Package ‘voson.tcn’

April 5, 2022

Title Twitter Conversation Networks and Analysis
Version 0.3.3
Description Collects tweets and metadata for threaded conversations and generates networks.
Type Package
Imports dplyr, httr, jsonlite, magrittr, openssl, rlang, stringr, tibble, tidyr
Depends R (>= 3.6)
Encoding UTF-8
Maintainer Bryan Gertzel <bryan.gertzel@anu.edu.au>
License GPL (>= 3)
RoxygenNote 7.1.2
NeedsCompilation no
URL https://github.com/vosonlab/voson.tcn
BugReports https://github.com/vosonlab/voson.tcn/issues
Author Bryan Gertzel [aut, cre], Robert Ackland [ctb] (https://orcid.org/0000-0002-0008-1766), Francisca Borquez [ctb]
Repository CRAN
Date/Publication 2022-04-05 07:00:02 UTC

R topics documented:

  tcn_counts ......................................................... 2
  tcn_network ...................................................... 3
  tcn_threads ....................................................... 4
  tcn_token ........................................................ 5
  tcn_tweets ......................................................... 6

Index 7
Description

Return the number of tweets for conversation ids.

Usage

tcn_counts(
    ids = NULL,
    token = NULL,
    endpoint = "recent",
    start_time = NULL,
    end_time = NULL,
    granularity = "day"
)

Arguments

ids List. Conversation ids.
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.
granularity Character string. Granularity or period for tweet counts. Can be "day", "minute" or "hour". Default is "day".

Value

A dataframe of conversation ids and counts.

Examples

## Not run:
# get tweet count for conversation thread over approximately 7 days
counts <-
tcn_counts(
    ids = "xxxxxxxx",
    token = token,
    endpoint = "all",
    start_time = "2020-09-30T01:00:00Z",
    end_time = "2020-10-07T01:00:00Z",
)
## 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

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

### Arguments

- **data**
  
  Named list. Dataframes of threaded conversation tweets and users collected by `get_threads` function.

- **type**

  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

```r
## Not run:
# generate twitter conversation network
tweets <- get_threads()

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 = 100,
    max_total = NULL,
    retry_on_limit = FALSE,
    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 per API v2 request. Up to 100 tweets for standard or 500 tweets for academic projects can be collected per request. Default is 100.

max_total  Numeric. Set maximum total number of tweets to collect as a cap limit precaution. Will only be accurate to within one search request count (being max_results, or 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 (reverse-chronological order), therefore the beginning part of the last conversation thread may be absent. Default is NULL.

retry_on_limit  Logical. When the API v2 rate-limit has been reached wait for reset time. Default is FALSE.
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

```r
## 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**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bearer</td>
<td>Character string. App bearer token.</td>
</tr>
<tr>
<td>consumer_key</td>
<td>Character string. App consumer key.</td>
</tr>
<tr>
<td>consumer_secret</td>
<td>Character string. App consumer secret.</td>
</tr>
</tbody>
</table>

**Value**

Named list containing the token.
Examples

```r
## Not run:

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

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

## End(Not run)
```

---

tcn_tweets

**Get tweets**

Description

Collects tweets for a list of tweet ids.

Usage

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

Arguments

- **tweet_ids**: List. Tweet ids or tweet URLs.
- **token**: List. Twitter API tokens.

Value

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

Examples

```r
## Not run:

# get twitter conversation threads by tweet ids or urls
tweet_ids <- c("xxxxxxxx",
               "https://twitter.com/xxxxxxxx/status/xxxxxxxx")
tweets <- tcn_tweets(tweet_ids, token)

## End(Not run)
```
Index

tcn_counts, 2
tcn_network, 3
tcn_threads, 4
tcn_token, 5
tcn_tweets, 6