Data Collection

Install Tweepy

To build the dataset for the '**Twitter Sentiment Analysis**' problem, we must first extract the tweets relevant to our topic – **Expo 2020**.

For this, we would be using the built-in python library '**Tweepy**' to extract the tweets related to Expo 2020.

Since Google Colab installs Version 3.10.0 of Tweepy by default, we will be downloading Version 4.4.0 instead to access the latest functions and features offered by Tweepy and ensure that we do not run into any problems while extracting the tweets.

A link to Tweepy's documentation - https://docs.tweepy.org/en/stable/ (https://docs.tweepy.org/en/stable/)

In [ ]:
```
# pip install tweepy==4.4.0
```

Imports

In [1]:
```
import tweepy
import configparser
import pandas as pd
import os
from os import access
from re import search
```

Accessing the API

For extracting the tweets using the Twitter API, we need to authenticate and provide access to the Twitter account. This can be done using the keys and tokens provided by Twitter.

We will be storing the keys in a local config file since they should not be shared with anyone but the developer. From this config file, we will be accessing our keys, which can be used to authenticate the Twitter account.

In [3]:
```
config = configparser.ConfigParser()      #(Spectrum, 2021)
config.read('../config.ini')
```

Out[3]:
```
['../config.ini']
```

In [2]:
```
consumerKey = config['twitter']['apiKey']
consumerSecret = config['twitter']['apiKeySecret']
accessToken = config['twitter']['accessToken']
accessTokenSecret = config['twitter']['accessTokenSecret']
```

Authenticating the Twitter Account

Using the functions offered by Tweepy and the keys provided by Twitter, we establish a connection with the Twitter API and provide access to our account.

After the connection and authentication are successful, we can begin collecting tweets related to Expo 2020.

To collect a large enough number of tweets, we needed to bypass the limit rate on the number of tweets allowed to be retrieved back. By default, the rate limit when searching for tweets matching a specific query is 450 tweets per 15 minutes (Twitter 2022b).

Thus, the wait_on_rate_limit has been set to True. This will allow the API initialized to wait automatically when a rate limit occur for a time period of 15 minutes (Tweepy 2022)

In [4]:
```
auth = tweepy.OAuthHandler(consumerKey, consumerSecret)
auth.set_access_token(accessToken, accessTokenSecret)
api = tweepy.API(auth, wait_on_rate_limit=True)   #provides access to t
```

Initial collector

In [ ]:
```
#themes = ["Innovation","Entertainment","Sustainability", "Golden Jubil
#themes = ["Entertainment"]

#termToSearch = "Expo 2020"
numOfTweets = 1000
engLanguage = "en"
expoTweets = []

#for theme in themes:
  #termToSearch = '#Expo2020+{}'.format(theme),
termToSearch = '(#Expo2020) AND (Entertainment OR Sustainability OR "Go
#termToSearch = "#Expo2020"
tweetCursor = tweepy.Cursor(
    api.search_tweets,
    #plenty of the duplicates come from the RT, hence we filter them ou
    q='{} -filter:retweets'.format(termToSearch),  #(Morrissey, Wasser
    lang = engLanguage,
    tweet_mode = "extended",     #(manoelhortaribeiro, 2017)
    result_type = "mixed",
    count = 100,
).items(numOfTweets)

for tweet in tweetCursor:          #(W3schools.com, 2022)
    tweetBody = tweet.full_text    #(manoelhortaribeiro, 2017)
    expoTweets.append(tweetBody)

tweetDataFrame = pd.DataFrame(expoTweets, columns=['Tweet Body'])
tweetDataFrame = tweetDataFrame.drop_duplicates()      #(Lungu, 2019)
tweetDataFrame.to_csv('extractedTweets.csv')
```

Collection of the Tweets

- As there is a possibility of duplicate tweets appearing when collecting the tweets, we decided to use a set. All the elements in a set are unique, as a set cannot contain two elements with the same value. Python's `set()` constructor was used to create the set to store the tweets.

- To collect a wide, diverse collection of tweets, a query was built to include various keywords related to Expo 2020. We started with the term "Expo" and its possible alternate forms that could be present in a tweet, such as a hashtag or a text, and stored all stored of them in the variable `expo2020_str`. Additionally, when people tweet about Expo 2020, sometimes they tag the official Expo 2020 Dubai twitter account (@expo2020dubai). Therefore, we have added `@expo` and `@Expo` as another possible other form of the term "Expo". As we aim to extract tweets about Expo 2020, we have added "2020" to `expo2020_str` to ensure we only get the tweets that are relevant to Expo 2020.
- Using Twitter (2022a) API documentation, the idea behind building the queries was as follows. Since the main keyword to be included in all tweets is `#Expo2020`, the `AND` operation was used to include the optional keywords we want to search for. These words range from `Entertainment` to `Culture`, thus, diversifying the collection of tweets to cover all the topics that were actively discussed in relation to Expo 2020. Any keyword in quotation marks such as "Golden Jubilee" when queried will return an exact match of the keyword used. In this instance, all tweets containing exactly #Expo and "Golden Jubilee" will be returned. This was how the query `'(#Expo2020) AND (Entertainment OR Sustainability OR "Golden Jubilee" OR Sports OR Tourism OR Arts OR Culture)'` was created.
- From there, we expanded the queries from that simple concept to include more keywords. We have chosen 43 different keywords, such as “Mobility”, “Sustainability”, “Opportunity”, etc., and mentioned them in alternative forms, inspired by the various ways people can mention a single word in a tweet. E.g. `(opportunity OR Opportunity OR #opportunity OR #Opportunity OR @opportunity OR @Opportunity)`. That gets concatenated with expo2020_str to return back specific tweets. According to Twitter (2022a), broad queries are not efficient, hence why we are not collecting only tweets where Expo 2020 appears but are being more specific.
- Retweeted content showed up multiple times in the dataset, since the original tweet was retweeted by multiple users. To avoid these duplicates, all retweets were filtered out. This was done using `-filter:retweets` in the query. Below we can see an instance of how retweets created duplicates within the dataset. The retweet does not differ from the original tweet hence why they were viewed as duplicates and were removed. (Twitter 2022a)

RT @expo2020dubai: Attention! See the preparations for the Union Fortress 8 parade on 4, 11, 18, and 25 February, along Ghaf Avenue and sto…
Attention! See the preparations for the Union Fortress 8 parade on 4, 11, 18, and 25 February, along Ghaf Avenue and stopping along Jubilee Park and Al Forsan Park at Expo 2020 Dubai. Get a taste of the parade itself, scheduled for March. Join us! https://t.co/lgx2OUyYIA

- Since some of the Expo 2020 tweets may be written in a different language, labeling these tweets by code or manually would not produce accurate results for the sentiment analysis problem. To fix this, we have set the `lang` parameter of `api.search_tweets` to `en`. By setting the language as English, we will be able to collect only tweets posted in English.
- To get the best results for the Twitter Sentiment Analysis problem, we aimed to obtain a mixture of tweets that included the most popular and recent Expo 2020 tweets posted. As a result, we set the `result_type` parameter to `mixed` to get a variety of tweets.
- After getting the tweets, we add them to a set to collect only the unique ones. We noticed that Tweepy had truncated the tweet body to around 140 characters while collecting the tweets. Since we needed the entire tweet body for performing sentiment analysis, we fixed the issue of truncated tweets by setting the `tweet_mode` parameter as `extended` and used the `.full_text` when adding the tweets to the set.

In [8]:

```
'''
Initializing a set to store the tweets
'''
expoTweets = set()
```

In [9]:

```
themes = ["Innovation","Entertainment","Sustainability", "Golden Jubile
termToSearch = "#Expo2020"
for theme in themes:
    termToSearch = '#Expo2020+{}'.format(theme)

#more on building a query: https://developer.twitter.com/en/docs/twitte
# themes = ["Innovation", "Entertainment", "Sustainability", "Golden Ju
expo2020_str = "(#Expo OR Expo OR expo OR #expo OR @Expo OR @expo) AND
termsToSearch = [
    "#Expo2020+Innovation",
    "#Expo2020+Entertainment",
    "#Expo2020+Sustainability",
    "#Expo2020+Golden Jubilee",
    "#Expo2020+Arts and Culture",
    "#Expo2020+Sports",
    "#Expo2020+Tourism",
    '"#Expo2020',
    '(#Expo2020) AND (Entertainment OR Sustainability OR "Golden Jubile
    "@expo2020Dubai",
    '(#Expo OR Expo OR expo OR #expo OR @Expo OR @expo) AND 2020',
    expo2020_str + ' AND (Dubai OR dubai OR #dubai OR #Dubai OR @Dubai
    expo2020_str + ' AND (uae OR UAE OR #uae OR #UAE OR @uae OR @UAE)',
    expo2020_str + ' AND (entertainment OR Entertainment OR #entertainm
    expo2020_str + ' AND (sustainability OR Sustainability OR #sustaina
    expo2020_str + ' AND (golden jubilee OR Golden Jubilee OR #golden j
    expo2020_str + ' AND (sports OR Sports OR #sports OR #Sports OR @sp
    expo2020_str + ' AND (Tourism OR tourism OR #Tourism OR #tourism OR
    expo2020_str + ' AND (art OR Art OR #art OR #Art OR @art OR @Art)',
    expo2020_str + ' AND (culture OR Culture OR #culture OR #Culture OR
    expo2020_str + ' AND (heritage OR Heritage OR #heritage OR #Heritag
    expo2020_str + ' AND (concert OR Concert OR #concert OR #Concert OR
    expo2020_str + ' AND (festival OR Festival OR #festival OR #Festiva
    expo2020_str + ' AND (innovation OR Innovation OR #innovation OR #I
    expo2020_str + ' AND (virtual OR Virtual OR #virtual OR #Virtual OR
    expo2020_str + ' AND (pavilion OR Pavilion OR #pavilion OR #Pavilio
    expo2020_str + ' AND (architecture OR Architecture OR #architecture
    expo2020_str + ' AND (dxb OR Dxb OR #dxb OR #Dxb OR @dxb OR @Dxb OR
    expo2020_str + ' AND (design OR Design OR #design OR #Design OR @de
    expo2020_str + ' AND (food OR Food OR #food OR #Food OR @food OR @F
    expo2020_str + ' AND (technology OR Technology OR #technology OR #T
    expo2020_str + ' AND (future OR Future OR #future OR #Future OR @fu
    expo2020_str + ' AND (mobility OR Mobility OR #mobility OR #Mobilit
    expo2020_str + ' AND (experience OR Experience OR #experience OR #E
    expo2020_str + ' AND (exhibition OR Exhibition OR #Exhibition OR #e
    expo2020_str + ' AND (passport OR Passport OR #passport OR #Passpor
    expo2020_str + ' AND (covid OR Covid OR #Covid OR #Covid OR @covid
    expo2020_str + ' AND (Cristiano OR cristiano OR #Cristiano OR #cris
    expo2020_str + ' AND (ronaldo OR Ronaldo OR #ronaldo OR #Ronaldo OR
    expo2020_str + ' AND (opportunity OR Opportunity OR #opportunity OR
    expo2020_str + ' AND (forsan OR Forsan OR #forsan OR #Forsan OR @fo
    expo2020_str + ' AND (opti OR OPTI OR Opti OR #opti OR #OPTI OR #Op
    expo2020_str + ' AND (emirates OR Emirates OR #emirates OR #Emirate
    expo2020_str + ' AND (robot OR Robot OR #robot OR #Robot OR @robot
    expo2020_str + ' AND (sustainable OR Sustainable OR #sustainable OR
    expo2020_str + ' AND (wasl OR Wasl OR #wasl OR #Wasl OR @wasl OR @W
    expo2020_str + ' AND (terra OR Terra OR #terra OR #Terra OR @terra
    expo2020_str + ' AND (music OR Music OR #music OR #Music OR @music
    expo2020_str + ' AND (water OR Water OR #water OR #Water OR @water
    expo2020_str + ' AND (digital OR Digital OR #digital OR #Digital OR
```

```
    expo2020_str + ' AND (climate OR Climate OR #climate OR #Climate OR
    expo2020_str + ' AND (tourists OR Tourists OR #tourists OR #Tourist
    expo2020_str + ' AND (country OR Country OR #country OR #Country OR
    expo2020_str + ' AND (countries OR Countries OR #countries OR #Coun
]

'''
Setting the maximum number of tweets to search as 10000.
'''
numOfTweets = 10000
engLanguage = "en"

'''
Initializing a counter to keep track of the number of tweets we were ab
'''
tweet_count = 0

# for termToSearch in termsToSearch:
for i in range(len(termsToSearch)):
    termToSearch = termsToSearch[i]
    tweetCursor = tweepy.Cursor(
        api.search_tweets,
        '''
         As plenty of duplicates appear from the Retweets (RT), we filt
        '''
        q='{} -filter:retweets'.format(termToSearch),  #(Morrissey, Was
        lang = engLanguage,
        tweet_mode = "extended",     #(manoelhortaribeiro, 2017)
        result_type = "mixed",
        count = 100,                '''We try to retrieve 100 tweets pe
        include_entities = False
    ).items(numOfTweets)

    for tweet in tweetCursor:          #(W3schools, 2022)
        expoTweets.add(tweet.full_text) #(manoelhortaribeiro, 2017)
        tweet_count += 1                    '''Incrementing the counte
    print(termToSearch)
    print("no. of tweets retrieved = {}, no. of unique tweets = {}".for
# 6613
```

```
#Expo2020+Innovation
no. of tweets retrieved = 79, no. of unique tweets = 77
#Expo2020+Entertainment
no. of tweets retrieved = 84, no. of unique tweets = 82
#Expo2020+Sustainability
no. of tweets retrieved = 114, no. of unique tweets = 108
#Expo2020+Golden Jubilee
no. of tweets retrieved = 116, no. of unique tweets = 110
#Expo2020+Arts and Culture
no. of tweets retrieved = 117, no. of unique tweets = 111
#Expo2020+Sports
no. of tweets retrieved = 131, no. of unique tweets = 124
#Expo2020+Tourism
no. of tweets retrieved = 154, no. of unique tweets = 146
"#Expo2020
no. of tweets retrieved = 2037, no. of unique tweets = 1884
(#Expo2020) AND (Entertainment OR Sustainability OR "Golden Jubilee" OR Sp
orts OR Tourism OR Arts OR Culture)
no. of tweets retrieved = 2149, no. of unique tweets = 1886
```

Saving the tweets collected

Initially, we created a Pandas DataFrame to store the set of unique tweets. We chose to use a DataFrame for the tweets as Pandas offers a function `to_csv()` , which helps to convert the DataFrame into a csv format.

After converting it to a csv format, we can then store the tweets in a .csv file, which will serve as our dataset.

In [10]:
```
'''
Initializing the DataFrame for the tweets, and setting the column of th
'''
tweetDataFrame = pd.DataFrame(expoTweets, columns=['Tweet Body'])
print(tweetDataFrame.shape)
```

```
(6613, 1)
```

In [11]:
```
'''
Converting the DataFrame into a csv format so that we can store it in a
'''
tweetDataFrame.to_csv('./data/tweets.csv', index=False)
```

References

L

- Lungu, C. (2019). Exploring Twitter data using Python: An Intro to NLP and Sentiment Analysis. Medium. [online]
  Available at: https://medium.com/analytics-vidhya/exploring-twitter-data-using-python-af1287ee65f1 (https://medium.com/analytics-vidhya/exploring-twitter-data-using-python-af1287ee65f1).

M

- Manoelhortaribeiro (2017). Tweepy not getting full text · Issue #935 · tweepy/tweepy. GitHub. [online]
  Available at: https://github.com/tweepy/tweepy/issues/935 (https://github.com/tweepy/tweepy/issues/935).
- Morrissey, M., Wasser, L. and Farmer, C., 2020. *Automate Getting Twitter Data in Python Using Tweepy and API Access.* [online] Earth Data Science.
  Available at: https://www.earthdatascience.org/courses/use-data-open-source-python/intro-to-apis/twitter-data-in-python/ (https://www.earthdatascience.org/courses/use-data-open-source-python/intro-to-apis/twitter-data-in-python/)

S

- Spectrum, A. (2021). How to get TWEETS by Python | Twitter API 2022. YouTube. [online]
  Available at: https://www.youtube.com/watch?v=Lu1nskBkPJU&t=601s (https://www.youtube.com/watch?v=Lu1nskBkPJU&t=601s).

T

- Tweepy. (2022). *Examples — tweepy 4.5.0 documentation.* [online] Available at: https://docs.tweepy.org/en/stable/examples.html?highlight=wait_on_rate_limit (https://docs.tweepy.org/en/stable/examples.html?highlight=wait_on_rate_limit)
- Twitter. (2022a). *Search Tweets - How to build a query.* [online] Available at: https://developer.twitter.com/en/docs/twitter-api/tweets/search/integrate/build-a-query (https://developer.twitter.com/en/docs/twitter-api/tweets/search/integrate/build-a-query).
- Twitter. (2022b). *Standard search API.* [online] Available at: https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/api-reference/get-search-tweets (https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/api-reference/get-search-tweets).

W

- W3schools. (2022). Python Iterators. [online] Available at: https://www.w3schools.com/python/python_iterators.asp (https://www.w3schools.com/python/python_iterators.asp)