Linhua Wang

TwiMine: Mining tweets from user of interest

1. Introduction and motivation

Twitter has 330 million monthly active users and 139 million daily active users (source). There are 500 million tweets being posted every day on average. That means, on average, one active daily user posts 3.6 tweets per day. The content of tweets must show some information about users' interests, concerns and current status. I am thus motivated to create a pipeline (software) to mine twitter data so that for a given user of interest (UOI), it is capable to show UOI's interests/concerns/status and how these change over time. The pipeline I created is able to show such information by showing word cloud plots (wordcloud) using graphical user interface (GUI).

2. Preparation

2.1 Apply for a twitter developer account

In order to get twitter public data, we need a developer account because twitter wants to make sure that we are not abusing data from users.

The application requires a twitter account and some answers to their questions that asking the purpose of getting a developer account. This process typically takes one day.

Given every question answered in the previous step is good, an email will be sent to the applicant's preset email account asking some follow-up questions. And then an approval for developer account should be granted.

2.2 Get credentials from the developer account

After getting the developer account approved, the following credentials could be generated and used for our purpose.

i. Consumer key
ii. Consumer secret
iii. Access token key
iv. Access token secret

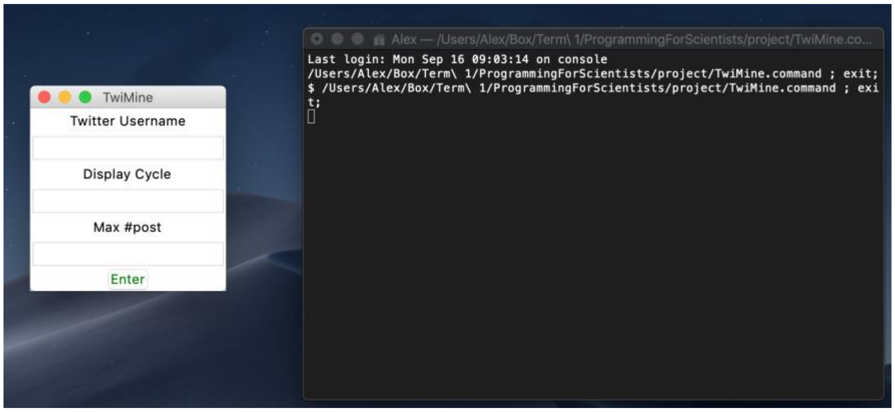
2.3 Python package Tweepy is used to mine tweets

Package *Tweepy* makes it possible for us to access Twitter API easily. In TwiMine, *Tweep*y is used to get the tweets from UOI.

3. Workflows

TwiMine is very easy to use and user friendly. It is just like a traditional app. A double click opens it. Enter some query information and hit “Enter” button gives us the results. The following steps show an example of mining tweets of UOI “USATODAY”.

Step 1: Double click on TwiMine.command.


TwiMine
Twitter Username
Display Cycle
Max #post
Enter
Alex — /Users/Alex/Box/Term\ 1/ProgrammingForScientists/project/TwiMine.co...

```
Last login: Mon Sep 16 09:03:14 on console
/Users/Alex/Box/Term\ 1/ProgrammingForScientists/project/TwiMine.command ; exit;
$ /Users/Alex/Box/Term\ 1/ProgrammingForScientists/project/TwiMine.command ; exi
t;
```

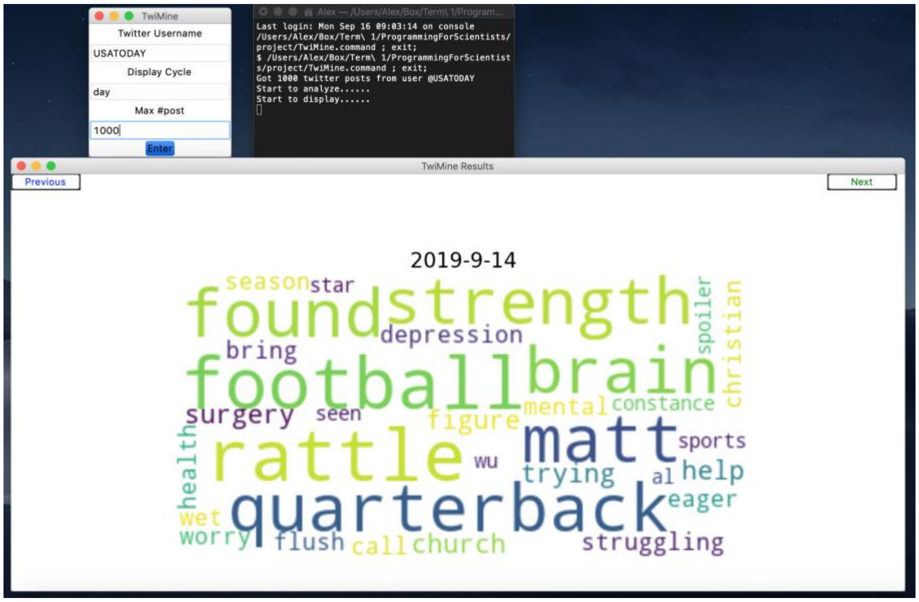
Screenshot 1: GUI

Two windows will pop up, one is the GUI of our TwiMine software, the other is a terminal showing the progress, command being executed, and standard output from Python scripts.

Step 2: Feed query information and hit “Enter”.

There are three entries to be filled in, only Twitter Username is required.

- Twitter Username: the user name of UOI, without “@”.
- Display Cycle: the period of analysis. Options are day, month and year. Default is day.
- Max #post: the maximum number of tweets to be analyzed from UOI. It should be integer. Default is 100.

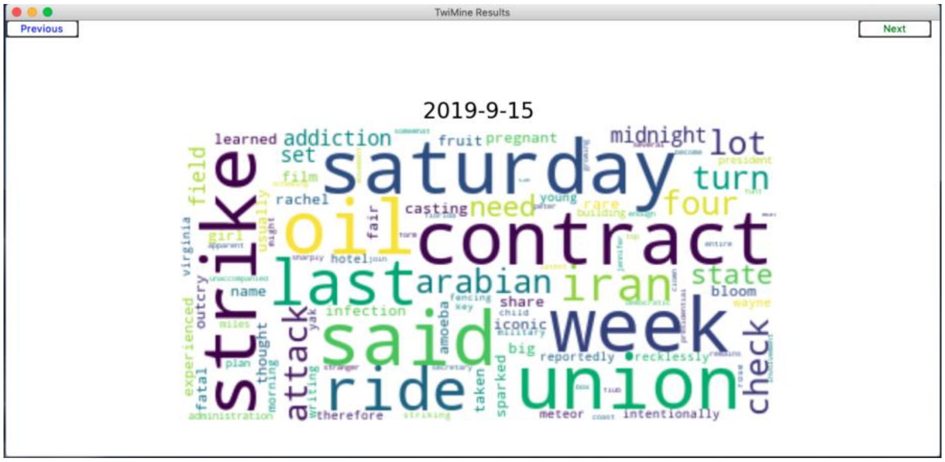


Twitter Username
USATODAY
Display Cycle
day
Max #post
1000

```
Last login: Mon Sep 16 09:03:14 on console
/Users/Alex/Box/Term\ 1/ProgrammingForScientists/
project/TwiMine.command ; exit;
$ /Users/Alex/Box/Term\ 1/ProgrammingForScientist
s/project/TwiMine.command ; exit;
Got 1000 twitter posts from user @USATODAY
Start to analyze......
Start to display......
```

Screenshot 2: Feed query information and result GUI after pressing "Enter" button.

After hitting the "Enter" button, we can get another GUI showing the result of analysis in word cloud figures. Because we are setting "display cycle = day", each word cloud figure shows the word cloud analysis result of a specific day, written at the top of the figure. There are two buttons on the figure: Previous and Next. When "Next" button is pressed, it will show the word cloud figure for the next available day if existed. If "Previous" button is pressed, it will show the word cloud figure for the previous available day if existed.



Screenshot 3: Result GUI after pressing "Next" button.

4. Conclusion

The resulted software *TwiMine* implemented the functions as written in the proposal of class project. It is able to get user-specific high-frequency words. Also, in the implementation, common words like “am”, “is”, “I”, “and” are removed from the text use for analysis because they provide little information and will be very high-frequency noises.

TwiMine is not a single mode analysis software. Depends on user’s interest, it can apply user-defined analysis. A “display cycle” entry specified by users could perform daily, monthly or yearly analysis on tweets of UOI. Additionally, for users whose computers don’t have too much RAM memory, *Tweepy* provides an option to customize the number of tweets to be analyzed. By doing this, users can fetch the number of tweets based on their concerns on hardware performance and computational time.

One limitation of *TwiMine* is that to operate it, users have to apply a developer account, which takes some time and effort. The credentials used in my class project won’t be shared to public. Thus, for people who want to run TwiMine, they should provide a credential.txt file containing the 4 credential codes as mentioned above.

The GUI of TwiMine can be further improved and the ways to show the results can be elaborated. For example, we could also show some other figures like pie chart and histograms. And these kinds of visualization could also be the options that users can tick in the main GUI.