

```
# -*- coding: utf-8 -*-
"""
Created on Thu Apr 20 16:12:50 2017

@author: Anonymized for peer review
"""

from twitter import *
from urllib import request
import re
import csv
import datetime

#enter path name for each newspaper to create distinct documents for each
#paper
with open("path.csv",mode="a") as file:
    file.close()

#Number of tweets to be scraped
n=200

#Username and location of account that shall be scraped (no "@")
twittername="Username"
source_location="location"

id_list=[]
article_id_list=[]
tweet_id_list=[]
time_scraped_list=[]
source_date_list=[]
source_list=[]
source_geo_list=[]
source_plz_list=[]
source_bl_list=[]
maintext_list=[]
n_hashtags_list=[]
first_hashtag_list=[]
second_hashtag_list=[]
third_hashtag_list=[]
lost_hashtags_list=[]
user_clear_list=[]
user_date_list=[]
username_list=[]
user_geo_list=[]
user_geo_strip_list=[]
user_plz_list=[]
user_bl_list=[]
rt_list=[]
fav_list=[]
reply_list=[]
user_text_list=[]
rep_text_list=[]
rep_receiver_list=[]
rep_receiver_source_list=[]
user_n_tweets_list=[]
user_n_followers_list=[]
user_n_following_list=[]
geo_match_list=[]
bl_match_list=[ ]
```

```
plz_diff_list=[]
first_engagement_filter_list=[]

a_article_id_list=[]
a_tweet_id_list=[]
a_time_scraped_list=[]
a_source_date_list=[]
a_source_list=[]
a_source_geo_list=[]
a_source_plz_list=[]
a_source_bl_list=[]
a_maintext_list=[]
a_tweet_url_list=[]
a_n_hashtags_list=[]
a_first_hashtag_list=[]
a_second_hashtag_list=[]
a_third_hashtag_list=[]
a_lost_hashtag_list=[]
a_rt_api_list=[]
a_fav_api_list=[]
a_rep_api_list=[]
a_rt_loc_list=[]
a_rt_loc_identified_list=[]
a_rt_loc_matching_list=[]
a_rt_loc_non_matching_list=[]
a_rt_bl_matching_list=[]
a_rt_bl_non_matching_list=[]
a_rt_plz_1000_greater_list=[]
a_rt_plz_1000_smaller_list=[]
a_rt_plz_10000_greater_list=[]
a_rt_plz_10000_smaller_list=[]
a_all_uniques_list=[]
a_all_loc_list=[]
a_all_loc_identified_list=[]
a_all_loc_matching_list=[]
a_all_loc_non_matching_list=[]
a_all_bl_matching_list=[]
a_all_bl_non_matching_list=[]
a_all_plz_1000_greater_list=[]
a_all_plz_1000_smaller_list=[]
a_all_plz_10000_greater_list=[]
a_all_plz_10000_smaller_list []

spillover_article_id_list=[]
spillover_tweet_id_list=[]
spillover_fav_count_list=[]
spillover_fav_saved_list=[]

#enter path to open list of city, federal state and zip code taken from
    datendieter(2012) [see main publication]
stadt_list=open("path.txt").read().splitlines()
bundesland_list=open("path.txt").read().splitlines()
plz_list=open("path.txt").read().splitlines()

tw_ids=[]
#enter path to current tweet id document (.txt / each id in new line)
tw_ids=open("path.txt").read().splitlines()
selected_tweets=[]
```

```
selection_counter=0
selection_goal=len(tw_ids)

#enter according information from your own Twitter accounts application
consumer_key = "consumer_key"
consumer_secret = "consumer_secret"
access_key = "access_key"
access_secret = "access_secret"

twitter=Twitter(auth = OAuth(access_key, access_secret, consumer_key,
    consumer_secret),retry=True)
posts=twitter.statuses.user_timeline(screen_name=twittername, count=n)

tweep=twittername
source_account=twitter.users.show(screen_name=twittername)

bl_counter=0
bl_goal=len(stadt_list)
quellbundesland="n.a."
while bl_counter<bl_goal:
    if source_location==stadt_list[bl_counter]:
        quellbundesland=bundesland_list[bl_counter]
    bl_counter=bl_counter+1

plz_counter=0
plz_goal=len(stadt_list)
quellplz="n.a."
while plz_counter<plz_goal:
    if source_location==stadt_list[plz_counter]:
        quellplz=plz_list[plz_counter]
    plz_counter=plz_counter+1

while selection_counter<selection_goal:
    p_counter=1
    for p in posts:
        if tw_ids[selection_counter]==p["id_str"]:
            selected_tweets.append(p)
            print("Selected Tweet number ",p_counter)
        p_counter=p_counter+1
    selection_counter=selection_counter+1

selected_list_counter=0
selected_list_goal=len(selected_tweets)

#enter path for document containing number of old articles
old_articles_list=open("path.txt").read().splitlines()
n_old_articles=int(old_articles_list[0])
article_counter=n_old_articles

#enter path for document containing number of old cases
old_cases_list=open("path.txt").read().splitlines()
n_old_cases=int(old_cases_list[0])
i=n_old_cases

while selected_list_counter<selected_list_goal:
    current_tweet=selected_tweets[selected_list_counter]
    current_id=current_tweet["id"]

    time_scraped=str(datetime.datetime.now())
```

```
current_tweets_user_list=[]
doubles=0

a_rt_loc=0
a_rt_loc_identified=0
a_rt_loc_matching=0
a_rt_loc_non_matching=0
a_rt_bl_matching=0
a_rt_bl_non_matching=0
a_rt_plz_1000_greater=0
a_rt_plz_1000_smaller=0
a_rt_plz_10000_greater=0
a_rt_plz_10000_smaller=0
a_all_uniques=0
a_all_loc=0
a_all_loc_identified=0
a_all_loc_matching=0
a_all_loc_non_matching=0
a_all_bl_matching=0
a_all_bl_non_matching=0
a_all_plz_1000_greater=0
a_all_plz_1000_smaller=0
a_all_plz_10000_greater=0
a_all_plz_10000_smaller=0

if article_counter<1000 and article_counter>99:
    article_id="A_"+twittername+'%d' % article_counter
elif article_counter<100 and article_counter>9:
    article_id="A_"+twittername+'0%d' % article_counter
elif article_counter<10:
    article_id="A_"+twittername+'00%d' % article_counter
else:
    article_id="A_+"+twittername+'%d' % article_counter
article_counter=article_counter+1

#Source Datum
source_date=current_tweet["created_at"]

#Source Name
source_name=source_account["name"]

#Source Geo und Bundesland
source_geo=source_location
source_bl=quellbundesland
source_plz=quellplz

#Maintext des Source Tweets
maintext=current_tweet["text"]
maintext=maintext.replace('\"','"')

#n_hashtags des Source Tweets
hashtags=re.findall(r'#\S*',maintext)
n_hashtags=len(hashtags)

#Inhalt der ersten drei Hashtags
first_hashtag="n.a."
second_hashtag="n.a."
third_hashtag="n.a."
lost_hashtags="nein"
```

```
current_tweet_ent=current_tweet["entities"]
current_tweet_hashtags=current_tweet_ent["hashtags"]
if len(current_tweet_hashtags)>0:
    hash1=current_tweet_hashtags[0]
    first_hashtag=hash1["text"]
if len(current_tweet_hashtags)>1:
    hash2=current_tweet_hashtags[1]
    second_hashtag=hash2["text"]
if len(current_tweet_hashtags)>2:
    hash3=current_tweet_hashtags[2]
    third_hashtag=hash3["text"]
if len(current_tweet_hashtags)>3:
    lost_hashtags="ja"

#Retweets
retweets=twitter.statuses.retweets._id(_id=current_id, count=100)
for r in retweets:
#Case ID
    case_counter=i
    if case_counter<1000 and case_counter>99:
        case_id=twittername+'%02d' % case_counter
    elif case_counter<100 and case_counter>9:
        case_id=twittername+'0%02d' % case_counter
    elif case_counter<10:
        case_id=twittername+'00%02d' % case_counter
    else:
        case_id=twittername+'%d' % case_counter

    i=i+1

    a_all_uniques=a_all_uniques+1

#Data of retweeting user:
user_date=r["created_at"]
rt=1
fav=0
reply=0

current_user=r["user"]
user_clear=current_user["name"]
username=current_user["screen_name"]
if len(current_user["location"])==0:
    user_geo="n.a."
else:
    user_geo=current_user["location"]
    user_geo=user_geo.replace("'", "")
a_rt_loc=a_rt_loc+1
a_all_loc=a_all_loc+1

user_geo_strippy=[]
strippy_counter=0
strippy_goal=len(stadt_list)-1
while strippy_counter<strippy_goal:
    stadtstring=stadt_list[strippy_counter]
    user_geo_strippy=re.findall(r'\b'+stadtstring+r'\b',user_geo,
                                flags=re.I)
    if len(user_geo_strippy)>0:
        strippy_counter=strippy_goal
    else:
        strippy_counter=strippy_counter+1
```

```
if len(user_geo_strippy)>0:
    user_geo_strip=user_geo_strippy[0]
    a_rt_loc_identified=a_rt_loc_identified+1
    a_all_loc_identified=a_all_loc_identified+1
else:
    user_geo_strip="n.a."

user_plz_counter=0
user_plz_goal=len(stadt_list)
user_plz="n.a."
while user_plz_counter<user_plz_goal:
    if user_geo_strip.lower()==stadt_list[user_plz_counter].lower():
        user_plz=plz_list[user_plz_counter]
    user_plz_counter=user_plz_counter+1

user_bl_counter=0
user_bl_goal=len(stadt_list)
user_bl="n.a."
while user_bl_counter<user_bl_goal:
    if user_geo_strip.lower()==stadt_list[user_bl_counter].lower():
        user_bl=bundesland_list[user_bl_counter]
    user_bl_counter=user_bl_counter+1

user_text=current_user["description"]
if user_text=="":
    user_text="n.a."
user_text=user_text.replace(' ', '')
user_n_tweets=current_user["statuses_count"]
user_n_followers=current_user["followers_count"]
user_n_following=current_user["friends_count"]

if source_geo.lower()==user_geo_strip.lower():
    geo_match=1
    a_rt_loc_matching=a_rt_loc_matching+1
    a_all_loc_matching=a_all_loc_matching+1
elif user_geo_strip=="n.a.":
    geo_match="n.a."
else:
    geo_match=0
    a_rt_loc_non_matching=a_rt_loc_non_matching+1
    a_all_loc_non_matching=a_all_loc_non_matching+1

if source_bl==user_bl:
    bl_match=1
    a_rt_bl_matching=a_rt_bl_matching+1
    a_all_bl_matching=a_all_bl_matching+1
elif user_bl=="n.a.":
    bl_match="n.a."
else:
    bl_match=0
    a_rt_bl_non_matching=a_rt_bl_non_matching+1
    a_all_bl_non_matching=a_all_bl_non_matching+1

if user_plz == "n.a.":
    plz_diff="n.a."
else:
    diff=int(source_plz)-int(user_plz)
    plz_diff=abs(diff)
    if int(plz_diff)>999:
        a_rt_plz_1000_greater=a_rt_plz_1000_greater+1
```

```
a_all_plz_1000_greater=a_all_plz_1000_greater+1
else:
    a_rt_plz_1000_smaller=a_rt_plz_1000_smaller+1
    a_all_plz_1000_smaller=a_all_plz_1000_smaller+1
if int(plz_diff)>9999:
    a_rt_plz_10000_greater=a_rt_plz_10000_greater+1
    a_all_plz_10000_greater=a_all_plz_10000_greater+1
else:
    a_rt_plz_10000_smaller=a_rt_plz_10000_smaller+1
    a_all_plz_10000_smaller=a_all_plz_10000_smaller+1

first_engagement_filter=1
rep_text="n.a."
rep_receiver="n.a."
rep_receiver_source="n.a."

id_list.append(case_id)
article_id_list.append(article_id)
tweet_id_list.append(current_id)
time_scraped_list.append(time_scraped)
source_date_list.append(source_date)
source_list.append(source_name)
source_geo_list.append(source_geo)
source_plz_list.append(source_plz)
source_bl_list.append(source_bl)
maintext_list.append(maintext)
n_hashtags_list.append(n_hashtags)
first_hashtag_list.append(first_hashtag)
second_hashtag_list.append(second_hashtag)
third_hashtag_list.append(third_hashtag)
lost_hashtags_list.append(lost_hashtags)
user_clear_list.append(user_clear)
user_date_list.append(user_date)
username_list.append(username)
user_geo_list.append(user_geo)
user_geo_strip_list.append(user_geo_strip)
user_plz_list.append(user_plz)
user_bl_list.append(user_bl)
rt_list.append(rt)
fav_list.append(fav)
reply_list.append(reply)
user_text_list.append(user_text)
rep_text_list.append(rep_text)
rep_receiver_list.append(rep_receiver)
rep_receiver_source_list.append(rep_receiver_source)
user_n_tweets_list.append(user_n_tweets)
user_n_followers_list.append(user_n_followers)
user_n_following_list.append(user_n_following)
geo_match_list.append(geo_match)
bl_match_list.append(bl_match)
plz_diff_list.append(plz_diff)
first_engagement_filter_list.append(first_engagement_filter)
print("Done with case: ",case_id," at: ",datetime.datetime.now())

current_tweets_user_list.append(username)

#from here: end of scraping retweets, Start of scraping favorites and
replies:
str_current_id=str(current_id)
tweet_url="https://twitter.com/"+twittername+"/status/"+str_current_id
```

```
#enter your Browser
req=request.Request(tweet_url, headers={'User-Agent' : "Mozilla/5.0"})
tweet_code=request.urlopen(req).read()

    tweet_code=tweet_code.decode(encoding="utf-8",errors="ignore").
    replace("\n"," ").replace("\t"," ").replace("<","<").replace(">",">
    ").replace("&Auml;","Ä").replace("&Ouml;","Ö").replace("&Uuml;","Ü")
    .replace("&auml;","ä").replace("&ouml;","ö").replace("&uuml;","ü").
    replace("&szlig;","ß").replace(""","'").replace("&nbsnbsp;"," ")

#Replies:
rep_code=re.findall(r'data-conversation-id(.*)stream-item-footer',
                     tweet_code)

rep_screen_name_list=[]
rep_maintext_list=[]
rep_y_receiver_list=[]

for rep_y in rep_code:
    str_rep=rep_y
    rep_y_screen_name=re.findall(r'data-screen-name="(.*?)"',str_rep)
    str_screen_name=rep_y_screen_name[0]
    rep_y_maintext=re.findall(r'TweetTextSize.*?>(.*?)</p>',str_rep)
    str_rep_maintext=rep_y_maintext[0]
    str_rep_maintext=str_rep_maintext.replace('\"','"')
    rep_y_receiver=re.findall(r'Antwort an.*?href="/(.*)"',str_rep)
    if len(rep_y_receiver)>0:
        str_rep_y_receiver=rep_y_receiver[0]
    else:
        str_rep_y_receiver="n.a."
    if str_screen_name!=twittername:
        rep_screen_name_list.append(str_screen_name)
        rep_maintext_list.append(str_rep_maintext)
        rep_y_receiver_list.append(str_rep_y_receiver)

rep_account_list=[]

for rep_user in rep_screen_name_list:
    rep_account=twitter.users.show(screen_name=rep_user)
    rep_account_list.append(rep_account)

rep_text_counter=0
for rep_user in rep_account_list:
#Case ID
    case_counter=i
    if case_counter<1000 and case_counter>99:
        case_id=twittername+'%d' % case_counter
    elif case_counter<100 and case_counter>9:
        case_id=twittername+'0%d' % case_counter
    elif case_counter<10:
        case_id=twittername+'00%d' % case_counter
    else:
        case_id=twittername+'%d' % case_counter

    i=i+1

#Data of replying user:
```

```
user_date="n.a."
rt=0
fav=0
reply=1

current_user=rep_user
user_clear=current_user[ "name" ]
username=current_user[ "screen_name" ]
if len(current_user[ "location" ])==0:
    user_geo="n.a."
else:
    user_geo=current_user[ "location" ]
    user_geo=user_geo.replace(' ','')

user_geo_strippy=[]
strippy_counter=0
strippy_goal=len(stadt_list)-1
while strippy_counter<strippy_goal:
    stadtstring=stadt_list[strippy_counter]
    user_geo_strippy=re.findall(r'\b'+stadtstring+r'\b',user_geo,
        flags=re.I)
    if len(user_geo_strippy)>0:
        strippy_counter=strippy_goal
    else:
        strippy_counter=strippy_counter+1
if len(user_geo_strippy)>0:
    user_geo_strip=user_geo_strippy[0]
else:
    user_geo_strip="n.a."

user_plz_counter=0
user_plz_goal=len(stadt_list)
user_plz="n.a."
while user_plz_counter<user_plz_goal:
    if user_geo_strip.lower()==stadt_list[user_plz_counter].lower():
        user_plz=plz_list[user_plz_counter]
    user_plz_counter=user_plz_counter+1

user_bl_counter=0
user_bl_goal=len(stadt_list)
user_bl="n.a."
while user_bl_counter<user_bl_goal:
    if user_geo_strip.lower()==stadt_list[user_bl_counter].lower():
        user_bl=bundesland_list[user_bl_counter]
    user_bl_counter=user_bl_counter+1

user_text=current_user[ "description" ]
if user_text=="":
    user_text="n.a."

rep_text=rep_maintext_list[rep_text_counter]

rep_receiver=rep_receiver_list[rep_text_counter]
if rep_receiver==twittername:
    rep_receiver_source=1
else:
    rep_receiver_source=0

rep_text_counter=rep_text_counter+1
```

```
user_text=user_text.replace('\"','"')
user_n_tweets=current_user["statuses_count"]
user_n_followers=current_user["followers_count"]
user_n_following=current_user["friends_count"]

if source_geo.lower()==user_geo_strip.lower():
    geo_match=1
elif user_geo_strip=="n.a.":
    geo_match="n.a."
else:
    geo_match=0

if source_bl==user_bl:
    bl_match=1
elif user_bl=="n.a.":
    bl_match="n.a."
else:
    bl_match=0

if user_plz == "n.a.":
    plz_diff="n.a."
else:
    diff=int(source_plz)-int(user_plz)
    plz_diff=abs(diff)

id_list.append(case_id)
article_id_list.append(article_id)
tweet_id_list.append(current_id)
time_scraped_list.append(time_scraped)
source_date_list.append(source_date)
source_list.append(source_name)
source_geo_list.append(source_geo)
source_plz_list.append(source_plz)
source_bl_list.append(source_bl)
maintext_list.append(maintext)
n_hashtags_list.append(n_hashtags)
first_hashtag_list.append(first_hashtag)
second_hashtag_list.append(second_hashtag)
third_hashtag_list.append(third_hashtag)
lost_hashtags_list.append(lost_hashtags)
user_clear_list.append(user_clear)
user_date_list.append(user_date)
username_list.append(username)
user_geo_list.append(user_geo)
user_geo_strip_list.append(user_geo_strip)
user_plz_list.append(user_plz)
user_bl_list.append(user_bl)
rt_list.append(rt)
fav_list.append(fav)
reply_list.append(reply)
user_text_list.append(user_text)
rep_text_list.append(rep_text)
rep_receiver_list.append(rep_receiver)
rep_receiver_source_list.append(rep_receiver_source)
user_n_tweets_list.append(user_n_tweets)
user_n_followers_list.append(user_n_followers)
user_n_following_list.append(user_n_following)
```

```
geo_match_list.append(geo_match)
bl_match_list.append(bl_match)
plz_diff_list.append(plz_diff)

if username not in current_tweets_user_list:
    first_engagement_filter=1
    a_all_uniques=a_all_uniques+1
    if user_geo!="n.a.":
        a_all_loc=a_all_loc+1
    if user_geo_strip!="n.a.":
        a_all_loc_identified=a_all_loc_identified+1
    if geo_match==1:
        a_all_loc_matching=a_all_loc_matching+1
    elif geo_match==0:
        a_all_loc_non_matching=a_all_loc_non_matching+1
    if bl_match==1:
        a_all_bl_matching=a_all_bl_matching+1
    elif bl_match==0:
        a_all_bl_non_matching=a_all_bl_non_matching+1
    if plz_diff!="n.a.":
        if int(plz_diff)>999:
            a_all_plz_1000_greater=a_all_plz_1000_greater+1
        else:
            a_all_plz_1000_smaller=a_all_plz_1000_smaller+1
        if int(plz_diff)>9999:
            a_all_plz_10000_greater=a_all_plz_10000_greater+1
        else:
            a_all_plz_10000_smaller=a_all_plz_10000_smaller+1
    else:
        first_engagement_filter=0
first_engagement_filter_list.append(first_engagement_filter)

current_tweets_user_list.append(username)
print("Done with case: ",case_id," at: ",datetime.datetime.now())

#Likes:
fav_code=re.findall(r'<li class="avatar-row(.*)?</div>',tweet_code)
if len(fav_code)>0:
    fav_code_str=fav_code[0]
else:
    fav_code_str="n.a."

fav_user_ids=re.findall(r'data-user-id="(.*?)"',fav_code_str)

fav_account_list=[]
for fav_user_id in fav_user_ids:
    int_fav_user_id=int(fav_user_id)
    fav_account=twitter.users.show(id=int_fav_user_id)
    fav_account_list.append(fav_account)

count_favs=0
for fav_user in fav_account_list:
#Case ID
    case_counter=i
    if case_counter<1000 and case_counter>99:
        case_id=twittername+'%03d' % case_counter
    elif case_counter<100 and case_counter>9:
        case_id=twittername+'00%d' % case_counter
    elif case_counter<10:
```

```
    case_id=twittername+'000%d' % case_counter
else:
    case_id=twittername+'%d' % case_counter

i=i+1

#Data of liking user:
user_date="n.a."
rt=0
fav=1
reply=0

current_user=fav_user
user_clear=current_user[ "name" ]
username=current_user[ "screen_name" ]
if len(current_user[ "location" ])==0:
    user_geo="n.a."
else:
    user_geo=current_user[ "location" ]
    user_geo=user_geo.replace(' ', '')

user_geo_strippy=[]
strippy_counter=0
strippy_goal=len(stadt_list)
while strippy_counter<strippy_goal:
    stadtstring=stadt_list[strippy_counter]
    user_geo_strippy=re.findall(r'\b'+stadtstring+r'\b',user_geo,
        flags=re.I)
    if len(user_geo_strippy)>0:
        strippy_counter=strippy_goal
    else:
        strippy_counter=strippy_counter+1
if len(user_geo_strippy)>0:
    user_geo_strip=user_geo_strippy[0]
else:
    user_geo_strip="n.a."

user_plz_counter=0
user_plz_goal=len(stadt_list)
user_plz="n.a."
while user_plz_counter<user_plz_goal:
    if user_geo_strip.lower()==stadt_list[user_plz_counter].lower():
        user_plz=plz_list[user_plz_counter]
    user_plz_counter=user_plz_counter+1

user_bl_counter=0
user_bl_goal=len(stadt_list)
user_bl="n.a."
while user_bl_counter<user_bl_goal:
    if user_geo_strip.lower()==stadt_list[user_bl_counter].lower():
        user_bl=bundesland_list[user_bl_counter]
    user_bl_counter=user_bl_counter+1

user_text=current_user[ "description" ]
if user_text=="":
    user_text="n.a."
user_text=user_text.replace(' ', '')
user_n_tweets=current_user[ "statuses_count" ]
```

```
user_n_followers=current_user["followers_count"]
user_n_following=current_user["friends_count"]

if source_geo.lower()==user_geo_strip.lower():
    geo_match=1
elif user_geo_strip=="n.a.":
    geo_match="n.a."
else:
    geo_match=0

if source_bl==user_bl:
    bl_match=1
elif user_bl=="n.a.":
    bl_match="n.a."
else:
    bl_match=0

if user_plz == "n.a.":
    plz_diff="n.a."
else:
    diff=int(source_plz)-int(user_plz)
    plz_diff=abs(diff)

first_engagement_filter=1
rep_text="n.a."
rep_receiver="n.a."
rep_receiver_source="n.a."

if username not in current_tweets_user_list:

    a_all_uniques=a_all_uniques+1
    if user_geo!="n.a.":
        a_all_loc=a_all_loc+1
    if user_geo_strip!="n.a.":
        a_all_loc_identified=a_all_loc_identified+1
    if geo_match==1:
        a_all_loc_matching=a_all_loc_matching+1
    elif geo_match==0:
        a_all_loc_non_matching=a_all_loc_non_matching+1
    if bl_match==1:
        a_all_bl_matching=a_all_bl_matching+1
    elif bl_match==0:
        a_all_bl_non_matching=a_all_bl_non_matching+1
    if plz_diff!="n.a.":
        if int(plz_diff)>999:
            a_all_plz_1000_greater=a_all_plz_1000_greater+1
        else:
            a_all_plz_1000_smaller=a_all_plz_1000_smaller+1
        if int(plz_diff)>9999:
            a_all_plz_10000_greater=a_all_plz_10000_greater+1
        else:
            a_all_plz_10000_smaller=a_all_plz_10000_smaller+1

    id_list.append(case_id)
    article_id_list.append(article_id)
    tweet_id_list.append(current_id)
    time_scraped_list.append(time_scraped)
    source_date_list.append(source_date)
    source_list.append(source_name)
    source_geo_list.append(source_geo)
```

```
source_plz_list.append(source_plz)
source_bl_list.append(source_bl)
maintext_list.append(maintext)
n_hashtags_list.append(n_hashtags)
first_hashtag_list.append(first_hashtag)
second_hashtag_list.append(second_hashtag)
third_hashtag_list.append(third_hashtag)
lost_hashtags_list.append(lost_hashtags)
user_clear_list.append(user_clear)
user_date_list.append(user_date)
username_list.append(username)
user_geo_list.append(user_geo)
user_geo_strip_list.append(user_geo_strip)
user_plz_list.append(user_plz)
user_bl_list.append(user_bl)
rt_list.append(rt)
fav_list.append(fav)
reply_list.append(reply)
user_text_list.append(user_text)
rep_text_list.append(rep_text)
rep_receiver_list.append(rep_receiver)
rep_receiver_source_list.append(rep_receiver_source)
user_n_tweets_list.append(user_n_tweets)
user_n_followers_list.append(user_n_followers)
user_n_following_list.append(user_n_following)
geo_match_list.append(geo_match)
bl_match_list.append(bl_match)
plz_diff_list.append(plz_diff)
first_engagement_filter_list.append(first_engagement_filter)

current_tweets_user_list.append(username)
count_favs=count_favs+1
print("Done with case: ",case_id," at: ",datetime.datetime.now())
)

else:
    doubles=doubles+1
    i=i-1

a_article_id_list.append(article_id)
a_tweet_id_list.append(current_id)
a_time_scraped_list.append(time_scraped)
a_source_date_list.append(source_date)
a_source_list.append(source_name)
a_source_geo_list.append(source_geo)
a_source_plz_list.append(source_plz)
a_source_bl_list.append(source_bl)
a_maintext_list.append(maintext)
a_tweet_url_list.append(tweet_url)
a_n_hashtags_list.append(n_hashtags)
a_first_hashtag_list.append(first_hashtag)
a_second_hashtag_list.append(second_hashtag)
a_third_hashtag_list.append(third_hashtag)
a_lost_hashtag_list.append(lost_hashtags)
a_rt_api_list.append(current_tweet["retweet_count"])
a_fav_api_list.append(current_tweet["favorite_count"])
a_rep_api_list.append(len(rep_account_list))
a_rt_loc_list.append(a_rt_loc)
a_rt_loc_identified_list.append(a_rt_loc_identified)
a_rt_loc_matching_list.append(a_rt_loc_matching)
```

```
a_rt_loc_non_matching_list.append(a_rt_loc_non_matching)
a_rt_bl_matching_list.append(a_rt_bl_matching)
a_rt_bl_non_matching_list.append(a_rt_bl_non_matching)
a_rt_plz_1000_greater_list.append(a_rt_plz_1000_greater)
a_rt_plz_1000_smaller_list.append(a_rt_plz_1000_smaller)
a_rt_plz_10000_greater_list.append(a_rt_plz_10000_greater)
a_rt_plz_10000_smaller_list.append(a_rt_plz_10000_smaller)
a_all_uniques_list.append(a_all_uniques)
a_all_loc_list.append(a_all_loc)
a_all_loc_identified_list.append(a_all_loc_identified)
a_all_loc_matching_list.append(a_all_loc_matching)
a_all_loc_non_matching_list.append(a_all_loc_non_matching)
a_all_bl_matching_list.append(a_all_bl_matching)
a_all_bl_non_matching_list.append(a_all_bl_non_matching)
a_all_plz_1000_greater_list.append(a_all_plz_1000_greater)
a_all_plz_1000_smaller_list.append(a_all_plz_1000_smaller)
a_all_plz_10000_greater_list.append(a_all_plz_10000_greater)
a_all_plz_10000_smaller_list.append(a_all_plz_10000_smaller)

selected_list_counter=selected_list_counter+1
if count_favs<current_tweet["favorite_count"]-doubles:
    spillover_article_id_list.append(article_id)
    spillover_tweet_id_list.append(current_id)
    spillover_fav_count_list.append(current_tweet["favorite_count"])
    spillover_fav_saved_list.append(count_favs)

#in CSV-Datei ablegen

#enter path for output file to save spillovers (likes that have not been
#scraped due to the rate limit of the preview window)
spillover_output=zip(spillover_article_id_list,spillover_tweet_id_list,
                     spillover_fav_count_list,spillover_fav_saved_list)
with open("path.csv",mode="a",encoding="utf-8") as fs:
    writer=csv.writer(fs)
#
    writer.writerow(("article_id","tweet_id","tweet_fav_count","favs_saved_b
y_us"))
writer.writerows(spillover_output)

#enter path for output file on tweet level
article_output=zip(a_article_id_list,a_tweet_id_list,a_time_scraped_list,
                   a_source_date_list,a_source_list,a_source_geo_list,a_source_plz_list,
                   a_source_bl_list,a_maintext_list,a_tweet_url_list,a_n_hashtags_list,
                   a_first_hashtag_list,a_second_hashtag_list,a_third_hashtag_list,
                   a_lost_hashtag_list,a_rt_api_list,a_fav_api_list,a_rep_api_list,
                   a_rt_loc_list,a_rt_loc_identified_list,a_rt_loc_matching_list,
                   a_rt_loc_non_matching_list,a_rt_bl_matching_list,
                   a_rt_bl_non_matching_list,a_rt_plz_1000_greater_list,
                   a_rt_plz_1000_smaller_list,a_rt_plz_10000_greater_list,
                   a_rt_plz_10000_smaller_list,a_all_uniques_list,a_all_loc_list,
                   a_all_loc_identified_list,a_all_loc_matching_list,
                   a_all_loc_non_matching_list,a_all_bl_matching_list,
                   a_all_bl_non_matching_list,a_all_plz_1000_greater_list,
                   a_all_plz_1000_smaller_list,a_all_plz_10000_greater_list,
                   a_all_plz_10000_smaller_list)
with open("path.csv",mode="a",encoding="utf-8") as fs:
    writer=csv.writer(fs)
#
    writer.writerow(("article_id","tweet_id","time_scraped","source_date","s
```

```
ource","source_geo","source_plz","source_bl","maintext","tweet_url","n_h
ashtags","first_hashtag","second_hashtag","third_hashtag","lost_hashtag"
,"rt_api","fav_api","rep","rt_loc","rt_loc_identified","rt_loc_matching"
,"rt_loc_non_matching","rt_bl_matching","rt_bl_non_matching","rt_plz_100
0_greater","rt_plz_1000_smaller","rt_plz_10000_greater","rt_plz_10000_sm
aller","all_uniques","all_loc","all_loc_identified","all_loc_matching","
all_loc_non_matching","all_bl_matching","all_bl_non_matching","all_plz_1
000_greater","all_plz_1000_smaller","all_plz_10000_greater","all_plz_100
00_smaller"))
writer.writerows(article_output)

#enter path for output file on interaction level
output=zip(id_list,article_id_list,tweet_id_list,time_scraped_list,
    source_date_list,source_list,source_geo_list,source_plz_list,
    source_bl_list,maintext_list,n_hashtags_list,first_hashtag_list,
    second_hashtag_list,third_hashtag_list,lost_hashtags_list,
    user_clear_list,user_date_list,username_list,user_geo_list,
    user_geo_strip_list,user_plz_list,user_bl_list,rt_list,fav_list,
    reply_list,user_text_list,rep_text_list,rep_receiver_list,
    rep_receiver_source_list,user_n_tweets_list,user_n_followers_list,
    user_n_following_list,geo_match_list,bl_match_list,plz_diff_list,
    first_engagement_filter_list)
with open("path.csv",mode="a",encoding="utf-8")as fo:
    writer=csv.writer(fo)
#
    writer.writerow(("case_id","article_id","tweet_id","time_scraped","sourc
e_date","source","source_geo","source_plz","source_bl","maintext","n_has
htags","first_hashtag","second_hashtag","third_hashtag","lost_hashtags",
"user_clear","user_date","username","user_geo","user_geo_strip","user_pl
z","user_bl","rt","fav","reply","user_text","rep_text_list","rep_receive
r","rep_receiver=source?","user_n_tweets","user_n_followers","user_n_fol
lowing","geo_match","bl_match","plz_diff","first_engagement_filter"))
writer.writerows(output)

#enter path to overwrite information about number of old articles
with open("path.txt",mode="w") as file:
    file.write("{}\n".format(article_counter))

#enter path to overwrite information about number of old cases
with open("path.txt",mode="w") as file:
    file.write("{}\n".format(i))

#Abschlussmeldung
print("done")
```