

Tweet Explorer

Introduction

The Tweet Explorer consists of three components:

- The collector, after having set up the credentials, allows for connection to the Twitter Search API and saves the collected tweets in `jsonl` format.
- They are then passed on to the visualizer (middle), where the user can get an overview of the content and then create retweet and hashtag networks.
- The interactive networks are generated as html files that can be explored in the web browser. The modular structure of the three components facilitates the development of new features which are suggested by the light grey boxes.

Installation

UNIX / macOS

The tweet explorer requires Python ≥ 3.6 to run. You most likely already have Python installed. To check which Python version you have, open a terminal window and type:

Python -V OR python3 -V

If your version is above 3.6, continue to the next step. Otherwise, please refer to the guides specific to your operating system to install Python ≥ 3.6 .

Now run the following command to install the necessary Python libraries (use `pip3` if you used `python3` before):

Pip install -r requirements.txt

You can now start the collector from within the same terminal window:

Streamlit run collector.py

You should see an error message that tells you to authenticate with your Twitter Developer credentials. Move on to the next section to generate the necessary keys.

To close the streamlit interface, hit `CTRL` + `C` in the terminal.

Windows

Download and install Python 3.8.2 from <https://www.python.org/ftp/python/3.8.2/python-3.8.2-amd64.exe>, making sure to tick the option of adding Python to your PATH variable.

Now, type in the following command to install the necessary packages, followed by `ENTER`:

pip3 install -r requirements.txt

After a while, all packages should be installed and you can start the collector with

streamlit run collector.py

To close the streamlit interface, hit `CTRL` + `C` in the PowerShell.

Data collection

Authentication

To use the tweet explorer, you need to apply for a Twitter Developer Account - <https://developer.twitter.com/en/use-cases/academic-researchers>.

Follow the steps on the link to create a new research account or to transform an existing account into one. Now, go to the Apps section - <https://developer.twitter.com/en/apps> of your Twitter account and click on `Create an app` in the upper right corner:

Go to your new app and enter the `Keys and tokens` section. Copy the Consumer API keys.

Create a new file in the tweet explorer folder called `twitter_apikeys.txt` with the following content:

```
# api_key
```

```
<Insert api_key here>
```

```
# api_secret_key
```

```
<Insert api_secret_key here>
```

Collecting tweets

The collector connects to the Twitter Search API - <https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets>, which allows users to collect tweets from the last 7 days based on an advanced search.

Change to the folder where you downloaded streamlit, open a terminal and start the data collector by typing:

```
Streamlit run collector.py
```

The collector interface will open in your browser. You can start a search based on a keyword. The tweets will be downloaded and continuously written into a new [json-lines](http://jsonlines.org/) file in ``./data/{currentdate_keyword.jsonl}``. Each line of this file contains one tweet object.

Data transformation

Start the visualizer, which will open the second interface in a browser window:

`Streamlit run visualizer.py`

You can select a previously collected dataset for further analysis from a drop-down menu. If you have your own Twitter dataset, please convert it to the ``json-lines`` format (every tweet dictionary in one line) and copy it to the ``./data`` folder.

The visualizer will create a new folder for every collection you make in the ``output`` folder.

Timeline of tweets

As a first step, the visualizer creates a time series showing the amount of tweets in the dataset over time, which will be saved in the project folder.

Retweet networks

The tweet explorer can generate retweet networks in which nodes are users. A link is drawn from node ``i`` to ``j`` if ``i`` retweets ``j``. The following methods are available:

Giant Component

When enabled, the graph will be reduced to its largest connected component.

Aggregation methods

- "Soft" aggregation

- * Removes all users that are never retweeted and only retweet one other user (and can therefore not be bridges in the network)

- "Hard" aggregation

- * Removes all users from the network that are retweeted less than t times.

Privacy option

Removes all accessible metadata of users that have less than 5000 followers (no public figures) from the interactive visualization in order to comply with current privacy standards. The nodes are visible and their links are taken into account, but they cannot be personally identified in the interface.

Community detection

The tweet explorer currently supports Louvain and InfoMap algorithms for community detection. The community assignments are saved as node metadata. Note that the Louvain community detection does not take into account link direction.

Hashtag networks

The **tweet explorer** can generate hashtag networks in which nodes are hashtags. A link is drawn between node i and j if i and j appear in the same tweet. The following methods are available:

Giant Component

When enabled, the graph will be reduced to its largest connected component.

Community detection

The **tweet explorer** currently supports Louvain community detection for hashtag networks.

Clustergraphs

If community detection is enabled, clustergraphs will be generated for both retweet and hashtag networks in which nodes are communities and links are weighted according to the cumulative links between users of the communities.

Export options

Its modular structure (division into collector/visualizer/explorer) and the ability to export the data makes the tool compatible with a variety of other data analysis tools. Both retweet and hashtag networks are saved as edgelist (`.csv`), GML (`.gml`) and GraphViz Dot (`.gv`).

Network exploration

Open the generated `html` files to explore the generated networks (we recommend using the latest version of Firefox for full feature support). The command palette on the left displays information about the network and can be interacted with. Currently, the following features are implemented:

- show information about the dataset
- show number of nodes and links
- recolor nodes according to community assignment
- change node size according to metadata values
- change node scaling (experimental)
- display user metadata on click
- search for users / hashtags

File structure

A summary of the file structure is found below:

COLLECTED DATA (created by the collector)

`.data/`

`.data/{date}_tweets_{keyword}.jsonl` <-- collected dataset

INTERACTIVE NETWORKS (created by the visualizer)

`./output/`

`./output/{date}_{keyword}/{date}_{keyword}_timeline.html` <-- timeline of tweets

`./output/{date}_{keyword}/{date}_{keyword}_RTN.html` <-- retweet network

`./output/{date}_{keyword}/{date}_{keyword}_HTN.html` <-- hashtag network

`./output/{date}_{keyword}/{date}_{keyword}_RTN_CG_{comdec_method}.html` <-- retweet network clustergraph

`./output/{date}_{keyword}/{date}_{keyword}_HTN_CG_{comdec_method}.html` <-- hashtag network clustergraph

EXPORTED NETWORKS (created by the visualizer)

`./output/{date}_{keyword}/export/`

`./output/{date}_{keyword}/export/RTN.csv` <-- retweet network as edgelist

`./output/{date}_{keyword}/export/RTN.gml` <-- retweet network as gml

`./output/{date}_{keyword}/export/RTN.gv` <-- retweet network as dot for graphviz

`./output/{date}_{keyword}/export/HTN.csv` <-- hashtag network as edgelist