



GTdownloader: A Python Package to Download, Visualize, and Export Georeferenced Tweets From the Twitter API

SOFTWARE
METAPAPER

JUAN ACOSTA-SEQUEDA 

SYBIL DERRIBLE 

*Author affiliations can be found in the back matter of this article

]u[ubiquity press

ABSTRACT

This article describes GTdownloader, a Python package that serves as both an API wrapper and as a geographic information pre-processing helper to facilitate the download of Twitter data from the Twitter API. Specifically, the package offers functions that enable the download of Twitter data through single functions that integrate access to the API call parameters in the form of familiar Python functions syntax. In addition, the data is available for download in common formats for further analysis, which includes standard Geographic Information Systems (GIS) vector format. This software package is especially useful for users with little to no experience with building API calls but who would highly benefit from access to Twitter data.

CORRESPONDING AUTHOR:

Juan Acosta-Sequeda

University of Illinois, Chicago

jugacostase@unal.edu.co

KEYWORDS:

Python; Data Science;
Geographical Information
Systems; Sentiment Analysis;
Twitter

TO CITE THIS ARTICLE:

Acosta-Sequeda J, Derrible S 2023 GTdownloader: A Python Package to Download, Visualize, and Export Georeferenced Tweets From the Twitter API. *Journal of Open Research Software*, 11: 7. DOI: <https://doi.org/10.5334/jors.443>

(1) OVERVIEW

INTRODUCTION

Social media data can provide insights into people's perception or preferences of specific topics [13], and thus have the possibility to impact many aspects of our society such as policies and infrastructure designs [1]. While getting a representative sample is often difficult or sometimes impossible [2, 6, 7, 10, 14], the text data available in platforms like Twitter can be valuable for research, especially given that many entities, from politicians and companies to influential individuals, use this platform to spread ideas, strategies, plans, and proposals. To make this data available to researchers, Twitter developed its own API [12], which is free for academic research. However, dealing with authentication, API calls, and data response handling can be overwhelming for researchers that have little to no experience in coding but who would still benefit from this type of data. In addition, the geographical component associated with geo-tagged tweets needs careful manipulation given that the text attributes of the tweets come packed in one list of tweets in the data entry of the response and the geographical attributes come in a separate list. For these reasons, we have developed GTdownloader, a high-level package that offers easy access to the full-archive-search Twitter API endpoint and compiles the retrieved data in standard formats so it can be easily manipulated and analyzed.

Although other interfaces exist to retrieve and analyze data from Twitter [5, 11], they are either not available in Python or they are not compatible with the current version of the API. The closest package identified in our search is TTLocVis [4], which also offers geographical data pre-visualization; however, for the most part, it offers static visualizations, and its main focus is topic modeling, which is out of the scope of GTdownloader.

IMPLEMENTATION AND ARCHITECTURE

The software implementation can be understood in terms of its two main classes: the *TweetDownloader()* class and the *GeoMethods()* class. The *TweetDownloader* class serves as the interface between the user and the API. Instead of having to build the entire body of the request, the user can interact with the API by means of Python functions that also follow the Python paradigm. For instance, a query in the form:

```
{'query': query,
 'start_time': start_time,
 'end_time': end_time,
 'expansions': 'geo.place_id,author_id',
 'place.fields': 'contained_within,country,country_
  code,full_name,geo,id'
 'tweet.fields': 'created_at,author_id,id,public_
  metrics,conversation_id',
```

```
'user.fields':
 'id,location,name,username,public_metrics',
 'max_results': max_page}
```

will be equivalent to the following implementation of a class method:

```
get_tweets(query=query, start_time=start_time,
           end_time=end_time, max_tweets=max_tweets
           )
```

Some of the query parameters of the request body are not part of the Python arguments. We implemented it this way because we consider these to be crucial for research and hence, are included in all requests in this package. In addition to simplifying the body of the requests into the *get_tweets()* method, the query optional parameters are also included as arguments in the same method. One way to illustrate this is the following query searching for tweets mentioning the FIFA World Cup that are just in English and retrieves only original tweets (i.e., re-tweets are not allowed):

```
query = "(FIFA World Cup) -is:retweet lang:en"
```

Using GTdownloader, it would translate into this:

```
get_tweets("FIFA World Cup", lang="en",
           include_retweets=False)
```

The Twitter API can retrieve a maximum of 500 tweets per call. If more tweets are needed, it is necessary to handle the API response pagination to get one page of results at a time by keeping track of the token generated in each response to identify the corresponding next page. GTdownloader takes care of this process so the user does not have to deal with this limitation regardless of the desired number of tweets.

Figure 1 illustrates how both classes interact with each other and the API, and it lists the possible outputs from their methods.

Aside from providing output files in standard formats for data analysis and Geographic Information Systems (GIS) post-processing, we leverage matplotlib [3] and Plotly [9] to include methods that would allow a user to visualize the tweets and their location in both static and interactive graphs. Further, we make use of the Wordcloud package [8] to generate a list of the most commonly used words and that takes stopwords (words not intended for the plot) as an argument of the plotting function.

DEMONSTRATION OF FUNCTIONALITY

Before downloading tweets from the API, users must ensure they have access to the API user keys provided by Twitter for researchers. Academic Researcher access applications can be submitted through the Twitter

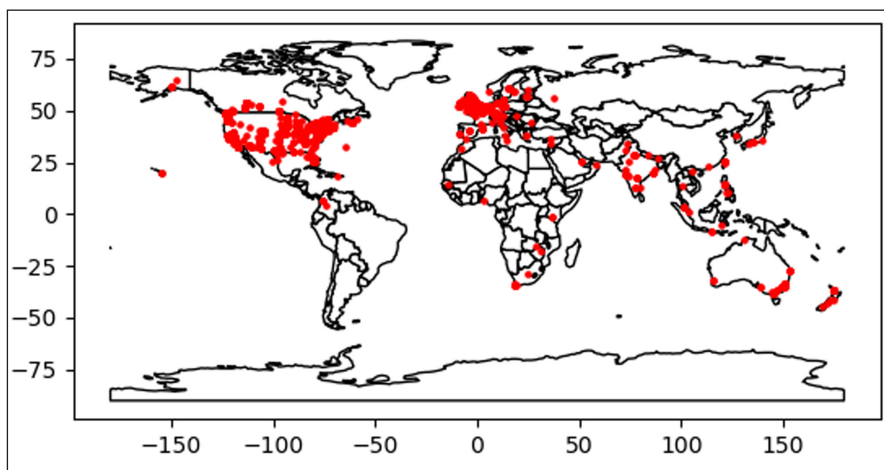


Figure 2 Simple map pre-visualization of tweets using bounding box centroids as geographical unit.

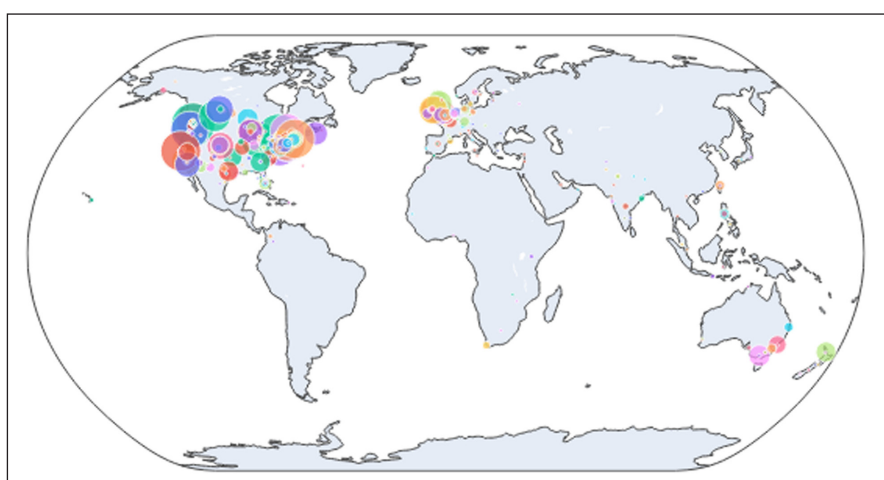


Figure 3 Interactive visualization displaying aggregated tweets using point size to represent tweet counts per location.

```
gtd.interactive_map_agg()
```

As shown in [Figure 4](#), a `time_unit` (year, month, day, hour, minute, or second) can be selected as well as the time unit in a map-based animation of the tweets.

```
gtd.map_animation(time_unit='month')
```

Finally, as shown in [Figure 5](#), a Wordcloud function lets us visualize the most common words in the downloaded tweets. Notice that we make use of the `custom_stopwords` parameter to exclude the query words and the `http` and `https` tags that may arise from url posting.

```
gtd.wordcloud(
    custom_stopwords=['bike', 'commuting',
                    'http', 'https'],
    background_color='white')
```

QUALITY CONTROL

GTDownloader is tested under the unit test framework. Unit tests have been included in the software repository and made available to all users. The tests are contained

in two main components: API transactions and data exports. The first one tests that all queries are performed correctly from the input parameters and that a successful response is obtained. The second tests component ensures the response is being handled correctly and that the output formats can be built correctly.

(2) AVAILABILITY

OPERATING SYSTEM

Works in all operating systems supporting Python.

PROGRAMMING LANGUAGE

Python 3.5 or higher

DEPENDENCIES

searchtweets-v2, Plotly, Geopandas, Wordcloud

LIST OF CONTRIBUTORS

Juan Acosta-Sequeda, Sybil Derrible

arguments, it is suitable to be incorporated into more complex pipelines that might include automated searches, text and sentiment analysis models, metrics tracking, and geographical dashboards.

The full documentation and reference of the package is provided online as well as installation instructions. Moreover, as authors we will do our best to provide support from users requests, which can be submitted in the form of GitHub issues.

ACKNOWLEDGEMENTS

Writing this functional piece of code and conducting the research that arose from it would not have been possible without the Academic Research access to the Twitter API v2, and hence the Twitter developers' team is gratefully acknowledged.

FUNDING INFORMATION


This research was supported in part by the National Science Foundation (NSF) EAGER grant no. 2014330. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funders.

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Juan Acosta-Sequeda  orcid.org/0000-0002-2940-1874
University of Illinois, Chicago

Sybil Derrible  orcid.org/0000-0002-2939-6016
University of Illinois, Chicago

REFERENCES

1. **Derrible S.** 2019, November 19. *Urban Engineering for Sustainability*. MIT Press. <https://mitpress.mit.edu/9780262043441/urban-engineering-for-sustainability/>.
2. **Hargittai E.** Potential Biases in Big Data: Omitted Voices on Social Media. *Social Science Computer Review*. 2020; 38(1): 10–24. DOI: <https://doi.org/10.1177/0894439318788322>
3. **Hunter JD.** Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*. 2007; 9(3): 90–95. DOI: <https://doi.org/10.1109/MCSE.2007.55>
4. **Kant G, Weisser C, Säfken B.** TLocVis: A Twitter Topic Location Visualization Package. *Journal of Open Source Software*. 2020; 5(54), 2507. DOI: <https://doi.org/10.21105/joss.02507>
5. **Kearney MW.** rtweet: Collecting and analyzing Twitter data. *Journal of Open Source Software*. 2019; 4(42): 1829. DOI: <https://doi.org/10.21105/joss.01829>
6. **Lock O, Pettit C.** Social media as passive geo-participation in transportation planning – how effective are topic modeling & sentiment analysis in comparison with citizen surveys? *Geo-Spatial Information Science*. 2020; 23(4): 275–292. DOI: <https://doi.org/10.1080/10095020.2020.1815596>
7. **Mellon J, Prosser C.** Twitter and Facebook are not representative of the general population: Political attitudes and demographics of British social media users. *Research & Politics*. 2017; 4(3): 205316801772000. DOI: <https://doi.org/10.1177/2053168017720008>
8. **Mueller A.** 2020. *WordCloud for Python*. http://amueller.github.io/word_cloud/.
9. **Plotly Technologies.** 2015. *Collaborative data science*. <https://plot.ly>.
10. **Reuter C, Kaufhold M-A, Spielhofer T, Hahne AS.** Social Media in Emergencies: A Representative Study on Citizens' Perception in Germany. *Proceedings of the ACM on Human-Computer Interaction*. 2017; 1(CSCW): 1–19. DOI: <https://doi.org/10.1145/3134725>
11. **Stojanovski D, Dimitrovski I, Madjarov G.** Tweetviz: Twitter data visualization. *Proceedings of the Data Mining and Data Warehouses*. 2014; 1–4.
12. **Twitter.** *Twitter API*; 2012. <https://developer.twitter.com/en/docs/twitter-api>.
13. **Verma S, Singh V.** Organizations and Employees Say “I do” to Work from Home during the Pandemic: A Sentiment Analysis of Twitter. *Journal of Information Systems and Technology Management*. 2022; 19: e202219008. DOI: <https://doi.org/10.4301/S1807-1775202219008>
14. **Wang Z, Hale S, Adelani DI, Grabowicz P, Hartman T, Flöck F, Jurgens D.** Demographic Inference and Representative Population Estimates from Multilingual Social Media Data. *The World Wide Web Conference*. 2019; 2056–2067. DOI: <https://doi.org/10.1145/3308558.3313684>

TO CITE THIS ARTICLE:

Acosta-Sequeda J, Derrible S 2023 GTdownloader: A Python Package to Download, Visualize, and Export Georeferenced Tweets From the Twitter API. *Journal of Open Research Software*, 11: 7. DOI: <https://doi.org/10.5334/jors.443>

Submitted: 20 October 2022

Accepted: 31 March 2023

Published: 08 June 2023

COPYRIGHT:

© 2023 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

Journal of Open Research Software is a peer-reviewed open access journal published by Ubiquity Press.

