

Mini-Project1 Liu Yue

Project Goals:

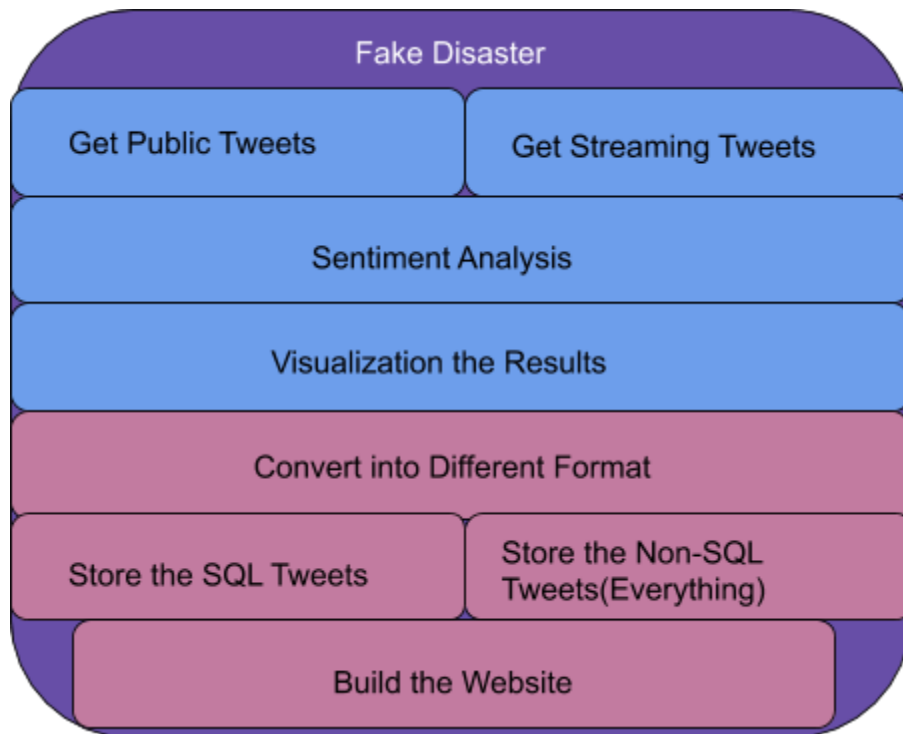
1. Build a library in Python
2. Analyze Twitter Feeds
3. Sentiment of Text Twitter Feeds

MVP & User Stories

1. Minimum Viable Product(MVP):
 - (1) Request for the target tweets and get the valid status of response via Twitter API;
 - (2) Request to analyze the sentiment of the tweets and get an answer via Google API;
 - (3) Get a small-scale dataset for labeling mini project2.
2. User Story:
 - (1) As a student learning social linguistics, I want to know users' attitudes to a real disaster;
 - (2) as a blogger, I want to know where is having a real disaster.

Modular Design

1. The blue components are done in Mini Project 1;
2. The discussion about sentiment analysis and visualization is in Mini Project 2;
3. The red components are done in Mini Project 3.



Understanding Different Twitter APIs

An API(Application Programming Interface) is an interface to program for specific tasks in the software. Twitter API and Google API are all service-based APIs in the industry, as well as Youtube API. Unlike the SOAP(Simple Object Access Protocol) APIs, they are all REST(Representational State Transfer) APIs.

Twitter APIs contain REST APIs and Streaming APIs. Twitter Developer Lab divided them into six different groups: Search Tweets, Account Activity API, Filter real-time Tweets, Direct Message API, Twitter for websites, Ads API. There are 293 APIs about the Twitter API searching in the ProgrammingWeb, and 31 of them are libraries.

Twitter API						
SEARCH						
APIs (293)	SDKs (115)	Articles (2158)	Libraries (31)	Source Code (27)	Frameworks (5)	Mashups (85)

There are nine different Python libraries built or tested by Twitter: five of them are deprecated or only have a single feature, the other four are active and are wrappers.

Table 1 Not Frequently Used Libraries

Library Name	Problem
TweetPony	No Documentation
Python Twitter Tools	Command Line Only
twitter-gobject	Search Only
TwitterSearch	Command Line Only
Birdy	No Documentation

Table 2 Active Libraries

Product	License	Participants	Oldest Comments	Latest Comments
tweepy	MIT License	174	03/06/2011	01/09/2020
python-twitter	Apache License 2.0	129	02/07/2013	11/30/2019
twython	MIT License	77	07/24/2013	01/05/2020
TwitterAPI	NA	18	NA	NA

We compare the remaining four libraries with REST API and Streaming API by achieving the same goals: (1) getting the public tweets; (2) getting the real-time tweets.

Table 3 Comparison

Library Name	Pros	Cons
tweepy	Systematic; Easy to split the tasks for an APP; Easy to get different kinds of users'	Hard to know the retweeters by using numeric id

	relationship	
python-twitter	Provide a Django tutorial; Help understand the numeric status id; Help understand the number limits	Not separate the consumer and access token
twython	Provide a Django example; Help understand the error condition	Not separate the consumer and access token
TwitterAPI	Help understand the number limits;	Hard to distinguish with Twitter API; Lacking active users

To sum up, tweepy is the prevailing product because it is well-organized. If we use other libraries, it will take more time to maintain the codes for the efficiency issue. It is unnecessary to provide Django examples in the library at this stage because there are other specific tutorial about it to make a better website.

We will store the data in a JSON(JavaScript Object Notation) file, which is a combination of list and dictionary.

Google Natural Language APIs

Searching in the ProgrammingWeb, we can find 904 different Google APIs, and over 33% of them are mashups.

APIs (904)	SDKs (416)	Articles (3987)	Libraries (204)	Source Code (395)	Frameworks (5)	Mashups (334)
----------------------	----------------------	---------------------------	---------------------------	-----------------------------	--------------------------	-------------------------

Product	Pros	Cons
Natural Language API	Easy to provide a measurement	Hard to analyze a large dataset

AutoML Natural Language	Good to understand Natural Language API; get results before programming; Proper for a large dataset.	Not actually using the API in Python
-------------------------	--	--------------------------------------

AutoML only accepts files stored in the Google Cloud Platform, and the file should be converted into '.csv' format. The dataset in mini project2 is from an ongoing Kaggle competition sponsored by Google AutoML.

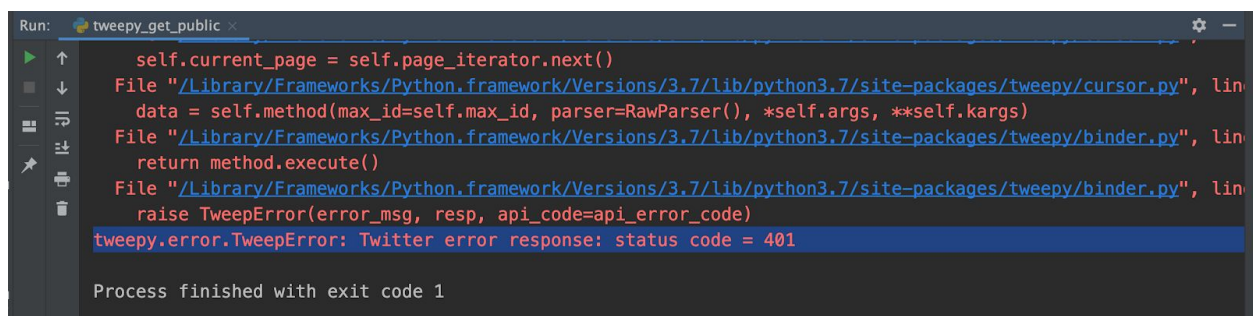
Error Conditions

There are two error conditions for the response codes: (1) HTTP status code; (2) JSON-based error codes and messages. It reminds us to be read the expiration and maximum count in the documentation.

It is easy to summarize the HTTP status code issue by using the following table.

Problem	Pattern
Informational	1XX
Successful	2XX
Redirection	3XX
Client Error	4XX
Server Error	5XX

For example, a 401 error represents Unauthorized Access, and we can solve it by regenerate the credentials.



```

Run: tweepy_get_public
self.current_page = self.page_iterator.next()
File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/tweepy/cursor.py", line
data = self.method(max_id=self.max_id, parser=RawParser(), *self.args, **self.kargs)
File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/tweepy/binder.py", line
return method.execute()
File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/tweepy/binder.py", line
raise TweepError(error_msg, resp, api_code=api_error_code)
tweepy.error.TweepError: Twitter error response: status code = 401

Process finished with exit code 1

```

There are too many requests if 429 goes back to your end, so you should simplify the code.



```
Run: tweepy_get_public <
return self.next()
File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/tweepy/cursor.py", line
self.current_page = self.page_iterator.next()
File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/tweepy/cursor.py", line
data = self.method(max_id=self.max_id, parser=RawParser(), *self.args, **self.kargs)
File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/tweepy/binder.py", line
return method.execute()
File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/tweepy/binder.py", line
raise TweepError(error_msg, resp, api_code=api_error_code)
tweepy.error.TweepError: Twitter error response: status code = 429
```

I also came across code 34 for a non-existing page. There is one exception, using the time-consuming tool. I tried to stream the data by TwitterAPI, but I can't get almost nothing because of the inefficiency of the library. The problem is caused by design. It is common if the code is not maintainable.



```
Run: TwitterAPI_streaming <
/usr/local/bin/python3.7 /Users/yue/Desktop/601mini1/step2_TwitterAPIs_stream/TwitterAPI_streaming.py
RT @0verfit: 一流虚飾AI人材をkaggleでポコポコにして負け惜しみ言わせたい
```