

Package ‘graphTweets’

July 22, 2025

Type Package

Title Visualise Twitter Interactions

Version 0.5.3

Date 2020-01-02

Description Allows building an edge table from data frame of tweets,
also provides function to build nodes and another create a temporal graph.

License MIT + file LICENSE

Depends R (>= 3.2.0)

Imports dplyr, igraph, purrr, rlang, magrittr, utils, tidyr, zeallot,
combinat

RoxygenNote 7.0.2

URL <http://graphTweets.john-coene.com>

BugReports <https://github.com/JohnCoene/graphTweets/issues>

Suggests rtweet, testthat

Encoding UTF-8

NeedsCompilation no

Author John Coene [aut, cre]

Maintainer John Coene <jcoenep@gmail.com>

Repository CRAN

Date/Publication 2020-01-08 09:00:08 UTC

Contents

gt_collect	2
gt_dyn	3
gt_edges	3
gt_edges_from_text	4
gt_graph	5
gt_nodes	6
gt_save	7

gt_collect	<i>Collect</i>
------------	----------------

Description

Collect

Usage

```
gt_collect(gt)
```

Arguments

gt An object of class `graphTweets` as returned by [gt_edges](#) and [gt_nodes](#).

Value

A named list of [tibble](#) 1) edges and 2) nodes.

Examples

```
# simulate dataset
tweets <- data.frame(
  text = c("I tweet @you about @him",
           "I tweet @me about @you"),
  screen_name = c("me", "him"),
  retweet_count = c(19, 5),
  status_id = c(1, 2),
  stringsAsFactors = FALSE
)

tweets %>%
  gt_edges(text, screen_name, status_id) %>%
  gt_nodes() %>%
  gt_collect() -> net
```

gt_dyn*Dynamise*

Description

Create a dynamic graph to import in Gephi.

Usage

```
gt_dyn(gt, lifetime = Inf)
```

Arguments

gt	An object of class graphTweets as returned by gt_edges and gt_nodes .
lifetime	Lifetime of a tweet in milliseconds, defaults to Inf.

Examples

```
## Not run:
# simulate dataset
tweets <- data.frame(
  text = c("I tweet @you about @him and @her",
           "I tweet @me about @you"),
  screen_name = c("me", "him"),
  created_at = c(Sys.time(), Sys.time() + 10000),
  status_id = c(1, 2),
  stringsAsFactors = FALSE
)

tweets %>%
  gt_edges(text, screen_name, status_id, "created_at") %>%
  gt_nodes() %>%
  gt_dyn() %>%
  gt_collect() -> net

## End(Not run)
```

gt_edges*Edges*

Description

Get edges from data.frame of tweets.

Usage

```
gt_edges(data, source, target, ..., tl = TRUE)

gt_preproc_edges(gt, func)

gt_edges_bind(gt, source, target, ..., tl = TRUE)

gt_co_edges(data, col, tl = TRUE)

gt_co_edges_bind(gt, col, tl = TRUE)
```

Arguments

data	Data.frame of tweets, usually returned by the rtweet package.
source	Author of tweets.
target	Edges target.
...	any other column name, see examples.
tl	Set to TRUE to convert source and target to lower case (recommended).
gt	An object of class graphTweets as returned by gt_edges and gt_nodes .
func	Function to pre-process edges, takes edges as constructed by gt_edges, includes columns named source target and others passed to the three dot construct.
col	Column containing co-mentions.

Functions

- `gt_edges`: Build edges
- `gt_preproc_edges`: Pre-process edges
- `gt_edges_bind`: Append edges

<code>gt_edges_from_text</code>	<i>Edges from text</i>
---------------------------------	------------------------

Description

Get edges from data.frame of tweets.

Usage

```
gt_edges_from_text(data, id, source, tweets, ...)

gt_edges_from_text_(
  data,
  id = "status_id",
  source = "screen_name",
  tweets = "text",
  ...
)
```

Arguments

data	Data.frame of tweets, usually returned by the <code>rtweet</code> package.
id	tweets unique id.
source	Author of tweets.
tweets	Column containing tweets.
...	any other column name.

Details

The `tl` arguments stands for `tolower` and allows converting the `#hashtags` to lower case as these often duplicated, i.e.: `#python` `#Python`.

Value

An object of class `graphTweets`.

Functions

- `gt_edges` - Build networks of users.
- `gt_co_edges` - Build networks of users to hashtags.

Examples

```
# simulate dataset
tweets <- data.frame(
  text = c("I tweet @you about @him and @her",
           "I tweet @me about @you"),
  screen_name = c("me", "him"),
  retweet_count = c(19, 5),
  status_id = c(1, 2),
  hashtags = c("rstats", "Python"),
  stringsAsFactors = FALSE
)

tweets %>%
  gt_edges_from_text(status_id, screen_name, text)
```

gt_graph

Graph

Description

Build `igraph` object.

Usage

```
gt_graph(gt)
```

Arguments

gt An object of class graphTweets as returned by [gt_edges](#) and [gt_nodes](#).

Value

An object of class igraph.

Examples

```
# simulate dataset
tweets <- data.frame(
  text = c("I tweet @you about @him",
           "I tweet @me about @you"),
  screen_name = c("me", "him"),
  retweet_count = c(19, 5),
  status_id = c(1, 2),
  stringsAsFactors = FALSE
)

tweets %>%
  gt_edges(text, screen_name, status_id) %>%
  gt_nodes() %>%
  gt_graph() -> net
```

gt_nodes	<i>Nodes</i>
----------	--------------

Description

Get nodes from a graphTweets object.

Usage

```
gt_nodes(gt, meta = FALSE)

gt_add_meta(gt, name, source, target)
```

Arguments

gt An object of class graphTweets as returned by [gt_edges](#) and [gt_nodes](#).
meta Set to TRUE to add meta data to nodes using [users_data](#).
name Name of column to create.
source, target Name of column too apply to edge source and target.

Value

An object of class graphTweets.

Functions

- `gt_nodes`: Builds nodes
- `gt_add_meta`: Add meta data to the nodes. The meta data is taken from the edges.

<code>gt_save</code>	<i>Save</i>
----------------------	-------------

Description

Save the graph to file.

Usage

```
gt_save(gt, file = "graphTweets.graphml", format = "graphml", ...)
```

Arguments

<code>gt</code>	An object of class <code>graphTweets</code> as returned by gt_edges and gt_nodes .
<code>file</code>	File name including extension (format).
<code>format</code>	Format file format, see write_graph .
<code>...</code>	Any other argument to pass to write_graph .

Examples

```
## Not run:
# simulate dataset
tweets <- data.frame(
  text = c("I tweet @you about @him",
           "I tweet @me about @you"),
  screen_name = c("me", "him"),
  retweet_count = c(19, 5),
  created_at = c(Sys.time(), Sys.time() + 15000),
  status_id = c(1, 2),
  stringsAsFactors = FALSE
)

tweets %>%
  gt_edges(text, screen_name, "created_at") %>%
  gt_nodes(TRUE) %>%
  gt_dyn() %>%
  gt_save()

## End(Not run)
```

Index

`gt_add_meta (gt_nodes)`, [6](#)
`gt_co_edges (gt_edges)`, [3](#)
`gt_co_edges_bind (gt_edges)`, [3](#)
`gt_collect`, [2](#)
`gt_dyn`, [3](#)
`gt_edges`, [2](#), [3](#), [3](#), [4](#), [6](#), [7](#)
`gt_edges_bind (gt_edges)`, [3](#)
`gt_edges_from_text`, [4](#)
`gt_edges_from_text_`
 (`gt_edges_from_text`), [4](#)
`gt_graph`, [5](#)
`gt_nodes`, [2–4](#), [6](#), [6](#), [7](#)
`gt_preproc_edges (gt_edges)`, [3](#)
`gt_save`, [7](#)

`tibble`, [2](#)
`tolower`, [5](#)

`users_data`, [6](#)

`write_graph`, [7](#)