisition machine is notably one of the more important pieces for data collection. It must be always on, have a stable network connection and be able to process data quickly. Maximizing these elements will maximize tweet gathering.

3.4.1 Java Environment Setup

The TweetScrape program in written in the Java programming language which allows for cross-platform compatibility. This means it can run in Windows, macOS or Linux. Just about any operating system supported by the Java Runtime Environment (JRE) will work. The JRE interprets the java code specific to your computer's architecture and passes the bytecode in a form your computer can understand. The technical details are unimportant, but having the JRE installed on your acquisition machine is key. For this documentation we will be using OpenJDK, an open-source version of the Java Developer Kit (JDK), which contains the JRE tools plus some tools on the developer side. If you are developing this application for commercial purposes, you are legally required to use Oracle's commercial JDK version instead.

1. Go to **https://jdk.java.net** and click on the *JDK* in **Ready for use**
2. Download the JDK specific to your acquisition machine's architecture
3. Install the JDK
4. Test the JDK by opening a command prompt/terminal and typing **java -version**
5. If everything goes well, you should see some helpful informational text regarding OpenJDK

3.4.2 Download and Configure TweetScrape

The TweetScrape application is text-based and requires no installer. However, there will be some configuration necessary. The user should have their following materials ready:

- OAuth consumer key and secret as well as access key and secret
- IP address or canonical location of their database
- Database username and password
- Database schema location for hashtags and tweets

The configuration can initially seem overwhelming. However, a careful review of these instructions should be helpful. If you find yourself stuck, reach out to the author.

The configuration file is broken into thirteen major parts:

- Location of your java interpreter (from OpenJDK you installed earlier)
- Classpath to the program libraries. The path should be identical for all of the jar files. Just a lot of copy/pasting or Edit/Replace
- The package.program to run, in this case it is robschumaker.tweetscrape.TwitterMain and does not need modified
- The next four parts in double quotes are OAuth consumer key and secret as well as access key and secret, respectively
- Database location and schema
- The next two parts are the database username and password respectively. Yeah it's plaintext.

- The next two parts are the database table of hashtags and the field within that table that contains the hashtags. If you didn't modify the sql script from earlier, this shouldn't need changed
- The last part is the table for tweets to be stored

What to do...

1. Download the *TweetScrape.zip* file from the ORS Resource webpage
2. Unzip the *TweetScrape.zip* file and move the TweetScrape folder to a directory location of your choosing
3. Open the *TweetScrape* folder and open the *TweetScrape.sh* file for editing. It is a text file, so just about any text editor will work
4. The first line is #**!/bin/bash**. This has special meaning in linux-based operating systems but not so much in Windows. If your acquisition machine is Windows-based, remove this line
5. The first section */usr/lib/jvm/jdk-17.0.2/bin/java* is the file location of your java interpreter. This file path **must be changed**. You must find the java interpreter and copy it's path. On my Windows machine I would change it to *"C:\Program Files\jdk-17.0.2\bin\java"* with the double-quotes
6. The next section is the path to the TweetScrape folder. My classpath in linux is */home/rschumaker/GoogleDrive/SourceCode/*. On my Windows machine it is *C:\Users\robsc\Desktop\TweetScrape\*. You need to find the exact path to your TweetScrape folder. Replace **all** instances of /home/rschumaker/GoogleDrive/SourceCode/ with your path
7. If you are using a Windows machine, change all colons in the classpath area to semicolons. Please do not use **Replace All** as there are colons outside of the classpath area
8. After *robschumaker.tweetscrape.TwitterMain* are four double-quote parameters for OAuth. Enter your consumer key, consumer secret, access key and access secret in each respectively
9. Next is the location of your database and schema. Enter the IP address or endpoint information to your database. If you are using a schema different from the *TweetScrape_ dbsetup.sql* schema, please enter it after the *3306:/*

10. Enter your database username and password. You set these values in *TweetScrape_ dbsetup.sql* near the bottom of the script

11. Enter your Hashtag table and Hashtag field from within your schema. If you didn't change these values in the *TweetScrape_ dbsetup.sql* then leave the defaults

12. Enter your Tweet table from within your schema. If you didn't change this value in the *TweetScrape_ dbsetup.sql* then leave the default

13. Save the file and close it

14. If you are using a Windows machine, rename the file extension to **.bat**

3.5 Running TweetScrape

Congratulations, you made it to this point. Assuming the database server is correctly setup, OAuth credentials are correct, OpenJDK is installed, TweetScrape configuration is perfect and God loves you, TweetScrape is ready to run.

1. On linux-based machines, use terminal to navigate to the TweetScrape directory and type *sh TweetScrape.sh*

2. On Windows-based machines, use the command prompt to navigate to the TweetScrape directory and type *TweetScrape.bat*

3. You should now see tweets matching your hashtags scrolling by. Every 100 tweets will be batch transacted to the database

4. If an error occurs, the error can sometimes be helpful in tracking down where the problem exists (database, OAuth or database credentials, file locations, table or field names, etc). You may need to be resourceful

3.6 Understanding the Data

Collected tweets can be found in the Tweet table of your database. The structure of the table is as follows:

1. TweetID - this is the unique identifier of a specific tweet assigned by Twitter. It is stored as a varchar(35), however, it can be treated as a bigint. It is also the primary key

2. Tweeter - the self-chosen name of the author of the tweet. It is stored as a varchar(15)
3. TweetDate - the timestamp of the tweet. It is stored as datetime and follows the yyyy-MM-dd hh:mm:ss format
4. TweetContent - the tweet message. It is stored as a varchar(300)
5. Retweet - whether the tweet is original (0) or a retweet of someone else's tweet (1). It is stored as a bit
6. TonePos - an integer count of positive terms found in the TweetContent. It is stored as an int
7. ToneNeg - an integer count of negative terms found in the TweetContent. It is stored as an int
8. Hashtag - the hashtag associated with the tweet. It is stored as a varchar(15)

To determine whether a tweet is positive vs negative (or neutral), you can use $TonePos > ToneNeg$ for positive tweets, $TonePos < ToneNeg$ for negative tweets and $TonePos = ToneNeg$ for neutral tweets.

References

George Washington University Libraries. (2017, September 14). *Where to get twitter data for academic research*. Retrieved July 21, 2021, from https://gwu-libraries.github.io/sfm-ui/posts/2017-09-14-twitter-data

Maddock, J., Starbird, K., & Mason, R. (2015). Using historical twitter data for research: Ethical challenges of tweet deletions, In *Cscw'15 workshop on ethics at the 2015 conference on computer supported cooperative work*, Vancouver, Canada.