ntities-from-social-media-posts-1

April 1, 2024

We start by preparing the environment by importing the relevant libraries, and setting some options:

[ ]:
```
%config InlineBackend.figure_format = 'retina' # high resolution plotting
import matplotlib.pyplot as plt
import pandas as pd
import advertools as adv
pd.set_option('display.max_columns', None)
pd.set_option('display.max_colwidth', 280)
adv.__version__
```

[116]:
```
tweets_users_df = pd.read_csv('../input/justdoit_tweets_2018_09_07_2.csv', )
print(tweets_users_df.shape)
tweets_users_df.head(3)
```

```
(5089, 72)
```

[116]:

	tweet_contributors	tweet_coordinates	tweet_created_at	tweet_display_text_range	tweet_entities
0	NaN	NaN	Fri Sep 07 16:25:06 +0000 2018	[0, 75]	{'hashtags': [{'text': 'quote', 'indices': [47, 53]}, {'text': 'motivation', 'indices': [54, 65]}, {'text': 'justdoit', 'indices': [66, 75]}], 'symbols': [], 'user_mentions': [], 'urls': [], 'media': [{'id': 1038100853872197632, 'id_str': '1038100853872197632', 'indices': [76…
1	NaN	NaN	Fri Sep 07 16:24:59 +0000 2018	[0, 237]	{'hashtags': [{'text': 'hero', 'indices': [90, 95]}, {'text': 'fdny', 'indices': [96, 101]}, {'text': 'likesforlikes', 'indices': [102, 116]}, {'text': 'promo', 'indices': [117, 123]}, {'text': 'music', 'indices': [124, 130]}, {'text': 'instagood', 'indices': [131, 141]}, {'t…
2	NaN	NaN	Fri Sep 07 16:24:50 +0000 2018	[0, 176]	{'hashtags': [{'text': 'JustDoIt', 'indices': [127, 136]}, {'text': '4YourMorning', 'indices': [137, 150]}, {'text': '4YourMemeCollection', 'indices': [151, 171]}], 'symbols': [], 'user_mentions': [], 'urls': [],

```
'media': [{'id': 1038100773396041728, 'id_str': '10381007733960…

                  tweet_extended_entities  \
0  {'media': [{'id': 1038100853872197632, 'id_str': '1038100853872197632',
'indices': [76, 99], 'media_url':
'http://pbs.twimg.com/media/DmgTOfwVAAAJqoh.jpg', 'media_url_https':
'https://pbs.twimg.com/media/DmgTOfwVAAAJqoh.jpg', 'url':
'https://t.co/J9lLdszdW6', 'display_url': '…
1
NaN
2  {'media': [{'id': 1038100773396041728, 'id_str': '1038100773396041728',
'indices': [177, 200], 'media_url':
'http://pbs.twimg.com/media/DmgTJz9UUAA57tu.jpg', 'media_url_https':
'https://pbs.twimg.com/media/DmgTJz9UUAA57tu.jpg', 'url':
'https://t.co/6ok9qR6k6M', 'display_url':…

   tweet_favorite_count  tweet_favorited  \
0                     0           False
1                     0           False
2                     0           False

tweet_full_text  \
0
Done is better than perfect. - Sheryl Sandberg #quote #motivation #justdoit
https://t.co/J9lLdszdW6
1  Shout out to the Great Fire Department and the tour!    Much love to NYC!
   \n•\n•\n•\n#hero #fdny #likesforlikes #promo #music #instagood #instadaily
#postoftheday #bestoftheday #justdoit #nike #picoftheday…
https://t.co/sFobQ2ukpo
2                                      There are some AMAZINGLY hilarious Nike
Ad memes happening on my newsfeed.  Soooo, I decided to get a little creative
too… \n\n#JustDoIt #4YourMorning #4YourMemeCollection \n\n
https://t.co/6ok9qR6k6M

  tweet_geo             tweet_id         tweet_id_str  \
0       NaN  1038100857932394496  1038100857932394496
1       NaN  1038100830807904256  1038100830807904256
2       NaN  1038100793147248640  1038100793147248640

  tweet_in_reply_to_screen_name  tweet_in_reply_to_status_id  \
0                           NaN                          NaN
1                           NaN                          NaN
2                           NaN                          NaN

   tweet_in_reply_to_status_id_str  tweet_in_reply_to_user_id  \
0                              NaN                        NaN
1                              NaN                        NaN
```

```
2                              NaN                          NaN

   tweet_in_reply_to_user_id_str  tweet_is_quote_status tweet_lang  \
0                            NaN                 False         en
1                            NaN                 False         en
2                            NaN                 False         en

                                       tweet_metadata tweet_place  \
0  {'iso_language_code': 'en', 'result_type': 'recent'}         NaN
1  {'iso_language_code': 'en', 'result_type': 'recent'}         NaN
2  {'iso_language_code': 'en', 'result_type': 'recent'}         NaN

  tweet_possibly_sensitive tweet_quoted_status  tweet_quoted_status_id  \
0                    False                NaN                     NaN
1                    False                NaN                     NaN
2                    False                NaN                     NaN

   tweet_quoted_status_id_str  tweet_retweet_count  tweet_retweeted  \
0                         NaN                    0            False
1                         NaN                    0            False
2                         NaN                    0            False

tweet_source  \
0                      <a href="https://statusbrew.com"
rel="nofollow">Statusbrew</a>
1               <a href="http://www.facebook.com/twitter"
rel="nofollow">Facebook</a>
2  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for
iPhone</a>

   tweet_truncated  \
0            False
1            False
2            False

                                tweet_user  \
0  {'id': 3188618684, 'id_str': '3188618684', 'name': 'Ultra YOU Woman',
'screen_name': 'UltraYOUwoman', 'location': 'California, USA', 'description': 'I
share tips to achieve your health goals and be your best self inside & out! Plus
healthy living, weight loss success stories,…
1  {'id': 18387174, 'id_str': '18387174', 'name': 'Yung Cut Up (Videos)',
'screen_name': 'yungcutup', 'location': 'Miami, Florida', 'description': 'All
Business inquiries contact cluuxx@gmail.com / Support & Download my new mixtape
"Clear Skies" https://t.co/0tOeBuJHHH', 'url': …
2  {'id': 32645612, 'id_str': '32645612', 'name': 'Rachel Bogle', 'screen_name':
'rachelbogle', 'location': 'Indianapolis, IN', 'description': 'Morning Traffic
Reporter @CBS4Indy | Traffic Authority | Radio  to TV  | Indiana Raised |
```

```
@IUBloomington Alum | Morkie Mom to Gizmo |…

   user_contributors_enabled                user_created_at  \
0                      False  Fri May 08 10:27:51 +0000 2015
1                      False  Fri Dec 26 09:30:23 +0000 2008
2                      False  Fri Apr 17 23:04:15 +0000 2009

   user_default_profile  user_default_profile_image  \
0                  True                      False
1                 False                      False
2                 False                      False

user_description  \
0  I share tips to achieve your health goals and be your best self inside & out!
Plus healthy living, weight loss success stories, skincare & post-birth snap
back!
1                                        All Business inquiries contact
cluuxx@gmail.com / Support & Download my new mixtape "Clear Skies"
https://t.co/0tOeBuJHHH
2    Morning Traffic Reporter @CBS4Indy | Traffic Authority | Radio   to TV   |
Indiana Raised | @IUBloomington Alum | Morkie Mom to Gizmo | Ms. USA Universal
2018

                            user_entities  \
0
{'url': {'urls': [{'url': 'https://t.co/jGlJswxjwS', 'expanded_url':
'https://about.me/ultrayouwoman', 'display_url': 'about.me/ultrayouwoman',
'indices': [0, 23]}]}, 'description': {'urls': []}}
1  {'url': {'urls': [{'url': 'http://t.co/lVm8vfDbfO', 'expanded_url':
'http://youtube.com/yungcutuptv', 'display_url': 'youtube.com/yungcutuptv',
'indices': [0, 22]}]}, 'description': {'urls': [{'url':
'https://t.co/0tOeBuJHHH', 'expanded_url': 'http://piff.me/6613310', 'displa…
2
{'url': {'urls': [{'url': 'https://t.co/g9exqgZp9x', 'expanded_url':
'http://www.cbs4indy.com', 'display_url': 'cbs4indy.com', 'indices': [0, 23]}]},
'description': {'urls': []}}

   user_favourites_count user_follow_request_sent  user_followers_count  \
0                  307.0                    False               57983.0
1                 1178.0                    False               13241.0
2                11864.0                    False               11377.0

  user_following  user_friends_count user_geo_enabled  \
0          False             48721.0            False
1          False              5489.0            False
2          False              2386.0            False
```

```
  user_has_extended_profile       user_id   user_id_str  \
0                     False  3.188619e+09  3.188619e+09
1                     False  1.838717e+07  1.838717e+07
2                     False  3.264561e+07  3.264561e+07

  user_is_translation_enabled user_is_translator user_lang  user_listed_count  \
0                       False              False        en              629.0
1                       False              False        en              150.0
2                       False              False        en              193.0

      user_location             user_name user_notifications  \
0   California, USA       Ultra YOU Woman              False
1    Miami, Florida  Yung Cut Up (Videos)              False
2  Indianapolis, IN          Rachel Bogle              False

  user_profile_background_color  \
0                        C0DEED
1                        131516
2                        FFFAFF

                  user_profile_background_image_url  \
0   http://abs.twimg.com/images/themes/theme1/bg.png
1  http://abs.twimg.com/images/themes/theme14/bg.gif
2   http://abs.twimg.com/images/themes/theme1/bg.png

             user_profile_background_image_url_https  \
0   https://abs.twimg.com/images/themes/theme1/bg.png
1  https://abs.twimg.com/images/themes/theme14/bg.gif
2   https://abs.twimg.com/images/themes/theme1/bg.png

  user_profile_background_tile  \
0                        False
1                         True
2                        False

                                     user_profile_banner_url  \
0  https://pbs.twimg.com/profile_banners/3188618684/1431170427
1    https://pbs.twimg.com/profile_banners/18387174/1488819752
2    https://pbs.twimg.com/profile_banners/32645612/1485823278

                                                  user_profile_image_url  \
0  http://pbs.twimg.com/profile_images/597000926272954368/eQ-8VrVk_normal.jpg
1  http://pbs.twimg.com/profile_images/945333114582298625/C8zA_uvh_normal.jpg
2  http://pbs.twimg.com/profile_images/986345956357615619/4zpa5kxF_normal.jpg

                                                 user_profile_image_url_https
\
```

```
0  https://pbs.twimg.com/profile_images/597000926272954368/eQ-8VrVk_normal.jpg
1  https://pbs.twimg.com/profile_images/945333114582298625/C8zA_uvh_normal.jpg
2  https://pbs.twimg.com/profile_images/986345956357615619/4zpa5kxF_normal.jpg

  user_profile_link_color user_profile_sidebar_border_color  \
0                  1DA1F2                            C0DEED
1                  3B94D9                            FFFFFF
2                  050505                            FFFFFF

  user_profile_sidebar_fill_color user_profile_text_color  \
0                          DDEEF6                  333333
1                          EFEFEF                  333333
2                          FC6A71                  050505

  user_profile_use_background_image user_protected user_screen_name  \
0                              True          False    UltraYOUwoman
1                              True          False        yungcutup
2                              True          False      rachelbogle

   user_statuses_count  user_time_zone user_translator_type  \
0              91870.0             NaN                 none
1             618822.0             NaN                 none
2              48075.0             NaN                 none

                 user_url  user_utc_offset user_verified
0  https://t.co/jGlJswxjwS              NaN         False
1   http://t.co/lVm8vfDbfO              NaN         False
2  https://t.co/g9exqgZp9x              NaN          True
```

[117]:
```
[x for x in dir(adv) if x.startswith('extract')]  # currently available extract␣
↪functions
```

[117]:
```
['extract',
 'extract_currency',
 'extract_emoji',
 'extract_hashtags',
 'extract_intense_words',
 'extract_mentions',
 'extract_questions',
 'extract_words']
```

[118]:
```
hashtag_summary = adv.extract_hashtags(tweets_users_df['tweet_full_text'])
hashtag_summary.keys()
```

[118]:
```
dict_keys(['hashtags', 'hashtags_flat', 'hashtag_counts', 'hashtag_freq',
'top_hashtags', 'overview'])
```

The most general one to get a quick idea about the data is the `overview` key.
This shows us how many posts we have, the total number of hashtags (or mentions, or emoji), the average number of hashtags per post, and the number of unique hashtags.

[119]:
```
hashtag_summary['overview']
```

[119]:
```
{'num_posts': 5089,
 'num_hashtags': 15483,
 'hashtags_per_post': 3.0424444881116135,
 'unique_hashtags': 4630}
```

Next, we can explore the extracted hashtags themselves. Here we are looking at the first ten.
As you can see for each post we get a list of hashtags. We get an empty list wherever there are no hashtags in the tweet.

[120]:
```
hashtag_summary['hashtags'][:10]
```

[120]:
```
[['#quote', '#motivation', '#justdoit'],
 ['#hero',
  '#fdny',
  '#likesforlikes',
  '#promo',
  '#music',
  '#instagood',
  '#instadaily',
  '#postoftheday',
  '#bestoftheday',
  '#justdoit',
  '#nike',
  '#picoftheday'],
 ['#justdoit', '#4yourmorning', '#4yourmemecollection'],
 ['#kapernickeffect', '#swoosh', '#justdoit'],
 ['#shaquem',
  '#nfl',
  '#seattle',
  '#seahawks',
  '#griffin',
  '#justdoit',
  '#nike'],
 ['#justdoit'],
 ['#registertovote', '#justdoit'],
 ['#justdoit'],
 ['#justdoit', '#takeaknee', '#takeakneeinnikes'],
 ['#fx',
  '#feelgoodfriday',
  '#fridayfeeling',
  '#tradermoni',
  '#justdoit',
```

```
 '#fridaymotivation']]
```

[121]:
```
hashtag_summary['hashtags_flat'][:10]
```

[121]:
```
['#quote',
 '#motivation',
 '#justdoit',
 '#hero',
 '#fdny',
 '#likesforlikes',
 '#promo',
 '#music',
 '#instagood',
 '#instadaily']
```

The count of hashtags for each tweet is given by the `hashtag_counts` key.
Later, we will combine all these in one DataFrame and do further analysis on them.

[122]:
```
hashtag_summary['hashtag_counts'][:20]
```

[122]:
```
[3, 12, 3, 3, 7, 1, 2, 1, 3, 6, 1, 2, 1, 1, 2, 1, 7, 1, 3, 2]
```

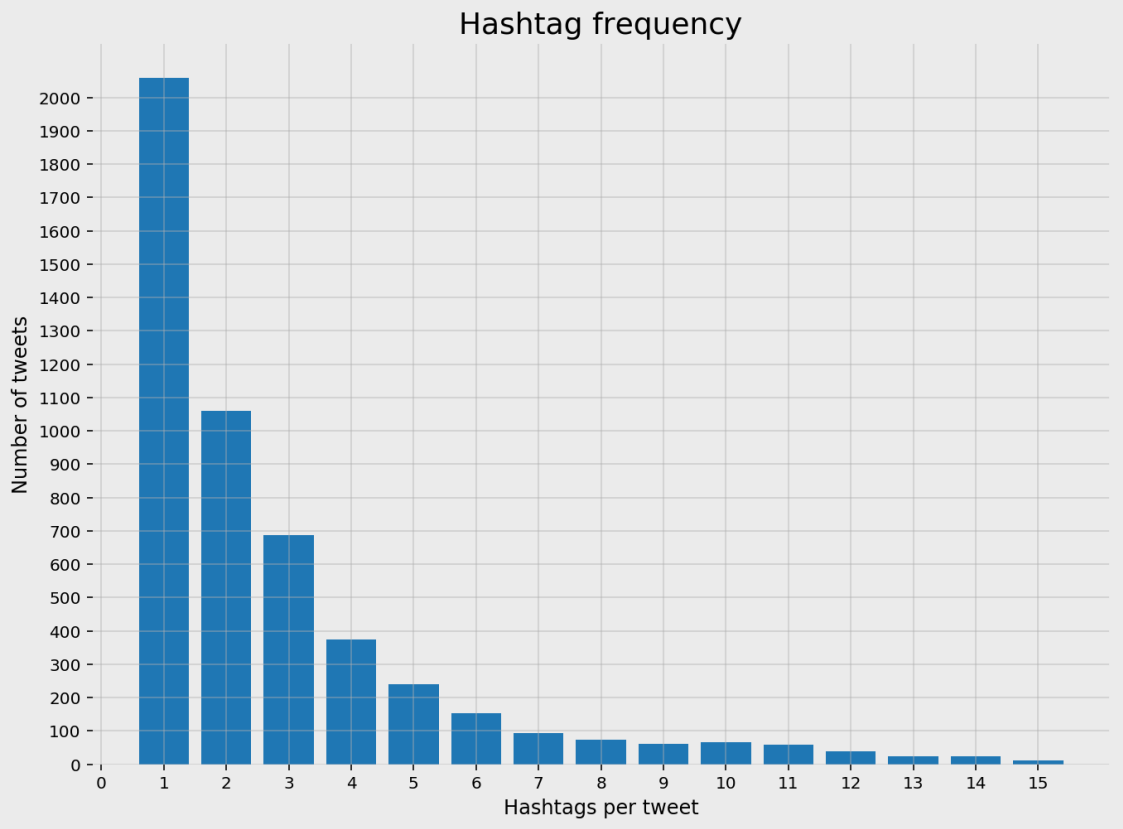
[123]:
```
hashtag_summary['hashtag_freq'][:15]
```

[123]:
```
[(1, 2058),
 (2, 1061),
 (3, 686),
 (4, 374),
 (5, 239),
 (6, 154),
 (7, 94),
 (8, 74),
 (9, 60),
 (10, 65),
 (11, 58),
 (12, 39),
 (13, 25),
 (14, 24),
 (15, 11)]
```

Visualizing the frequencies to get a better overview of how they are distributed, we plot the top fifteen frequencies:

[124]:
```
plt.figure(facecolor='#ebebeb', figsize=(11, 8))
plt.bar([x[0] for x in hashtag_summary['hashtag_freq'][:15]],
        [x[1] for x in hashtag_summary['hashtag_freq'][:15]])
plt.title('Hashtag frequency', fontsize=18)
plt.xlabel('Hashtags per tweet', fontsize=12)
```

```
plt.ylabel('Number of tweets', fontsize=12)
plt.xticks(range(16))
plt.yticks(range(0, 2100, 100))
plt.grid(alpha=0.5)
plt.gca().set_frame_on(False)
```



You are probably wonderig which are the top hashtags, and how popular they are.
This is provided by the `top_hashtags` key.
As mentioned above, it shouldn't be a surprise that #justdoit is the top one, and that #nike and #colinkaepernick are in the top positions as well.

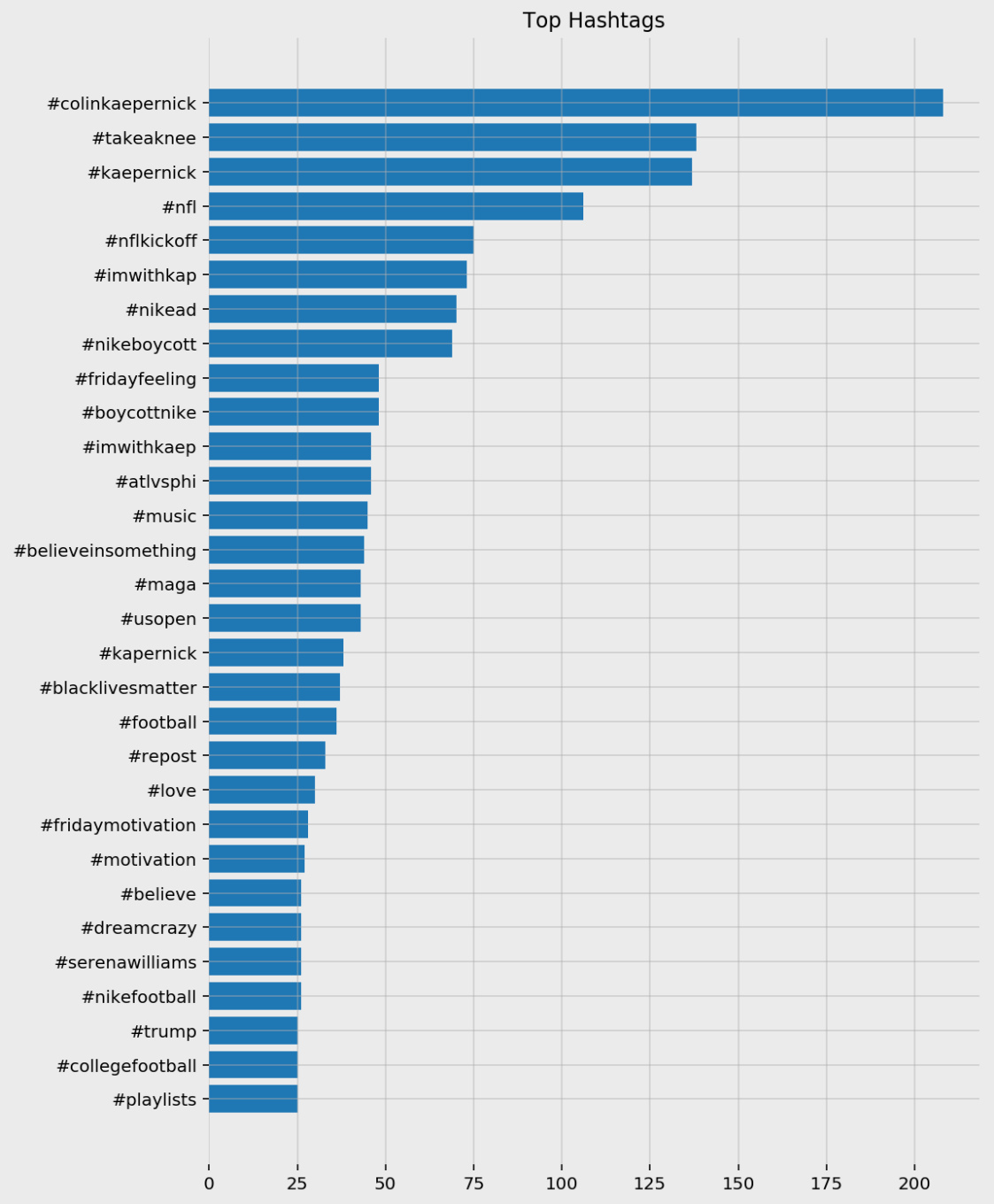
[125]:
```
hashtag_summary['top_hashtags'][:10]
```

[125]:
```
[('#justdoit', 5150),
 ('#nike', 1112),
 ('#colinkaepernick', 208),
 ('#takeaknee', 138),
 ('#kaepernick', 137),
 ('#nfl', 106),
 ('#nflkickoff', 75),
 ('#imwithkap', 73),
```

```
 ('#nikead', 70),
 ('#nikeboycott', 69)]
```

Visualizing the same data (excluding #justdoit and #nike):

[126]:
```
plt.figure(facecolor='#ebebeb', figsize=(8, 12))
plt.barh([x[0] for x in hashtag_summary['top_hashtags'][2:][:30]][::-1],
         [x[1] for x in hashtag_summary['top_hashtags'][2:][:30]][::-1])
plt.title('Top Hashtags')
plt.grid(alpha=0.5)
plt.gca().set_frame_on(False)
```



Emoji

You will see that the `extract_emoji` function is pretty much the same as `extract_hashtags`. The only difference is that it has emoji both as images and their textual counterparts.

[127]:
```
emoji_summary = adv.extract_emoji(tweets_users_df['tweet_full_text'])
emoji_summary.keys()
```

[127]:
```
dict_keys(['emoji', 'emoji_text', 'emoji_flat', 'emoji_flat_text',
'emoji_counts', 'emoji_freq', 'top_emoji', 'top_emoji_text', 'overview'])
```

[128]:
```
emoji_summary['overview']
```

[128]:
```
{'num_posts': 5089,
 'num_emoji': 3205,
 'emoji_per_post': 0.6297897425820397,
 'unique_emoji': 407}
```

[129]:
```
emoji_summary['emoji'][:20]
```

[129]:
```
[[],
 [' \u200d ', ' \u200d ', ' ', ' ', ' ', ' '],
 [' ', ' '],
 [],
 [],
 [],
 [],
 [],
 [],
 [],
 [],
 [],
 [],
 [],
 [],
 [],
 [],
 [' ', ' ', ' '],
 [],
 [],
 []]
```

[130]:
```
emoji_summary['emoji_text'][:20]
```

[130]:
```
[[],
 ['man firefighter',
  'man firefighter',
  'hundred points',
  'movie camera',
  'fire',
  'raised fist dark skin tone'],
 ['cookie', 'face with tears of joy'],
 [],
 [],
 [],
 [],
```

```
 [],
 [],
 [],
 [],
 [],
 [],
 [],
 [],
 [],
 ['thinking face', 'face with rolling eyes', 'flushed face'],
 [],
 [],
 []]
```

[131]:
```
emoji_summary['emoji_flat'][:10]
```

[131]:
```
[' \u200d ', ' \u200d ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ']
```

[132]:
```
emoji_summary['emoji_flat_text'][:10]
```

[132]:
```
['man firefighter',
 'man firefighter',
 'hundred points',
 'movie camera',
 'fire',
 'raised fist dark skin tone',
 'cookie',
 'face with tears of joy',
 'thinking face',
 'face with rolling eyes']
```

Putting them side by side to get a better idea, and taking a look at the first ten:

[133]:
```
list(zip(emoji_summary['emoji_flat'][:10], emoji_summary['emoji_flat_text'][:
 ↪10]))
```

[133]:
```
[(' \u200d ', 'man firefighter'),
 (' \u200d ', 'man firefighter'),
 (' ', 'hundred points'),
 (' ', 'movie camera'),
 (' ', 'fire'),
 (' ', 'raised fist dark skin tone'),
 (' ', 'cookie'),
 (' ', 'face with tears of joy'),
 (' ', 'thinking face'),
 (' ', 'face with rolling eyes')]
```

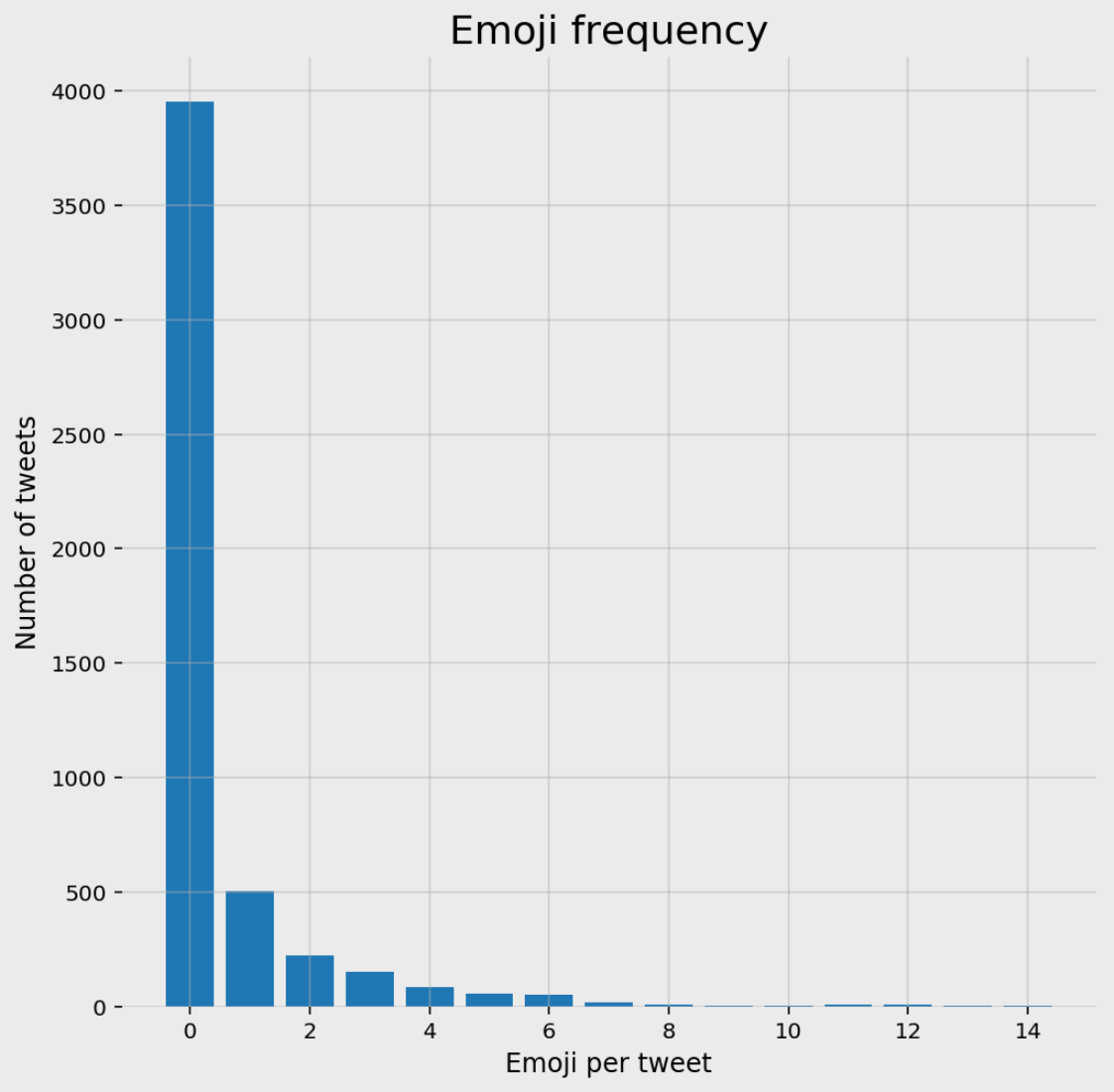
[134]:
```
emoji_summary['emoji_counts'][:15]
```

[134]:
```
[0, 6, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

[135]:
```
emoji_summary['emoji_freq'][:15]
```

[135]:
```
[(0, 3952),
 (1, 502),
 (2, 223),
 (3, 153),
 (4, 82),
 (5, 54),
 (6, 52),
 (7, 18),
 (8, 8),
 (9, 5),
 (10, 4),
 (11, 7),
 (12, 7),
 (13, 4),
 (14, 3)]
```

[136]:
```
plt.figure(facecolor='#ebebeb', figsize=(8, 8))
plt.bar([x[0] for x in emoji_summary['emoji_freq'][:15]],
        [x[1] for x in emoji_summary['emoji_freq'][:15]])
plt.title('Emoji frequency', fontsize=18)
plt.xlabel('Emoji per tweet', fontsize=12)
plt.ylabel('Number of tweets', fontsize=12)
plt.grid(alpha=0.5)
plt.gca().set_frame_on(False)
```



[137]:
```
emoji_summary['top_emoji'][:20]
```

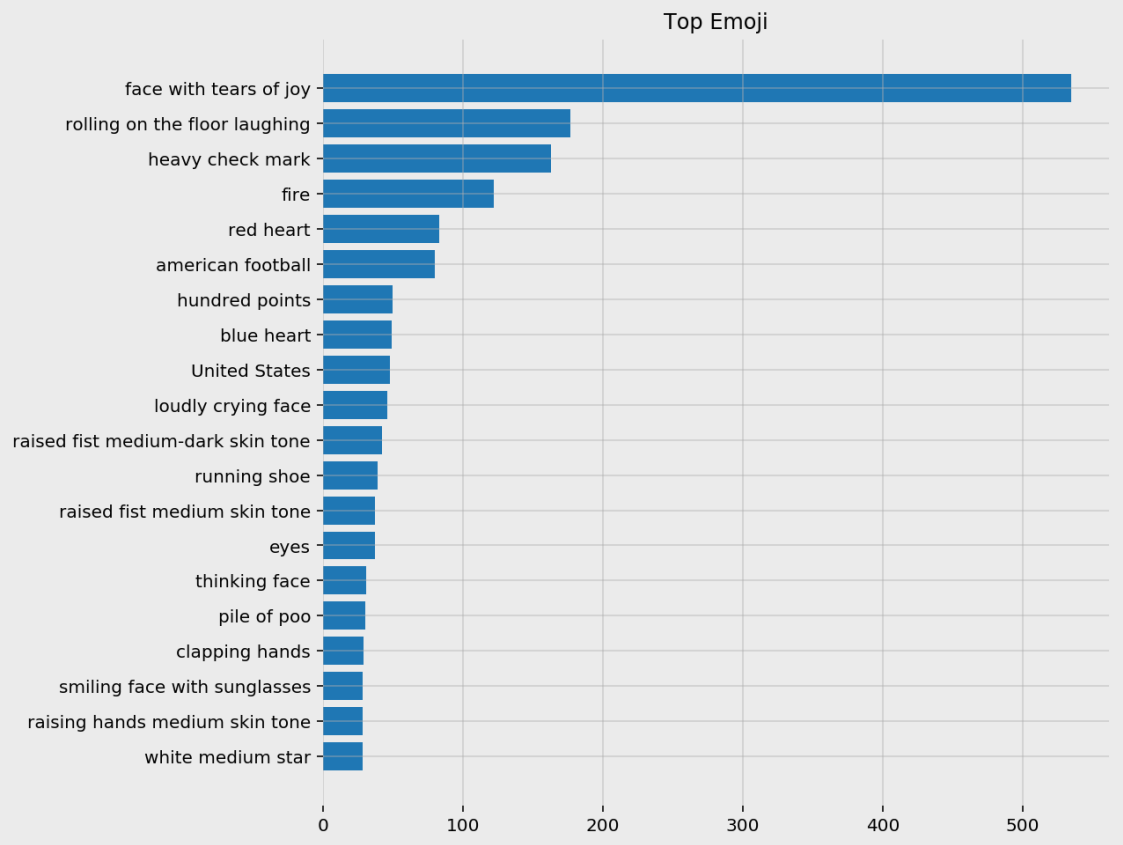
[137]:
```
[(' ', 535),
 (' ', 177),
 (' ', 163),
 (' ', 122),
 (' ', 83),
 (' ', 80),
 (' ', 50),
 (' ', 49),
 (' ', 48),
 (' ', 46),
 (' ', 42),
```

```
 (' ', 39),
 (' ', 37),
 (' ', 37),
 (' ', 31),
 (' ', 30),
 (' ', 29),
 (' ', 28),
 (' ', 28),
 (' ', 28)]
```

[138]:
```
emoji_summary['top_emoji_text'][:20]
```

[138]:
```
[('face with tears of joy', 535),
 ('rolling on the floor laughing', 177),
 ('heavy check mark', 163),
 ('fire', 122),
 ('red heart', 83),
 ('american football', 80),
 ('hundred points', 50),
 ('blue heart', 49),
 ('United States', 48),
 ('loudly crying face', 46),
 ('raised fist medium-dark skin tone', 42),
 ('running shoe', 39),
 ('raised fist medium skin tone', 37),
 ('eyes', 37),
 ('thinking face', 31),
 ('pile of poo', 30),
 ('clapping hands', 29),
 ('smiling face with sunglasses', 28),
 ('raising hands medium skin tone', 28),
 ('white medium star', 28)]
```

[139]:
```
plt.figure(facecolor='#ebebeb', figsize=(8, 8))
plt.barh([x[0] for x in emoji_summary['top_emoji_text'][:20]][::-1],
         [x[1] for x in emoji_summary['top_emoji_text'][:20]][::-1])
plt.title('Top Emoji')
plt.grid(alpha=0.5)
plt.gca().set_frame_on(False)
```



[140]:
```
mention_summary = adv.extract_mentions(tweets_users_df['tweet_full_text'])
mention_summary.keys()
```

[140]:
```
dict_keys(['mentions', 'mentions_flat', 'mention_counts', 'mention_freq',
'top_mentions', 'overview'])
```

[141]:
```
mention_summary['overview']
```

[141]:
```
{'num_posts': 5089,
 'num_mentions': 4863,
 'mentions_per_post': 0.9555904892906268,
 'unique_mentions': 1624}
```

[142]:
```
mention_summary['mentions'][:15]
```

[142]:
```
[[],
 [],
 [],
 [],
 [],
 ['@realdonaldtrump'],
```

```
 [],
 ['@nike'],
 ['@nike', '@nikestore', '@kaepernick7'],
 [],
 [],
 ['@nike'],
 ['@cspensions'],
 ['@realdonaldtrump', '@colinkaperneck7'],
 ['@repadamschiff', '@repadamschiff']]
```

[143]:
```
mention_summary['mentions_flat'][:10]
```

[143]:
```
['@realdonaldtrump',
 '@nike',
 '@nike',
 '@nikestore',
 '@kaepernick7',
 '@nike',
 '@cspensions',
 '@realdonaldtrump',
 '@colinkaperneck7',
 '@repadamschiff']
```

[144]:
```
mention_summary['mention_counts'][:20]
```

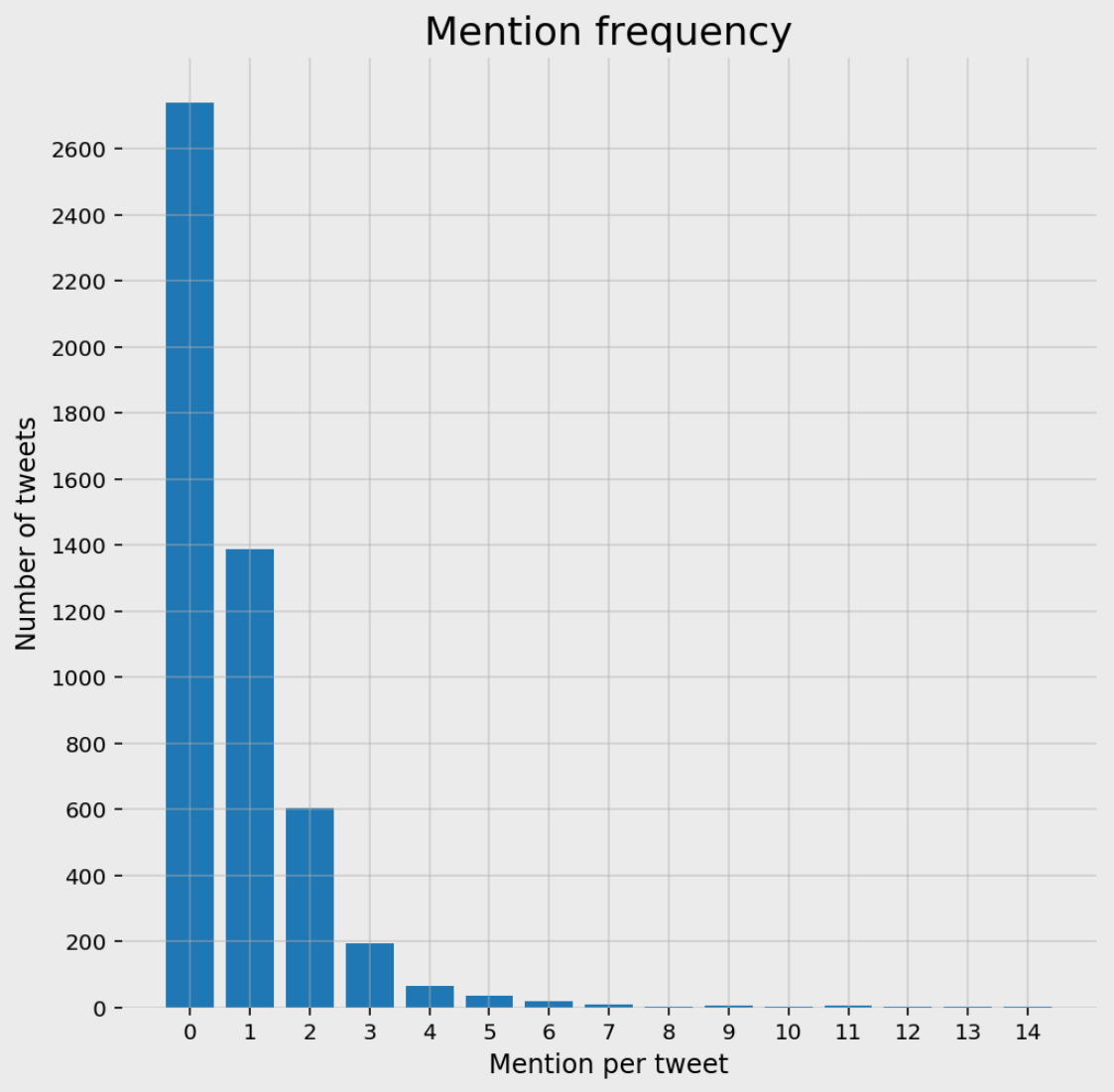
[144]:
```
[0, 0, 0, 0, 0, 1, 0, 1, 3, 0, 0, 1, 1, 2, 2, 0, 0, 1, 2, 0]
```

[145]:
```
mention_summary['mention_freq'][:15]
```

[145]:
```
[(0, 2738),
 (1, 1386),
 (2, 602),
 (3, 193),
 (4, 66),
 (5, 34),
 (6, 18),
 (7, 8),
 (8, 3),
 (9, 4),
 (10, 1),
 (11, 4),
 (12, 2),
 (13, 1),
 (14, 2)]
```

[146]:
```
plt.figure(facecolor='#ebebeb', figsize=(8, 8))
plt.bar([x[0] for x in mention_summary['mention_freq'][:15]],
```

```
        [x[1] for x in mention_summary['mention_freq'][:15]])
plt.title('Mention frequency', fontsize=18)
plt.xlabel('Mention per tweet', fontsize=12)
plt.ylabel('Number of tweets', fontsize=12)
plt.xticks(range(15))
plt.yticks(range(0, 2800, 200))
plt.grid(alpha=0.5)
plt.gca().set_frame_on(False)
```

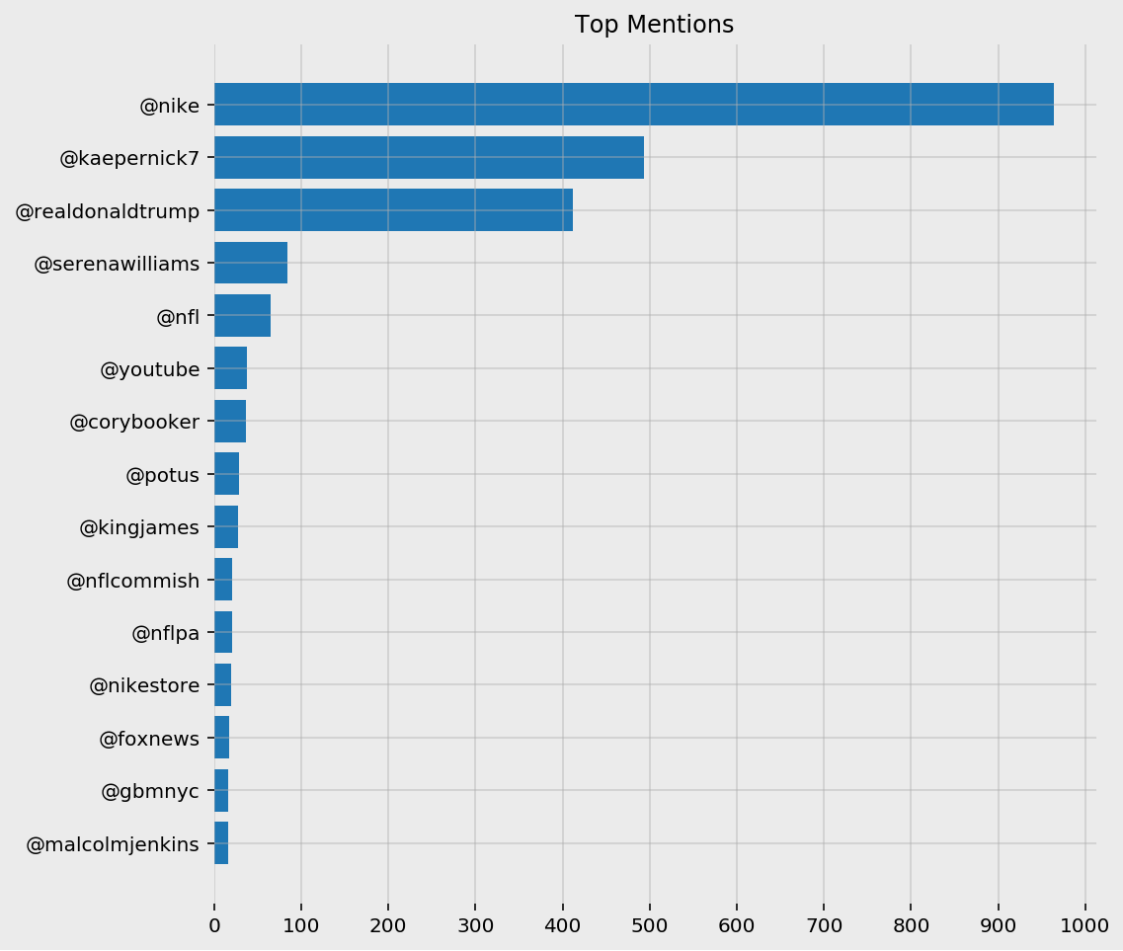


[147]:
```
mention_summary['top_mentions'][:10]
```

[147]:
```
[('@nike', 964),
 ('@kaepernick7', 493),
```

```
 ('@realdonaldtrump', 412),
 ('@serenawilliams', 84),
 ('@nfl', 65),
 ('@youtube', 38),
 ('@corybooker', 36),
 ('@potus', 29),
 ('@kingjames', 27),
 ('@nflcommish', 21)]
```

[148]:
```
plt.figure(facecolor='#ebebeb', figsize=(8, 8))
plt.barh([x[0] for x in mention_summary['top_mentions'][:15]][::-1],
         [x[1] for x in mention_summary['top_mentions'][:15]][::-1])
plt.title('Top Mentions')
plt.grid(alpha=0.5)
plt.xticks(range(0, 1100, 100))
plt.gca().set_frame_on(False)
```



Questions

[149]:
```
question_summary = adv.extract_questions(tweets_users_df['tweet_full_text'])
```

[150]:
```
question_summary.keys()
```

[150]:
```
dict_keys(['question_marks', 'question_marks_flat', 'question_mark_counts',
'question_mark_freq', 'top_question_marks', 'overview', 'question_mark_names',
'question_text'])
```

[151]:
```
question_summary['overview']
```

[151]:
```
{'num_posts': 5089,
 'num_question_marks': 674,
 'question_marks_per_post': 0.13244252308901552,
 'unique_question_marks': 1}
```

13.2% of the tweets contained questions.

[152]:
```
question_summary['question_mark_freq']
```

[152]:
```
[(0, 4590), (1, 391), (2, 62), (3, 34), (4, 7), (5, 3), (6, 1), (8, 1)]
```

[153]:
```
question_summary['top_question_marks'] # this is more interesting if you have␣
↪questions in different languages where different question marks are used.
```

[153]:
```
[('?', 674)]
```

Here are of some questions that were asked.

[154]:
```
[(i,x) for i, x in  enumerate(question_summary['question_text']) if x][:15]
```

[154]:
```
[(6, ["Why won't Trump protect our elections?"]),
 (17, ['You want to impress me?']),
 (30,
  ['But what exactly does sacrifice mean to a multi-billion dollar
corporation?']),
 (45, ['Do I have your back?']),
 (50, ['Invest in #Mojo50?']),
 (58, ['Were you surprised?']),
 (81, ['Owned Yet, Libs?']),
 (87,
  ["Slave owners and private prisons have used black men and women as a
political and financial piece for gain\n\nBut people are mad that nike is
putting money in Kaepernick's pockets giving a world platform for a message of
racial injustice?"]),
 (105, ['Can commerce and activism coexist?']),
 (106, ['See the difference?']),
 (107, ['School shooting?']),
 (125,
```

```
  ['@Kaepernick7 I salute #justdoit and to those human beings burning their
#bikes that they already paid for - why couldn't they all just donate those
shoes to those that don't even know what a shoe is?']),
 (126, ['Her solution?']),
 (131,
  ['200 smooooooooth writing/drawing pages are calling your name - have you
ordered your custom, handcrafted journal?']),
 (137, ['BTW- have you watched the movie: #AllThePresidentsMen lately?'])]
```

Intense Words

[155]:
```
intense_summary = adv.extract_intense_words(tweets_users_df['tweet_full_text'],␣
↪min_reps=3)
```

[156]:
```
intense_summary['overview']
```

[156]:
```
{'num_posts': 5089,
 'num_intense_words': 1274,
 'intense_words_per_post': 0.25034387895460797,
 'unique_intense_words': 979}
```

It seems a quarter of tweets have people intensely expressing their feelings.

[157]:
```
intense_summary['top_intense_words'][:20]
```

[157]:
```
[('…', 41),
 ('  ', 27),
 ('  ', 20),
 ('!!!', 15),
 ('@MatthewWolfff', 13),
 ('   ', 12),
 ('!!!!', 10),
 ('  ', 9),
 ('  ', 9),
 ('it…', 9),
 ('…', 7),
 ('Nike!!!', 5),
 ('it!!!', 5),
 ('    ', 5),
 ('@PaylessInsider!!!', 5),
 ('@Baby___Del', 4),
 ('@Mongo444444', 4),
 ('@Briteeye777', 4),
 ('   ', 4),
 ('crazy…', 4)]
```

Currency Symbols

[158]:
```
currency_summary = adv.extract_currency(tweets_users_df['tweet_full_text'])
```

[159]:
```
currency_summary.keys()
```

[159]:
```
dict_keys(['currency_symbols', 'currency_symbols_flat',
'currency_symbol_counts', 'currency_symbol_freq', 'top_currency_symbols',
'overview', 'currency_symbol_names', 'surrounding_text'])
```

[160]:
```
currency_summary['overview']
```

[160]:
```
{'num_posts': 5089,
 'num_currency_symbols': 65,
 'currency_symbols_per_post': 0.012772646885439182,
 'unique_currency_symbols': 1}
```

It seems there isn't much talk about money, with 1.2% of the tweets containing currency symbols. Let's see what they are.

[161]:
```
currency_summary['top_currency_symbols']
```

[161]:
```
[('$', 65)]
```

[162]:
```
[x for x in currency_summary['surrounding_text'] if x][:20]
```

[162]:
```
[['Nike got $43 million of free p'],
 [' confirm Kavanaugh. $20.20 is a small pri'],
 ['ending any leftover $ after funding @Sena'],
 ['Tier 1: $50/year ', 'Tier 2: $100/year '],
 ['oor foreign workers $0.20 an hour #JustDo'],
 [' spend considerable $ are gen x conservat',
  'pending, #justdoit, $NKE lol.'],
 ["just Doing It'  for $ 0.23 an hour ",
  'he #NikeAd ? Was it $0.23 an hour? '],
 ['flip flops for only $25! #JustDoIt'],
 ['urrent #stock price $80 and some change. '],
 ['t black kids buying $200 Jordan's and you',
  'rofessionals buying $350 Nike Apple Watch',
  'some hick burning a $10 pair of socks tha'],
 ['e "Buzz" equated to $163.5 million in val'],
 ['livery service. Get $7 off your next orde'],
 ['m credited and some $ for their art?!!! h'],
 ['#JustDoIt @elonmusk $tsla'],
 ['ainability and save $26 trillion. Why are'],
 ['ille Jury Fines Man $1 for Punching White'],
 ['founder gave nearly $400,000 to Trump. @N'],
 ['w…and he just got $30 mil for this ad??'],
 ['on! I will spend my $ w/companies that ba'],
```

```
['Of Richard Nixon's A$$! ']]
```

[163]:
```
word_summary = adv.extract_words(tweets_users_df['tweet_full_text'],
                                 words_to_extract=['sport', 'football',␣
↪'athlet',],
                                 entire_words_only=False) # when set to False,␣
↪it extracts the words and show how they appear within a larger word if any
                                                          # if set to True, is␣
↪only extracts the exact words specified only if they appear as entire words
```

[164]:
```
word_summary.keys()
```

[164]:
```
dict_keys(['words', 'words_flat', 'word_counts', 'word_freq', 'top_words',
'overview'])
```

[165]:
```
word_summary['overview']
```

[165]:
```
{'num_posts': 5089,
 'num_words': 355,
 'words_per_post': 0.06975830222047553,
 'unique_words': 80}
```

Almost 7% of the tweets contained any of the words that we specified. This indicates that this was not a very sports-oriented discussion.
Below are the top words.

[166]:
```
word_summary['top_words'][:20]
```

[166]:
```
[('football', 39),
 ('#football', 36),
 ('#nikefootball', 25),
 ('#collegefootball', 25),
 ('#highschoolfootball', 25),
 ('#adidasfootball', 25),
 ('athletes', 17),
 ('#sports', 15),
 ('sports', 13),
 ('athlete', 13),
 ('#thursdaynightfootball', 12),
 ('#sport', 7),
 ('sporting', 7),
 ('athletic', 6),
 ('#athlete', 6),
 ('@nikesportswear', 4),
 ('sport.', 4),
 ('athletes.', 4),
 ('@nikefootball', 3),
```

```
 ('@usnikefootball', 2)]
```

[167]:
```
word_summary_politics = adv.extract_words(tweets_users_df['tweet_full_text'],
                                          ['politic', 'polic', 'trump',␣
↪'donald'])
```

[168]:
```
word_summary_politics['overview']
```

[168]:
```
{'num_posts': 5089,
 'num_words': 780,
 'words_per_post': 0.1532717626252702,
 'unique_words': 133}
```

[169]:
```
word_summary_politics['top_words'][:20]
```

[169]:
```
[('@realdonaldtrump', 400),
 ('police', 56),
 ('trump', 53),
 ('#trump', 24),
 ('political', 13),
 ('#impeachtrump', 12),
 ('#fucktrump', 10),
 ('#trumpresign', 8),
 ('donald', 8),
 ('politics', 8),
 ('trump.', 6),
 ('#policebrutality', 6),
 ('#dumptrump', 5),
 ('trump!', 4),
 ('@donaldjtrumpjr', 4),
 ('#vetsagainsttrump', 4),
 ('#melaniatrump', 4),
 ('#trumpwh', 4),
 ('trump's', 4),
 ('#trumpsupporters', 4)]
```

Combine tweets, usernames, followers counts, with extracted entities

Now that we have extracted the entities that we want, we can now create a new DataFrame showing tweets, usernames, followers count, and the extracted entities:

[170]:
```
extracted_tweets =  (tweets_users_df[['tweet_full_text', 'user_screen_name',␣
↪'user_followers_count']]
.assign(hashtags=hashtag_summary['hashtags'],
        hashcounts=hashtag_summary['hashtag_counts'],
        mentions=mention_summary['mentions'],
        mention_count=mention_summary['mention_counts'],
        emoji=emoji_summary['emoji'],
```

```
        emoji_text=emoji_summary['emoji_text'],
        emoji_count=emoji_summary['emoji_counts'],))
extracted_tweets.head()
```

[170]:
```
                                        tweet_full_text  \
0
Done is better than perfect. - Sheryl Sandberg #quote #motivation #justdoit
https://t.co/J9lLdszdW6
1  Shout out to the Great Fire Department and the tour!    Much love to NYC!
   \n•\n•\n•\n#hero #fdny #likesforlikes #promo #music #instagood #instadaily
#postoftheday #bestoftheday #justdoit #nike #picoftheday…
https://t.co/sFobQ2ukpo
2                                    There are some AMAZINGLY hilarious Nike
Ad memes happening on my newsfeed.  Soooo, I decided to get a little creative
too… \n\n#JustDoIt #4YourMorning #4YourMemeCollection \n\n
https://t.co/6ok9qR6k6M
3
#kapernickeffect #swoosh #justdoit @ Lucas Bishop's Cigar Lounge
https://t.co/BhPBnjOkuU
4
One Hand, One Dream: The Shaquem Griffin Story  https://t.co/OEbEmwULLF
#shaquem  #NFL #Seattle #Seahawks #griffin #JustDoIt #Nike
https://t.co/pr8eosDZS7

  user_screen_name  user_followers_count  \
0    UltraYOUwoman               57983.0
1        yungcutup               13241.0
2      rachelbogle               11377.0
3      ErvGotti609                 218.0
4     NoLuckNeeded               13731.0

                                                hashtags  \
0
[#quote, #motivation, #justdoit]
1  [#hero, #fdny, #likesforlikes, #promo, #music, #instagood, #instadaily,
#postoftheday, #bestoftheday, #justdoit, #nike, #picoftheday]
2
[#justdoit, #4yourmorning, #4yourmemecollection]
3
[#kapernickeffect, #swoosh, #justdoit]
4
[#shaquem, #nfl, #seattle, #seahawks, #griffin, #justdoit, #nike]

   hashcounts mentions  mention_count                    emoji  \
0           3       []              0                       []
1          12       []              0  [ ,  ,  ,  ,  ,  ]
2           3       []              0                  [ ,  ]
```

```
3           3       []              0                       []
4           7       []              0                       []

           emoji_text  \
0
[]
1  [man firefighter, man firefighter, hundred points, movie camera, fire, raised
fist dark skin tone]
2                                                                    [cookie,
face with tears of joy]
3
[]
4
[]

   emoji_count
0            0
1            6
2            2
3            0
4            0
```

[171]:
```
word_freq_hash = adv.word_frequency(extracted_tweets['hashtags'].str.join(' '),
                                    extracted_tweets['user_followers_count'].
 ↪fillna(0))
word_freq_hash.head(10)
```

[171]:

	word	abs_freq	wtd_freq	rel_value
0	#justdoit	5150	17020680.0	3305.0
1	#drjanegoodall	1	2896006.0	2896006.0
2	#nike	1112	2076009.0	1867.0
3	#itstrue	1	1057047.0	1057047.0
4	#takeaknee	138	1004787.0	7281.0
5	#imwithkaep	46	814506.0	17707.0
6	#colinkaepernick	208	481859.0	2317.0
7	#nfl	106	400158.0	3775.0
8	#believeinsomething	44	379901.0	8634.0
9	#nflkickoff	75	373529.0	4980.0

The first one is of course going to be #justdoit because this is what all tweets contain, but the second and fourth are surprising, because we don't see them anywhere in the lists above, and they both have an absolute frequency of 1 (they were used only once).
This means that this one time where they were used they were tweeted by someone with a very large number of followers, and therefore, the tweet(s) containing these hashtags have achieved more reach than others, that have been tweeted more frequently.
Let's see who these tweets were tweeted by:

[172]:
```
extracted_tweets[extracted_tweets['hashtags'].str.join(' ').str.
↪contains('drjanegoodall|itstrue',case=False)]
```

[172]:
```
                          tweet_full_text  \
1465  When I see a video like this I have to keep repeating "they belong in the
wild, they belong in the wild…", resist the urge to get one as a pet &amp;
instead make a donation to the great work of #DrJaneGoodall.  #JustDoIt at
https://t.co/NKndhJu9np https://t.co/3vDHe4hqYh
4527
Believe in the 3 Is'. Intensity. Integrity. Intelligence. #itstrue #justdoit
https://t.co/rDZ29gYKTd

     user_screen_name  user_followers_count                    hashtags  \
1465    HamillHimself            2896006.0  [#drjanegoodall, #justdoit]
4527    RealKurtAngle            1057047.0        [#itstrue, #justdoit]

      hashcounts mentions  mention_count emoji emoji_text  emoji_count
1465           2       []              0   [ ]   [monkey]            1
4527           2       []              0    []         []            0
```

Apparently, there are two tweets by two different accounts who have 2,896,006 and 1,057,047 followers, respectively.
This is a very good example where you are able to extract hidden information in a data set. Had we not looked at the weighted frequency, we would have left out two tweets by users with 2.8M and 1.05M users, a massive amount of users. I'll leave it to you to explore further other findings in the table, and I'll close by getting the word frequency for mentions and emoji using the same technique.

[173]:
```
word_freq_mention = adv.word_frequency(extracted_tweets['mentions'].str.join('␣
↪'),
                                       extracted_tweets['user_followers_count'].
↪fillna(0))
word_freq_mention.head(10)
```

[173]:
```
              word  abs_freq   wtd_freq  rel_value
0            @nike       964  2576473.0     2673.0
1      @kaepernick7       493  1592478.0     3230.0
2  @realdonaldtrump       412  1422724.0     3453.0
3             @nfl        65   619220.0     9526.0
4           @nflpa        21   611288.0    29109.0
5       @nflcommish        21   574331.0    27349.0
6        @kingjames        27   565001.0    20926.0
7     @mosesbread72        16   541067.0    33817.0
8          @kstills        16   541067.0    33817.0
9   @malcolmjenkins        16   541067.0    33817.0
```

It seems there isn't much of a surprise here. The accounts that were mentioned the most are the ones you would expect based on the above findings. In some cases the most used words (mentions

in this case) are also the most used, on a weighted basis.

Let's see how things are with emoji:

[174]:
```
word_freq_emoji = adv.word_frequency(extracted_tweets['emoji'].str.join(' '),
                                     extracted_tweets['user_followers_count'].
 ↪fillna(0))
word_freq_emoji.head(10)
```

[174]:

	word	abs_freq	wtd_freq	rel_value
0		1	2896006.0	2896006.0
1		535	762943.0	1426.0
2		163	506331.0	3106.0
3		177	359634.0	2032.0
4		48	322516.0	6719.0
5		80	288040.0	3600.0
6		122	232236.0	1904.0
7		13	199547.0	15350.0
8		2	189520.0	94760.0
9		13	173217.0	13324.0

It seems we have two surprises here, where the monkey emoji reached more people (counting on a weighted basis) even though it was only used once. This is the same tweet we saw above with the top hashtag.
The police officer is also another surprise, because it is ranked 9 on a weighted basis, even though it was only twice in this dataset of 5,000 tweets. Another example of hidden, important, and surprising information, that can be easily overlooked.

`advertools` has a convenience dictionary to translate any emoji and provide you with the name of that emoji

[175]:
```
[adv.emoji_dict.emoji_dict[k] for k in word_freq_emoji['word'][:10]]
```

[175]:
```
[':monkey:',
 ':face_with_tears_of_joy:',
 ':heavy_check_mark:',
 ':rolling_on_the_floor_laughing:',
 ':United_States:',
 ':american_football:',
 ':fire:',
 ':face_blowing_a_kiss:',
 ':man_police_officer:',
 ':grinning_face_with_big_eyes:']
```

Adding to the same DataFrame for easier reading:

[176]:
```
word_freq_emoji[:10].assign(emoji_text=[adv.emoji_dict.emoji_dict[k] for k in␣
 ↪word_freq_emoji['word'][:10]])
```

[176]:

	word	abs_freq	wtd_freq	rel_value	emoji_text
0		1	2896006.0	2896006.0	:monkey:
1		535	762943.0	1426.0	:face_with_tears_of_joy:
2		163	506331.0	3106.0	:heavy_check_mark:
3		177	359634.0	2032.0	:rolling_on_the_floor_laughing:
4		48	322516.0	6719.0	:United_States:
5		80	288040.0	3600.0	:american_football:
6		122	232236.0	1904.0	:fire:
7		13	199547.0	15350.0	:face_blowing_a_kiss:
8		2	189520.0	94760.0	:man_police_officer:
9		13	173217.0	13324.0	:grinning_face_with_big_eyes: