**Twitter Sentiment Analysis - Documentation**

**Overview**

This Python tool fetches tweets from Twitter, performs sentiment analysis on each tweet, and then categorizes them into positive, negative, and neutral sentiments. The program provides the following:

1. Fetches tweets based on a search query (e.g., "Donald Trump").
2. Analyzes the sentiment of each tweet using the TextBlob library.
3. Outputs the percentage of positive, negative, and neutral tweets.
4. Displays the first few positive and negative tweets.

**Requirements**

To run this program, you'll need the following Python libraries installed:

* tweepy for interacting with the Twitter API.
* textblob for performing sentiment analysis.
* re (regular expression) for cleaning up the tweet text.

Install the necessary libraries using:

bash

Copy code

pip install tweepy textblob

You will also need valid **Twitter Developer API credentials** (consumer key, consumer secret, access token, and access token secret) to access the Twitter API.

**How It Works**

**1. Authentication**

The program starts by authenticating with the Twitter API using the credentials provided by you. The TwitterClient class handles the authentication using the OAuthHandler from tweepy.

python

Copy code

# Initialize authentication

self.auth = OAuthHandler(consumer\_key, consumer\_secret)

self.auth.set\_access\_token(access\_token, access\_token\_secret)

self.api = tweepy.API(self.auth)

* If authentication is successful, the program connects to Twitter's API.
* If authentication fails (invalid credentials), it will raise an error.

**2. Fetching Tweets**

The get\_tweets() method is used to fetch tweets based on a search query (e.g., "Donald Trump"). You can specify the number of tweets you want to retrieve by passing the count parameter. The default is 10 tweets.

python

Copy code

fetched\_tweets = self.api.search(q=query, count=count)

The method returns a list of tweets containing the query term. The program processes each tweet one by one.

**3. Cleaning Tweets**

The clean\_tweet() method is used to preprocess each tweet. This function removes mentions (e.g., @username), special characters, and links from the tweet to make it ready for sentiment analysis.

python

Copy code

return ' '.join(re.sub(r"(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)", " ", tweet).split())

* **@username**: Mentions (Twitter handles) are removed.
* **Special Characters**: Non-alphanumeric characters are removed.
* **URLs**: Links (starting with http://, https://, or similar) are removed.

**4. Sentiment Analysis**

The sentiment analysis is performed using TextBlob. The get\_tweet\_sentiment() method creates a TextBlob object for each cleaned tweet and analyzes its polarity score.

python

Copy code

analysis = TextBlob(self.clean\_tweet(tweet))

* **Polarity > 0**: The tweet is classified as **positive**.
* **Polarity == 0**: The tweet is classified as **neutral**.
* **Polarity < 0**: The tweet is classified as **negative**.

**5. Displaying Sentiment Results**

After fetching and analyzing the tweets, the program calculates and prints the percentage of positive, negative, and neutral tweets.

python

Copy code

positive\_percentage = (100 \* len(ptweets) / len(tweets))

negative\_percentage = (100 \* len(ntweets) / len(tweets))

neutral\_percentage = (100 \* (len(tweets) - (len(ntweets) + len(ptweets))) / len(tweets))

Finally, the program prints the first 5 positive and negative tweets.

**Main Classes and Functions**

**1. TwitterClient Class**

This class encapsulates all the logic for connecting to Twitter, fetching tweets, cleaning them, and analyzing sentiment.

**\_\_init\_\_(self)**

* Initializes the authentication process with Twitter's API.

**clean\_tweet(self, tweet)**

* Cleans the tweet by removing mentions, special characters, and URLs.

**get\_tweet\_sentiment(self, tweet)**

* Analyzes the sentiment of a tweet using TextBlob. Returns 'positive', 'neutral', or 'negative'.

**get\_tweets(self, query, count=10)**

* Fetches tweets containing the search query and processes them.
* Returns a list of dictionaries with the tweet's text and sentiment.

**2. Main Function**

The main() function orchestrates the entire flow:

1. It creates an object of the TwitterClient class.
2. It fetches tweets based on a search query (e.g., "Donald Trump").
3. It calculates and prints the percentage of positive, negative, and neutral tweets.
4. It displays the first 5 positive and negative tweets.

**Expected Outputs**

**1. Percentage of Positive, Negative, and Neutral Tweets**

After running the main() function, you will see the percentage of positive, negative, and neutral tweets.

plaintext

Copy code

Positive tweets percentage: 45.0 %

Negative tweets percentage: 35.0 %

Neutral tweets percentage: 20.0 %

This shows the sentiment distribution for the fetched tweets.

**2. Positive Tweets**

The first 5 positive tweets will be displayed:

plaintext

Copy code

Positive tweets:

"Donald Trump speaks about the future of America."

"Exciting news! Trump's policies are making an impact."

"Trump's leadership is revitalizing the economy."

"His economic plan is bringing growth."

"Trump's plan for the future looks promising."

**3. Negative Tweets**

The first 5 negative tweets will be displayed:

plaintext

Copy code

Negative tweets:

"Donald Trump is ruining America."

"Trump's policies are disastrous for the country."

"Another poor decision by Trump."

"The world is better off without Trump's leadership."

"Trump has failed to live up to his promises."

**Error Handling**

* **Authentication Error**: If Twitter API credentials are invalid, the program will print an authentication error message.

plaintext

Copy code

Error: Authentication Failed. [Error details]

* **API Rate Limit Exceeded**: If the program exceeds Twitter's API rate limits, it may raise an exception, which will print an error message like:

plaintext

Copy code

Error : [Error details]

**Running the Code**

To run the code, follow these steps:

1. Ensure you have installed tweepy and textblob using pip install tweepy textblob.
2. Replace the placeholders (XXXXXXXXXXXXXXXXXXXXXXXX) in the consumer\_key, consumer\_secret, access\_token, and access\_token\_secret variables with your actual Twitter Developer API credentials.
3. Run the script in a Python environment, such as a Jupyter notebook, to see the output of the sentiment analysis.

**Conclusion**

This Twitter Sentiment Analysis tool helps you fetch tweets on a specific topic and analyze their sentiment using TextBlob. It's a great starting point for anyone looking to perform social media sentiment analysis or build sentiment-based models using Twitter data.