Package 'vosonSML'

July 18, 2020

Version 0.29.13

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 data.table, dplyr (>= 0.7.8), Hmisc, httpuv, httr, igraph (>=
1.2.2), jsonlite, lubridate, magrittr, methods, purrr, RCurl,
rlang (>= 0.3.0.1), rtweet (>= 0.6.8), stats, stringr,
textutils, tibble, utils

Suggests knitr, rmarkdown, stopwords, testthat, tictoc, tidyr,
tidytext

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 7.1.1

NeedsCompilation no

VignetteBuilder knitr

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

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

Repository CRAN

Date/Publication 2020-07-18 05:00:03 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 provided 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 typically requires four developer API keys generated when you create a twitter app via the twitter developer web site.

There is another method available commonly used by third-party apps in which an app can be authorized by a user to use the twitter API on their behalf. The implementation of this method in vosonSML does not require a developer account but does still require the user to have access to a developers apps two consumer API keys. This allows multiple users to access the twitter API with vosonSML via a single developer account and app.

The twitter OAuth process is 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 using 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$apiSecret, accessToken = myDevKeys$accessToken,
  accessTokenSecret = myDevKeys$accessTokenSecret)

# twitter authentication via authorization of an app to their user account
# requires the apps consumer API keys
# apiKey and apiSecret parameters are equivalent to the apps consumer key and secret
# will open a web browser to twitter prompting the user to log in and authorize the app
twitterAuth <- Authenticate("twitter", appName = "An App",
  apiKey = "xxxxxxxxxxxx", apiSecret = "xxxxxxxxxxxx"
)

## 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 <- "xxxxxxxxxxxx"

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 = c(3, 10),
  ua = getOption("HTTPUserAgent"),
  writeToFile = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

credential	A credential object generated from Authenticate with class name "reddit".
threadUrls	Character vector. Reddit thread urls to collect data from.
waitTime	Numeric vector. Time range in seconds to select random wait from in-between url collection requests. Minimum is 3 seconds. Default is c(3,10) for a wait time chosen from between 3 and 10 seconds.
ua	Character string. Override User-Agent string to use in Reddit thread requests. Default is option("HTTPUserAgent") value as set by vosonSML.
writeToFile	Logical. Write collected data to file. Default is FALSE.
verbose	Logical. Output additional information about the data collection. Default is TRUE.
...	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, 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+10,
    ...
)
```

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 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.

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 four types of networks that can be created from collected data: activity, actor, twomode or semantic.

For activity networks refer to [Create.activity.twitter](), [Create.activity.youtube]() and [Create.activity.reddit]() for parameters and usage.

For actor networks refer to [Create.actor.twitter](), [Create.actor.youtube]() and [Create.actor.reddit]().

For twomode and semantic networks refer to [Create.twomode.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 "activity", "actor", "twomode" or "semantic".
...	Optional parameters to pass to functions providied by supporting R packages that are used for social media network creation.

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, 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″.
verbose	Logical. Output additional information about the network creation. Default is TRUE.
...	Additional parameters passed to function. Not used in this method.

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.twitter
```
                    Create twitter activity network

Description

Creates a twitter activity network from collected tweets. Nodes are tweets and directed edges represent the relationship of tweets to one another. For example, there is a directed edge from a quote tweet towards the tweet that was quoted. Stand-alone tweets that are not replies, retweets or quote tweets have no relation to others and will be isolates.

Usage

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

Arguments

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

Value

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

Examples

```
## Not run:
# create a twitter activity network graph
activityNetwork <- twitterData %>% Create("activity")

# network
# activityNetwork$nodes
# activityNetwork$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, 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".
verbose	Logical. Output additional information about the network creation. Default is TRUE.
...	Additional parameters passed to function. Not used in this method.

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.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, ...)
```

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.

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.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, verbose = TRUE, ...)
```

Arguments

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

Value

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

Note

When creating twitter actor networks, a network with additional user information can be generated using the [AddUserData](#) 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")

# network
# actorNetwork$nodes
```

```
# actorNetwork$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, ...)
```

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.

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)
```

`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 hashtags extracted from the tweet text. Network edges are weighted and represent occurrence of words and hashtags in the same tweets.

The creation of twitter semantic networks requires text processing and the tokenization of tweets. As such this function requires the additional installation of the **tidyr** and **tidytext** packages to achieve this.

Usage

```
## S3 method for class 'semantic.twitter'
Create(
  datasource,
  type,
  removeTermsOrHashtags = NULL,
  stopwords = TRUE,
  stopwordsLang = "en",
  stopwordsSrc = "smart",
  removeNumbers = TRUE,
  removeUrls = TRUE,
  termFreq = 5,
  hashtagFreq = 50,
  assoc = "limited",
  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. Words 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.
`stopwords`	Logical. Removes stopwords from the tweet data. Default is `TRUE`.
`stopwordsLang`	Character string. Language of stopwords to use. Refer to the **stopwords** package for further information on supported languages. Default is `"en"`.

stopwordsSrc	Character string. Source of stopwords list. Refer to the **stopwords** package for further information on supported sources. Default is `"smart"`.
removeNumbers	Logical. Removes whole numerical tokens from the tweet text. For example, a year value such as `2020` will be removed but not mixed values such as `G20`. Default is TRUE.
removeUrls	Logical. Removes twitter shortened URL tokens from the tweet text. Default is TRUE.
termFreq	Numeric integer. Specifies the percentage of most frequent words to include. For example, `termFreq = 20` means that the 20 percent most frequently occurring `words` 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 words are used.
hashtagFreq	Numeric integer. Specifies the percentage of most frequent `hashtags` to include. For example, `hashtagFreq = 20` means that the 20 percent most frequently occurring hashtags will be included in the semantic network as nodes. The default value is 50.
assoc	Character string. Association of nodes. A value of `"limited"` includes only edges between most frequently occurring hashtags and terms. A value of `"full"` includes ties between most frequently occurring hashtags and terms, hashtags and hashtags, and terms and terms. Default is `"limited"`.
verbose	Logical. Output additional information about the network creation. Default is FALSE.
...	Additional parameters passed to function. Not used in this method.

Value

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

Note

The words and hashtags passed to the function in the `removeTermsOrHashtags` parameter are removed before word frequencies are calculated and are therefore excluded from top percentage of most frequent terms completely rather than simply filtered out of the final network.

The top percentage of frequently occurring hashtags `hashtagFreq` and words `termFreq` are calculated to a minimum frequency and all terms that have an equal or greater frequency than the minimum are included in the network as nodes. For example, of unique hashtags of varying frequencies in a dataset the top 50 frequency or most common hashtags may calculate to being the first 20 hashtags. The frequency of the 20th hashtag is then used as the minimum and all hashtags of equal or greater frequency are included as part of the top 50 most frequently occurring hashtags. So the number of top hashtags may end up being greater than 20 if there is more than one hashtag that has frequency matching the minimum. The exception to this is if the minimum frequency is 1 and the `hashtagFreq` is set to less than 100, in this case only the first 20 hashtags will be included.

Hashtags and words in the top percentages are included in the network as isolates if there are no instances of them occurring in tweet text with other top percentage frequency terms.

Examples

```
## Not run:
# twitter semantic network creation additionally requires the tidyr, tidytext and stopwords packages
# for working with text data
install.packages(c("tidyr", "tidytext", "stopwords"))

# create a twitter semantic network graph removing the hashtag '#auspol' and using the
# top 2% frequently occurring words and 10% most frequently occurring hashtags as nodes
semanticNetwork <- twitterData %>%
                   Create("semantic", removeTermsOrHashtags = c("#auspol"),
                               termFreq = 2, hashtagFreq = 10, verbose = TRUE)

# network
# semanticNetwork$nodes
# semanticNetwork$edges

## End(Not run)
```

Create.twomode.twitter

Create twitter 2-mode network

Description

Creates a 2-mode network from tweets returned from the twitter search query. In this network there are two types of nodes, twitter users who authored or were mentioned in collected tweets and hashtags found within tweets. Network edges represent a users tweets that contain hashtags or mention users screen names.

The creation of twitter 2-mode networks requires text processing and the tokenization of tweets. As such this function requires the additional installation of the **tidytext** package to achieve this.

Usage

```
## S3 method for class 'twomode.twitter'
Create(
  datasource,
  type,
  removeTermsOrHashtags = NULL,
  weighted = TRUE,
  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 "twomode".

removeTermsOrHashtags

> Character vector. Users or hashtags to remove from the twomode network. For example, this parameter could be used to remove the user or hashtag that was used to collect the data by removing any nodes with matching name. Default is NULL to remove none.

weighted
> Logical. Add weights to network edges. If set to FALSE tweet status_id and created_at fields will be preserved for edges in the dataframe. Default is TRUE.

verbose
> Logical. Output additional information about the network creation. Default is FALSE.

...
> Additional parameters passed to function. Not used in this method.

Value

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

Examples

```
## Not run:
# twitter 2-mode network creation additionally requires the tidytext package
# for working with text data
install.packages("tidytext")

# create a twitter 2-mode network graph removing the hashtag '#auspol' as it was used in
# the twitter search query
twomodeNetwork <- twitterData %>%
                  Create("twomode", removeTermsOrHashtags = c("#auspol"), verbose = TRUE)

# network
# twomodeNetwork$nodes
# twomodeNetwork$edges

## End(Not run)
```

---

vosonSML::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, ...)
```

Arguments

net	A named list of dataframes nodes and edges generated by `Create`.
data	A dataframe generated by `Collect`.
...	Additional parameters passed to function.

Value

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

Note

Supports all `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
activityNetwork <- collectData %>% Create("activity") %>% AddText(collectData)

# network
# activityNetwork$nodes
# activityNetwork$edges

## End(Not run)
```

---

vosonSML::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, ...)
```

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.

Value

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

---

vosonSML::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, ...)
```

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.

Value

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

```
vosonSML::AddText.actor.youtube
```
Add columns containing text data to youtube actor network dataframes

---

Description

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 'actor.youtube'
AddText(net, data, repliesFromText = FALSE, atRepliesOnly = TRUE, ...)
```

Arguments

net	A named list of dataframes nodes and edges generated by `Create`.
data	A dataframe generated by `Collect`.
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`.
...	Additional parameters passed to function. Not used in this method.

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
activityNetwork <- collectData %>% Create("activity") %>%
                                   AddText(collectData, repliesFromText = FALSE)

# network
# activityNetwork$nodes
# activityNetwork$edges

## End(Not run)
```

---

vosonSML::AddUserData *Add columns of user information as node attributes to network dataframes*

Description

Network is supplemented with additional downloaded social media user information applied as node attributes.

Usage

```
AddUserData(net, data, ...)
```

Arguments

net	A named list of dataframes nodes and edges generated by Create.
data	A dataframe generated by Collect.
...	Additional parameters passed to function.

Value

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

Note

Only supports twitter actor networks at this time. Refer to AddUserData.actor.twitter.

---

vosonSML::AddUserData.actor.twitter
Supplement twitter actor network by adding user profile attributes to nodes

---

Description

Network is supplemented with additional downloaded user information applied as actor node attributes.

Usage

```
## S3 method for class 'actor.twitter'
AddUserData(
  net,
  data,
  lookupUsers = TRUE,
  twitterAuth = NULL,
  verbose = TRUE,
  ...
)
```

Arguments

`net`	A named list of dataframes nodes and edges generated by `Create`.
`data`	A dataframe generated by `Collect`.
`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`.
`verbose`	Logical. Output additional information about the network creation. Default is `TRUE`.
`...`	Additional parameters passed to function. Not used in this method.

Value

Network as a named list of two dataframes containing $nodes, $edges and $users. Nodes include columns for additional user data.

Note

Using the standard twitter API this function is limited to collecting profiles of 90000 users per 15 mins before hitting the rate limit. It does not wait and retry upon hitting rate limit.

Examples

```
## Not run:
# add user info to a twitter actor network
actorNetwork <- collectData %>%
  Create("actor") %>% AddUserData(collectData, twitterAuth = myTwitterAuth)

# network
# actorNetwork$nodes
# actorNetwork$edges

## End(Not run)
```

```
vosonSML::AddVideoData
```
                    Add columns of video information to network dataframes

---

Description

Network is supplemented with additional downloaded video information.

Usage

```
AddVideoData(net, youtubeAuth = NULL, ...)
```

Arguments

net	A named list of dataframes nodes and edges generated by Create.
youtubeAuth	Youtube Authenticate object.
...	Additional parameters passed to function.

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.

```
vosonSML::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,
  ...
)
```

Arguments

net	A named list of dataframes nodes and edges generated by `Create`.
youtubeAuth	Youtube Authenticate object.
videoIds	List. Video ID's 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.

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)
```

---

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=xxxxxxxx`.

---

vosonSML::Graph   *Create an igraph graph from network*

Description

Create an igraph graph from network

Usage

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

Arguments

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

Value

An igraph object.

---

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(path, socialmedia, type = NULL)
```

Arguments

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

Value

A dataframe with datasource class attributes.

Index