Twitter Data Scraping Tutorial

Abstract
This tutorial walks you through installing Anaconda, GetOldTweets3, and details how to scrape data and then manipulate it within Excel to prepare the dataset for analysis.
Table of Contents

Introduction ................................................................................................................ 2
Install Anaconda with Jupyter Notebook ............................................................................. 4
Install GetOldTweets3 Library ............................................................................................ 9
Launch the Data Scraper Jupyter Notebook ............................................................................. 11
Run the Data Scraper ........................................................................................................ 13
  Initialization of the Process ............................................................................................ 13
  Text-based Query .......................................................................................................... 14
  Username Query .......................................................................................................... 16
Open the Dataset in Excel and Prepare it for Analysis ...................................................... 18
  Unique ID Changes ...................................................................................................... 19
  Splitting Date and Time ............................................................................................... 21
  Fixing the Mentions Error ........................................................................................... 25
Concluding Thoughts ....................................................................................................... 26
Introduction

This Twitter Data Scraping tutorial will step you through the process of setting up a Python environment and how to use the supplied Jupyter Notebook to collect tweet data. The instructions and screen shots are shown in a Microsoft Windows environment, but the programs used also exist for Mac and Linux operating systems.

The concept behind using this Python data scraper is to remove the need for you to register for a Twitter developer account, and also to give you access to all past Twitter data. Many of the other methods of tweet collection limit you to retrieving only the past week’s tweets and you would need to plan far ahead and set up a retriever, such as TAGS, to collect data over time.

This tutorial and the methods it will teach you will allow you to retrieve historical Twitter data from any point since Twitter’s inception. The following is a snapshot of the first tweets of the developers when Twitter launched in 2006. These tweets were retrieved using the methods detailed in this tutorial.

The first tweets (in reverse date/time order)

Some assumptions were made when designing this tutorial to allow it to be as comprehensive as possible for a very broad audience. The explanations are written for users who:

1. have little to no experience with Python
2. do not have Python and Jupyter Notebook installed
3. are familiar with Twitter and the concept of hashtags and usernames
4. have a hashtag or username and date range of interest for research
5. have some familiarity with Excel

If you are already familiar with parts of these concepts (Python/Excel) please just skim the instructions where you feel confident with the process.
There are a set of outcomes for this Twitter data scraping tutorial and the instructional material is separated into these goals:

1. Install Anaconda with Jupyter Notebook
2. Install GetOldTweets3 library
3. Launch the data scraper Jupyter Notebook
4. Run the data scraper
5. Open the dataset in Excel and prepare it for analysis

Some of the steps to follow are embedded in the main text of the instructional material, while others are in the captions of the figures. **Bolded text** has been used to draw your attention to items to do. In some of the illustrations, a **yellow cursor** is visible to indicate what to select.
Install Anaconda with Jupyter Notebook

Anaconda is an open-source distribution package for Python, R, and other data science resources. Included with the bundled Anaconda installation is Jupyter Notebook, a web-based application which allows users to create documents that include live code and other documentary materials.

Navigate to: https://www.anaconda.com/products/individual

At the bottom of the page are the links to the installers for the various operating systems.

<table>
<thead>
<tr>
<th>Anaconda Installers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Windows</strong></td>
</tr>
<tr>
<td>Python 3.7</td>
</tr>
<tr>
<td>64-Bit Graphical Installer (456 MB)</td>
</tr>
<tr>
<td>32-Bit Graphical Installer (423 MB)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>MacOS</strong></td>
</tr>
<tr>
<td>Python 3.7</td>
</tr>
<tr>
<td>64-Bit Graphical Installer (442 MB)</td>
</tr>
<tr>
<td>64-Bit Command Line Installer (430 MB)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Linux</strong></td>
</tr>
<tr>
<td>Python 3.7</td>
</tr>
<tr>
<td>64-Bit (x86) Installer (522 MB)</td>
</tr>
<tr>
<td>64-Bit (Power8 and Power9) Installer (276 MB)</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

**Anaconda Installer Options**

Select the appropriate Python 3.7 installer for your operating system to save the installer to your computer.

Launch the installer. The installation wizard will move you through the various steps and allow you to adjust a few options. The sequence of screens is shown below.
Welcome Screen – Click Next

Licensing Screen – Click Next
Installation Type – If you share your computer with other people and have admin privileges, select All Users, otherwise leave the default. Click Next

Installation Path – Typically leave the default unless you need to install it on another drive. Click Next
**Advanced Options** – leave these options as is and click **Install**

**Progress Bar** – let the installation run to completion
Completion Screen – **Uncheck** the tutorial/learn more boxes and select **Finish**

This completes the “Installation of Anaconda and Jupyter Notebook” section of the tutorial.
Install GetOldTweets3 Library

The GetOldTweets3 library is open-source and contains the Python code needed to scrape the data using the methods contained in this tutorial. Dmitry Mottl branched this library from Jefferson Henrique’s code. We will be using the Anaconda Powershell command prompt to install the library, but if you are interested in reading about the library, you can access the documentation on GitHub (https://github.com/Mottl/GetOldTweets3).

Select the Anaconda Powershell Prompt from the Start menu.

Windows Start Menu – Click Anaconda Powershell Prompt

The Anaconda Powershell prompt is a command window that allows you to run programs as you would from a normal command prompt. However, it has the environment set to be able to run Python and other scripts. The figure below shows the command you need to enter to install the GetOldTweets3 library.

Enter the following command: `pip install GetOldTweets3`
Installation messages from pip

The installation will display the steps it took to download and install the library. When it is completed, enter exit at the prompt to close the window.

This completes the “Install GetOldTweets3 Library” section of the tutorial.
Launch the Data Scraper Jupyter Notebook

The Jupyter Notebook that you will use to scrape Twitter data was originally created by Martin Beck (https://towardsdatascience.com/@its.martin.beck). I have modified it for this tutorial to include different options and to document more fully the steps needed to run the scraper.

**Navigate to:** [http://chdr.cah.ucf.edu/neh-digculture/NEH-DigCulture-TweetScraper.zip](http://chdr.cah.ucf.edu/neh-digculture/NEH-DigCulture-TweetScraper.zip) to download the notebook and unzip the file. Place it on your desktop for ease of access.

**Select** Jupyter Notebook from the Start menu.

*Windows Start Menu – click on Jupyter Notebook*

The notebook will launch a local server instance to support the process and then it will launch the web app for Jupyter Notebook.
The web app for the notebook will launch in your default browser and display folder navigation options. It defaults to the Desktop and if you unzipped the notebook there you should see it in the list. If you placed it elsewhere, click on the folder next to the Desktop link and navigate to the correct location.

This completes “Launch the Data Scraper Jupyter Notebook” section of the tutorial.
Run the Data Scraper

The tweet scraper notebook contains two options for collecting tweets; one that uses a text query search and another to collect the tweets of a specific user.

Jupyter notebooks are a set of “cells” that may contain comments (such as the title block below) or code to execute (the highlighted cell).

The code in a cell can be run by pressing the Run button in the toolbar, or by pressing Ctrl-Enter on the keyboard.

The following explanations will step through each of the executable cells in the notebook and describe what to expect for outcomes and what to do if something fails.

Initialization of the Process

The compartmentalization of code within a notebook allows you to run sections separately. The first section, Initialization of the process, loads two libraries, GetOldTweets3 (which we installed) and pandas (which was installed with Anaconda). When a code cell is executed using the Run button or Ctrl-Enter, the square brackets to the left of the code (showing [1] in this example) will change to an
asterisk (*) while the code executes. Since this cell is only two lines of code you may blink and miss the asterisk. You will notice it more later when the data is being scraped.

Text-based Query

As each of the appropriate code cells are executed, the code is loaded into memory and is then available to other code within the notebook. The next code cell in the notebook contains a function that runs the query on the Twitter data and creates a CSV file that contains the results.

This function will be run (loaded into memory) before we execute the code that defines what our search parameters will contain.

By looking at the comments within this code cell, you will see that 4 parameters are passed to the function: text_query (the search terms), start_date (beginning of date range), end_date (ending of date range) and a count. The count constrains the number of tweets requested through the Twitter API. There is a variable limit for the number of tweets you can ask for in a single query. Some documentation says you can retrieve up to 18,000 per query. Typically, I can retrieve about 10,000 every 15-20 minutes without the process failing.

As you work with this scraper and find you have criteria that push against this limit, think about breaking the queries up by day, by a single hashtag, etc. to reduce the size of the dataset retrieved in a single query. You can combine multiple datasets in Excel afterwards as described later in this tutorial.
Text query code cell – **Click within this cell to make it active and Run (Ctrl-Enter)**

Once the text query code cell has been run, we will set the criteria for the tweet scraping and retrieve the data.

In the code cell below, you will see the 4 parameters to set. In this example, I am retrieving tweets that use the hashtag #covid19 and include the textrealDonaldTrump. This combination query looks for Trump in any context: username, tweet content, or mentions, regardless of whether someone used the @ username or the # hashtag symbol.

The date parameters require some caveats.

1. The until_date (Twitter’s variable name) needs to be your end date + 1. In the example below, the last date in the range that the API will send back will be 15 March 2020.
2. The Twitter API will return query results from the until_date back towards the since_date (i.e., end date to start date). This means that if your query hits the count limit before the query finishes traversing all of the dates in your range, you may get, in this example, say 5,000 tweets from 15 March 2020, 5,000 tweets from 14 March 2020, and none from the rest of the days in the range. You will need to examine your dataset if you are querying over multiple days to ensure that all your requested data is retrieved. If you do not get all the expected data, run single days individually and combine the data afterwards.

---

**Text query process**

This is the actual process that calls the above query function

```python
# This cell will be modified for your search criteria and then run to gather the data
# Make sure to Ctrl-S after changing the criteria within this function to save your changes
# text_query can contain hashtags (e.g. #metoo, #covid19)
# it can also include boolean AND/OR (e.g. #covid19 OR #coronavirus)
text_query = '#covid19 AND realDonaldTrump'

# the count is the maximum to return, Twitter will stop large requests and the process will error out
# typically, 16,000 processes successfully. Up to 12,000 has run to completion also. If larger counts are run
# please approximately 15 minutes between runs to ensure twitter doesn't respond with an error
# count = 10000

# the until_date variable should be set to the day following your range
# for example, to include data from 1 March 2020 to 15 March 2020, set since_date = '2020-03-01' and until_date = '2020-03-15'
since_date = '2020-03-01'
until_date = '2020-03-15'

# call to the query function that will create the resultant CSV file
# text_query_to_csv(text_query, since_date, until_date, count)
```

---

**Process to retrieve tweets**

Modify the text_query, since_date, and until_date in this code cell to be your research parameters.

Save your changes to the code by using the **Save icon** in the toolbar or pressing **Ctrl-S**.

Run the cell.
As this function takes time to run, you will notice the [*] displayed as the code executes and you will see that it is completed when the asterisk is replaced with a number. This number denotes the number of times cells are executed in a session.

You will also notice a CSV file will be saved to your desktop. The name of the file will be your text_query and the number (in thousands) of the count you requested (not the actual count of returned tweets).

If the process has an error, the asterisk will be replaced by a number, but your CSV file will not appear. If you look below the code cell, you will find the error information. As mentioned previously, Twitter constrains the number of queries. If you stay within the 10,000 count and a 1520 minute interval between large queries (ones that approach the 10,000 tweet boundary) you shouldn’t have any issues.

If you see this error, you are running too many large queries to quickly. Wait 20 minutes and try again.

Too Many Requests error

You may also get this one if Twitter just had a momentary glitch. You can retry immediately if you get this error.

Service Temporarily unavailable error

The errors that may be displayed list out all of the Python traceback code. Just ignore it. The main thing to note is whether the issue is a 429 (wait 20 minutes and try again) or a 503 (just try again). When you run the code cell again, the error information will be cleared.

Username Query

The remaining two code cells in the notebook are used for queries by Twitter username. To collect the tweets of a specific user, the code cell that runs the query by username needs to be loaded into memory like we did for the text-based query function.

The difference here is that the parameter is the username instead of the text criteria.
### Using a username-based search to collect tweets

The following notebook cell sets up a function that collects the tweets based on a specific username and saves to a CSV file.

```python
Username query – Click in the code cell and Run (Ctrl-Enter)
```

The last cell in the notebook is the function to set the query by username parameters and run the actual retrieval. Modify the username, count, and date parameters as necessary for your research.

```python
Query by username – Click in code cell and Run (Ctrl-Enter)
```

You will now have one or more CSV files with Twitter data. These files will need some manipulation on Excel to make them ready for analysis.

Remember to close the Jupyter notebook server window by clicking on the X to close it.

This completes “Run the Data Scraper” section of the tutorial.
Open the Dataset in Excel and Prepare it for Analysis

The data scrapers described above create CSV files containing the tweet data. When you initially open the file in Excel it will look like this:

<table>
<thead>
<tr>
<th>Col.</th>
<th>Content</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>count</td>
<td>It is an incremental number of the count of tweets in the file and is zero-based</td>
</tr>
<tr>
<td>B</td>
<td>ID</td>
<td>The unique ID of the tweet. This column is useful for de-duplicating data that has been collected via multiple queries. This technique will be outlined later in the tutorial. When the CSV is opened initially, this appears in exponential notation.</td>
</tr>
<tr>
<td>C</td>
<td>Datetime</td>
<td>This is the date/time of the tweet. Typically, I am interested in just the date and will explain later how to split this column into its constituent parts.</td>
</tr>
<tr>
<td>D</td>
<td>Text</td>
<td>The actual textual content of the tweet without emoji (these were excluded)</td>
</tr>
<tr>
<td>E</td>
<td>User</td>
<td>The username on the account sending the tweet</td>
</tr>
<tr>
<td>F</td>
<td>To</td>
<td>The username(s) the tweet was sent to in a reply</td>
</tr>
<tr>
<td>G</td>
<td>Retweets</td>
<td>The number of times the tweet was retweeted</td>
</tr>
<tr>
<td>H</td>
<td>Favorites</td>
<td>The number of times the tweet was liked</td>
</tr>
</tbody>
</table>

CSV in Excel

The table below describes the content of these columns and some comments outlining manipulation we will do to make them more useful/visually understandable before saving the data as an Excel spreadsheet.
### Twitter Data Scraping Tutorial

Amy Larner Giroux

#### Mentions

Other Twitter usernames mentioned in the tweet. Currently the column in the example shows an Excel error of #NAME?. This will be explained below.

#### Hashtags

A list of the hashtags included in the tweet.

### Unique ID Changes

The ID is shown in exponential notation. To change, highlight the column, click on the format dropdown and select Number, and then use the decimal place decreasing button to remove the two decimal places.

![Changing the ID column from exponential to integer](image)

When combining multiple datasets in one spreadsheet, you will want to remove any duplicates. Copy and paste the rows into one spreadsheet as below:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1994</td>
<td>1234567890</td>
<td>1234567891</td>
<td>1234567892</td>
<td>1234567893</td>
<td>1234567894</td>
<td>1234567895</td>
<td>1234567896</td>
<td>1234567897</td>
</tr>
<tr>
<td>1995</td>
<td>1234567898</td>
<td>1234567899</td>
<td>1234567900</td>
<td>1234567901</td>
<td>1234567902</td>
<td>1234567903</td>
<td>1234567904</td>
<td>1234567905</td>
</tr>
<tr>
<td>1996</td>
<td>1234567906</td>
<td>1234567907</td>
<td>1234567908</td>
<td>1234567909</td>
<td>1234567910</td>
<td>1234567911</td>
<td>1234567912</td>
<td>1234567913</td>
</tr>
<tr>
<td>1997</td>
<td>1234567914</td>
<td>1234567915</td>
<td>1234567916</td>
<td>1234567917</td>
<td>1234567918</td>
<td>1234567919</td>
<td>1234567920</td>
<td>1234567921</td>
</tr>
<tr>
<td>1998</td>
<td>1234567922</td>
<td>1234567923</td>
<td>1234567924</td>
<td>1234567925</td>
<td>1234567926</td>
<td>1234567927</td>
<td>1234567928</td>
<td>1234567929</td>
</tr>
<tr>
<td>1999</td>
<td>1234567930</td>
<td>1234567931</td>
<td>1234567932</td>
<td>1234567933</td>
<td>1234567934</td>
<td>1234567935</td>
<td>1234567936</td>
<td>1234567937</td>
</tr>
<tr>
<td>2000</td>
<td>1234567938</td>
<td>1234567939</td>
<td>1234567940</td>
<td>1234567941</td>
<td>1234567942</td>
<td>1234567943</td>
<td>1234567944</td>
<td>1234567945</td>
</tr>
<tr>
<td>2001</td>
<td>1234567946</td>
<td>1234567947</td>
<td>1234567948</td>
<td>1234567949</td>
<td>1234567950</td>
<td>1234567951</td>
<td>1234567952</td>
<td>1234567953</td>
</tr>
<tr>
<td>2002</td>
<td>1234567954</td>
<td>1234567955</td>
<td>1234567956</td>
<td>1234567957</td>
<td>1234567958</td>
<td>1234567959</td>
<td>1234567960</td>
<td>1234567961</td>
</tr>
<tr>
<td>2003</td>
<td>1234567962</td>
<td>1234567963</td>
<td>1234567964</td>
<td>1234567965</td>
<td>1234567966</td>
<td>1234567967</td>
<td>1234567968</td>
<td>1234567969</td>
</tr>
<tr>
<td>2004</td>
<td>1234567970</td>
<td>1234567971</td>
<td>1234567972</td>
<td>1234567973</td>
<td>1234567974</td>
<td>1234567975</td>
<td>1234567976</td>
<td>1234567977</td>
</tr>
<tr>
<td>2005</td>
<td>1234567978</td>
<td>1234567979</td>
<td>1234567980</td>
<td>1234567981</td>
<td>1234567982</td>
<td>1234567983</td>
<td>1234567984</td>
<td>1234567985</td>
</tr>
<tr>
<td>2006</td>
<td>1234567986</td>
<td>1234567987</td>
<td>1234567988</td>
<td>1234567989</td>
<td>1234567990</td>
<td>1234567991</td>
<td>1234567992</td>
<td>1234567993</td>
</tr>
<tr>
<td>2007</td>
<td>1234567994</td>
<td>1234567995</td>
<td>1234567996</td>
<td>1234567997</td>
<td>1234567998</td>
<td>1234567999</td>
<td>1234568000</td>
<td>1234568001</td>
</tr>
</tbody>
</table>

#### Merged data files

Select all rows in the file (Ctrl-A) and from the Data tab, click on Remove Duplicates.
Remove Duplicates

When the Remove Duplicates dialog is shown, make sure that the **My data has headers** is checked, and then uncheck (Column A) as that column is not unique.

Remove Duplicates – **Uncheck column A and then click on OK**

Excel will show you how many duplicate rows were removed.

Results of Deduplication

If you have merged multiple files together, Column A is no longer unique. You may delete the column. The figures throughout the rest of the tutorial still have Column A in place.
Splitting Date and Time

By creating separate columns for date and time, you can more easily cluster your data by date. To do this, we will use the **Text to Columns** function in Excel.

First, **select the column** and change it to a **Text** format using the dropdown in the Home tab. This action will help to retain the YYYY-MM-DD format of the date portion of the field.

Next, **insert two empty columns** to the right of the Datetime column as shown above. The Text to Column function needs these columns to hold the separated data.

On the Data tab, **select the Text to Columns** function.
The Text to Columns Wizard will step you through the process of splitting the Datetime field. The wizard will default to Delimited, which is fine since the two fields are separated by a space.

_Step 1 of Text to Column_ – make sure **Delimited** is selected and click **Next**
Step 2 of Text to Column — check the box for Space and click Next

The third screen of the Text to Column wizard will require multiple changes. We need to set the destination to the two columns we inserted and set the data format for the new columns.

Step 3(a) – Click the arrow on the right end of Destination
Step 3(b) – **highlight the two new columns** (D & E) and once the =D$E appears in the bar, **click the down arrow** to return to the wizard.

Next, we need to format the two new columns so that the date format is retained.

Step 3(c) – **Use Ctrl-click to select both columns** in the preview and then **click on Text**
After completing the three parts of Step 3, **click on Finish** and enter column names for the two new columns.

![Completed Date and Time columns](image)

**Fixing the Mentions Error**

The final change to make to the data is to fix the error in the Mentions column. Since a username in Twitter begins with @, Excel thinks that it is a formula and when it opens the CSV the Mentions column is prefaced by an equal sign as seen here in the formula bar.

![Mentions formula error](image)

The simplest way to fix this error is to **highlight the column**, press Ctrl-H to bring up the Find/Replace dialog and **replace the equal sign with nothing**. This removes the formula and allows the Mentions to be viewed.

![Mentions replacement](image)
Find and Replace the equal sign to fix Mentions formula error

Remember to save the file as a spreadsheet (XLSX format) so that your changes are retained.

This completes “Open the Dataset in Excel and Prepare it for Analysis” section of the tutorial.

Concluding Thoughts

By following the methods outlined in this tutorial, you will be able to create a dataset of tweets that can be used as input for a textual analysis program such as Orange (https://orange.biolab.si/).

Once Anaconda and Jupyter Notebook have been installed, creating new datasets is as simple as changing the text or user query criteria in the notebook and running the code cells as needed. It is not a complicated process and the installation of the software is straightforward.

Scraping Twitter data is a simple process:

1. Decide whether to query by text criteria or username.
2. Run the first cell in the notebook to load the libraries.
3. If using a text-based query:
   a. Run the “Using a text-based search to collect tweets” code cell
   b. Modify the search parameters in the “Text query process” code cell and then run the cell
4. If using a username-based query:
   a. Run the “Using a username-based search to collect tweets” code cell
   b. Modify the search parameters in the “Username query process” code cell and the run the cell
5. Modify your CSV file to prepare it for data analysis.

The takeaways from this tutorial are to remember the following:

1. Make sure that the end date of your range is one greater than the date you want
2. Only try for 10,000 maximum tweets per query to keep Twitter from restricting you
3. Wait approximately 15-20 minutes between queries that return close to the 10,000 maximum. If your queries are returning a few thousand each time, you can run them more frequently.
4. Make the suggested changes to the data in Excel before further analysis of your data.
5. Enjoy the wealth of data retrievable using this Python-based data scraping method!

If you have any questions, I can be reached at AmyG@ucf.edu.

Understanding Digital Culture: Humanist Lenses for Internet Research

NEH Summer Institute, University of Central Florida, 1–5 June 2020