Project Documentation: Tweet Scraper using ntscraper

By-

Ayush Tripathi, 21BCE1148

Akhil Ajithkumar, 21BAI1177

1. Introduction

1.1 Purpose

This project aims to collect tweets on specific topics based on user input using the ntscraper library. The user provides a topic, and the program retrieves tweets containing the specified topic using the Nitter API.

1.2 Dependencies

- **ntscraper**
- **pandas**

2. Installation

2.1 Install Required Packages

To install the necessary packages, run the following commands in the terminal:

pip install ntscraper

pip install pandas

3. Code Implementation

3.1 Overview

The code is implemented in a Python script named **Code.py**. It uses the ntscraper library to interact with the Nitter API and retrieve tweets based on user input. The collected data is then organized into a Pandas DataFrame and saved as a CSV file.

3.2 Code

```
# Importing required libraries
import pandas as pd
from ntscraper import Nitter

# Initializing the ntscraper
scraper = Nitter()

# Function to get tweets based on user input
def get_tweets(name, modes, no):
    tweets = scraper.get_tweets(name, mode=modes, number=no)
    final_tweets = []

    # Extracting relevant information from each tweet
    for tweet in tweets['tweets']:
        data = [tweet['link'], tweet['text'], tweet['date'], tweet['stats']['likes'],
tweet['stats']['comments']]
        final_tweets.append(data)

    # Creating a Pandas DataFrame from the extracted data
    df = pd.DataFrame(final_tweets, columns=['link', 'text', 'date', 'likes', 'comments'])
    return df

# Taking user input for the topic
topic = input("Enter a topic you want tweets on: ")

# Getting tweets for the specified topic
data = get_tweets(topic, 'hashtag', 100)

# Saving the collected data to a CSV file
data.to_csv('tweets.csv')
```

4. Flowchart

4.1 Description

1. User provides a topic as input.
2. The **get_tweets** function is called with the specified topic, mode ('hashtag' in this case), and the number of tweets to retrieve.
3. The ntscraper library interacts with the Nitter API to fetch tweets.
4. Relevant information from each tweet is extracted and stored in a Pandas DataFrame.
5. The collected data is saved to a CSV file named 'tweets.csv'.

5. Output

The output of the script is a CSV file named 'tweets.csv', containing the collected tweets' information, including link, text, date, likes, and comments.

This documentation provides an overview of the project, details about the code implementation, a flowchart depicting the process, and information about the output generated by the script.