



“Polarizer”

PRESENTED BY:

BABHULGAONKAR SHASHWAT
BENDKULE PRITAM
SURYAVANSHI JAYESH
SANAP VIVEK

Overview



What is sentiment analysis?

Sentiment Analysis is the process of 'computationally' determining whether a piece of writing is positive, negative or neutral. It's also known as opinion mining, deriving the opinion or attitude of a speaker.

Use Cases



- **Business:** In marketing field companies use it to develop their strategies, to understand customers' feelings towards products or brand, how people respond to their campaigns or product launches and why consumers don't buy some products.
- **Politics:** In political field, it is used to keep track of political view, to detect consistency and inconsistency between statements and actions at the government level. It can be used to predict election results as well!
- **Public Actions:** Sentiment analysis also is used to monitor and analyse social phenomena, for the spotting of potentially dangerous situations and determining the general mood of the blogosphere.

Installation



Tweepy: [tweepy](#) is the python client for the official [Twitter API](#).
Install it using following pip command:

```
pip install tweepy
```

TextBlob: [textblob](#) is the python library for processing textual data.

Install it using following pip command:

```
pip install textblob
```

Also, we need to install some NLTK corpora using following command:

```
python -m textblob.download_corpora
```


Authentication




In order to fetch tweets through Twitter API, we register for an App through our twitter account. Below are the steps for the same:

- Open this [link](#) and click the button: **'Create New App'**
- Fill the application details. You can leave the callback url field empty.
- Once the app is created, you will be redirected to the app page.
- Open the **'Keys and Access Tokens'** tab.
- Copy **'Consumer Key'**, **'Consumer Secret'**, **'Access token'** and **'Access Token Secret'**.

Screenshots of Developer API's

 Developer Use cases Products Docs More


Dashboard Polarizer ▾ 

Apps > PolarizerSentimentAnalyzer

App details Keys and tokens Permissions

App details Edit ▾

Details and URLs

 **App icon**
App icon is default, click edit to upload.

You are using the default icon now,
change it in app editing mode. ×

PolarizerSentimentAnalyzer

Description
Polarizer is a smart python script to bifurcate positive and negative tweets. This program uses tweepy, textblob , NLTK and pandas as the dependencies.

Website URL
<https://github.com/JayeshSuryavanshi/Polarizer--Sentiment-analyzer-for-Twitter>

Sign in with Twitter
Disabled

Callback URL
None

Screenshots of Keys and tokens

The screenshot shows the Twitter Developer Portal interface. At the top is a purple navigation bar with links for Developer, Use cases, Products, Docs, and More. On the right of this bar are links for Dashboard and Polarizer, along with a user profile picture. Below the navigation bar, the breadcrumb 'Apps > PolarizerSentimentAnalyzer' is visible. The main content area has three tabs: 'App details', 'Keys and tokens' (which is selected), and 'Permissions'. The 'Keys and tokens' tab contains a section titled 'Keys and tokens' with a subtitle 'Keys, secret keys and access tokens management.' Below this, there is a 'Consumer API keys' section displaying an API key '1qP7HdLTa2kIKAv0xNZEpd3fZ' and an API secret key 'RRgdq5eTWwuP67ZrDIEAEPR5MTP93dE8W3P7tsBNRniiDamjJu', each with a 'Regenerate' button. The next section is 'Access token & access token secret', showing an access token '3002739264-5dQllrzigtJu8kDpMSyfk4Kzs8xeVXJzXKf07GB' and an access token secret 'hQlYdpBCVPRvnJISpjZGmH4826IKRx1wEIMiEMzrYQXe', both with 'Regenerate' buttons. Below these is the 'Read and write' access level with a 'Revoke' button. At the bottom of the page is a purple footer bar containing links for 'Developer policy and terms', 'Follow @twitterdev', and a 'Subscribe to developer news' button.

Developer

Use cases

Products

Docs

More

Dashboard

Polarizer

Apps > PolarizerSentimentAnalyzer

App details

Keys and tokens

Permissions

Keys and tokens

Keys, secret keys and access tokens management.

Consumer API keys

1qP7HdLTa2kIKAv0xNZEpd3fZ (API key)

RRgdq5eTWwuP67ZrDIEAEPR5MTP93dE8W3P7tsBNRniiDamjJu (API secret key)

Regenerate

Access token & access token secret

3002739264-5dQllrzigtJu8kDpMSyfk4Kzs8xeVXJzXKf07GB (Access token)

hQlYdpBCVPRvnJISpjZGmH4826IKRx1wEIMiEMzrYQXe (Access token secret)

Read and write (Access level)

Revoke


Regenerate


Developer policy and terms

Follow @twitterdev

Subscribe to developer news

Permissions

 Developer Use cases Products Docs More

Dashboard Polarizer 

Apps > [PolarizerSentimentAnalyzer](#)

App details Keys and tokens **Permissions**

Permissions

Changes to the app permissions will be reflected in access tokens generated after the permissions are saved. You will need to regenerate existing access tokens to alter permissions levels.

Access permission
Read and write

Additional permissions
None

Edit

Developer policy and terms Follow @twitterdev [Subscribe to developer news](#)

About
Let's go Twitter
Company
Values

Business
About Twitter Ads
Targeting
Analytics

Developers
Documentation
Forums
Communities

Help Center
Using Twitter
Managing your account
Safety and security

Marketing
Insights
Success Stories
Solutions

Implementation & Results



Link for the project:

<https://github.com/JayeshSuryavanshi/Polarizer--Sentiment-analyzer-for-Twitter>

We followed these 3 major steps in our program:

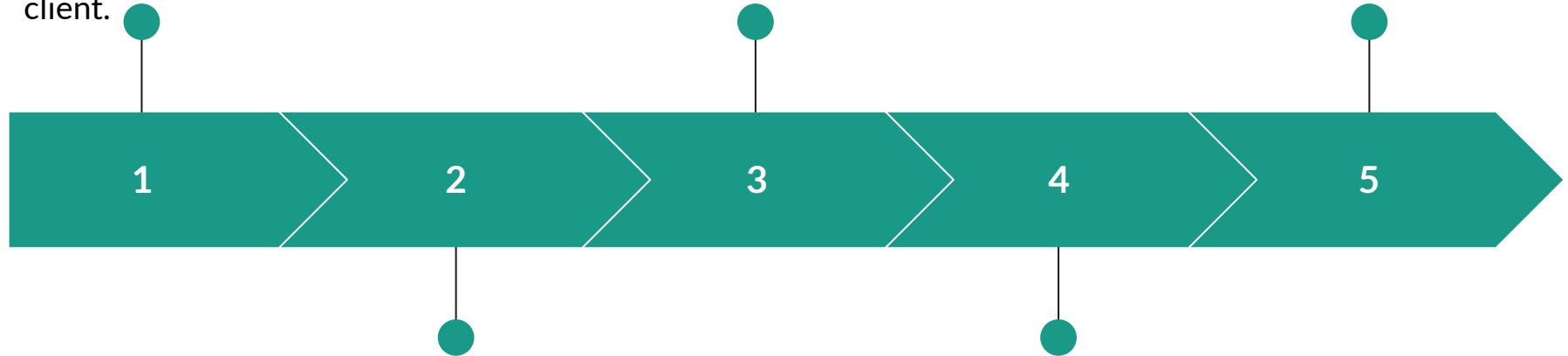
- **Authorization:** Authorize twitter API client.
- **Fetching:** Make a GET request to Twitter API to fetch tweets for a particular query.
- **Parsing:** Parse the tweets. Classify each tweet as positive, negative or neutral.

create a **TwitterClient** class

__init__ function to handle the authentication of API client.

In **get_tweet_sentiment** we use **textblob** module.

we use **sentiment.polarity** method of **TextBlob** class



get_tweets function

Twitter API to fetch tweets.

Parsing of Tweets using **TextBlob** library

Procedure:



- First of all, we create a **TwitterClient** class. This class contains all the methods to interact with Twitter API and parsing tweets.

We use `__init__` function to handle the authentication of API client.

- In `get_tweets` function, we use:

```
fetches_tweets = self.api.search(q = query, count = count)
```

to call the Twitter API to fetch tweets.


- In `get_tweet_sentiment` we use `textblob` module.

```
analysis = TextBlob(self.clean_tweet(tweet))
```

`TextBlob` is actually a high level library built over top of [NLTK](#) library. First we call `clean_tweet` method to remove links, special characters, etc. from the tweet using some simple regex.

Then, as we pass `tweet` to create a **TextBlob** object, following processing is done over text by `textblob` library:

Procedure:

- 
- Tokenize the tweet ,i.e split words from body of text.
 - Remove stopwords from the tokens.(stopwords are the commonly used words which are irrelevant in text analysis like I, am, you, are, etc.)
 - Do POS(part of speech) tagging of the tokens and select only significant features/tokens like adjectives, adverbs, etc.
 - Pass the tokens to a **sentiment classifier** which classifies the tweet sentiment as positive, negative or neutral by assigning it a polarity between -1.0 to 1.0 .

Here is how **sentiment classifier** is created:

- **TextBlob** uses a Movies Reviews dataset in which reviews have already been labelled as positive or negative.
- Positive and negative features are extracted from each positive and negative review respectively.

Then, we use **sentiment.polarity** method of **TextBlob** class to get the polarity of tweet between -1 to 1.

Procedure:



Then, we classify polarity as:

```
if analysis.sentiment.polarity > 0:
```

```
    return 'positive'
```

```
elif analysis.sentiment.polarity == 0:
```

```
    return 'neutral'
```

```
else:
```

- ```
 return 'negative'
```
- Finally, parsed tweets are returned. Then, we can do various type of statistical analysis on the tweets. For example, in above program, we tried to find the percentage of positive, negative and neutral tweets about a query.

# References



- <http://www.ijcaonline.org/research/volume125/number3/dandrea-2015-ijca-905866.pdf>
- <https://textblob.readthedocs.io/en/dev/quickstart.html#sentiment-analysis>
- [textblob.readthedocs.io/en/dev/modules/textblob/en/sentiments.html](https://textblob.readthedocs.io/en/dev/modules/textblob/en/sentiments.html)