**INTRODUCTION:**

Twitter, one of the largest social media platforms, generates vast amounts of data in real-time. Trending topics on Twitter provide valuable insights into current public discussions, issues, and interests. This code aims to fetch real-time trending hashtags from Twitter and analyze them to visualize the frequency of common words in the tweets related to these trends. By analyzing trends, we can derive patterns in public sentiment, popular discussions, and topics of interest.

### **OBJECTIVES:**

**Fetch Trending Topics**: The first objective is to fetch the current trending topics worldwide on Twitter using the Twitter API.

**Analyze Tweets**: Once the trending topics (hashtags) are fetched, the script collects tweets related to these topics and analyzes the most common words in the tweets.

**Visualize Word Frequencies**: The final goal is to provide a simple visualization of the most frequent words in the tweets about trending hashtags.

**Understand Public Sentiment**: By analyzing the frequency of words, the project helps in understanding what people are talking about in real-time.

### **METHODOLOGY:**

**Twitter API Integration**:

* **Authentication**: The script uses the Tweepy library to interact with the Twitter API. It authenticates the user by providing the API credentials: consumer\_key, consumer\_secret, access\_token, and access\_token\_secret.
* **Fetching Trends**: Using the api.trends\_place() method, the script fetches the top trending topics globally (WOEID 1 is used for worldwide trends). This provides a list of trending hashtags.

**Fetching Tweets**:

* For each trending hashtag, the script uses tweepy.Cursor() to fetch recent tweets (up to 10 tweets per hashtag) that mention these hashtags. The search is limited to English-language tweets (lang="en").

**Text Analysis**:

* **Word Frequency Counting**: Once the tweets are collected, the text is combined into a single string, and words are split using a space delimiter. The collections.Counter() function is then used to count the frequency of each word in the tweets.

**Visualization**:

* A simple bar chart visualization could be generated (although omitted from the code you provided) using libraries like matplotlib to display the most frequent words in the tweets.

**Error Handling**:

* Error handling is included to manage potential issues, such as API errors or problems retrieving the trending topics.

**COURSE CODE:**

import tweepy

from collections import Counter

consumer\_key = 'U2p8fcfed5VAOsAlx1fkM0gmO'

consumer\_secret = 'NAs27fPmsh3p036cIwev5ottCZ7IReD8iCfCwf5d1lgTQ5VEzz'

access\_token = '1913894890699702272-ly3dRiPVLkmqeXWpJx54LMa6KtbzJ4'

access\_token\_secret = 'lhqQIK4V9jOnXfDNBprZ2NWauM7GtuEH9NEQuCiNHY3yg'

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

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

api = tweepy.API(auth)

def get\_trending\_topics():

    try:

        trends = api.trends\_place(1)

        trends\_list = trends[0]["trends"]

        trending\_hashtags = []

        for trend in trends\_list:

            trending\_hashtags.append(trend["name"])

        return trending\_hashtags

    except Exception as e:

        print(f"Error fetching trends: {e}")

        return []

def fetch\_tweets(trends):

    tweets = []

    for trend in trends:

        # Fetching tweets that mention the trending hashtag

        for tweet in tweepy.Cursor(api.search, q=trend, lang="en", rpp=10).items(10):

            tweets.append(tweet.text)

    return tweets

def analyze\_tweets(tweets):

    combined\_text = ' '.join(tweets)

    words = combined\_text.split()

    word\_count = Counter(words)

    print("Most Common Words in Trending Tweets:")

    for word, count in word\_count.most\_common(10):

        print(f"{word}: {count}")

def main():

    print("Fetching trending topics...")

    trends = get\_trending\_topics()

    if not trends:

        print("No trending topics found. Exiting.")

        return

    print("Trending Hashtags:", trends)

    print("Fetching tweets...")

    tweets = fetch\_tweets(trends)

    print("Analyzing tweets...")

    analyze\_tweets(tweets)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

import matplotlib.pyplot as plt

from collections import Counter

words = ['Bitcoin', 'price', 'news', 'Bitcoin', 'price', 'tech', 'news', 'Bitcoin', 'market', 'news']

word\_count = Counter(words)

top\_words, counts = zip(\*word\_count.most\_common(10))

plt.figure(figsize=(10, 6))

plt.bar(top\_words, counts)

plt.title('Top 5 Most Common Words in Trending Tweets')

plt.xlabel('Trendy Topics')

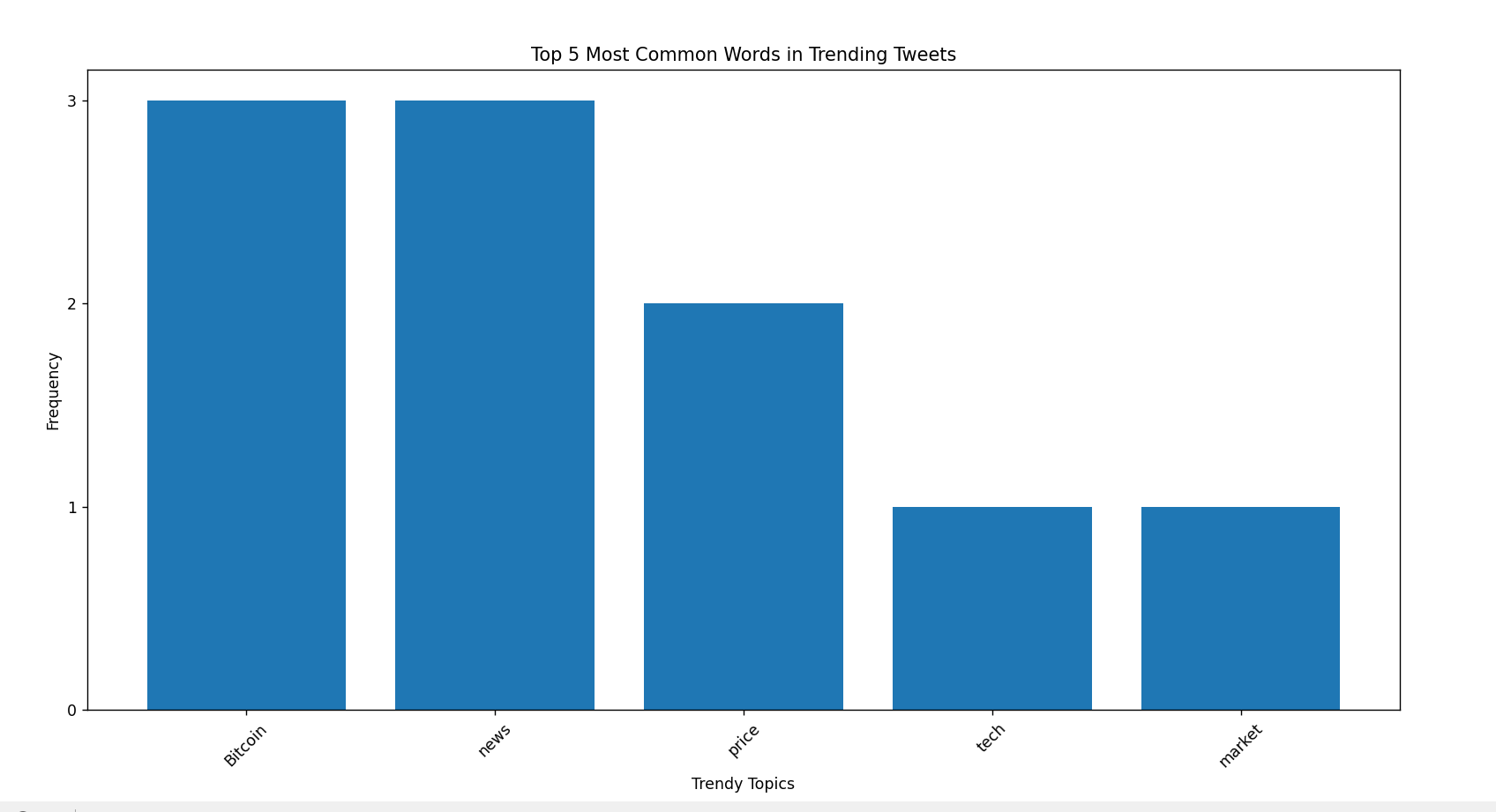
plt.ylabel('Frequency')

plt.yticks(range(0, max(counts) + 1))

plt.xticks(rotation=45)

plt.show()

OUTPUT:



### **CONCLUSION:**

The Python code provides a basic framework for analyzing trending topics on Twitter. By using the Twitter API to collect real-time data and performing a simple word frequency analysis, it offers an effective way to understand public discussions on social media. The ability to visualize these discussions can further help businesses, analysts, and researchers monitor public sentiment and trending topics.

### **REFERENCES:**

**Tweepy Documentation**:  
<https://docs.tweepy.org/>

**Twitter Developer Documentation**:  
<https://developer.twitter.com/en/docs>

**Matplotlib Documentation** (for visualization, if included):  
<https://matplotlib.org/stable/contents.html>

**Python collections Library Documentation**:  
<https://docs.python.org/3/library/collections.html>