Package 'topics'

November 23, 2024

Type Package

Title Creating and Significance Testing Language Features for Visualisation

Version 0.21.0

Description
Implements differential language analysis with statistical tests and offers various language visualization techniques for n-grams and topics. It also supports the 'text' package. For more information, visit <https://r-topics.org/> and <https://www.r-text.org/>.

License GPL-3

URL https://r-topics.org/

BugReports https://github.com/theharmonylab/topics/issues

Encoding UTF-8

Archs x64

SystemRequirements Python (>= 3.6.0)

LazyData true

BuildVignettes true

Imports textmineR, ggplot2, dplyr, mallet, rJava, ggwordcloud, effsize, purrr, tibble, methods, readr, stopwords, Matrix, ngram, stringr, rlang, tidyr

RoxygenNote 7.3.2

Suggests text, knitr, rmarkdown, testthat, covr

VignetteBuilder knitr

Depends R (>= 4.00)

NeedsCompilation no

Author Leon Ackermann [aut] (<https://orcid.org/0009-0008-8583-8748>), Zhuojun Gu [aut] (<https://orcid.org/0009-0000-1610-4830>), Oscar Kjell [aut, cre] (<https://orcid.org/0000-0002-2728-6278>)

Maintainer Oscar Kjell <oscar.kjell@psy.lu.se>

Repository CRAN

Date/Publication 2024-11-23 15:20:02 UTC

Contents

dep_wor_data 2
topicsDtm 3
topicsDtmEval 4
topicsGrams 5
topicsModel 6
topicsPlot 7
topicsPreds 9
topicsTest 11

Index **13**

`dep_wor_data` *Example data about mental health descirptions .*

Description

Example data about mental health descirptions .

Usage

```
dep_wor_data
```

Format

A data frame with 500 participants and 13 variables:

Depselect Words that respondents have selected from a pre-defined list

Worselect Words that respondents have selected from a pre-defined list

Depword Wrods where respondents describe their experience with depression in life

Worword Words where respondents describe their experience with depression in life

Depphrase phrases where respondents describe their experience with depression in life

Worphrase Phrases where respondents describe their experience with anxiety in life

Deptext Text where respondents describe their experience with depression in life

Wortext Text where respondents describe their experience with anxiety in life

Gender Respondents gender 0=male, 1=female

Age respondents age in years

PHQ9tot total score of the respondents PHQ-9

GAD7tot total score of the respondents GAD-7

Source

<https://osf.io/preprints/psyarxiv/p67db>

topicsDtm *Document Term Matrix*

Description

The function for creating a document term matrix

Usage

```
topicsDtm(
  data,
  ngram_window = c(1, 3),
  stopwords = stopwords::stopwords("en", source = "snowball"),
  removalword = "",
  occ_rate = 0,
  removal_mode = "none",
  removal_rate_most = 0,
  removal_rate_least = 0,
  split = 1,
  seed = 42L,
  save_dir,
  load_dir = NULL,
  threads = 1
)
```

Arguments

data	(list) the list containing the text data with each entry belonging to a unique id
ngram_window	(list) the minimum and maximum n-gram length, e.g. c(1,3)
stopwords	(stopwords) the stopwords to remove, e.g. stopwords::stopwords("en", source = "snowball")
removalword	(string) the word to remove
occ_rate	(integer) the rate of occurence of a word to be removed
removal_mode	(string) the mode of removal -> "none", "frequency", "term" or "percentage", frequency removes all words under a certain frequency or over a certain frequency as indicated by removal_rate_least and removal_rate_most, term removes an absolute amount of terms that are most frequent and least frequent, percentage the amount of terms indicated by removal_rate_least and removal_rate_most relative to the amount of terms in the matrix
removal_rate_most	(integer) the rate of most frequent words to be removed, functionality depends on removal_mode
removal_rate_least	(integer) the rate of least frequent words to be removed, functionality depends on removal_mode

split	(float) the proportion of the data to be used for training
seed	(integer) the random seed for reproducibility
save_dir	(string) the directory to save the results, if NULL, no results are saved.
load_dir	(string) the directory to load from.
threads	(integer) the number of threads to use

Value

the document term matrix

Examples

```
# Create a Dtm and remove the terms that occur less than 4 times and more than 500 times.
save_dir_temp <- tempfile()

dtm <- topicsDtm(data = dep_wor_data$Depphrase,
                 removal_mode = "frequency",
                 removal_rate_least = 4,
                 removal_rate_most = 500,
                 save_dir = save_dir_temp)

# Create Dtm and remove the 5 least and 5 most frequent terms.
dtm <- topicsDtm(data = dep_wor_data$Depphrase,
                 removal_mode = "term",
                 removal_rate_least = 1,
                 removal_rate_most = 1,
                 save_dir = save_dir_temp)

# Create Dtm and remove the 5% least frequent and 1% most frequent terms.
dtm <- topicsDtm(data = dep_wor_data$Depphrase,
                 removal_mode = "percentage",
                 removal_rate_least = 1,
                 removal_rate_most = 1,
                 save_dir = save_dir_temp)

# Load precomputed Dtm from directory
dtm <- topicsDtm(load_dir = save_dir_temp,
                 seed = 42,
                 save_dir = save_dir_temp)
```

---

topicsDtmEval *Summarize and Visualize your Document Term Matrix*

Description

This function creates a frequency table of your DTM and generates up to four plots for visualization

Usage

```
topicsDtmEval(dtm)
```

Arguments

dtm	(R_obj) The document term matrix - output from topicsDtm

Value

A named list containing:

dtm_summary A dataframe of terms and their frequencies.

frequency_plot A bar plot of all term frequencies with example terms.

frequency_plot_30_least A bar plot of the 30 least frequent terms (if numer of terms > 30).

frequency_plot_30_most A bar plot of the 30 most frequent terms (if numer of terms > 30).

historgam_of_frequencies A histogram of term frequencies (this is the same information as in the frequency_plot but presented differently).

---

topicsGrams *N-grams*

Description

The function computes ngrams from a text

Usage

```
topicsGrams(data, n = 2, sep = " ", top_n = NULL, pmi_threshold = 0)
```

Arguments

data	(tibble) The data
n	(integer) The length of ngram
sep	(string) The separator
top_n	(integer) The number of top ngrams to be displayed
pmi_threshold	(integer) The pmi threshold, if it shall not be used set to 0

Value

A list containing tibble of the ngrams with the frequency and probability and a tibble containing the relative frequency of the ngrams for each user

topicsModel *Topic modelling*

Description

The function to create and train and an LDA model.

Usage

```
topicsModel(
  dtm,
  num_topics = 20,
  num_top_words = 10,
  num_iterations = 1000,
  seed = 42,
  save_dir,
  load_dir = NULL
)
```

Arguments

dtm	(R_obj) The document term matrix
num_topics	(integer) The number of topics to be created
num_top_words	(integer) The number of top words to be displayed
num_iterations	(integer) The number of iterations to run the model
seed	(integer) The seed to set for reproducibility
save_dir	(string) The directory to save the model, if NULL, the model will not be saved
load_dir	(string) The directory to load the model from, if NULL, the model will not be loaded

Value

A list of the model, the top terms, the labels, the coherence, and the prevalence

Examples

```
# Create LDA Topic Model
save_dir_temp <- tempfile()
dtm <- topicsDtm(
data = dep_wor_data$Depphrase,
save_dir = save_dir_temp)

model <- topicsModel(
dtm = dtm, # output of topicsDtm()
num_topics = 20,
num_top_words = 10,
```

```
num_iterations = 1000,
seed = 42,
save_dir = save_dir_temp)

# Load precomputed LDA Topic Model
model <- topicsModel(
load_dir = save_dir_temp,
seed = 42,
save_dir = save_dir_temp)
```

---

topicsPlot *Plot word clouds*

Description

This function create word clouds and topic fugures

Usage

```
topicsPlot(
  model = NULL,
  ngrams = NULL,
  test = NULL,
  p_threshold = 0.05,
  color_scheme = "default",
  scale_size = FALSE,
  plot_topics_idx = NULL,
  save_dir,
  figure_format = "svg",
  width = 10,
  height = 8,
  max_size = 10,
  seed = 42,
  scatter_legend_dot_size = 15,
  scatter_legend_bg_dot_size = 9,
  scatter_legend_n = c(1, 1, 1, 1, 0, 1, 1, 1, 1),
  scatter_legend_method = c("mean"),
  scatter_legend_specified_topics = NULL,
  scatter_legend_topic_n = FALSE,
  grid_legend_title = "legend_title",
  grid_legend_title_size = 5,
  grid_legend_title_color = "black",
  grid_legend_x_axes_label = "legend_x_axes_label",
  grid_legend_y_axes_label = "legend_y_axes_label",
  grid_legend_number_color = "black",
  grid_legend_number_size = 5
)
```

Arguments

model	(list) A trained topics model. For examples from topicsModel(). Should be NULL if plotting ngrams.
ngrams	(list) The output from the the topicsGram() function . Should be NULL if plotting topics.
test	(list) The test results; if plotting according to dimension(s) include the object from topicsTest() function.
p_threshold	(integer) The p-value threshold to use for significance testing.
color_scheme	(string 'default' or vector) The color scheme. For plots not including a test, the color_scheme should in clude 2 colours (1 gradient pair), such as: c("lightgray", "darkblue) For 1 dimensional plots of n-grams it should contain 4 colours (2 gradient pairs), such as: c("#EAEAEA", "darkred", # negative ngrams colors "#EAEAEA", "darkgreen" # positve ngrams colors) For 1-dimension plots of topics, it should contain 6 colours (3 gradient pairs), such as c("#EAEAEA", "darkred", # negative topics colors "#EAEAEA", "darkgray", # colours of topics not significantly associated "#EAEAEA", "darkgreen" # positve topics colors) For 2-dimensional plots of topics, the color scheme should contain 18 colours (9 gradient pairs), such as: c("lightgray", "#398CF9", # quadrant 1 (upper left corner) "lightgray", "#60A1F7", # quadrant 2 "lightgray", "#5dc688", # quadrant 3 (upper right corner) "lightgray", "#e07f6a", # quadrant 4 "lightgray", "darkgray", # quadrant 5 (middle square) "lightgray", "#40DD52", # quadrant 6 "lightgray", "#FF0000", # quadrant 7 (bottom left corner) "lightgray", "#EA7467", # quadrant 8 "lightgray", "#85DB8E") # quadrant 9 (bottom right corner)
scale_size	(logical) Whether to scale the size of the words.
plot_topics_idx	(vector) The index or indeces of the topics to plot (e.g., look in the model-object for the indices; can for example, be c(1, 3:5) to plot topic t_1, t_3, t_4 and t_5) (optional).
save_dir	(string) The directory to save the plots.
figure_format	(string) Set the figure format, e.g., ".svg", or ".png".
width	(integer) The width of the topic (units = "in").
height	(integer) The width of the topic (units = "in").
max_size	(integer) The max size of the words.

`seed` (integer) The seed to set for reproducibility

`scatter_legend_dot_size`
(integer) The size of dots in the scatter legend.

`scatter_legend_bg_dot_size`
(integer) The size of background dots in the scatter legend.

`scatter_legend_n`
(numeric or vector) A vector determining the number of dots to emphasis in each quadrant of the scatter legend. For example: c(1,1,1,1,0,1,1,1,1) result in one dot in each quadrant except for the middle quadrant.

`scatter_legend_method`
(string) The method to filter topics to be emphasised in the scatter legend. Can be either "mean", "max_x", or "max_y"

`scatter_legend_specified_topics`
(vector) Specify which topic(s) to be emphasised in the scatter legend. For example c("t_1", "t_2"). If set, scatter_legend_method will have no effect.

`scatter_legend_topic_n`
(boolean) Allow showing the topic number or not in the scatter legend

`grid_legend_title`
The title of grid topic plot.

`grid_legend_title_size`
The size of the title of the plot.

`grid_legend_title_color`
The color of the legend title.

`grid_legend_x_axes_label`
The label of the x axes.

`grid_legend_y_axes_label`
The label of the y axes.

`grid_legend_number_color`
The color in the text in the legend.

`grid_legend_number_size`
The color in the text in the legend.

Value

The function saves figures in the save_dir.

`topicsPreds` *Predict topic distributions*

Description

The function to predict the topics of a new document with the trained model.

Usage

```
topicsPreds(
  model,
  data,
  num_iterations = 100,
  seed = 42,
  save_dir,
  load_dir = NULL
)
```

Arguments

model	(list) The trained model
data	(tibble) The new data
num_iterations	(integer) The number of iterations to run the model
seed	(integer) The seed to set for reproducibility
save_dir	(string) The directory to save the model, if NULL, the predictions will not be saved
load_dir	(string) The directory to load the model from, if NULL, the predictions will not be loaded

Value

A tibble of the predictions

Examples

```
# Predict topics for new data with the trained model
save_dir_temp <- tempfile()

dtm <- topicsDtm(
data = dep_wor_data$Depphrase,
save_dir = save_dir_temp)

model <- topicsModel(dtm = dtm, # output of topicsDtm()
                     num_topics = 20,
                     num_top_words = 10,
                     num_iterations = 1000,
                     seed = 42,
                     save_dir = save_dir_temp)

preds <- topicsPreds(
model = model, # output of topicsModel()
data = dep_wor_data$Depphrase,
save_dir = save_dir_temp)
```

topicsTest *Statistically test topics*

Description

The function to test the lda model for multiple dimensions, e.g., 2.

Usage

```
topicsTest(
  data,
  model = NULL,
  preds = NULL,
  ngrams = NULL,
  pred_var_x = NULL,
  pred_var_y = NULL,
  group_var = NULL,
  control_vars = c(),
  test_method = "linear_regression",
  p_alpha = 0.05,
  p_adjust_method = "fdr",
  seed = 42,
  load_dir = NULL,
  save_dir
)
```

Arguments

data	(tibble) The data to test on
model	(list) The trained model
preds	(tibble) The predictions
ngrams	(list) output of the ngram function
pred_var_x	(string) The x variable name to be predicted, and to be plotted (only needed for regression or correlation)
pred_var_y	(string) The y variable name to be predicted, and to be plotted (only needed for regression or correlation)
group_var	(string) The variable to group by (only needed for t-test)
control_vars	(vector) The control variables (not supported yet)
test_method	(string) The test method to use, either "correlation","t-test", "linear_regression","logistic_regression", or "ridge_regression"
p_alpha	(numeric) Threshold of p value set by the user for visualising significant topics
p_adjust_method	(character) Method to adjust/correct p-values for multiple comparisons (default = "none"; see also "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr").

seed	(integer) The seed to set for reproducibility
load_dir	(string) The directory to load the test from, if NULL, the test will not be loaded
save_dir	(string) The directory to save the test, if NULL, the test will not be saved

Value

A list of the test results, test method, and prediction variable

Examples

```
# Test the topic document distribution in respect to a variable
save_dir_temp <- tempfile()

dtm <- topicsDtm(
  data = dep_wor_data$Depphrase,
  save_dir =  save_dir_temp)

model <- topicsModel(
  dtm = dtm, # output of topicsDtm()
  num_topics = 20,
  num_top_words = 10,
  num_iterations = 1000,
  seed = 42,
  save_dir = save_dir_temp)

preds <- topicsPreds(
 model = model, # output of topicsModel()
 data = dep_wor_data$Depphrase,
 save_dir = save_dir_temp)

test <- topicsTest(
  model = model, # output of topicsModel()
  data=dep_wor_data,
  preds = preds, # output of topicsPreds()
  test_method = "linear_regression",
  pred_var_x = "Age",
  save_dir = save_dir_temp)
```

Index

∗ **datasets**
dep_wor_data, 2

dep_wor_data, 2

topicsDtm, 3
topicsDtmEval, 4
topicsGrams, 5
topicsModel, 6
topicsPlot, 7
topicsPreds, 9
topicsTest, 11