Prise en main de l'API

Permet d'installer l'API Tweepy sur la machine

In [1]:
```
import sys
!{sys.executable} -m pip install tweepy
```

```
Requirement already satisfied: tweepy in /Users/arnaudbascop/opt/anaconda3/lib/python3.7/site-packages (3.9.0)
Requirement already satisfied: six>=1.10.0 in /Users/arnaudbascop/opt/anaconda3/lib/python3.7/site-packages (from twe
epy) (1.12.0)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /Users/arnaudbascop/opt/anaconda3/lib/python3.7/site-packa
ges (from tweepy) (1.3.0)
Requirement already satisfied: requests[socks]>=2.11.1 in /Users/arnaudbascop/opt/anaconda3/lib/python3.7/site-packag
es (from tweepy) (2.22.0)
Requirement already satisfied: oauthlib[...]
```

Importer la bibliothèque Tweepy

In [2]:
```
import tweepy
```

In [3]:
```
from tweepy import OAuthHandler
```

Informations nécessaires à son fonctionnement : ids TwitterApp

In [4]:
```
consumer_key = 'L2mPv8Oxjs2mLtBEWZZiyDy0W'
consumer_secret = 'idPHQogjNLSKdo7ns8exk9ZXSbIsqcd4UsXd1vofliwj1s2iZj'
access_token = '1413867470-bWJRTSz2Ece90tG1Z5IU9pmPZQvdWR10xrWqNBL'
access_secret = '6Z0X9eJ0dLlYJj8JrqX6VcNMOOsTWpaxaFosJhiv5Xkoa'

auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)

api = tweepy.API(auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True, compression=True)
```

Exemple de boucle pour afficher les dix derniers tweets de ma timeline

In [5]:
```
for status in tweepy.Cursor(api.home_timeline).items(10):
    # Process a single status
    print(status.text)
```

```
RT @CherrySlllll: Non: pas consentant.e -j'ai pas envie: pas consentant.e -je sais pas: pas consentant.e -sous l'empr
ise de drogue.alcool:…
RT @syvzwvn: RT for extreme luck https://t.co/0U3aaOx9ZH
RT @HumansNoContext: https://t.co/8r7Aa4IEhq
RT @alexadem1e: https://t.co/5HWT0OdWIp
RT @SabrinaBabySlut: Im so glad we have a flawless voting system in Aus https://t.co/v7JkPMVj78
RT @moraberly: envie d'avoir un calendrier de l'avant mac ou nyx ou sephora
RT @Dedodante: Je crois que Leclerc envoit des messages subliminaux. https://t.co/nOAHuydcXW
Une série sur les agents, vous pensez que ça peut marcher ? □

Retrouvez l'intégralité de la saison 4 de… https://t.co/IfSk4Oso25
RT @sasouumk: QUI A PRIS MON CHARG...

ah non c'est bon j'ai trouvé
J'ai rêvé de ma twingo □ RIP
```

Exemple de code pour récupérer un @ et le nombre de ses followers

In [6]:
```
user = api.get_user('lemondefr')
print(user.screen_name)
print(user.followers_count)
```

```
lemondefr
9000537
```

Exemple de code pour récupérer mes 274 followers

In [7]:
```
import time
```

In [8]:
```
# Fonction qui permet de contourner la sécuriter de l'API twitter
def limit_handled(cursor):
    while True:
        try:
            yield cursor.next()
        except tweepy.RateLimitError:
            time.sleep(15 * 60)

# On parcours dans mes followers
nb_follower = 0
for follower in limit_handled(tweepy.Cursor(api.followers).items()):
    if follower.friends_count < 300:
        print(follower.screen_name)
        nb_follower = nb_follower + 1
print(nb_follower)
```

```
FERON03876207
Hana__heh
imen2tp
SakuinM
PierreMaxxx
AdrianaCiuffra1
mcb1708
ElliiiseeeElise
farfadome
MJulie50350766
kvdvg
[...]
```

In [9]:
```
print(nb_follower)
# nombre de mes followers qui ont moins de 300 abonnements
```

```
178
```

Essayons d'établir une petite base de données

In [10]:
```
# un pôle arbitraire: supposons que ceux qui s'intéressent un minimum à la politique suivent le président actuel
pole = api.get_user('EmmanuelMacron')
print(pole.screen_name)
print(pole.followers_count)
```

```
EmmanuelMacron
6220443
```

In [11]:
```
# On va fouiller et récupérer les ids de 0.1% de ses followers, soit d'environ 6217 personnes
# En récupérant seulement les identifiants de follow des utilisateurs,
# on atteint un ratio de récupération de 75 000 followers toutes les 15 minutes. Ici, on stop à plus de 6000

followerids =[]
for user in tweepy.Cursor(api.followers_ids, screen_name=pole.screen_name,count=5000).items():
    followerids.append(user)
    if len(followerids) > 6217:
        break
print(len(followerids))
```

```
6218
```

In [12]:
```
# On vérifie les ids en affichant le tableau
for id in followerids:
    print(id)
```

```
951205235674009600
1324280478489288707
1324281664302886915
1324277771842060289
1324281079868674053
889339478673940481
1324281700533358593
1324279659299819525
1264001281531748352
1324100889330749441
1324097882878840836
1263163433094197248
1324099004171489281
1323995743603068928
430024785
1324100978124197889
1324100867813969920
[...]
```

In [13]:
```
# Grâce aux ids, on peut accéder aux informations de ces 6218 utilisateurs
# On peut aller jusqu'à 18 000 utilisateurs toutes les 15 minutes
users = []
for i in range(int(len(followerids)/100) + 1):
    fin_cycle = min((i + 1) * 100, len(followerids))
    users.extend(
        api.lookup_users(user_ids=followerids[i * 100:fin_cycle])
    )
```

In [14]:
```
# On vérifie les utilisateurs en affichant la taille du tableau et leur @
print(len(users))
for i in range(len(users)):
    print(users[i].screen_name)
```

```
6218
SlqzzFTN
DominiqueLegoff
FerielHaddouche
JasmineLauraIb1
ElisabethYaho
HemalDabgar
bessard_emilie
celya26565303
SamoukaDiarra20
BrunaSi98616742
marion_mwimmer
SumitGuha621
tehb0htehlipt0n
CottierVincent
[...]
```

In [15]:
```
# Super : on a 0.1% de la communauté Twitter de Macron ! On peut maintenant en savoir plus sur eux
# Cherchons leurs liens, ce qui connecte cette communauté entre elle
# On détermine un lien : le fait de suivre la même personne
# On devrait avoir Macron en mega-hub, mais également la découverte d'autres
```

In [17]:
```
# On va parcourir chaque utilisateur et récupérer leurs abonnements,
# pour ce faire on utilise le dictionnaire afin de stocker les bonnes infos
# Le problème, c'est que faire une requête tweepy.cursor pour chaque user, c'est explosé la limite
# et cette fois, il n'y a pas de moyen de contourner
# Alors ça peut être long : la liste des abonnements pour 90 utilisateurs sur 6218 utilisateurs a pris 1h30 !
# Si on voulait 1000 personnes, il faudrait 15h. Nos 6218 : presque 4 jours.
# (L'erreur est dûe au fait que j'ai arrêté l'algo pour exploiter seulement les 90 utilisateurs)
# Ainsi, on se retrouve pour la suite avec les abonnements de 0.001% de la communauté de Macron

abonnementsids = {}
for i in range(len(users)):
    # Pour chaque utilisateur, on récupère ses abonnements avec la même méthode
    tmpids = []
    # Ne pas oublier la gestion d'erreur pour les comptes privés (jusque 13%), il y en a beaucoup !!
    try:
        for abonnement in tweepy.Cursor(api.friends_ids, screen_name=users[i].screen_name, count=5000).items():
            tmpids.append(abonnement)
    except tweepy.TweepError:
        print("Erreur utilisateur, au suivant")
    abonnementsids[users[i].screen_name] = tmpids
len(abonnementsids)
```

```
Erreur utilisateur, au suivant
Rate limit reached. Sleeping for: 845
Erreur utilisateur, au suivant
Erreur utilisateur, au suivant
Rate limit reached. Sleeping for: 892
Erreur utilisateur, au suivant
Rate limit reached. Sleeping for: 892
Erreur utilisateur, au suivant
Erreur utilisateur, au suivant
Erreur utilisateur, au suivant
Rate limit reached. Sleeping for: 893
Erreur utilisateur, au suivant
Rate limit reached. Sleeping for: 891
Erreur utilisateur, au suivant
Erreur utilisateur, au suivant
Rate limit reached. Sleeping for: 892
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-17-987030d65ec9> in <module>
     13     # Ne pas oublier la gestion d'erreur pour les comptes privés (jusque 13%), il y en a beaucoup !!
     14     try:
---> 15         for abonnement in tweepy.Cursor(api.friends_ids, screen_name=users[i].screen_name, count=5000).items(
):
     16             tmpids.append(abonnement)
     17     except tweepy.TweepError:
```

In [46]:
```
# On vérifie que le nombre des informations récupérées est cohérent

# Nombre d'abonnements des 90 users récupérés
res = 0
for k, v in abonnementsids.items():
    for j in range(len(v)):
        res = res + 1
print(res)
# Nombre d'abonnements des +6000 users qu'on devrait avoir
res = 0
for i in range(len(users)):
    res += users[i].friends_count
print(res)
```

```
10319
612735
```

In [58]:
```
# On va éliminer tous les utilisateurs qui ont posé problème ou qui sont inutiles (0 abonnement ou compte privé)
abonnementsids_final = {}
for key, value in abonnementsids.items():
        if len(value) > 0:
            abonnementsids_final[key] = value;
print(len(abonnementsids))
print(len(abonnementsids_final))
# On en a éliminé 17
# Nombre d'abonnements des 73 users récupérés
res = 0
for k, v in abonnementsids_final.items():
    for j in range(len(v)):
        res = res + 1
print(res)
```

```
90
73
10319
```

In [90]:
```
def existe(cle, liste):
    if cle in liste:
        return "true"
    return "false"
```

In [105]:
```
# Il est temps d'établir les liens qui unissent 0.001% de la communauté de Macron
# Nous travaillons pour l'instant avec les ids des followers (pas de @)
# Dictionnaire avec clé : id, valeur : nbLien
liens = {}
liste_visite_ids = []
for k, v in abonnementsids_final.items():
    nbLiens = 0
    for i in range(len(v)):
        for key, value in abonnementsids_final.items():
            for j in range(len(value)):
                if(v[i] == value[j] and k != key and existe(key, liste_visite_ids) == "false"):
                    nbLiens = nbLiens + 1
        if(existe(v[i], liens) == "true"):
            liens[v[i]] += nbLiens
        else:
            liens[v[i]] = nbLiens
    liste_visite_ids.append(k)
liens
```

Out[105]:
```
{600345740: 50,
 1976143068: 4064,
 1231659627818995714: 75,
 1284808847048286208: 107,
 1177162175263182848: 76,
 759049329197846528: 76,
 971820228: 76,
 122333150: 76,
 984709076876713984: 108,
 11348282: 961,
 44196397: 741,
 953315323075268608: 93,
 895640528238374912: 93,
 3296914397: 93,
 975716046819807232: 94,
 28351525: 416,
 30479919: 332,
 1147115632669941760: 96,
 3309184942: 96,
 1315380544008581126: 96,
 926257982: 96,
 1124012369657651204: 96,
 1305044987516682241: 96,
 421333672: 222,
 735416416434851845: 97,
 25069929: 823,
 1053068199632224256: 102,
    [...]
```

In [108]:
```
# Vérifions que chaque lien est unique
from collections import Counter
print(Counter(liens.keys()))
```

```
Counter({600345740: 1, 1976143068: 1, 1231659627818995714: 1, 1284808847048286208: 1, 1177162175263182848: 1, 7590493
29197846528: 1, 971820228: 1, 122333150: 1, 984709076876713984: 1, 11348282: 1, 44196397: 1, 953315323075268608: 1, 8
95640528238374912: 1, 3296914397: 1, 1026937440752361472: 1, 1318685948356460545: 1, 848323309473320962: 1, 121604838
9688827905: 1, 1320286347735826432: 1, 1027898750210072576: 1, 1214315619031478272: 1, 1078020752975237121: 1, 105872
2410860634114: 1, 1148610660374388738: 1, 1129875967965573122: 1, 798269733036523521: 1, 859468269358641152: 1, 93013
7039227506691: 1, 975716046819807232: 1, 989179899025018885: 1, 1004014467015995393: 1, 1105143949155598338: 1, 12573
51083065782273: 1, 28351525: 1, 30479919: 1, 1147115632669941760: 1, 1226863927147061248: 1, 1126107773119365120: 1,
1000779873458229249: 1, 1078483119626428416: 1, 1078614259284787200: 1,[...]
```

In [135]:
```
# Récupérons nos 10 premiers mega-hubs
liensTries = sorted(liens.values())
usersid_tries = []
for i in range(10):
    for k, v in liens.items():
        if liensTries[len(liensTries)-1-i] == v:
            usersid_tries.append(k)
hubs = []
try:
    hubs.extend(
        api.lookup_users(user_ids=usersid_tries)
    )
except tweepy.TweepError:
    print('oops')
```

In [136]:
```
for i in hubs:
    print(i.screen_name)
    print(i.followers_count)
len(users)
```

```
EmmanuelMacron
6222617
BarackObama
124966386
paulpogba
8014080
Cristiano
89012082
rihanna
99356667
KimKardashian
67278519
ChampionsLeague
31108520
KMbappe
5469203
lemondefr
9001409
neymarjr
50469394
```

Out[136]:
```
20
```

Conclusion

Si on admet que 0.001% soit significatif des followers de Macron (lol), le plus grand nombre de liens se dirige vers Barack Obama, Paul Pogba, Christiano Ronaldo, Le Monde, ou même encore Kim Kardashian La communauté apprecierait donc le foot (50% des 10 plus gros hubs) Un léger penchant politique (20% des plus gros hubs) Et un penchant pour la culture pop (20% des plus gros hubs)

Cette expérimentation a montré que la difficulté de notre projet se trouve dans la limite imposée par l'API de twitter, mais également dans la complexité de l'algorithme déterminant les liens avec une imbrication de 4 boucles. On sait à présent qu'il va falloir correctement anticiper les algorithmes pour pouvoir les faire tourner assez longtemps afin de récupérer des données exploitables (environ 1 semaine pour 8000 liste d'abonnements de d'utilisateurs) Aussi, la sélection des hubs de départ est très importante, mais ne pas sélectionner trop de hubs ! Cela aurait un facteur multiplicatif important sur toutes les durées des algorithmes Cependant, à notre échelle, le travail est tout de même faisable et ça c'est top