Collecting / Storing and analyzing Tweets with Python and MongoDB

Twitter analysis applications:

1. Event detection

2. Sentiment analyais

3. Recommender system

The Twitter Streaming API is ideal for grabbing data in real-time and storing it for analysis.

I tend to use Tweepy python library to connect with API due to its ease of use and simple structure.

For storing we are using MongoDB because it is perfectly suited for the unstrucured and large volume data.

Steps involved to write python code for downloading tweets and storing in mongo database:

Step1: Import the libraries:

```
import tweepy
import json
from pymongo import MongoClient
```

Step2: Next you will need to have a Twitter developer account and obtain credentials:

```
CONSUMER_KEY = "KEY"
CONSUMER_SECRET = "SECRET"
ACCESS_TOKEN = "TOKEN"
ACCESS_TOKEN_SECRET = "TOKEN_SECRET"
```

- Create a Twitter developer account if you do not already have one from : https://developer.twitter.com/
- Go to https://developer.twitter.com/en/apps and log in with your Twitter user account.
- Click “Create an app”
- Fill out the form, and click “Create”

- A pop up window will appear for reviewing Developer Terms. Click the “Create” button again.
- In the next page, click on “Keys and Access Tokens” tab, and copy your “API key” and “API secret” from the `Consumer API keys` section.

Scroll down to `Access token & access token secret` section and click “Create”. Then copy your “Access token” and “Access token secret”.

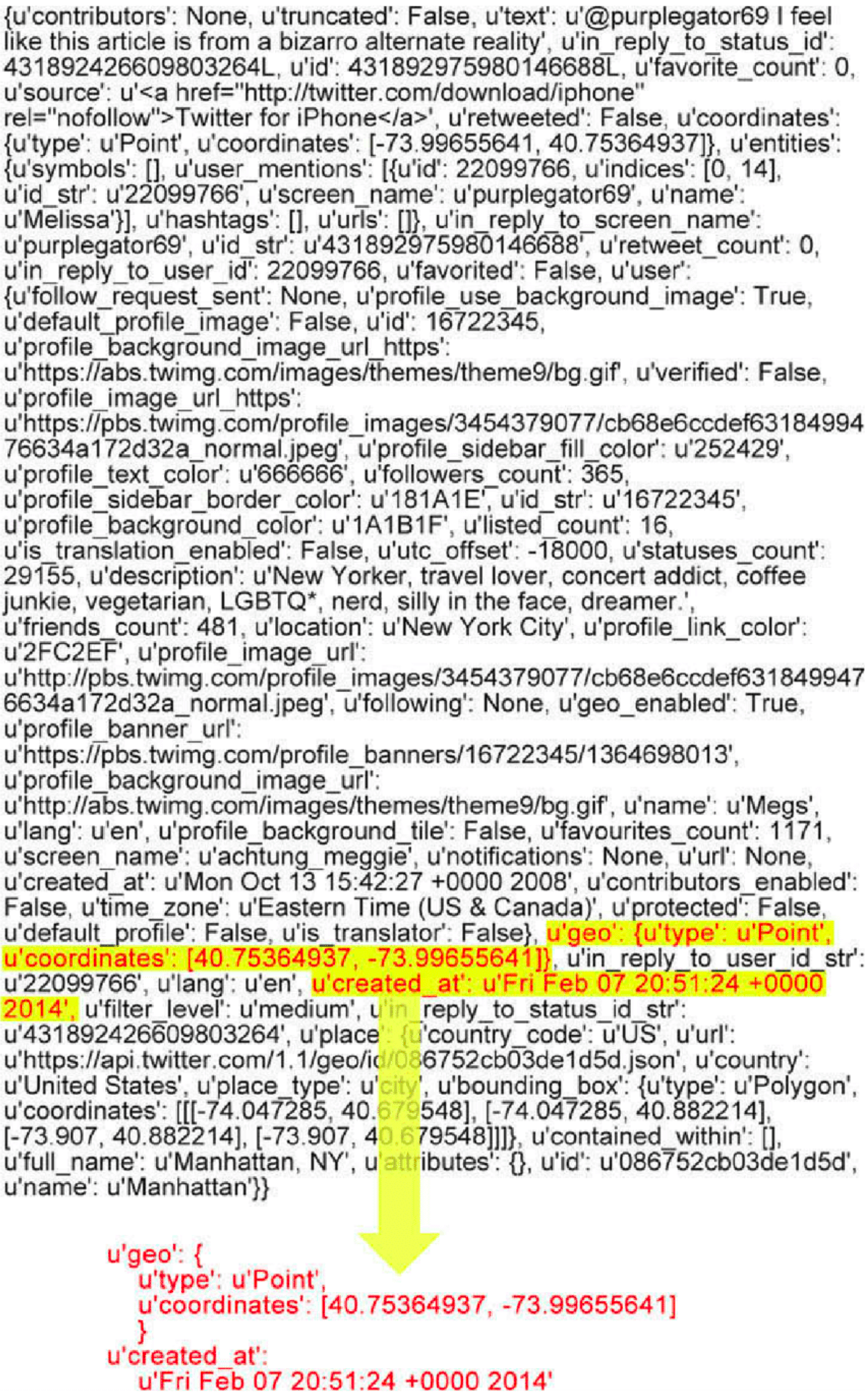
Step3: Build the StreamListener class provided by tweepy to access the Twitter Streaming API

```
class StreamListener(tweepy.StreamListener):

        # Called initially to connect to the Streaming API
        def on_connect(self):
             print("You are now connected to the streaming API.")

       #It connects to your mongoDB and stores the tweet
        def on_data(self, data):
        try:
                  client = MongoClient(MONGO_HOST)
              # Use twitterdb database. If it doesn't exist, it will be created.
                  db = client.twitterdb
              # Decode the JSON from Twitter
                  datajson = json.loads(data)
             #grab the 'created_at' data from the Tweet to use for display
                  created_at = datajson['created_at']
              #print out a message to the screen that we have collected a tweet
                 print("Tweet collected at " + str(created_at))
             #insert the data into the mongoDB into a collection called twitter_search
              #if twitter_search doesn't exist, it will be created.
                 db.twitter_search.insert(datajson)
      except Exception as e:
          print(e)
auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
#Set up the listener. The 'wait_on_rate_limit=True' is needed to help with Twitter API
rate limiting.
listener = StreamListener(api=tweepy.API(wait_on_rate_limit=True)) streamer =
tweepy.Stream(auth=auth, listener=listener)
```

Streaming API returns tweets in JSON format:



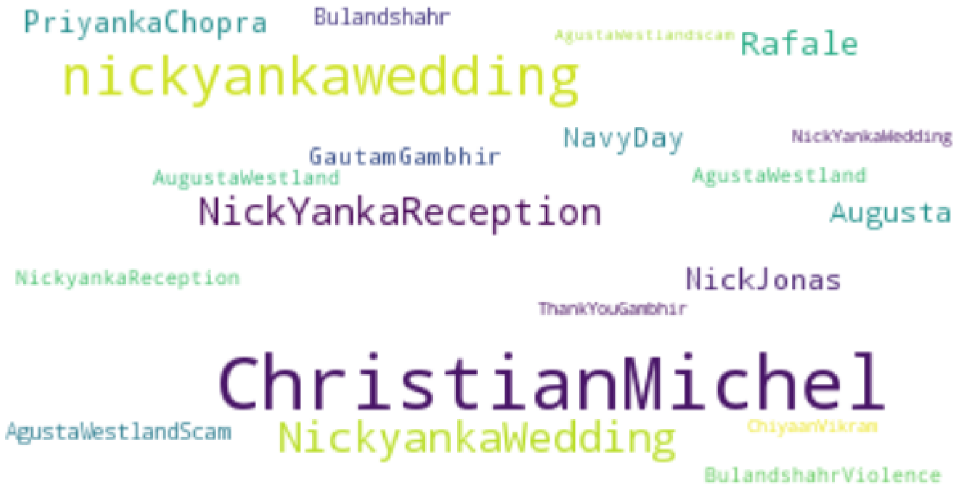
```
{u'contributors': None, u'truncated': False, u'text': u'@purplegator69 I feel
like this article is from a bizarro alternate reality', u'in_reply_to_status_id':
431892426609803264L, u'id': 431892975980146688L, u'favorite_count': 0,
u'source': u'<a href="http://twitter.com/download/iphone"
rel="nofollow">Twitter for iPhone</a>', u'retweeted': False, u'coordinates':
{u'type': u'Point', u'coordinates': [-73.99655641, 40.75364937]}, u'entities':
{u'symbols': [], u'user_mentions': [{u'id': 22099766, u'indices': [0, 14],
u'id_str': u'22099766', u'screen_name': u'purplegator69', u'name':
u'Melissa'}], u'hashtags': [], u'urls': []}, u'in_reply_to_screen_name':
u'purplegator69', u'id_str': u'431892975980146688', u'retweet_count': 0,
u'in_reply_to_user_id': 22099766, u'favorited': False, u'user':
{u'follow_request_sent': None, u'profile_use_background_image': True,
u'default_profile_image': False, u'id': 16722345,
u'profile_background_image_url_https':
u'https://abs.twimg.com/images/themes/theme9/bg.gif', u'verified': False,
u'profile_image_url_https':
u'https://pbs.twimg.com/profile_images/3454379077/cb68e6ccdef63184994
76634a172d32a_normal.jpeg', u'profile_sidebar_fill_color': u'252429',
u'profile_text_color': u'666666', u'followers_count': 365,
u'profile_sidebar_border_color': u'181A1E', u'id_str': u'16722345',
u'profile_background_color': u'1A1B1F', u'listed_count': 16,
u'is_translation_enabled': False, u'utc_offset': -18000, u'statuses_count':
29155, u'description': u'New Yorker, travel lover, concert addict, coffee
junkie, vegetarian, LGBTQ*, nerd, silly in the face, dreamer.',
u'friends_count': 481, u'location': u'New York City', u'profile_link_color':
u'2FC2EF', u'profile_image_url':
u'http://pbs.twimg.com/profile_images/3454379077/cb68e6ccdef631849947
6634a172d32a_normal.jpeg', u'following': None, u'geo_enabled': True,
u'profile_banner_url':
u'https://pbs.twimg.com/profile_banners/16722345/1364698013',
u'profile_background_image_url':
u'http://abs.twimg.com/images/themes/theme9/bg.gif', u'name': u'Megs',
u'lang': u'en', u'profile_background_tile': False, u'favourites_count': 1171,
u'screen_name': u'achtung_meggie', u'notifications': None, u'url': None,
u'created_at': u'Mon Oct 13 15:42:27 +0000 2008', u'contributors_enabled':
False, u'time_zone': u'Eastern Time (US & Canada)', u'protected': False,
u'default_profile': False, u'is_translator': False}, u'geo': {u'type': u'Point',
u'coordinates': [40.75364937, -73.99655641]}, u'in_reply_to_user_id_str':
u'22099766', u'lang': u'en', u'created_at': u'Fri Feb 07 20:51:24 +0000
2014', u'filter_level': u'medium', u'in_reply_to_status_id_str':
u'431892426609803264', u'place': {u'country_code': u'US', u'url':
u'https://api.twitter.com/1.1/geo/id/086752cb03de1d5d.json', u'country':
u'United States', u'place_type': u'city', u'bounding_box': {u'type': u'Polygon',
u'coordinates': [[[-74.047285, 40.679548], [-74.047285, 40.882214],
[-73.907, 40.882214], [-73.907, 40.679548]]]}, u'contained_within': [],
u'full_name': u'Manhattan, NY', u'attributes': {}, u'id': u'086752cb03de1d5d',
u'name': u'Manhattan'}}
```

```
u'geo': {
    u'type': u'Point',
    u'coordinates': [40.75364937, -73.99655641]
    }
u'created_at':
    u'Fri Feb 07 20:51:24 +0000 2014'
```

Some queries are required to extract useful information from large amount of data:

1. Extract most popular 20 hashtags and represent using wordcloud plot: Information present in tweet['entities']['hashtags']

[(u'ChristianMichel', 6293), (u'nickyankawedding', 2600), (u'NickyankaWedding', 1722), (u'NickYankaReception', 1252), (u'Rafale', 894), (u'NickJonas', 720) (u'PriyankaChopra', 692), (u'Augusta', 572), (u'NavyDay', 539) ,(u'GautamGambhir', 452) ,(u'Bulandshahr', 332) ,(u'AgustaWestlandScam', 308) ,(u'NickyankaReception', 266) ,(u'AugustaWestland', 265) ,(u'AgustaWestland', 261) ,(u'BulandshahrViolence', 252) ,(u'NickYankaWedding', 208) ,(u'ChiyaanVikram', 192) ,(u'ThankYouGambhir', 179),(u'AgustaWestlandscam', 147), (u'WATCH', 145)]



2. Extract most popular 5 tweets based on retweet_count: Information present in tweet['retweeted_status']

Inspector Subodh kumar singh sacrificed his life to uphold law &amp; order in the district. With his demise, we have l… https://t.co/JgiCG1jcfH : 4187
My goodness⭐#nickyankawedding https://t.co/uVnES8izgH : 3736

Priyanka literally had a fairytale wedding #NickyankaWedding https://t.co/75QicCqfqX :   1207
Gautam Gambhir Announces Retirement From All Form… https://t.co/Uftiw7Wk83 : 685
An amazing performance of that quintessentially British Royal Navy tradition of the 'Sailor's Hornpipe' by the Indi… https://t.co/cAnvj36nBz: 663

3. Extract most active users: Information present in tweet['user']['screen_name']

[(u'sarathmonicvf', 78), (u'nitya12', 61), (u'SarlaSungroya', 58), (u'PCsDRAG0N', 55),(u'AbsumMuggle', 52), (u'Pcglobaldomina', 48), (u'PCsFanKartik', 47), (u'ria_xoxo3', 43), (u'chaudhary5665', 42)]