

```

1 # Some helpful resources:
2 # https://twython.readthedocs.io/en/latest/api.html
3 # https://twython.readthedocs.io/en/latest/usage/basic_usage.html
4 # https://twython.readthedocs.io/en/latest/usage/streaming_api.html
5
6 from twython import Twython, exceptions
7 from datetime import datetime, timedelta
8 import time
9 import os
10
11 # This are the codes that must be requested to Twitter in order to use
12 # their API.
13 # This codes are confidential and non-transferable.
14 CONSUMER_KEY = *** # Python character string
15 CONSUMER_SECRET = *** # Python character string
16 ACCESS_TOKEN = *** # Python character string
17 ACCESS_TOKEN_SECRET = *** # Python character string
18
19 # Get a new Twython instance using them.
20 twitter = Twython(CONSUMER_KEY, CONSUMER_SECRET,
21                  ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
22
23
24 def follow_rate(user):
25     '''Get the rate of followers and friends.'''
26
27     followers = twitter.get_followers_ids(screen_name=user)['ids']
28     friends = twitter.get_friends_ids(screen_name=user)['ids']
29
30     print('followers/friends = ', len(followers), '/',
31          len(friends), ' = ', len(followers) / len(friends))
32
33     match_number = 0
34     for id in followers:
35         if id in friends:
36             match_number += 1
37     print('match number = ', match_number)
38
39
40 def get_users_groups(user, new_path, path):
41     '''Get the social groups where this user takes part.
42     It searches in his/hers followers and friends to check if there are
43     matches between them... This could be much more exhaustive.'''
44     if not path:
45         # followers = twitter.get_followers_ids(screen_name=user)['ids']
46         friends_ids = twitter.get_friends_ids(screen_name=user)['ids'] # #
47         get_friends_list(screen_name=user)['users']
48         file_temp = open(new_path, 'w')
49         file_temp.write(repr(friends_ids))
50         file_temp.close()
51     else:
52         file = open(path)
53         friends_ids = eval(file.readline().strip())
54         file.close()
55     max_attempt_number = 8
56     friends_net = []
57
58     # If an exception occurs in user n, then the variable c should be
59     # assigned that number (n) in order to recatch the process
60     # The number in the iterator should also be set to that number (n)
61     c = 1
62     for friend_id in friends_ids[1:]: # THIS ITERATOR
63         c += 1
64         print('----- ', c, ':', friend_id, '-----')
65         friends_net_file = open(new_path + '_netdata', 'a')
66         for other_id in friends_ids[c:]:
67             attempt = 0
68             while attempt < max_attempt_number:
69                 try:
70                     friendship = twitter.show_friendship(source_id=friend_id,
71                                                         target_id=other_id)
72                 except Exception as exc:
73                     attempt += 1
74                     assert attempt < max_attempt_number, 'attempt time {} exceeded'.
75                     format(max_attempt_number)
76             friends_net_file.close()

```

```

75         print('failed attempt at: ', datetime.now())
76         print('exception: ', exc)
77         time.sleep(6 * 60)
78         print('reattempting at: ', datetime.now())
79         friends_net_file = open(new_path + '_netdata', 'a')
80     else:
81         break
82
83     print('keep going: ', other_id)
84     if friendship['relationship']['source']['following'] \
85         and friendship['relationship']['source']['followed_by']:
86         friends_net.append((friend_id, other_id))
87         print(repr((friend_id, other_id)))
88         friends_net_file.write(repr((friend_id, other_id)) + '\n')
89     friends_net_file.close()
90     return friends_net
91
92
93 def get_relationship_srctrgt(source, targets, get_matched_likes=False):
94     """
95     Checks the interaction the source user has with
96     the targets provided in targets.
97     """
98     liked = twitter.get_favorites(id=source, count=199)
99     timeline = twitter.get_user_timeline(id=source, count=199)
100    interaction_list = []
101
102    for target in targets:
103        interaction_dict = {'source_user_id': source,
104                           'target_user_id': target,
105                           'favorites': [],
106                           'retweeted': [],
107                           'replies': [],
108                           'matched_likes': None}
109        if get_matched_likes:
110            interaction_dict['matched_likes'] = []
111
112        if get_matched_likes:
113            trgt_likes_ids = []
114            for tweet in twitter.get_favorites(id=target, count=199):
115                trgt_likes_ids.append(tweet['id'])
116
117        for tweet in liked:
118            if target == tweet['user']['id']:
119                interaction_dict['favorites'].append(tweet['id'])
120
121            if get_matched_likes:
122                if tweet['id'] in trgt_likes_ids:
123                    interaction_dict['matched_likes'].append(tweet['id'])
124
125        for tweet in timeline:
126            if target == tweet['in_reply_to_user_id']:
127                interaction_dict['replies'].append((tweet['id'], tweet['
in_reply_to_status_id']))
128
129            if tweet.get('retweeted_status', False) \
130                and target == tweet['retweeted_status']['user']['id']:
131                interaction_dict['retweeted'].append((tweet['id'], tweet['
retweeted_status']['id']))
132
133        interaction_list.append(interaction_dict)
134
135    if len(interaction_list) == 1:
136        return interaction_list[0]
137    else:
138        return interaction_list
139
140
141 def get_relationship_interactions(net, storage_path=None):
142     """
143     Checks the interaction that every user in net has
144     with the other users in net.
145     """
146     liked = {}
147     timeline = {}
148     for user_id in net:

```

```

149         # get_favorites limited to 75 req./15 min, 200 most recent
150         liked[user_id] = twitter.get_favorites(id=user_id, count=199)
151         # get_favorites limited to 900 req./15 min, 200 most recent
152         timeline[user_id] = twitter.get_user_timeline(id=user_id, count=199)
153         timemark = str(datetime.now())
154         file = open('twitter_raw/' + str(user_id) + '_liked_' + timemark, 'w')
155         file.write(repr(liked[user_id]))
156         file.close()
157         file = open('twitter_raw/' + str(user_id) + '_timeline_' + timemark, 'w')
158         file.write(repr(timeline[user_id]))
159         file.close()
160
161     interaction_list = []
162
163     for source_id in net:
164         for target_id in net:
165             if source_id == target_id:
166                 continue
167             friendship = twitter.show_friendship(source_id=source_id, target_id=
target_id)
168
169             interaction_dict = {'source_user_id': source_id,
170                               'target_user_id': target_id,
171                               'source_follows_target': friendship['relationship']
172                               ['source']['following'],
173                               'target_follows_source': friendship['relationship']
174                               ['source']['followed_by'],
175                               'favorites': [],
176                               'retweeted': [],
177                               'replies': [],
178                               'matched_likes': [],
179                               'matched_rts': None}
180
181             for tweet in liked[source_id]:
182                 if target_id == tweet['user']['id']:
183                     interaction_dict['favorites'].append(tweet['id'])
184
185                 if tweet['id'] in liked[target_id]:
186                     interaction_dict['matched_likes'].append(tweet['id'])
187
188             for tweet in timeline[source_id]:
189                 if target_id == tweet['in_reply_to_user_id']:
190                     interaction_dict['replies'].append((tweet['id'], tweet['
in_reply_to_status_id']))
191
192                 if tweet.get('retweeted_status', False) \
and target_id == tweet['retweeted_status']['user']['id']:
193                     interaction_dict['retweeted'].append((tweet['id'], tweet['
retweeted_status']['id']))
194
195             interaction_list.append(interaction_dict)
196     return interaction_list
197
198 def report_interactions(test_names, test_ids):
199     relations = get_relationship_interactions(test_ids)
200     timemark = str(datetime.now())
201     backup = open('twitter_out/net_interactions_' + timemark + '.data', 'w')
202     print(relations, file=backup)
203     backup.close()
204
205     # SOMETHING IS WRONG FROM HERE ON
206     report = open('twitter_out/net_report_' + timemark + '.data', 'w')
207     i = 0
208     for source_user, source_id in zip(test_names, test_ids):
209         i += 1
210         for target_user, target_id in zip(test_names[i:], test_ids[i:]):
211             print('Interactions between {} and {}'.format(source_user, target_user
), file=report)
212             for relation in relations:
213                 if relation['source_user_id'] == source_id:
214                     src_to_trgt = relation
215                 elif relation['source_user_id'] == target_id:
216                     trgt_to_src = relation
217             print('
{:15} follows {:15} : {}'.format(
218                 source_user, target_user, src_to_trgt['source_follows_target']),

```

```

file=report)
218 file=report)
219     print('    {:15} follows {:15} : {}'.format(
220           target_user, source_user, trgt_to_src['source_follows_target']),
file=report)
221
222     print('    {:15} liked {:15} : {} times'.format(
223           source_user, target_user, len(src_to_trgt['favorites'])), file=
report)
224     print('    {:15} liked {:15} : {} times'.format(
225           target_user, source_user, len(trgt_to_src['favorites'])), file=
report)
226
227     print('    {:15} retweeted {:15}: {} times'.format(
228           source_user, target_user, len(src_to_trgt['retweeted'])), file=
report)
229     print('    {:15} retweeted {:15}: {} times'.format(
230           target_user, source_user, len(trgt_to_src['retweeted'])), file=
report)
231
232     print('    {:15} replied {:15} : {} times'.format(
233           source_user, target_user, len(src_to_trgt['replies'])), file=report)
234     print('    {:15} replied {:15} : {} times'.format(
235           target_user, source_user, len(trgt_to_src['replies'])), file=report)
236
237     print('    liked the same tweet: {} times'.format(
238           len(src_to_trgt['matched_likes'])), file=report)
239     print('    retweeted the same tweet: {} times'.format(
240           '_not_supported_yet_'), file=report)
241     print('\n', file=report)
242
243     report.close()
244
245
246 def download_timeline_and_likes(ids):
247     """
248     Downloads the timeline and likes of the users in
249     list ids.
250     """
251     max_attempt_number = 4
252     timemark = str(datetime.now())
253     os.mkdir('twitter_raw/' + timemark)
254
255     for user_id in ids:
256         attempt = 0
257         while attempt < max_attempt_number:
258             try:
259                 # get_favorites limited to 75 req./15 min, 200 most recent
260                 liked = twitter.get_favorites(id=user_id, count=199)
261                 # get_favorites limited to 900 req./15 min, 200 most recent
262                 timeline = twitter.get_user_timeline(id=user_id, count=199)
263
264                 file = open('twitter_raw/' + timemark + '/' + str(user_id) + '_liked
', 'w')
265                 file.write(repr(liked))
266                 file.close()
267                 file = open('twitter_raw/' + timemark + '/' + str(user_id) + '
_timeline', 'w')
268                 file.write(repr(timeline))
269                 file.close()
270             except Exception as exc:
271                 attempt += 1
272                 assert attempt < max_attempt_number, 'attempt time {} exceeded'.
format(max_attempt_number)
273                 if '(Unauthorized)' in str(exc):
274                     print('exception: ', exc)
275                     print('ignoring: ', user_id)
276                     break
277                 else:
278                     print('failed attempt at: ', datetime.now())
279                     print('exception: ', exc)
280                     time.sleep(6 * 60)
281                     print('reattempting at: ', datetime.now())
282             else:
283                 break
284
285

```

```
File: /usr/local/lib/python3.8/dist-packages/urllib3/util.py
286 test_names = *** # Python list of length 10 which contains the Twitter user names,
    as strings, of each member in the test group
287 test_ids = *** # Python list of length 10 which contains the Twitter user IDs, as
    integers, of each member in the test group
288
289 if __name__ == '__main__':
290     report_interactions(test_names, test_ids)
```