README: Twitter Data Fetching and Sentiment Analysis

This script fetches recent tweets using Twitter API v2, analyzes the sentiment of the tweets using TextBlob, and stores the results in a CSV file.

Main Features

1. **Fetch Tweets**:
   Retrieves recent tweets based on a specified keyword using the Twitter API v2.
2. **Sentiment Analysis**:
   Uses **TextBlob** to calculate the sentiment polarity of each tweet:
   - Polarity ranges from **-1 (negative)** to **1 (positive)**.
3. **Save Results**:
   Stores the tweets along with their sentiment scores in a CSV file (tweets_with_sentiment.csv).

Setup and Requirements

1. **Python 3.8+**:
   Ensure Python is installed on your system.
2. **Twitter Developer Account**:
   Get a Bearer Token from Twitter's Developer Portal. Set up a project here.
3. **Required Libraries**:
   Install the following libraries using pip:
4. pip install tweepy pandas textblob

To use **TextBlob**, ensure that nltk is set up by downloading its corpora if not already done:

python -m textblob.download_corpora

How to Use

1. **Replace Bearer Token**:
   Update the bearer_token variable in the script with your own Bearer Token:
2. bearer_token = "YOUR_BEARER_TOKEN"
3. **Run the Script**:
   Use the command:
4. python sentiment_analysis.py
5. **Input Parameters**:
   Modify the following variables:
   - keyword: The search term for fetching tweets (e.g., "Python").
   - count: The number of tweets to fetch (max 100 per request).
6. **Output**:
   - The fetched tweets with their sentiment scores are displayed in the terminal.

   - A CSV file (tweets_with_sentiment.csv) containing tweet text, creation time, and sentiment scores is saved.

Script Workflow

1. **Initialize Twitter API v2 Client**:
   - Authenticates the request using your Bearer Token.
2. **Fetch Tweets**:
   - Uses the search_recent_tweets endpoint to collect tweets containing the specified keyword.
3. **Analyze Sentiment**:
   - Uses TextBlob to calculate the sentiment polarity of each tweet.
4. **Save and Display**:
   - Outputs the tweets and sentiment scores.
   - Optionally saves them to a CSV file.

Notes

1. **Rate Limits**:
   - The Twitter API limits the number of requests. For the search_recent_tweets endpoint, the cap is **450 requests per 15 minutes**.
2. **Empty Results**:
   - If no tweets match the keyword, the script will display an appropriate message.
3. **Extensibility**:
   - Enhance by adding features like advanced sentiment metrics, keyword automation, or real-time monitoring.
4. **Data Privacy**:
   - Respect Twitter's Developer Agreement when using or sharing data.

Future Enhancements

- Integrate a machine learning model for advanced sentiment analysis.
- Store data in a database for scalability.
- Create a web app for real-time sentiment visualization using Flask or Django.