Package: vosonSML (via r-universe)

August 20, 2024

Version 0.35.0

Title Collecting Social Media Data and Generating Networks for Analysis

Description A suite of easy to use functions for collecting social media data and generating networks for analysis. Supports Mastodon, YouTube, Reddit and Web 1.0 data sources.

Type Package

Imports data.table, dplyr (>= 1.0), httr2, jsonlite, lubridate, methods, purrr, rlang (>= 1.0), stringr, textutils, tidyr, tibble

Suggests glue, igraph (>= 1.2.2), knitr, readr, rmarkdown, robotstxt, rtoot, rvest, stringi, testthat, urltools, vctrs, xml2

Depends R (>= 4.1)

Encoding UTF-8

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

License GPL (>= 3)

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

NeedsCompilation no

VignetteBuilder knitr

URL https://vosonlab.github.io/vosonSML

BugReports https://github.com/vosonlab/vosonSML/issues

Repository https://vosonlab.r-universe.dev

RemoteUrl https://github.com/vosonlab/vosonsml

RemoteRef HEAD

RemoteSha cd0e52fde45e35093d73679766766fc64aa53506

Contents

`AddText` *Add columns containing text data to network dataframes*

Description

Network is supplemented with additional social media text data applied as node or edge attributes.

Usage

```
AddText(net, data, ..., writeToFile = FALSE, verbose = TRUE)

add_text(net, data, ..., writeToFile = FALSE, verbose = TRUE)
```

Arguments

net	A named list of dataframes nodes and edges generated by Create.
data	A dataframe generated by Collect.
...	Additional parameters passed to function.
writeToFile	Logical. Write data to file. Default is FALSE.
verbose	Logical. Output additional information. Default is TRUE.

Value

Network as a named list of two dataframes containing $nodes and $edges including columns containing text data.

Note

Supports social media activity and actor networks. Refer to AddText.activity.reddit and AddText.actor.reddit for additional reddit parameters. Refer to AddText.actor.youtube for additional YouTube actor network parameters.

Examples

```
## Not run:
# add text to an activity network
net_activity <- collect_data |>
  Create("activity") |> AddText(collect_data)

# network
net_activity$nodes
net_activity$edges

## End(Not run)
```

AddText.activity.mastodon

Add columns containing text data to mastodon activity network dataframes

---

Description

Add columns containing text data to mastodon activity network dataframes

Usage

```
## S3 method for class 'activity.mastodon'
AddText(net, data, ..., writeToFile = FALSE, verbose = TRUE)
```

Arguments

net	A named list of dataframes nodes and edges generated by Create.
data	A dataframe generated by Collect.
...	Additional parameters passed to function. Not used in this method.
writeToFile	Logical. Write data to file. Default is FALSE.
verbose	Logical. Output additional information. Default is TRUE.

Value

Network as a named list of two dataframes containing $nodes and $edges including columns containing text data.

Examples

```
## Not run:
# add text to an activity network
net_activity <- collect_mdn |>
  Create("activity") |>
  AddText(collect_mdn)

# network
net_activity$nodes
net_activity$edges

## End(Not run)
```

AddText.activity.reddit

Add columns containing text data to reddit activity network dataframes

---

Description

Add columns containing text data to reddit activity network dataframes

Usage

```
## S3 method for class 'activity.reddit'
AddText(net, data, cleanText = FALSE, ..., writeToFile = FALSE, verbose = TRUE)
```

Arguments

net	A named list of dataframes nodes and edges generated by Create.
data	A dataframe generated by Collect.
cleanText	Logical. Simple removal of problematic characters for XML 1.0 standard. Implemented to prevent reddit specific XML control character errors when generating graphml files. Default is FALSE.
...	Additional parameters passed to function. Not used in this method.
writeToFile	Logical. Write data to file. Default is FALSE.
verbose	Logical. Output additional information. Default is TRUE.

Value

Network as a named list of two dataframes containing $nodes and $edges including columns containing text data.

Examples

```
## Not run:
# add text to an activity network
net_activity <- collect_rd |>
  Create("activity") |>
  AddText(collect_rd)

# network
net_activity$nodes
net_activity$edges

## End(Not run)
```

AddText.actor.mastodon

Add columns containing text data to mastodon actor network dataframes

---

Description

Add columns containing text data to mastodon actor network dataframes

Usage

```
## S3 method for class 'actor.mastodon'
AddText(net, data, ..., writeToFile = FALSE, verbose = TRUE)
```

Arguments

net	A named list of dataframes nodes and edges generated by Create.
data	A dataframe generated by Collect.
...	Additional parameters passed to function. Not used in this method.
writeToFile	Logical. Write data to file. Default is FALSE.
verbose	Logical. Output additional information. Default is TRUE.

Value

Network as a named list of two dataframes containing $nodes and $edges including columns containing text data.

Examples

```
## Not run:
# add text to an actor network ignoring references to actors at the beginning of
# comment text
net_actor <- collect_mdn |>
  Create("actor") |>
  AddText(collect_mdn)

# network
net_actor$nodes
net_actor$edges

## End(Not run)
```

AddText.actor.reddit *Add columns containing text data to reddit actor network dataframes*

Description

Add columns containing text data to reddit actor network dataframes

Usage

```
## S3 method for class 'actor.reddit'
AddText(net, data, cleanText = FALSE, ..., writeToFile = FALSE, verbose = TRUE)
```

Arguments

net	A named list of dataframes nodes and edges generated by Create.
data	A dataframe generated by Collect.
cleanText	Logical. Simple removal of problematic characters for XML 1.0 standard. Implemented to prevent reddit specific XML control character errors when generating graphml files. Default is FALSE.

...	Additional parameters passed to function. Not used in this method.
writeToFile	Logical. Write data to file. Default is FALSE.
verbose	Logical. Output additional information. Default is TRUE.

Value

Network as a named list of two dataframes containing $nodes and $edges including columns containing text data.

Examples

```
## Not run:
# add text to an actor network ignoring references to actors at the beginning of
# comment text
net_actor <- collect_rd |>
  Create("actor") |>
  AddText(collect_rd)

# network
net_actor$nodes
net_actor$edges

## End(Not run)
```

AddText.actor.youtube *Add columns containing text data to YouTube activity network dataframes*

Description

Text comments are added to the network as node attributes.

Text comments are added to the network as edge attributes. References to actors are detected at the beginning of comments and edges redirected to that actor instead if they differ from the top-level comment author.

Usage

```
## S3 method for class 'activity.youtube'
AddText(net, data, ..., writeToFile = FALSE, verbose = TRUE)

## S3 method for class 'actor.youtube'
AddText(
  net,
  data,
  repliesFromText = FALSE,
  atRepliesOnly = TRUE,
```

```
  ...,
  writeToFile = FALSE,
  verbose = TRUE
)
```

Arguments

net	A named list of dataframes nodes and edges generated by Create.
data	A dataframe generated by Collect.
...	Additional parameters passed to function. Not used in this method.
writeToFile	Logical. Write data to file. Default is FALSE.
verbose	Logical. Output additional information. Default is TRUE.
repliesFromText	Logical. If comment text for an edge begins with screen_name change the edge to be directed to screen_name - if different from the top level comment author that the reply comment was posted to. Default is FALSE.
atRepliesOnly	Logical. Comment screen_names must begin with an '@' symbol to be redirected. Default is TRUE.

Value

Network as a named list of two dataframes containing $nodes and $edges including columns containing text data.

Network as a named list of two dataframes containing $nodes and $edges including columns containing text data.

Examples

```
## Not run:
# add text to an activity network
net_activity <- collect_yt |>
  Create("activity") |> AddText(collect_yt)

# network
net_activity$nodes
net_activity$edges

## End(Not run)

## Not run:
# add text to an actor network ignoring references to actors at
# the beginning of comment text
net_actor <- collect_yt |>
  Create("actor") |>
  AddText(collect_yt, repliesFromText = FALSE)

# network
net_actor$nodes
net_actor$edges
```

```
## End(Not run)
```

AddVideoData *Add columns of video information to network dataframes*

Description

Network is supplemented with additional downloaded video information.

Usage

```
AddVideoData(net, youtubeAuth = NULL, ..., writeToFile = FALSE, verbose = TRUE)

add_videos(net, youtubeAuth = NULL, ..., writeToFile = FALSE, verbose = TRUE)
```

Arguments

net	A named list of dataframes nodes and edges generated by Create.
youtubeAuth	YouTube Authenticate object.
...	Additional parameters passed to function.
writeToFile	Logical. Write data to file. Default is FALSE.
verbose	Logical. Output additional information. Default is TRUE.

Value

Network as a named list of three dataframes containing $nodes, $edges and $videos nodes and edges include columns for additional video data.

Note

Only supports YouTube actor networks. Refer to AddVideoData.actor.youtube.

`AddVideoData.actor.youtube` *Add video information to youtube actor network dataframes*

Description

YouTube actor network is supplemented with additional downloaded video information. Adds video id, title, description and publish time as edge attributes. Nodes or actor references to video id's in the network are substituted with the actor id (video channel id) retrieved from the video details.

Usage

```
## S3 method for class 'actor.youtube'
AddVideoData(
  net,
  youtubeAuth = NULL,
  videoIds = NULL,
  actorSubOnly = FALSE,
  ...,
  writeToFile = FALSE,
  verbose = TRUE
)
```

Arguments

`net`	A named list of dataframes `nodes` and `edges` generated by `Create`.
`youtubeAuth`	YouTube Authenticate object.
`videoIds`	List. Video id's for which to download video information.
`actorSubOnly`	Logical. Only substitute video id's for their publishers channel id. Don't add additional video data to edge list.
`...`	Additional parameters passed to function.
`writeToFile`	Logical. Write data to file. Default is `FALSE`.
`verbose`	Logical. Output additional information. Default is `TRUE`.

Value

Network as a named list of three dataframes containing `$nodes`, `$edges` and `$videos` nodes and edges include columns for additional video data.

Examples

```
## Not run:
# replace video id references with actors and add video id, title, description and plublish time
# to an actor network
actorNetwork <- collectData |> Create("actor") |> AddVideoData(youtubeAuth)
```

```
# only replace video id references with actors that published videos in network
actorNetwork <- collectData |> Create("actor") |> AddVideoData(youtubeAuth, actorSubOnly = TRUE)

# network
# actorNetwork$nodes
# actorNetwork$edges

# dataframe of downloaded video data
# actorNetwork$videos

## End(Not run)
```

`Authenticate` *Create a credential object to access social media APIs*

Description

`Authenticate` creates a `credential` object that enables R to make authenticated calls to social media APIs. A `credential` object is a S3 object containing authentication related information such as an access token or key, and a class name identifying the social media that grants authentication. `Authenticate` is the first step of the `Authenticate`, `Collect` and `Create` workflow.

Refer to `Authenticate.mastodon`, `Authenticate.youtube`, `Authenticate.reddit` and `Authenticate.web` for parameters and usage.

Usage

```
Authenticate(socialmedia, ..., verbose = TRUE)
```

Arguments

`socialmedia`	Character string. Identifier for social media API to authenticate with. Supported social media are `"mastodon"`, `"youtube"`, `"reddit"` and `"web"`.
`...`	Optional parameters to pass to functions providied by supporting R packages that are used for social media API access.
`verbose`	Logical. Print messages to console. Default is `TRUE`.

`Authenticate.mastodon` *Mastodon API authentication*

Description

Mastodon OAuth authentication.

Usage

```
## S3 method for class 'mastodon'
Authenticate(
  socialmedia,
  instance = NULL,
  type = "public",
  ...,
  verbose = TRUE
)
```

Arguments

socialmedia	Character string. Identifier for social media API to authenticate, set to "mastodon".
instance	Character string. Server to authenticate against and create token.
type	Character string. Type of access, can be "public" or "user". Default is "public".
...	Additional parameters passed to function. Not used in this method.
verbose	Logical. Output additional information. Default is TRUE.

Value

A credential object containing an access token $auth and social media type descriptor $socialmedia set to "mastodon". Object has the class names "credential" and "mastodon".

Note

vosonSML uses the **rtoot** package for Mastodon data collection and API access tokens.

Examples

```
## Not run:
# mastodon API public access bearer token
mastodon_auth <- Authenticate(
  "mastodon",
  instance = "mastodon.social"
)

# mastodon API user access bearer token
mastodon_auth_user <- Authenticate(
  "mastodon",
  instance = "mastodon.social",
  type = "user"
)

# if thread collection API token not required
mastodon_auth <- Authenticate("mastodon")

## End(Not run)
```

`Authenticate.reddit` *Reddit API authentication*

Description

Reddit does not require authentication in this version of vosonSML.

Usage

```
## S3 method for class 'reddit'
Authenticate(socialmedia, ..., verbose = TRUE)
```

Arguments

`socialmedia`	Character string. Identifier for social media API to authenticate, set to `"reddit"`.
`...`	Additional parameters passed to function. Not used in this method.
`verbose`	Logical. Output additional information. Default is `TRUE`.

Value

A `credential` object containing a `$auth = NULL` value and social media type descriptor `$socialmedia` set to `"reddit"`. Object has the class names `"credential"` and `"reddit"`.

Note

Even though reddit does not require authentication in this version of vosonSML the `Authenticate` function must still be called to set the `socialmedia` identifier. This is used to route to the appropriate social media `Collect` function.

Examples

```
## Not run:
# reddit authentication
redditAuth <- Authenticate("reddit")

## End(Not run)
```

`Authenticate.web` *Web crawler authentication*

Description

Web crawler does not require authentication in this version of vosonSML.

Usage

```
## S3 method for class 'web'
Authenticate(socialmedia, ..., verbose = TRUE)
```

Arguments

`socialmedia`	Character string. Identifier for social media API to authenticate, set to `"web"`.
`...`	Additional parameters passed to function. Not used in this method.
`verbose`	Logical. Output additional information. Default is `TRUE`.

Value

A `credential` object containing a `$auth = NULL` value and social media type descriptor `$socialmedia` set to `"web"`. Object has the class names `"credential"` and `"web"`.

Note

Even though the web crawler does not require authentication in this version of vosonSML the `Authenticate` function must still be called to set the `socialmedia` identifier. This is used to route to the appropriate social media `Collect` function.

Examples

```
## Not run:
# web authentication
webAuth <- Authenticate("web")

## End(Not run)
```

Authenticate.youtube *YouTube API authentication*

Description

YouTube authentication uses OAuth2 and requires a Google Developer API key as described here: https://developers.google.com/youtube/v3/docs/.

Usage

```
## S3 method for class 'youtube'
Authenticate(socialmedia, apiKey, ..., verbose = TRUE)
```

Arguments

socialmedia	Character string. Identifier for social media API to authenticate, set to "youtube".
apiKey	Character string. Google developer API key to authenticate.
...	Additional parameters passed to function. Not used in this method.
verbose	Logical. Output additional information. Default is TRUE.

Value

A credential object containing an api key $auth and social media type descriptor $socialmedia set to "youtube". Object has the class names "credential" and "youtube".

Examples

```
## Not run:
# youtube authentication with google developer api key
myAPIKey <- "xxxxxxxxxxx"

youtubeAuth <- Authenticate("youtube", apiKey = myAPIKey)

## End(Not run)
```

Collect *Collect data from social media for generating networks*

Description

This function collects data from social media and structures it into a dataframe that can be used for creating networks for further analysis. Collect is the second step of the Authenticate, Collect, and Create workflow.

Usage

```
Collect(credential, ..., writeToFile = FALSE, verbose = TRUE)
```

Arguments

credential	A credential object generated from Authenticate.
...	Optional parameters to pass to functions providied by supporting R packages that are used for social media API collection.
writeToFile	Logical. Write data to file. Default is FALSE.
verbose	Logical. Output additional information. Default is TRUE.

Collect.listing.reddit

Collect reddit thread listings from subreddits

Description

Collects thread listings for one or more specified subreddits and structures the data into a dataframe.

Usage

```
## S3 method for class 'listing.reddit'
Collect(
  credential,
  endpoint,
  subreddits,
  sort = "hot",
  period = "all",
  max = 25,
  waitTime = c(6, 8),
  ua = getOption("HTTPUserAgent"),
  ...,
  writeToFile = FALSE,
  verbose = TRUE
)

collect_reddit_listings(
  subreddits,
  sort = "new",
  period = NULL,
  max = 25,
  waitTime = c(6, 8),
  ua = vsml_ua(),
  writeToFile = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

credential	A credential object generated from Authenticate with class name "reddit".
endpoint	API endpoint.
subreddits	Character vector. Subreddit names to collect thread listings from.
sort	Character vector. Listing thread sort order. Options are "hot", "top", "new", and "rising". Default is "hot".
period	Character vector. Listing top threads by time period. Only applicable to sort order by "top". Options are "hour", "day", "week", "month", "year" and "all". Default is "all".
max	Numeric vector. Maximum number of threads in listing to return. Default is 25.
waitTime	Numeric vector. Time range in seconds to select random wait from in-between url collection requests. Minimum is 3 seconds. Default is c(6, 8) for a wait time chosen from between 6 and 8 seconds.
ua	Character string. Override User-Agent string to use in Reddit thread requests. Default is option("HTTPUserAgent") value as set by vosonSML.
...	Additional parameters passed to function. Not used in this method.
writeToFile	Logical. Write collected data to file. Default is FALSE.
verbose	Logical. Output additional information. Default is TRUE.

Value

A tibble object with class names "listing" and "reddit".

Note

The reddit endpoint used for collection has maximum limit of 25 per listing.

Examples

```
## Not run:
# subreddit url to collect threads from
subreddits  <- c("datascience")

redditListing <- redditAuth |>
  Collect(endpoint = "listing", subreddits = subreddits, sort = "new", writeToFile = TRUE)

## End(Not run)
```

Collect.search.mastodon *Collect post data from mastodon search*

Description

This function collects posts based on search terms and structures the data into a dataframe with the class names "datasource" and "mastodon".

Usage

```
## S3 method for class 'search.mastodon'
Collect(
  credential,
  endpoint,
  hashtag = NULL,
  instance = NULL,
  local = FALSE,
  numPosts = 100,
  anonymous = TRUE,
  retryOnRateLimit = TRUE,
  ...,
  writeToFile = FALSE,
  verbose = TRUE
)
```

Arguments

credential	A credential object generated from Authenticate with class name "mastodon".
endpoint	API endpoint.
hashtag	Character string. Specifies a mastodon query to search on e.g #hashtag. Set to NULL for unfiltered public posts. Default is NULL.
instance	Character string. Server to collect posts from. Default is NULL.
local	Logical. Search the local server or global timeline. FALSE.
numPosts	Numeric. Specifies how many tweets to be collected. Default is 100.
anonymous	Logical. Collect public posts without authenticating. Default is TRUE.
retryOnRateLimit	Logical. When the API rate-limit is reached should the collection wait and resume when it resets. Default is TRUE.
...	Arguments passed on to rtoot::get_timeline_hashtag only_media logical, Show only statuses with media attached? min_id character, Return results immediately newer than this id
writeToFile	Logical. Write collected data to file. Default is FALSE.
verbose	Logical. Output additional information. Default is TRUE.

Value

A tibble object with class names "datasource" and "mastodon".

`Collect.thread.mastodon` *Collect posts data from mastodon threads*

Description

Collects public posts for one or more specified mastodon conversation threads and structures the data into a dataframe with the class names "datasource" and "mastodon".

Usage

```
## S3 method for class 'thread.mastodon'
Collect(
  credential,
  endpoint,
  threadUrls,
  ...,
  writeToFile = FALSE,
  verbose = TRUE
)
```

Arguments

credential	A credential object generated from Authenticate with class name "mastodon".
endpoint	API endpoint.
threadUrls	Character vector. Mastodon thread post urls to collect data from.
...	Additional parameters passed to function. Not used in this method.
writeToFile	Logical. Write collected data to file. Default is FALSE.
verbose	Logical. Output additional information about the data collection. Default is TRUE.

Value

A tibble object with class names "datasource" and "mastodon".

Examples

```
## Not run:
# post urls to collect threads from
threadUrls <- c("https://mastodon.social/@xxxxxx/xxxxxxxxx")

mastodonData <- Authenticate("mastodon") |>
  Collect(threadUrls = threadUrls, writeToFile = TRUE)
```

```
## End(Not run)
```

`Collect.thread.reddit` *Collect comments data from reddit threads*

Description

Collects comments made by users on one or more specified subreddit conversation threads and structures the data into a dataframe with the class names `"datasource"` and `"reddit"`.

Usage

```
## S3 method for class 'thread.reddit'
Collect(
  credential,
  endpoint,
  threadUrls,
  sort = NA,
  waitTime = c(6, 8),
  ua = getOption("HTTPUserAgent"),
  ...,
  writeToFile = FALSE,
  verbose = TRUE
)

collect_reddit_threads(
  threadUrls,
  sort = "best",
  waitTime = c(6, 8),
  ua = vsml_ua(),
  writeToFile = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

`credential`	A `credential` object generated from `Authenticate` with class name `"reddit"`.
`endpoint`	API endpoint.
`threadUrls`	Character vector. Reddit thread urls to collect data from.
`sort`	Character vector. Reddit comment sort order. Options are `"best"`, `"top"`, `"new"`, `"controversial"`, `"old"`, and `"qa"`. Default is `NA`.
`waitTime`	Numeric vector. Time range in seconds to select random wait from in-between url collection requests. Minimum is 3 seconds. Default is `c(6, 8)` for a wait time chosen from between 6 and 8 seconds.

ua	Character string. Override User-Agent string to use in Reddit thread requests. Default is option("HTTPUserAgent") value as set by vosonSML.
...	Additional parameters passed to function. Not used in this method.
writeToFile	Logical. Write collected data to file. Default is FALSE.
verbose	Logical. Output additional information about the data collection. Default is TRUE.

Value

A tibble object with class names "datasource" and "reddit".

Note

The reddit web endpoint used for collection has maximum limit of 500 comments per thread url.

Examples

```
## Not run:
# subreddit url to collect threads from
threadUrls <- c("https://www.reddit.com/r/xxxxxx/comments/xxxxxx/x_xxxx_xxxxxxxxx/")

redditData <- redditAuth |>
  Collect(threadUrls = threadUrls, writeToFile = TRUE)

## End(Not run)
```

Collect.web *Collect hyperlinks from web pages*

Description

Collects hyperlinks from web pages and structures the data into a dataframe with the class names "datasource" and "web".

Usage

```
## S3 method for class 'web'
Collect(credential, pages = NULL, ..., writeToFile = FALSE, verbose = TRUE)

collect_web_hyperlinks(pages = NULL, writeToFile = FALSE, verbose = TRUE, ...)
```

Arguments

credential	A credential object generated from Authenticate with class name "web".
pages	Dataframe. Dataframe of web pages to crawl. The dataframe must have the columns page (character), type (character) and max_depth (integer). Each row is a seed web page to crawl, with the page value being the page URL. The type value is type of crawl as either "int", "ext" or "all", directing the crawler to follow only internal links, follow only external links (different domain to the seed page) or follow all links. The max_depth value determines how many levels of hyperlinks to follow from the seed site.
...	Additional parameters passed to function. Not used in this method.
writeToFile	Logical. Write collected data to file. Default is FALSE.
verbose	Logical. Output additional information. Default is TRUE.

Value

A tibble object with class names "datasource" and "web".

Examples

```
## Not run:
pages <- tibble::tibble(page = c("http://vosonlab.net",
                                 "https://rsss.cass.anu.edu.au"),
                        type = c("int", "all"),
                        max_depth = c(2, 2))

webData <- webAuth |>
  Collect(pages, writeToFile = TRUE)

## End(Not run)
```

Collect.youtube *Collect comments data for YouTube videos*

Description

This function collects public comments data for one or more YouTube videos using the YouTube Data API v3 and structures the data into a dataframe with the class names "datasource" and "youtube".

YouTube has a quota unit system as a rate limit with most developers having either 10,000 or 1,000,000 units per day. Many read operations cost a base of 1 unit such as retrieving individual comments, plus 1 or 2 units for text snippets. Retrieving threads or top-level comments with text costs 3 units per request (maximum 100 comments per request). Using this function a video with 250 top-level comments and 10 of those having reply comments of up to 100 each, should cost (9 + 20) 29 quota units and return between 260 and 1260 total comments. There is currently a limit of 100 reply comments collected per top-level comment.

More information about the YouTube Data API v3 can be found here: https://developers.google.com/youtube/v3/getting-started

Usage

```
## S3 method for class 'youtube'
Collect(
  credential,
  videoIDs = c(),
  maxComments = 1e+10,
  ...,
  writeToFile = FALSE,
  verbose = TRUE
)
```

Arguments

credential	A credential object generated from Authenticate with class name "youtube".
videoIDs	Character vector. Specifies YouTube video URLs or IDs. For example, if the video URL is https://www.youtube.com/watch?v=xxxxxxxxxxx then use URL or ID videoIDs = c("xxxxxxxxxxx").
maxComments	Numeric integer. Specifies how many top-level comments to collect from each video. This value does not consider replies to top-level comments. The total number of comments returned for a video will usually be greater than maxComments depending on the number of reply comments present.
...	Additional parameters passed to function. Not used in this method.
writeToFile	Logical. Write data to file. Default is FALSE.
verbose	Logical. Output additional information. Default is TRUE.

Value

A tibble object with class names "datasource" and "youtube".

Note

Due to specifications of the YouTube Data API it is currently not efficient to specify the exact number of comments to return from the API using maxComments parameter. The maxComments parameter is applied to top-level comments only and not the replies to these comments. As such the number of comments collected is usually greater than expected. For example, if maxComments is set to 10 and one of the videos 10 top-level comments has 5 reply comments then the total number of comments collected will be 15 for that video. Comments data for multiple YouTube videos can be requested in a single operation, maxComments is applied to each individual video and not the combined total of comments.

Examples

```
## Not run:
# list of YouTube video urls or ids to collect
video_ids <- c("https://www.youtube.com/watch?v=xxxxxxxx",
               "https://youtu.be/xxxxxxxx",
               "xxxxxxx")
```

```
# collect approximately 200 comments for each YouTube video
youtubeData <- youtubeAuth |>
  Collect(videoIDs = video_ids, maxComments = 200)

## End(Not run)
```

`Create` *Create networks from social media data*

Description

This function creates networks from social media data as produced from `Collect`. `Create` is the final step of the `Authenticate`, `Collect` and `Create` workflow.

Usage

```
Create(datasource, type, ..., writeToFile = FALSE, verbose = TRUE)
```

Arguments

`datasource`	Collected social media data of class `"datasource"` and `socialmedia`.
`type`	Character string. Type of network to be created, can be `"activity"`, `"actor"`, `"twomode"` or `"semantic"`.
`...`	Optional parameters to pass to functions providied by supporting R packages that are used for social media network creation.
`writeToFile`	Logical. Write data to file. Default is `FALSE`.
`verbose`	Logical. Output additional information. Default is `TRUE`.

`Create.activity.mastodon`

Create mastodon activity network

Description

Creates a mastodon activity network from collected posts. Nodes are posts and directed edges represent the relationship of posts to one another.

Usage

```
## S3 method for class 'activity.mastodon'
Create(
  datasource,
  type,
  subtype = NULL,
  ...,
  writeToFile = FALSE,
  verbose = TRUE
)
```

Arguments

datasource	Collected social media data with "datasource" and "mastodon" class names.
type	Character string. Type of network to be created, set to "activity".
subtype	Character string. Subtype of activity network to be created. Can be set to "tag". Default is NULL.
...	Additional parameters passed to function. Not used in this method.
writeToFile	Logical. Write data to file. Default is FALSE.
verbose	Logical. Output additional information. Default is TRUE.

Value

Network as a named list of two dataframes containing $nodes and $edges.

Examples

```
## Not run:
# create a mastodon activity network
activity_net <- mastodon_data |> Create("activity")

# create a mastodon tag relations network
activity_net <- mastodon_data |> Create("activity", "tag")

## End(Not run)
```

Create.activity.reddit

Create reddit activity network

Description

Creates a reddit activity network from subreddit thread comments. Nodes are comments and initial thread posts, edges form the discussion structure and signify to which comment or post a comment has been made to.

Usage

```
## S3 method for class 'activity.reddit'
Create(datasource, type, ..., writeToFile = FALSE, verbose = TRUE)
```

Arguments

datasource	Collected social media data with "datasource" and "reddit" class names.
type	Character string. Type of network to be created, set to "activity".
...	Additional parameters passed to function. Not used in this method.
writeToFile	Logical. Write data to file. Default is FALSE.
verbose	Logical. Output additional information. Default is TRUE.

Value

Network as a named list of two dataframes containing $nodes and $edges.

Examples

```
## Not run:
# create a reddit activity network graph
activityNetwork <- redditData |> Create("activity")

# network
# activityNetwork$nodes
# activityNetwork$edges

## End(Not run)
```

Create.activity.web *Create web activity network*

Description

Creates a web page activity network from pages. Nodes are web pages.

Usage

```
## S3 method for class 'activity.web'
Create(
  datasource,
  type,
  lcase = TRUE,
  ...,
  writeToFile = FALSE,
  verbose = TRUE
)
```

Arguments

datasource	Collected social media data with "datasource" and "web" class names.
type	Character string. Type of network to be created, set to "activity".
lcase	Logical. Convert urls and page names to lowercase.
...	Additional parameters passed to function. Not used in this method.
writeToFile	Logical. Write data to file. Default is FALSE.
verbose	Logical. Output additional information. Default is TRUE.

Value

Network as a named list of two dataframes containing $nodes and $edges.

Examples

```
## Not run:
# create a web activity network graph
net_activity <- data_collect |> Create("activity")

# network
# net_activity$nodes
# net_activity$edges

## End(Not run)
```

Create.activity.youtube

Create YouTube activity network

Description

Creates an activity network from collected YouTube video comment threads. Nodes are top-level comments, reply comments and videos. Edges are directed between the nodes and represent commenting activity.

Usage

```
## S3 method for class 'activity.youtube'
Create(datasource, type, ..., writeToFile = FALSE, verbose = TRUE)
```

Arguments

datasource	Collected social media data with "datasource" and "youtube" class names.
type	Character string. Type of network to be created, set to "activity".
...	Additional parameters passed to function. Not used in this method.
writeToFile	Logical. Write data to file. Default is FALSE.
verbose	Logical. Output additional information. Default is TRUE.

Value

Network as a named list of two dataframes containing `$nodes` and `$edges`.

Examples

```
## Not run:
# create a YouTube activity network graph
activityNetwork <- youtubeData |> Create("activity")

# network
# activityNetwork$nodes
# activityNetwork$edges

## End(Not run)
```

Create.actor.mastodon *Create mastodon actor network*

Description

Creates a mastodon actor network from posts. Mastodon users who have posted are actor nodes. The created network is directed with edges representing replies.

Usage

```
## S3 method for class 'actor.mastodon'
Create(
  datasource,
  type,
  subtype = NULL,
  inclMentions = TRUE,
  ...,
  writeToFile = FALSE,
  verbose = TRUE
)
```

Arguments

`datasource`	Collected social media data with `"datasource"` and `"mastodon"` class names.
`type`	Character string. Type of network to be created, set to `"actor"`.
`subtype`	Character string. Subtype of actor network to be created. Can be set to `"server"`. Default is `NULL`.
`inclMentions`	Logical. Create edges for users mentioned or tagged in posts. Default is `TRUE`.
`...`	Additional parameters passed to function. Not used in this method.
`writeToFile`	Logical. Write data to file. Default is `FALSE`.
`verbose`	Logical. Output additional information. Default is `TRUE`.

Value

Network as a named list of two dataframes containing $nodes and $edges.

Examples

```
## Not run:
# create a mastodon actor network
actor_net <- mastodon_data |> Create("actor")

# create a mastodon server relations network
actor_net <- mastodon_data |> Create("actor", "server")

## End(Not run)
```

Create.actor.reddit *Create reddit actor network*

Description

Creates a reddit actor network from thread comments on subreddits. Users who have commented on a thread are actor nodes and comment replies to each other are represented as directed edges.

Usage

```
## S3 method for class 'actor.reddit'
Create(datasource, type, ..., writeToFile = FALSE, verbose = TRUE)
```

Arguments

datasource	Collected social media data with "datasource" and "reddit" class names.
type	Character string. Type of network to be created, set to "actor".
...	Additional parameters passed to function. Not used in this method.
writeToFile	Logical. Write data to file. Default is FALSE.
verbose	Logical. Output additional information. Default is TRUE.

Value

Network as a named list of two dataframes containing $nodes and $edges.

Examples

```
## Not run:
# create a reddit actor network graph with comment text as edge attributes
actorNetwork <- redditData |> Create("actor")

# network
# actorNetwork$nodes
# actorNetwork$edges

## End(Not run)
```

`Create.actor.web` *Create web actor network*

Description

Creates a web page domain network from pages. Nodes are site domains.

Usage

```
## S3 method for class 'actor.web'
Create(datasource, type, ..., writeToFile = FALSE, verbose = TRUE)
```

Arguments

`datasource`	Collected social media data with `"datasource"` and `"web"` class names.
`type`	Character string. Type of network to be created, set to `"activity"`.
`...`	Additional parameters passed to function. Not used in this method.
`writeToFile`	Logical. Write data to file. Default is `FALSE`.
`verbose`	Logical. Output additional information. Default is `TRUE`.

Value

Network as a named list of two dataframes containing `$nodes` and `$edges`.

Examples

```
## Not run:
# create a web actor network graph
net_activity <- data_collect |> Create("actor")

# network
# net_activity$nodes
# net_activity$edges

## End(Not run)
```

`Create.actor.youtube` *Create YouTube actor network*

Description

Creates a YouTube actor network from comment threads on YouTube videos. Users who have made comments to a video (top-level comments) and users who have replied to those comments are actor nodes. The comments are represented as directed edges between the actors. The video id is also included as an actor node, representative of the videos publisher with top-level comments as directed edges towards them.

Usage

```
## S3 method for class 'actor.youtube'
Create(datasource, type, ..., writeToFile = FALSE, verbose = TRUE)
```

Arguments

`datasource`	Collected social media data with `"datasource"` and `"youtube"` class names.
`type`	Character string. Type of network to be created, set to `"actor"`.
`...`	Additional parameters passed to function. Not used in this method.
`writeToFile`	Logical. Write data to file. Default is `FALSE`.
`verbose`	Logical. Output additional information. Default is `TRUE`.

Value

Network as a named list of two dataframes containing `$nodes` and `$edges`.

Examples

```
## Not run:
# create a YouTube actor network graph
actorNetwork <- youtubeData |> Create("actor")

# network
# actorNetwork$nodes
# actorNetwork$edges

## End(Not run)
```

Graph *Create an igraph graph from network*

Description

Create an igraph graph from network

Usage

```
Graph(net, directed = TRUE, ..., writeToFile = FALSE, verbose = TRUE)
```

Arguments

net	A named list of dataframes nodes and edges generated by Create.
directed	Logical. Create a directed graph. Default is TRUE.
...	Additional parameters passed to function. Not used in this method.
writeToFile	Logical. Save graph to a file in the current working directory. Default is FALSE.
verbose	Logical. Output additional information. Default is TRUE.

Value

An igraph object.

ImportRtoot *Import rtoot collected data*

Description

Imports **rtoot** collected data from rda or rds saved object file or from an rtoot dataframe. Ensures datasource and specified socialmedia type are set so data is usable by Create functions. Not required if collected data was collected by vosonSML and saved as an rds file, use readRDS instead.

Usage

```
ImportRtoot(data)

import_rtoot(data)
```

Arguments

data	Character string or dataframe. File path to or tibble of collected data from **rtoot**.

Value

A dataframe suitable for input into mastodon network Create functions.

Note

Only supports **rtoot** data collected using the get_timeline_hashtag, get_timeline_public, get_status, get_context functions.

Examples

```
## Not run:
# import rtoot collected data from dataframe
collect_mast <- ImportRtoot(rtoot_data)

# import rtoot collected data from file
collect_mast <- ImportRtoot("./rtoot_search_n100.rds")

## End(Not run)
```

Merge *Merge collected data*

Description

Merge collected data

Usage

```
Merge(..., unique = TRUE, rev = TRUE, writeToFile = FALSE, verbose = TRUE)

merge_data(..., unique = TRUE, rev = TRUE, writeToFile = FALSE, verbose = TRUE)
```

Arguments

...	Collect data to merge.
unique	Logical. Remove duplicates based on observation id. Default is TRUE.
rev	Logical. Reverses order of observations before removing duplicates. If collect data is provided chronologically then this should ensure the most recent copy of a duplicate is kept. Default is TRUE.
writeToFile	Logical. Save data to a file in the current working directory. Default is FALSE.
verbose	Logical. Output additional information. Default is TRUE.

Value

A merged Collect object.

`MergeFiles` *Merge collected data files*

Description

Merge collected data files

Usage

```
MergeFiles(
  path = ".",
  pattern = "(?-i).+?\\.rds$",
  unique = TRUE,
  rev = TRUE,
  writeToFile = FALSE,
  verbose = TRUE
)

merge_files(
  path = ".",
  pattern = "(?-i).+?\\.rds$",
  unique = TRUE,
  rev = TRUE,
  writeToFile = FALSE,
  verbose = TRUE
)
```

Arguments

`path`	Directory path of Collect data to merge. Default is the working directory.
`pattern`	Regular expression (regex) for matching file names to merge.
`unique`	Logical. Remove duplicates based on observation id. Default is `TRUE`.
`rev`	Logical. Reverses order of observations before removing duplicates. If collect data is provided chronologically then this should ensure the most recent copy of a duplicate is kept. Default is `TRUE`.
`writeToFile`	Logical. Save data to a file in the current working directory. Default is `FALSE`.
`verbose`	Logical. Output additional information. Default is `TRUE`.

Value

A merged Collect object.

Index