Package ‘graphTweets’

October 13, 2022

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

R topics documented:

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**

### 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

```r
# 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**
```r
## 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

\[
gt\_edges(data, source, target, \ldots, tl = \text{TRUE})
\]

\[
gt\_preproc\_edges(gt, func)
\]

\[
gt\_edges\_bind(gt, source, target, \ldots, tl = \text{TRUE})
\]

\[
gt\_co\_edges(data, col, tl = \text{TRUE})
\]

\[
gt\_co\_edges\_bind(gt, col, tl = \text{TRUE})
\]

Arguments

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

Functions

- \texttt{gt\_edges}: Build edges
- \texttt{gt\_preproc\_edges}: Pre-process edges
- \texttt{gt\_edges\_bind}: Append edges

\texttt{gt\_edges\_from\_text} \hspace{1cm} \textit{Edges from text}

Description

Get edges from data.frame of tweets.

Usage

\[
gt\_edges\_from\_text(data, id, source, tweets, \ldots)
\]

\[
gt\_edges\_from\_text_(
  data,
  id = "status\_id",
  source = "screen\_name",
  tweets = "text",
  \ldots
)
\]
Arguments

data  Data.frame of tweets, usually returned by the rtweet 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)
Arguments

gt

An object of class graphTweets as returned by \texttt{gt\_edges} and \texttt{gt\_nodes}.

Value

An object of class \texttt{igraph}.

Examples

```r
# 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
```

---

\textbf{gt\_nodes}

<table>
<thead>
<tr>
<th>\textit{Nodes}</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textbf{Description}</td>
</tr>
<tr>
<td>Get nodes from a graphTweets object.</td>
</tr>
</tbody>
</table>

\textbf{Usage}

```r
gt\_nodes(gt, meta = FALSE)
```

```r
gt\_add\_meta(gt, name, source, target)
```

\textbf{Arguments}

<table>
<thead>
<tr>
<th>\textit{gt}</th>
</tr>
</thead>
<tbody>
<tr>
<td>An object of class graphTweets as returned by \texttt{gt_edges} and \texttt{gt_nodes}.</td>
</tr>
</tbody>
</table>

| \textit{meta} |
| Set to \texttt{TRUE} to add meta data to nodes using \texttt{users\_data}. |

| \textit{name} |
| Name of column to create. |

| \textit{source, target} |
| Name of column too apply to edge source and target. |

\textbf{Value}

An object of class graphTweets.
gt_save

Functions

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

Description

Save the graph to file.

Usage

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

Arguments

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

Examples

```r
## 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