README: Twitter Data Fetching Using Tweepy API v2

This script demonstrates how to fetch recent tweets using the Twitter API v2 via the Tweepy library. It collects tweets based on a keyword and stores them in a CSV file for further analysis.

Requirements

1. **Python 3.8 or higher**
   Ensure Python is installed on your system. You can download it from python.org.
2. **Twitter Developer Account**
   You need a Bearer Token from Twitter's Developer Portal. Set up a project and get the API credentials here.
3. **Dependencies**
   Install the required libraries using pip:
4. pip install tweepy pandas

How to Use

1. **Clone or Copy the Script**
   Save the provided script as fetch_tweets.py or any filename of your choice.
2. **Replace the Bearer Token**
   Replace the bearer_token variable in the script with your own Bearer Token:
3. bearer_token = "YOUR_BEARER_TOKEN"
4. **Run the Script**
   Use the following command in your terminal or command prompt:
5. python fetch_tweets.py
6. **Specify the Keyword and Count**
   - Update the keyword variable in the script to search for specific topics, hashtags, or phrases.
   - Modify the count variable to specify the number of tweets to fetch (max 100 per request).
7. **Output**
   - The fetched tweets are displayed in the terminal as a DataFrame.
   - If tweets are found, they are saved to a CSV file named tweets.csv in the script's directory.

Script Overview

1. Initialization

- **Twitter API v2 Client Setup:**
  The script initializes the tweepy.Client using your Bearer Token.
- **Bearer Token:** Required to authenticate requests to the API.

2. Fetching Tweets

- **Function:** fetch_tweets_v2 takes a keyword and count as input.
- **Request:** Uses the search_recent_tweets endpoint to fetch the most recent tweets.

- **Response Handling:** Extracts text and created_at fields from the API response.

3. Error Handling

- Handles common errors such as:
  - Invalid or expired Bearer Token.
  - Exceeding rate limits (HTTP 429 error).
  - Empty responses when no tweets match the search query.

4. CSV Export

- Saves the fetched tweets into a CSV file (tweets.csv) for future use.

Points to Note

1. **Rate Limits:**
   Twitter API v2 has usage caps:
   - 450 requests per 15-minute window for the search_recent_tweets endpoint.
   - Plan upgrades on the Developer Portal increase limits.
2. **Max Results Per Request:**
   The max_results parameter allows fetching up to 100 tweets in one request.
3. **Data Privacy:**
   Do not share or misuse collected tweets, as Twitter's Developer Agreement requires responsible use of data.
4. **Error Handling:**
   - Check API credentials if you encounter authentication errors.
   - Handle rate limits gracefully by adding delays or retry mechanisms.

Customization Ideas

- **Keyword Automation:** Integrate a system to dynamically update keywords based on trends or user input.
- **Sentiment Analysis:** Use NLP libraries (e.g., TextBlob, NLTK) to analyze tweet sentiment.
- **Database Integration:** Save tweets to a database (e.g., SQLite, MySQL) for scalable applications.
