Title  Twitter Conversation Networks and Analysis
Version  0.5.0
Description  Collects tweets and metadata for threaded conversations and generates networks.
Type  Package
Imports  dplyr, httr, jsonlite, openssl, progress, rlang, stringr, tibble, tidyr
Depends  R (>= 4.1)
Encoding  UTF-8
Maintainer  Bryan Gertzel <bryan.gertzel@anu.edu.au>
License  GPL (>= 3)
RoxygenNote  7.2.1
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  2022-08-30 13:50:02 UTC

R topics documented:

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

Index  8
tcn_counts  

*Get conversation tweet counts*

**Description**

Return the tweet count per day, hour or minute for conversation ids.

**Usage**

```r
tcn_counts(
  ids = NULL,
  token = NULL,
  endpoint = "recent",
  start_time = NULL,
  end_time = NULL,
  granularity = "day",
  retry_on_limit = TRUE,
  verbose = TRUE
)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ids</td>
<td>List. Conversation ids.</td>
</tr>
<tr>
<td>token</td>
<td>List. Twitter API tokens.</td>
</tr>
<tr>
<td>endpoint</td>
<td>Character string. Twitter API v2 search endpoint. Can be either &quot;recent&quot; for</td>
</tr>
<tr>
<td></td>
<td>the last 7 days or &quot;all&quot; if users app has access to historical &quot;full-archive&quot;</td>
</tr>
<tr>
<td></td>
<td>tweets. Default is &quot;recent&quot;.</td>
</tr>
<tr>
<td>start_time</td>
<td>Character string. Earliest tweet timestamp to return (UTC in ISO 8601 format).</td>
</tr>
<tr>
<td>end_time</td>
<td>Character string. Latest tweet timestamp to return (UTC in ISO 8601 format).</td>
</tr>
<tr>
<td>granularity</td>
<td>Character string. Granularity or period for tweet counts. Can be &quot;day&quot;, &quot;minute&quot;</td>
</tr>
<tr>
<td></td>
<td>or &quot;hour&quot;. Default is &quot;day&quot;.</td>
</tr>
<tr>
<td>retry_on_limit</td>
<td>Logical. When the API v2 rate-limit has been reached wait for reset time. Default is TRUE.</td>
</tr>
<tr>
<td>verbose</td>
<td>Logical. Output additional information. Default is TRUE.</td>
</tr>
</tbody>
</table>

**Value**

A dataframe of conversation ids and counts.

**Note**

A rate-limit of 300 requests per 15 minute window applies. If a conversation count request contains more than 31 days-worth of results it will use more than one request as API results will be paginated.
tcn_network

Examples

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

# total tweets per conversation id for period
counts$counts |> dplyr::count(conversation_id, wt = tweet_count)

## End(Not run)
```

### Description

Generates a network from conversation tweets. 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
network <- tcn_network(tweets, "activity")

# network nodes and edges
```
tn_threads

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,
annotations = FALSE,
max_results = NULL,
max_total = NULL,
retry_on_limit = TRUE,
skip_list = NULL,
verbose = TRUE
)

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.
annotations    Logical. Include tweet context annotation field in results. If TRUE Twitter imposes a max_results limit of 100. Default is FALSE.
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 NULL to set maximum allowed.
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 TRUE.

skip_list Character vector. List of tweet conversation IDs to skip searching if found.

verbose Logical. Output additional information. Default is TRUE.

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)
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

```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,
  referenced_tweets = FALSE,
  retry_on_limit = TRUE,
  clean = TRUE,
  verbose = TRUE
)
```

Arguments

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

  referenced_tweets
      Logical. Also retrieve tweets referenced by requested tweets. Default is FALSE.

  retry_on_limit
      Logical. When the API v2 rate-limit has been reached wait for reset time. De-
      fault is TRUE.

  clean
      Logical. Clean results.

  verbose
      Logical. Output additional information. Default is TRUE.

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_tweets(tweet_ids, token)

## End(Not run)
Index

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