Mining (together with a bit of web scraping) of large social networks from Twitter using Python (and Ruby)

Moses Boudourides*[1]

[1]University of Patras – Greece

Abstract

This workshop is going to focus on how to construct certain networks from Twitter data after mining them from the Twitter API or/and possibly using a bit of web scraping.

An API (Application Programming Interface) of a social networking service is, roughly speaking, a set of routines that has been set up to allowing users to access specific chunks of data hosted in the social media. Needless to say that nowadays social media provide easily accessed sources of big data among which those displaying relational features may supply easily to access examples of large empirical social networks representing the underlying structures of action. The reason we are using Python to implement such data mining tasks is because Python exhibits a number of advantageous qualities in data gathering, data manipulation and data visualization and analysis.

Social media allow content that is created in one place can be dynamically posted and updated on the web. For instance, content (including texts, photos and videos) can be embedded, dynamically posted and shared together with certain user information (profile). In general, since the Twitter API is more open when comes to sharing information and given the existing restrictions in the Facebook and LinkedIn APIs, we are going to focus here just on data mining from Twitter.

The main mining tool in Twitter includes two RESTful APIs. Through the Twitter REST API methods users may access and interact with core Twitter data (such as update timelines, status data and user information) and Twitter Search data. Through the Streaming API method, users may access streaming tweets in real time as they happen. In all cases, retrieved data are in the JSON data format. JSON is the acronym of JavaScript Object Notation, i.e. an open standard format that uses human-readable text to transmit data objects consisting of attribute-value pairs.

Twitter users may access the API through an authorization provided by the OAuth tool, which is an authentication protocol that allows users to approve their application to act on their behalf without sharing their password. After getting authorization, users may employ different API methods for accessing information on tweets (including the occurrence of hashtags, search terms, embedded media etc.), users, following relationships (friends, followers), retweets, etc.

*Speaker

Furthermore, in certain situations, when the API of a service does not sufficiently provide all of the functionality that one requires, there is an alternative resort to collect data directly, the way they are actually displayed in a website, through web scraping, i.e., a programmatic method of extracting data from websites. For this purpose, we are using a Ruby script together with a few libraries (ruby germs, like Nokogiri) that grab the Twitter server's response to a browser's request (like a Twitter search) and parse it in such a way that that all the contents of the served Twitter data might be retrieved in a nicely-formatted list.

From the social network analysis point of view, our purpose is to extracted from the mined (or/and scrapped) Twitter data a multilayer network consisting of the following three layers: (i) the layer of retweeted data (or RTs) among Twitter users, (ii) the layer of friendship (follower-following) relationships among these users and (iii) the layer of co-occurring hashtags included in the tweets sent by the Twitter users.

For example, the following network visualization:

https://github.com/mboudour/TwitterMining/blob/master/ThreelayerCommunitiesTwitter.jpg

shows the 3–layer network extracted from Twitter data (about 500 K tweets), retrieved under the search term "Obamacare" and mined in the period October 18–31, 2013:

Furthermore, the main Python scripts we are using for Twitter mining are accessible here:

https://github.com/mboudour/TwitterMining

Of course, the above repository will be constantly updated to fit the needs of the Sunbelt 2016 workshop (and also to include the required Ruby scripts for Twitter scraping).

Prerequisites: Very elementary familiarity with Python. Workshop Length: 1 session (3 hours)

Attendance Limit: 21-30

Submitting Instructors: Moses Boudourides and Sergios Lenis

Institution: University of Patras

Email: Moses.Boudourides@gmail.com

Keywords: Twitter networks, mining