

# Package ‘voston.tcn’

September 30, 2021

**Title** Twitter Conversation Networks and Analysis

**Description** Collects tweets and metadata for threaded conversations and generates networks.

**Version** 0.2.4

**Type** Package

**Imports** dplyr, httr, jsonlite, magrittr, openssl, rlang, stringr,  
tibble, tidyr

**Depends** R (>= 3.6)

**License** GPL (>= 3)

**URL** <https://github.com/vostonlab/voston.tcn>

**BugReports** <https://github.com/vostonlab/voston.tcn/issues>

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**NeedsCompilation** no

**Author** Bryan Gertzel [aut, cre],  
Robert Ackland [ctb] (<<https://orcid.org/0000-0002-0008-1766>>),  
Francisca Borquez [ctb]

**Maintainer** Bryan Gertzel <bryan.gertzel@anu.edu.au>

**Repository** CRAN

**Date/Publication** 2021-09-30 09:20:02 UTC

## R topics documented:

tcn_network . . . . .	2
tcn_threads . . . . .	2
tcn_token . . . . .	4
tcn_tweets . . . . .	5

<b>Index</b>	<b>6</b>
--------------	----------

---

`tcn_network`*Generate network from conversation tweets*

---

**Description**

Creates the nodes and edges for a Twitter conversation network. An "activity" type network with tweets as nodes, or an "actor" type with users as nodes can be created.

**Usage**

```
tcn_network(data = NULL, type = "actor")
```

**Arguments**

<code>data</code>	Named list. Dataframes of threaded conversation tweets and users collected by <code>get_threads</code> function.
<code>type</code>	Character string. Type of network to generate, either "actor" or "activity". Default is "actor".

**Value**

Named list of dataframes for network nodes and edges.

**Examples**

```
## Not run:  
# generate twitter conversation network  
network <- tcn_network(tweets, "activity")  
  
# network nodes and edges  
network$nodes  
network$edges  
  
## End(Not run)
```

---

`tcn_threads`*Get threaded conversation tweets*

---

**Description**

Collects tweets that share the same Twitter conversation ID as supplied tweets.

**Usage**

```
tcn_threads(
  tweet_ids = NULL,
  token = NULL,
  endpoint = "recent",
  start_time = NULL,
  end_time = NULL,
  max_results = NULL,
  skip_list = NULL
)
```

**Arguments**

tweet_ids	List. Tweet ids of any tweet that are part of the threaded conversations of interest. Also accepts a list of tweet URLs or a mixed list.
token	List. Twitter API tokens.
endpoint	Character string. Twitter API v2 search endpoint. Can be either "recent" for the last 7 days or "all" if users app has access to historical "full-archive" tweets. Default is "recent".
start_time	Character string. Earliest tweet timestamp to return (UTC in ISO 8601 format). If NULL API will default to 30 days before end_time. Default is NULL.
end_time	Character string. Latest tweet timestamp to return (UTC in ISO 8601 format). If NULL API will default to now - 30 seconds. Default is NULL.
max_results	Numeric. Set maximum number of tweets to collect as a cap limit precaution. Will only be accurate to within one search request count (100 for standard or 500 tweets for academic project). This will not be ideal for most cases as an API search generally retrieves the most recent tweets first, therefore the beginning part of the last conversation thread may be absent. Default is NULL.
skip_list	Character vector. List of tweet conversation IDs to skip searching if found. This list is automatically appended with conversation_id's when collecting multiple conversation threads to prevent search duplication.

**Value**

A named list. Dataframes of tweets, users, errors and request metadata.

**Examples**

```
## Not run:
# get twitter conversation threads by tweet ids or urls
tweet_ids <- c("xxxxxxxx",
              "https://twitter.com/xxxxxxxx/status/xxxxxxxx")
tweets <- tcn_threads(tweet_ids, token, endpoint = "recent")

# get twitter conversation threads by tweet ids or urls using historical endpoint
# starting from May 01, 2021.
tweet_ids <- c("xxxxxxxx",
              "https://twitter.com/xxxxxxxx/status/xxxxxxxx")
```

```

tweets <- tcn_threads(tweet_ids,
                     token = token,
                     endpoint = "all",
                     start_time = "2021-05-01T00:00:00Z")

## End(Not run)

```

---

tcn\_token

*Get a twitter API access token*


---

### Description

Assigns a bearer token to the token object or retrieves a bearer token from the Twitter API using a Twitter apps consumer keys.

### Usage

```
tcn_token(bearer = NULL, consumer_key = NULL, consumer_secret = NULL)
```

### Arguments

```

bearer           Character string. App bearer token.
consumer_key     Character string. App consumer key.
consumer_secret  Character string. App consumer secret.

```

### Value

Named list containing the token.

### Examples

```

## Not run:

# assign bearer token
token <- tcn_token(bearer = "xxxxxxx")

# retrieve twitter app bearer token
token <- tcn_token(consumer_key = "xxxxxxx",
                  consumer_secret = "xxxxxxx")

## End(Not run)

```

---

`tcn_tweets`*Get tweets*

---

**Description**

Collects tweets for a list of tweet ids.

**Usage**

```
tcn_tweets(tweet_ids = NULL, token = NULL)
```

**Arguments**

<code>tweet_ids</code>	List. Tweet ids or tweet URLs.
<code>token</code>	List. Twitter API tokens.

**Value**

A named list. Dataframes of tweets, users, errors and request metadata.

**Examples**

```
## Not run:  
# get twitter conversation threads by tweet ids or urls  
tweet_ids <- c("xxxxxxx",  
              "https://twitter.com/xxxxxxx/status/xxxxxxx")  
tweets <- tcn_tweets(tweet_ids, token)  
  
## End(Not run)
```

# Index

[tcn\\_network](#), [2](#)  
[tcn\\_threads](#), [2](#)  
[tcn\\_token](#), [4](#)  
[tcn\\_tweets](#), [5](#)