Package 'vosonSML'

July 18, 2019

Version 0.27.2

Title Collecting Social Media Data and Generating Networks for Analysis

Description A suite of tools for collecting and constructing networks from social media data. Provides easy-to-use functions for collecting data across popular platforms (Twitter, YouTube and Reddit) and generating different types of networks for analysis.

Type Package

Imports tm, stringr, RCurl, igraph (>= 1.2.2), Hmisc, data.table, httpuv, methods, httr, magrittr, dplyr (>= 0.7.8), rlang (>= 0.3.0.1), RedditExtractoR (>= 2.1.2), rtweet (>= 0.6.8), textutils, tictoc

Depends R (>= 3.2.0)

Encoding UTF-8

Author Timothy Graham, Robert Ackland, Chung-hong Chan, Bryan Gertzel

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

License GPL (>= 3)

RoxygenNote 6.1.1

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2019-07-18 08:42:05 UTC

R topics documented:

`vosonSML-package` *Collection and network analysis of social media data*

Description

The goal of the **vosonSML** package is to provide a suite of easy-to-use tools for collecting data from social media and generating different types of networks suited to Social Network Analysis (SNA) and text analytics. It offers tools to create unimodal, multimodal, and semantic networks. Excellent packages such as **rtweet**, **RedditExtractoR**, **magrittr**, **dplyr** and **igraph** were drawn on to provide an integrated work flow for creating different types of networks out of social media data. Creating networks from online social media is often non-trivial and time consuming. This package simplifies such tasks so users can focus on analysis.

vosonSML uses a straightforward S3 class system. Data collected with this package produces `data.frame` inheritable objects that are assigned the class `"datasource"`. Additionally, `"datasource"` objects are attributed a class identifying the source of data, such as `"twitter"` or `"youtube"`. In this way `datasource` objects are fast, easy to work with, and can be used as input to easily construct different kinds of networks. For example, the function `Collect` can be used to collect twitter data, which is then passed to the `Create` function resulting in a twitter network (as igraph object) that is ready for analysis.

Author(s)

Created by Timothy Graham and Robert Ackland with major contributions by Chung-hong Chan. The current lead developer and maintainer is Bryan Gertzel.

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.twitter`, `Authenticate.youtube` and `Authenticate.reddit` for parameters and usage.

Usage

```
Authenticate(socialmedia, ...)
```

Arguments

`socialmedia`	Character string. Identifier for social media API to authenticate with. Supported social media are `"twitter"`, `"youtube"` and `"reddit"`.
`...`	Optional parameters to pass to functions providied by supporting R packages that are used for social media API access.

Authenticate.reddit *Reddit API authentication*

Description

Reddit does not require authentication in this version of vosonSML.

Usage

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

Arguments

`socialmedia`	Character string. Identifier for social media API to authenticate, set to `"reddit"`.
`...`	Additional parameters passed to function. Not used in this method.

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.twitter` *Twitter API authentication*

Description

Twitter authentication uses OAuth and either requires authorization of the rtweet package rstats2twitter client app by a registered twitter user or twitter app developer API keys as described here: `https://developer.twitter.com/en/docs/basics/authentication/overview/oauth`.

Usage

```
## S3 method for class 'twitter'
Authenticate(socialmedia, appName, apiKey, apiSecret,
  accessToken, accessTokenSecret, ...)
```

Arguments

`socialmedia`	Character string. Identifier for social media API to authenticate, set to `"twitter"`.
`appName`	Character string. Registered twitter app name associated with the API keys.
`apiKey`	Character string. API consumer key to authenticate.
`apiSecret`	Character string. API consumer secret to authenticate.
`accessToken`	Character string. API access token to authenticate.
`accessTokenSecret`	Character string. API access token secret to authenticate.
`...`	Additional parameters passed to function. Not used in this method.

Value

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

Examples

```
## Not run:
# twitter authentication via user authorization of app on their account
# will open a web browser to twitter prompting the user to log in and authorize the app
# apiKey and apiSecret are equivalent to a twitter apps consumer key and secret
twitterAuth <- Authenticate("twitter", appName = "An App",
  apiKey = "xxxxxxxxxxxx", apiSecret = "xxxxxxxxxxxx"
)

# twitter authentication with developer app api keys
myDevKeys <- list(appName = "My App", apiKey = "xxxxxxxxxxxx",
  apiSecret = "xxxxxxxxxxxx", accessToken = "xxxxxxxxxxxx",
  accessTokenSecret = "xxxxxxxxxxxx")

twitterAuth <- Authenticate("twitter", appName = myDevKeys$appName,
  apiKey = myDevKeys$apiKey, apiSecret = myDevKeys$apiSecret, accessToken = myDevKeys$access
  accessTokenSecret = myDevKeys$accessTokenSecret)

## 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, ...)
```

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.

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.

Refer to `Collect.twitter`, `Collect.youtube` and `Collect.reddit` for parameters and usage.

Usage

```
Collect(credential, ...)
```

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.

`Collect.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 'reddit'
Collect(credential, threadUrls, waitTime = 5,
  writeToFile = FALSE, ...)
```

Arguments

credential	A credential object generated from Authenticate with class name "reddit".
threadUrls	Character vector. Reddit thread urls to collect data from.
waitTime	Numeric integer. Time in seconds to wait in-between url collection requests.
writeToFile	Logical. Write collected data to file. Default is FALSE.
...	Additional parameters passed to function. Not used in this method.

Value

A data.frame 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, waitTime = 3, writeToFile = TRUE)

## End(Not run)
```

Collect.twitter *Collect tweet data from twitter search*

Description

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

The twitter Standard search API sets a rate limit of 180 requests every 15 minutes. A maximum of 100 tweets can be collected per search request meaning the maximum number of tweets per operation is 18000 / 15 minutes. More tweets can be collected by using retryOnRateLimit = TRUE parameter which will cause the collection to pause if the rate limit is reached and resume when the rate limit resets (in approximately 15 minutes). Alternatively the twitter API parameter since_id can be used in a later session to resume a twitter search collection from the last tweet previously collected as tweet status id's are sequential. The Standard API only returns tweets for the last 7 days.

All of the search query operators available through the twitter API can be used in the searchTerm field. For example, to search for tweets containing the term "love" or "hate" the "OR" operator can be used in the term field: searchTerm = "love OR hate". For more information refer to the twitter API documentation for query operators: https://developer.twitter.com/en/docs/tweets/search/guides/standard-operators.

Usage

```
## S3 method for class 'twitter'
Collect(credential, searchTerm = "",
  searchType = "recent", numTweets = 100, includeRetweets = TRUE,
  retryOnRateLimit = FALSE, writeToFile = FALSE, verbose = FALSE,
  ...)
```

Arguments

credential	A credential object generated from Authenticate with class name "twitter".
searchTerm	Character string. Specifies a twitter search term. For example, "Australian politics" or the hashtag "#auspol".
searchType	Character string. Returns filtered tweets as per search type recent, mixed or popular. Default type is recent.
numTweets	Numeric. Specifies how many tweets to be collected. Defaults is 100.
includeRetweets	Logical. Specifies if the search should filter out retweets. Defaults is TRUE.
retryOnRateLimit	Logical. Default is FALSE.
writeToFile	Logical. Write collected data to file. Default is FALSE.
verbose	Logical. Output additional information about the data collection. Default is FALSE.
...	Arguments passed on to rtweet::search_tweets

geocode Geographical limiter of the template "latitude,longitude,radius" e.g., geocode = "37.78,-122.40,1mi".

max_id Character, returns results with an ID less than (that is, older than) or equal to 'max_id'. Especially useful for large data returns that require multiple iterations interrupted by user time constraints. For searches exceeding 18,000 tweets, users are encouraged to take advantage of rtweet's internal automation procedures for waiting on rate limits by setting retryonratelimit argument to TRUE. It some cases, it is possible that due to processing time and rate limits, retrieving several million tweets can take several hours or even multiple days. In these cases, it would likely be useful to leverage retryonratelimit for sets of tweets and max_id to allow results to continue where previous efforts left off.

parse Logical, indicating whether to return parsed data.frame, if true, or nested list, if false. By default, parse = TRUE saves users from the wreck of time and frustration associated with disentangling the nasty nested list returned from Twitter's API. As Twitter's APIs are subject to change, this argument would be especially useful when changes to Twitter's APIs affect performance of internal parsers. Setting parse = FALSE also ensures the maximum amount of possible information is returned. By default, the rtweet parse process returns nearly all bits of information returned from Twitter. However, users may occasionally encounter new or omitted variables. In these rare cases, the nested list object will be the only way to access these variables.

Value

A data.frame object with class names "datasource" and "twitter".

Note

Additional parameters passed to this function in the ellipsis ... will also be passed to the Twitter search API request. Most parameters have been covered but a complete list can be found here: https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets A useful additional parameter is language allowing the user can restrict tweets returned to a particular language using an ISO 639-1 code. For example, to restrict a search to tweets in English the value language = "en" can be passed to this function.

Examples

```
## Not run:
# search and collect 100 recent tweets for the hashtag #auspol
myTwitterData <- twitterAuth %>%
  Collect(searchTerm = "#auspol", searchType = "recent", numTweets = 100, verbose = TRUE,
          includeRetweets = FALSE, retryOnRateLimit = TRUE, 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, verbose = FALSE,
  writeToFile = FALSE, maxComments = 1e+13, ...)
```

Arguments

credential	A credential object generated from Authenticate with class name "youtube".
videoIDs	Character vector. Specifies one or more youtube video IDs. For example, if the video URL is https://www.youtube.com/watch?v=xxxxxxxxxxx then use videoIDs = c("xxxxxxxxxxx").
verbose	Logical. Output additional information about the data collection. Default is FALSE.
writeToFile	Logical. Write collected data to file. Default is FALSE.
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.

Value

A data.frame 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.

To help extract video ids for videos the function GetYoutubeVideoIDs can be used. It accepts input of a vector or file containing video urls and creates a chracter vector suitable as input for the videoIDs parameter.

Examples

```
## Not run:
# create a list of youtube video ids to collect on
videoIDs <- GetYoutubeVideoIDs(c("https://www.youtube.com/watch?v=xxxxxxxx",
                                 "https://youtu.be/xxxxxxxx"))

# collect approximately 200 threads/comments for each youtube video
youtubeData <- youtubeAuth %>%
  Collect(videoIDs = videoIDs, writeToFile = TRUE, verbose = FALSE, 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.

There are three types of networks that can be created from collected data: `actor`, `bimodal` or `semantic`.

For `actor` networks refer to `Create.actor.twitter`, `Create.actor.youtube` and `Create.actor.reddit` for parameters and usage.

For `bimodal` and `semantic` networks refer to `Create.bimodal.twitter` and `Create.semantic.twitter` functions for parameters and usage respectively.

Usage

```
Create(datasource, type, ...)
```

Arguments

`datasource`	Collected social media data of class `"datasource"` and `socialmedia`.
`type`	Character string. Type of network to be created, can be `"actor"`, `"bimodal"` or `"semantic"`.
`...`	Optional parameters to pass to functions providied by supporting R packages that are used for social media network creation.

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. Optionally a graph with edge weights (`weight`) or edge text attributes (`vosonTxt_comment`) can be created.

Usage

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

Arguments

datasource	Collected social media data with "datasource" and "reddit" class names.
type	Character string. Type of network to be created, set to "actor".
weightEdges	Logical. Combine and weight directed network edges. Default is FALSE.
textData	Logical. Include comment text as an edge attributes of returned igraph network. Is set to FALSE if the weightEdges parameter is TRUE as text merging is not supported. Default is FALSE.
cleanText	Logical. Simple removal of non-alphanumeric, non-punctuation, and non-space characters from the comment text data applied as graph edge attribute. Not suitable in some cases requiring capture of utf or emoji characters. Implemented to support basic text analysis and to prevent reddit specific XML control character errors in the graphml files created by this function. Alternatively custom cleaning of text data can be performed on the datasource dataframe before being passed to this function. Default is TRUE.
writeToFile	Logical. Save network data to a file in the current working directory. Default is FALSE.
...	Additional parameters passed to function. Not used in this method.

Value

Named list containing generated network as igraph object $graph.

Examples

```
## Not run:
# create a reddit actor network graph with comment text as edge attributes
actorNetwork <- redditData %>%
  Create("actor", includeTextData = TRUE, writeToFile = TRUE)

# igraph object
# actorNetwork$graph

## End(Not run)
```

Create.actor.twitter

Create twitter actor network

Description

Creates a twitter actor network from tweets returned from the twitter search query. Twitter users who have tweeted / retweeted or been mentioned in a tweet are actor nodes. The created network is directed with edges of different types representing retweets, quote tweets, mentions and replies to other users. Users who have tweeted without relations to other users will appear in the network graph as isolate nodes.

Usage

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

Arguments

datasource	Collected social media data with "datasource" and "twitter" class names.
type	Character string. Type of network to be created, set to "actor".
writeToFile	Logical. Save network data to a file in the current working directory. Default is FALSE.
verbose	Logical. Output additional information about the network creation. Default is FALSE.
...	Additional parameters passed to function. Not used in this method.

Value

Named list containing dataframes with the relations between actors (directed edges) $relations, the actors (including isolates) $users and generated actor network as igraph object $graph.

Note

When creating twitter actor networks, a network with additional user information can be generated using the AddTwitterUserData function. Additional calls can be made to the twitter API to get information about users that were identified as nodes during network creation but did not tweet (meaning no user profile information was initially collected for them).

Examples

```
## Not run:
# create a twitter actor network graph and output to console additional information
# during network creation (verbose)
actorNetwork <- twitterData %>% Create("actor", writeToFile = TRUE, verbose = TRUE)

# igraph object
# actorNetwork$graph

## 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,
  ...)
```

Arguments

`datasource`	Collected social media data with `"datasource"` and `"youtube"` class names.
`type`	Character string. Type of network to be created, set to `"actor"`.
`writeToFile`	Logical. Save network data to a file in the current working directory. Default is `FALSE`.
`...`	Additional parameters passed to function. Not used in this method.

Value

Named list containing generated network as igraph object `$graph`.

Examples

```
## Not run:
# create a youtube actor network graph
actorNetwork <- youtubeData %>% Create("actor", writeToFile = TRUE)

# igraph object
# actorNetwork$graph

## End(Not run)
```

`Create.bimodal.twitter` *Create twitter bimodal network*

Description

Creates a bimodal network from tweets returned from the twitter search query. In this network there are two types of nodes, twitter users who have tweeted (actors) and hashtags found within their tweets. Network edges are weighted and represent hashtag usage by the actor or specifically their tweets that contain a hashtag matching the name of the node they are directed towards.

Usage

```
## S3 method for class 'bimodal.twitter'
Create(datasource, type,
  removeTermsOrHashtags = NULL, writeToFile = FALSE, verbose = FALSE,
  ...)
```

Arguments

`datasource`	Collected social media data with `"datasource"` and `"twitter"` class names.
`type`	Character string. Type of network to be created, set to `"bimodal"`.
`removeTermsOrHashtags`	Character vector. Terms or hashtags to remove from the bimodal network. For example, this parameter could be used to remove the search term or hashtag that was used to collect the data by removing any nodes with matching name. Default is `NULL` to remove none.
`writeToFile`	Logical. Save network data to a file in the current working directory. Default is `FALSE`.
`verbose`	Logical. Output additional information about the network creation. Default is `FALSE`.
`...`	Additional parameters passed to function. Not used in this method.

Value

Named list containing bimodal network as igraph object `$graph`.

Examples

```
## Not run:
# create a twitter bimodal network graph removing the hashtag '#auspol' as it was used in
# the twitter search query
bimodalNetwork <- twitterData %>%
                  Create("bimodal", removeTermsOrHashtags = c("#auspol"), writeToFile = TRUE
                         verbose = TRUE)

# igraph object
```

```
# bimodalNetwork$graph

## End(Not run)
```

`Create.semantic.twitter` *Create twitter semantic network*

Description

Creates a semantic network from tweets returned from the twitter search query. Semantic networks describe the semantic relationships between concepts. In this network the concepts are significant words and terms (hashtags) extracted from the text corpus of the tweet data, and actors represented as nodes. Network edges are weighted and represent usage of frequently occurring terms and hashtags by the actors.

Usage

```
## S3 method for class 'semantic.twitter'
Create(datasource, type,
  removeTermsOrHashtags = NULL, stopwordsEnglish = TRUE,
  termFreq = 5, hashtagFreq = 50, writeToFile = FALSE,
  verbose = FALSE, ...)
```

Arguments

`datasource`	Collected social media data with `"datasource"` and `"twitter"` class names.
`type`	Character string. Type of network to be created, set to `"semantic"`.
`removeTermsOrHashtags`	Character vector. Terms or hashtags to remove from the semantic network. For example, this parameter could be used to remove the search term or hashtag that was used to collect the data by removing any nodes with matching name. Default is `NULL` to remove none.
`stopwordsEnglish`	Logical. Removes English stopwords from the tweet data. Default is `TRUE`.
`termFreq`	Numeric integer. Specifies the percentage of most frequent terms to include. For example, a `termFreq = 20` means that the 20 percent most frequently occurring `terms` will be included in the semantic network as nodes. A larger percentage will increase the number of nodes and therefore the size of graph. The default value is `5`, meaning the top 5 percent most frequent terms are used.
`hashtagFreq`	Numeric integer. Specifies the percentage of most frequent `hashtags` to include. For example, a `hashtagFreq = 80` means that the 80 percent most frequently occurring hashtags will be included in the semantic network as nodes. The default value is `50`.

writeToFile	Logical. Save network data to a file in the current working directory. Default is FALSE.
verbose	Logical. Output additional information about the network creation. Default is FALSE.
...	Additional parameters passed to function. Not used in this method.

Value

Named list containing semantic network as igraph object $graph.

Examples

```
## Not run:
# create a twitter semantic network graph removing the hashtag '#auspol' and using the
# top 2% frequently occurring terms and 10% frequently occurring hashtags as additional
# concepts or nodes
semanticNetwork <- twitterData %>%
                   Create("semantic", removeTermsOrHashtags = c("#auspol"), termFreq = 2,
                          hashtagFreq = 10, writeToFile = TRUE, verbose = TRUE)

# igraph object
# semanticNetwork$graph

## End(Not run)
```

---

vosonSML::AddTwitterUserData

Enhances twitter actor network graph by adding user attributes to nodes

---

Description

Creates a network from the relations and users dataframes generated by Create. Network is supplemented with additional downloaded twitter user information applied as node attributes.

Usage

```
AddTwitterUserData(collectData, networkData, lookupUsers = TRUE,
  twitterAuth = NULL, writeToFile = FALSE)
```

Arguments

collectData	A dataframe containing the collected tweet data from the Collect function.
networkData	A named list containing the relations $relations and users $users data returned from the Create actor network function.

lookupUsers	Logical. Lookup user profile information using the twitter API for any users data missing from the collect data set. For example fetches profile information for users that became nodes during network creation because they were mentioned in a tweet but did not author any tweets themselves. Default is TRUE.
twitterAuth	A twitter authentication object from Authenticate.
writeToFile	Logical. If TRUE a data frame of user information and the resulting network graph will be written to file in rds and graphml formats respectively. Default is FALSE.

Value

A named list containing a dataframe with user information $users and an igraph object of the twitter actor network with supplemental user node attributes $graph.

Note

Only supports twitter actor network at this time. Bimodal network support could be achieved by the filtering of the twitter user ids from nodes of other types in the networkData. Refer to S3 methods Authenticate.twitter, Collect.twitter and Create.actor.twitter to first create twitter actor network and data to pass as input into this function.

Examples

```
## Not run:
# add additional twitter user profile information to actor network graph as node attributes
# requires twitterAuth from Authenticate, twitterData from Collect and actorNetwork from
# Create actor network
actorNetWithUserAttr <- AddTwitterUserData(twitterData, actorNetwork,
                                           lookupUsers = TRUE,
                                           twitterAuth = twitterAuth, writeToFile = TRUE)
# igraph object
# actorNetWithUserAttr$graph

## End(Not run)
```

---

vosonSML::GetYoutubeVideoIDs

Extract the ids from a list of youtube video URLs

Description

This function reads youtube video urls from a list and or a text file and converts them to a vector of video ids. For example, URL https://www.youtube.com/watch?v=73I5dRucCds returns the id 73I5dRucCds. This function can be used to create a vector for the youtube Collect.youtube functions videoIDs parameter.

Usage

```
GetYoutubeVideoIDs(urls = NULL, file = NULL)
```

Arguments

urls	Character vector. List of youtube URLs.
file	Character string. Text file containing youtube URLs.

Value

A vector of youtube video ids as character strings that were extracted from input video urls.

Note

Accepts youtube URL formats https://youtu.be/xxxxxxxx and https://www.youtube.com/watch?v=xx

---

vosonSML::ImportData

Import collected data previously saved to file

Description

Imports collected data from file into a dataframe of class datasource and specified socialmedia type that is usable by Create functions.

Usage

```
ImportData(file, type = "csv", socialmedia)
```

Arguments

file	Character string. Collected data file path.
type	Character string. Type of file or file format of file to import csv or rds. Default is csv.
socialmedia	Character string. Social media type of collected data twitter, youtube or reddit.

Value

A dataframe with datasource class attributes.