segalgouldn / **Senior-Project-Subtweets**

Branch: master ▾ Senior-Project-Subtweets / development / Non-Subtweets Downloader.md Find file Copy path

segalgouldn Markdown fixes. f46f7c4 an hour ago

1 contributor

84 lines (61 sloc) | 2.83 KB

Script for downloading a ground truth non-subtweets dataset

Import libraries for accessing the API and managing JSON data

```
import tweepy
import json
```

Load the API credentials

```
consumer_key, consumer_secret, access_token, access_token_secret = (open("../../credentials.txt")
                                                                    .read().split("\n"))
```

Authenticate the connection to the API using the credentials

```
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
```

Connect to the API

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

Define a function for recursively accessing parent tweets

```
def first_tweet(tweet_status_object):
    try:
        return first_tweet(api.get_status(tweet_status_object.in_reply_to_status_id_str, tweet_mode="extended"))
    except tweepy.TweepError:
        return tweet_status_object
```

Define a function for finding tweets with replies that specifically do not call them subtweets

```
def get_non_subtweets(max_tweets=10000000,
                      query=("-subtweet AND @ since:2018-03-01 exclude:retweets filter:replies")):
    non_subtweets_ids_list = []
    non_subtweets_list = []
    i = 0
    for potential_non_subtweet_reply in tweepy.Cursor(api.search, lang="en",
                                                      tweet_mode="extended", q=query).items(max_tweets):
        i += 1
        potential_non_subtweet_original = first_tweet(potential_non_subtweet_reply)
        if (not potential_non_subtweet_original.in_reply_to_status_id_str
            and potential_non_subtweet_original.user.lang == "en"):
            if (potential_non_subtweet_original.id_str in non_subtweets_ids_list
                or "subtweet" in potential_non_subtweet_original.full_text
                or "Subtweet" in potential_non_subtweet_original.full_text
                or "SUBTWEET" in potential_non_subtweet_original.full_text):
                continue
            else:
                non_subtweets_ids_list.append(potential_non_subtweet_original.id_str)
                non_subtweets_list.append({"tweet_data": potential_non_subtweet_original._json,
                                           "reply": potential_non_subtweet_reply._json})
                with open("../data/other_data/non_subtweets.json", "w") as outfile:
                    json.dump(non_subtweets_list, outfile, indent=4)
                print(("Tweet #{0} was a reply to a non-subtweet: {1}\n"
                       .format(i, potential_non_subtweet_original.full_text.replace("\n", " "))))
    return non_subtweets_list
```

Show the results

```
non_subtweets_list = get_non_subtweets()
print(len(non_subtweets_list))
```