

Sentiment analysis using Twitter streaming API and Python

Course: AE-663, Group: P13

Monika Sahai: 133079022 Zeal Sheth: 133079023

Indian Institute Of Technology, Bombay

April 29, 2015

Outline

1 Introduction

- Sentiment analysis
- Twitter API
- REST vs Streaming API

2 Packages used

- NLTK
- Tweepy
- Matplotlib

3 Control flow

4 Results

- Time wise sentiment graph
- Percentage
- State wise happiness distribution

5 Conclusions

6 Bibliography

Sentiment Analysis

① What is sentiment analysis?

Process of identifying and characterizing the opinions expressed in a text to determine if the writer's emotion is positive, negative or neutral.

② Why is it useful?

- ① Companies use sentiment analysis to improve their business.

Ex: Customer responses(feedback forms) can be analyzed to calculate the customer satisfaction index.

- ② Powerful method for analysis of business in share market.

Twitter API

① Steps to connect to API

- I Create a twitter account.
- II Go to <https://apps.twitter.com/> and log in with your twitter credentials.
- III Click "Create New App"
- IV Fill out the form, agree to the terms, and click "Create your Twitter application"
- V In the next page, click on "API keys" tab, and copy your "API key" and "API secret".
- VI Scroll down and click "Create my access token", and copy your "Access token" and "Access token secret".

② These keys and tokens are used for connecting to twitter and streaming live tweets.

③ API returns the result in json format

Twitter API

1 Rest and Streaming API

1 Search/REST API

- ★ Search goes back in time (up to a week) to find tweets that have already been sent.
- ★ HTTP stream is short lived.

2 Streaming API

- ★ Stream goes forward in time (starting from when you initiate the call) to capture new tweets in (more or less) real time as they are sent.
- ★ Requires keeping a persistent HTTP connection open.

REST vs Streaming API

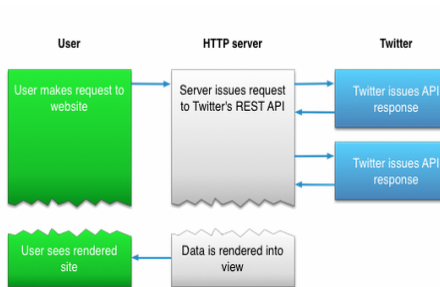


Figure : REST API

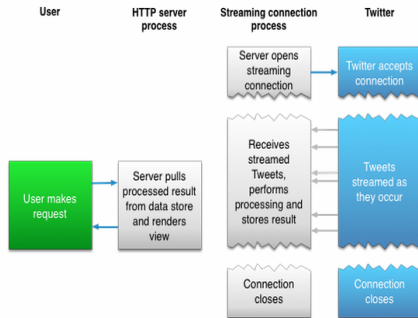


Figure : Streaming API

Packages used

- ① nltk : used for data mining
- ② re : used for filtering tweets
- ③ tweepy : Python library for twitter API
- ④ json : for reading the data collected by twitter streaming.
- ⑤ matplotlib : Package for visualizing the data in graphical form.
- ⑥ Matplotlib Basemap toolkit : Library for geo-plotting

- ① NLTK: Natural Language Processing Toolkit
- ② Phases of classifier:
 - ① Phase-I : Training of the classifier
 - ② Phase-II : Testing of the classifier
- ③ We have used a database of 2500 tweets as sample data whose sentiments are known.
- ④ This data is used to extract features for sentiment analysis.
- ⑤ Feature list is then given to classifier for training.
- ⑥ Testing of the classifier is done by calling the trained classifier on data to be analysed i.e. tweets.

Sentiment analysis flow

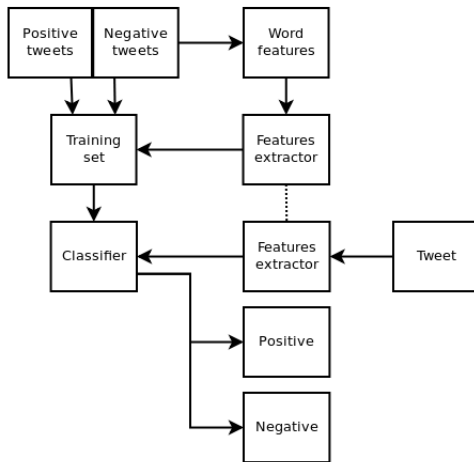


Figure : Training and Testing of Bayes classifier

- 1 Provides Classes and methods for connecting to API and streaming namely OAuthHandler

```
from tweepy.streaming import StreamListener
from tweepy import OAuthHandler as OA
from tweepy import Stream
```

```
l=StdOutListener()
auth = OA(consumer_key , consumer_secret)
auth.set_access_token(access_token , access_token_secret)
stream = Stream(auth , l)
```

Matplotlib

① Basemap

```
from mpl_toolkits.basemap import Basemap as BM
```

```
m = BM(llcrnrlon=-119,llcrnrlat=20,urcrnrlon=-64,  
      urcrnrlat=49,projection='laea',lat_1=33,lat_2=45,  
      lon_0=-95,lat_0=50)
```

- ① provides the facilities to transform coordinates to one of 25 different map projections.

② Shapefiles:

- ① Contains geographical data.
- ② It is developed and regulated by Esri (Environmental Systems Research Institute)
- ③ The shapefile format is a digital vector storage format for storing geometric location and associated attribute information.

Matplotlib contd.

- 1 Matplotlib is then used to plot the points in the transformed coordinates.
 - 1 Shape file is read and polygons are constructed using the parameters obtained from the shape file.
 - 2 Polygon library is used to generate polygon from shapefiles.
 - 3 Used modules colors and patches to fill the polygons states on map with different colors.

Control Flow

- 1 Streaming
- 2 Feature extraction
- 3 Sentiment categorization as positive,negative or neutral
- 4 Constructing polygons for states using shapefile
- 5 Assigning colors to polygons based on the happiness score
- 6 Plotting the map with set properties

Happiness score distribution

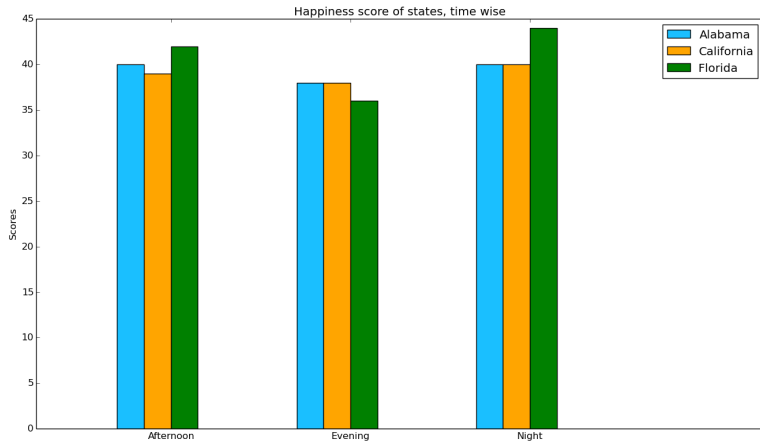


Figure : Happiness score for different times of the day for sample of 50 tweets

Happiness score distribution

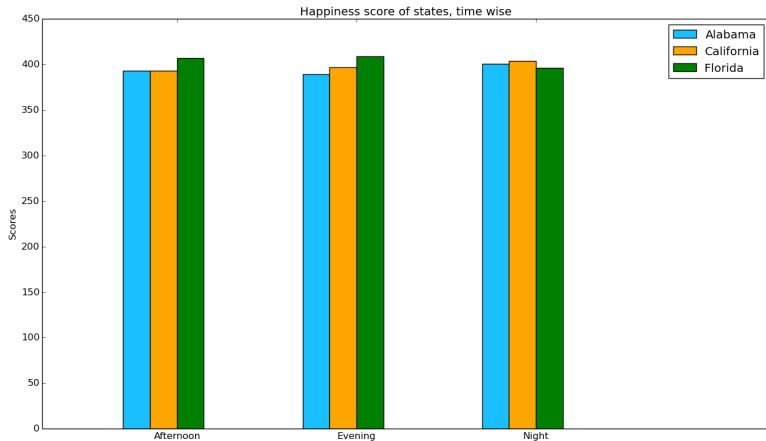


Figure : Happiness score for different times of the day for sample of 500 tweets

Happiness score distribution contd.

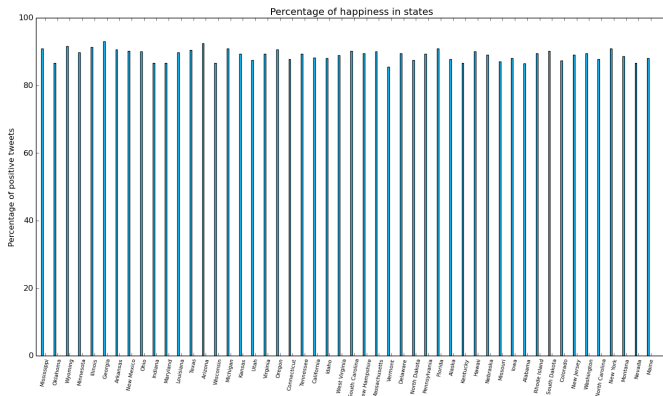


Figure : Percentage of happiness in each state

Happiness score distribution contd.

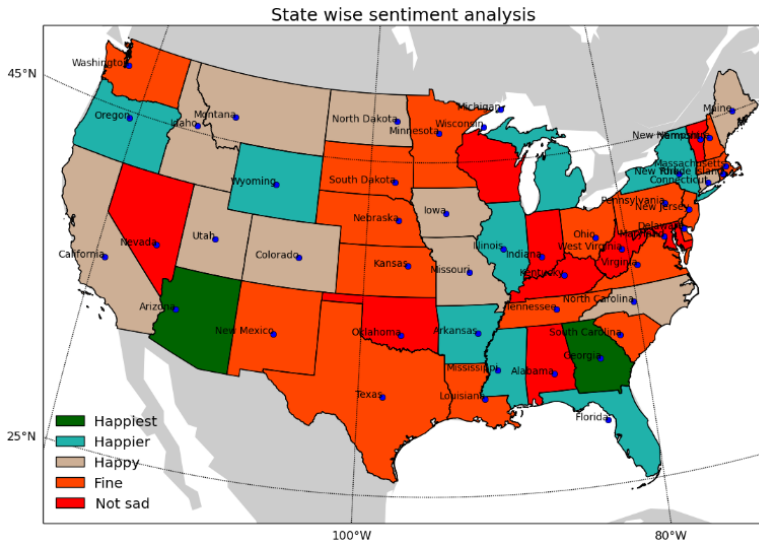


Figure : State wise geo-distribution of the happiness score

Conclusions

- ① Happiest states are Arizona and Georgia
- ② Almost all states are fairly positive since out of 500 tweets , minimum happy score is 320.
- ③ States with score above 331(category fine) is maximum. So on an average, maximum states are positive.

Bibliography

- ① <https://dev.twitter.com/streaming/overview>
- ② <http://www.laurentluce.com/posts>
- ③ https://www.census.gov/geo/maps-data/data/cbf/cbf_state.html
- ④ <http://stackoverflow.com>
- ⑤ <http://www.pythoncentral.io/introduction-to-tweepy-twitter-for-python>
- ⑥ <https://class.coursera.org/datasci-001/lecture/55>
- ⑦ <http://matplotlib.org/basemap/>