Package: textplot (via r-universe)

October 22, 2024

Type Package

Title Text Plots

Version 0.2.2

Maintainer Jan Wijffels <jwijffels@bnosac.be>

Description Visualise complex relations in texts. This is done by providing functionalities for displaying text co-occurrence networks, text correlation networks, dependency relationships as well as text clustering and semantic text 'embeddings'. Feel free to join the effort of providing interesting text visualisations.

License GPL-2

URL https://github.com/bnosac/textplot

LazyData true

Imports utils, methods, lattice, stats, Matrix, graphics, data.table (>= 1.9.6)

Suggests knitr, udpipe, BTM, igraph, graph, Rgraphviz, qgraph, glasso, ggplot2, ggraph, ggforce, concaveman, ggrepel, ggalt, uwot

VignetteBuilder knitr

RoxygenNote 7.1.2

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

RemoteUrl https://github.com/bnosac/textplot

RemoteRef HEAD

RemoteSha 6097d2b0453ad19d01b03e55dd17e88bccbbec8d

Contents

example_btm 2
example_embedding 2
example_embedding_clusters 3
example_udpipe 3

plot.BTM 3
textplot_bar 5
textplot_bitermclusters 7
textplot_cooccurrence 9
textplot_correlation_glasso 10
textplot_correlation_lines 11
textplot_correlation_lines_attrs 13
textplot_dependencyparser 14
textplot_embedding_2d 16

Index **18**

`example_btm` *Example Biterm Topic Model*

Description

The object is a BTM topic model created with the BTM package. It was created on a subset of all CRAN packages, namely package which are part of the NaturalLanguageProcessing and MachineLearning task views.
Timepoint of creation was 2020-04-10.

Examples

```
library(BTM)
data(example_btm, package = 'textplot')
example_btm
str(example_btm)
```

`example_embedding` *Example word embedding matrix*

Description

A matrix with 25-dimensional word embeddings, constructed upon the be_parliament_2020 dataset in the doc2vec R package

Examples

```
data(example_embedding, package = 'textplot')
head(example_embedding)
```

example_embedding_clusters *Example words emitted in a ETM text clustering model*

Description

Example words emitted in a ETM text clustering model constructed upon the be_parliament_2020 dataset in the doc2vec R package

Examples

```
data(example_embedding_clusters, package = 'textplot')
head(example_embedding_clusters)
terminology <- split(example_embedding_clusters, example_embedding_clusters$cluster)
lapply(terminology, head, n = 5)
```

example_udpipe *Example annotation of text using udpipe*

Description

The object is a data.frame of the annotation of the text: "UDPipe provides tokenization, tagging, lemmatization and dependency parsing of raw text"

Examples

```
data(example_udpipe)
str(example_udpipe)
```

plot.BTM *Plot function for a BTM object*

Description

Plot biterms as a clustered graph. The graph is constructed by assigning each word to a topic and within a topic of words biterm frequencies are shown.

Usage

```
## S3 method for class 'BTM'
plot(
  x,
  biterms = terms(x, type = "biterms")$biterms,
  top_n = 7,
  which,
  labels = seq_len(x$K),
  title = "Biterm topic model",
  subtitle = list(),
  ...
)
```

Arguments

x	an object of class BTM with a biterm topic model
biterms	a data.frame with columns term1, term2, topic with all biterms and the topic these were assigned to. Defaults to the biterms used to construct the model.
top_n	integer indicating to limit to displaying the top_n terms for each topic. Defaults to 7.
which	integer vector indicating to display only these topics. See the examples.
labels	a character vector of names. Should be of the same length as the number of topics in the data.
title	character string with the title to use in the plot
subtitle	character string with the subtitle to use in the plot
...	not used

Value

an object of class ggplot

See Also

BTM, textplot_bitermclusters.default

Examples

```
library(igraph)
library(BTM)
library(ggraph)
library(ggforce)
library(concaveman)
data(example_btm, package = 'textplot')

model <- example_btm

plot(model, title = "BTM model", top_n = 3)
plot(model, title = "BTM model", top_n = 3, labels = 1:model$K)
```

```
plot(model, title = "BTM model", which = 7:15)
plot(model, title = "BTM model", subtitle = "First 5 topics",
     which = 1:5, top_n = 10)
plot(model, title = "Biterm topic model", subtitle = "First 8 topics",
     which = 1:8, top_n = 7)


topiclabels <- c("Garbage",
  "Data Mining", "Gradient descent", "API's",
  "Random Forests", "Stat models", "Text Mining / NLP",
  "GLM / GAM / Bayesian", "Machine learning", "Variable selection",
  "Regularisation techniques", "Optimisation", "Fuzzy logic",
  "Classification/Regression trees", "Text frequencies",
  "Neural / Deep learning", "Variable selection",
  "Text file handling", "Text matching", "Topic modelling")
plot(model, title = "Biterm topic model", subtitle = "some topics",
     top_n = 7,
     which = c(3, 4, 5, 6, 7, 9, 12, 16, 20),
     labels = topiclabels)


library(BTM)
library(data.table)
library(udpipe)
## Annotate text with parts of speech tags
data("brussels_reviews", package = "udpipe")
anno <- subset(brussels_reviews, language %in% "nl")
anno <- data.frame(doc_id = anno$id, text = anno$feedback, stringsAsFactors = FALSE)
anno <- udpipe(anno, "dutch", trace = 10)
## Get cooccurrences of nouns / adjectives and proper nouns
biterms <- as.data.table(anno)
biterms <- biterms[, cooccurrence(x = lemma,
                                  relevant = upos %in% c("NOUN", "PROPN", "ADJ"),
                                  skipgram = 2),
                   by = list(doc_id)]
## Build the BTM model
set.seed(123456)
x <- subset(anno, upos %in% c("NOUN", "PROPN", "ADJ"))
x <- x[, c("doc_id", "lemma")]
model <- BTM(x, k = 5, beta = 0.01, iter = 2000, background = TRUE,
             biterms = biterms, trace = 100)
plot(model)
```

`textplot_bar` *Barplot of a frequency table using lattice*

Description

Barplot of a frequency table using lattice

Usage

```
textplot_bar(x, ...)

## Default S3 method:
textplot_bar(
  x,
  panel = "Effect",
  total = sum(x),
  top = 40,
  col.panel = "lightgrey",
  col.line = "lightblue",
  lwd = 3,
  cextext = 0.5,
  addpct = FALSE,
  cexpct = 0.75,
  textpos = 3,
  pctpos = 1,
  v = NULL,
  col.abline = "red",
  ...
)
```

Arguments

x	a table to plot or a data.frame with the first column the label and the second column the frequency
...	other arguments passed on to `lattice::dotplot`
panel	character string what to put into the panel
total	integer with the total. Defaults to sum(x). Is used to plot the table counts as a percentage. In which case this is divided by the total.
top	integer indicating to plot only the first 'top' table elements. Defaults to 40.
col.panel	color of the panel. Defaults to lightgrey.
col.line	color of the line. Passed on to the col argument in `lattice::panel.lines`
lwd	width of the line. Passed on to the lwd argument in `lattice::panel.lines`
cextext	numeric with the cex of the text with the counts plotted. Passed on to `lattice::panel.text`.
addpct	logical indicating to add the percent with `lattice::panel.text`
cexpct	numeric with the cex of the text plotted when using addpct. Passed on to `lattice::panel.text`.
textpos	passed on to the pos argument of panel.text to indicate where to put the text of the frequencies
pctpos	passed on to the pos argument of panel.text to indicate where to put the text of the percentages
v	passed on to `lattice::panel.abline` to draw a vertical line
col.abline	passed on to `lattice::panel.abline` to draw a vertical line

Value

the result of a call to `lattice::dotplot`

Examples

```
data(brussels_listings, package = 'udpipe')
x <- table(brussels_listings$neighbourhood)
x <- sort(x)
textplot_bar(x,
 panel = "Locations", col.panel = "darkgrey", xlab = "Listings",
 cextext = 0.75, addpct = TRUE, cexpct = 0.5)


x <- sample(LETTERS, 1000, replace = TRUE)
textplot_bar(sort(table(x)), panel = "Frequencies", xlab = "Frequency",
   cextext = 0.75, main = "Freq stats")
textplot_bar(sort(table(x)), panel = "Frequencies", addpct = TRUE, top = 15)

## x can also be a data.frame where the first column
## is the label and the second column the frequency
x <- data.frame(l = LETTERS, amount = rnorm(26))
textplot_bar(x)
textplot_bar(x, v = 0)
```

---

textplot_bitermclusters

Plot biterm cluster groups

Description

Plot biterms as a clustered graph. The graph is constructed by assigning each word to a topic and within a topic of words biterm frequencies are shown.

Usage

```
textplot_bitermclusters(x, ...)

## Default S3 method:
textplot_bitermclusters(
  x,
  biterms,
  which,
  labels = seq_len(length(table(biterms$topic))),
  title = "Biterm topic model",
  subtitle = list(),
  ...
)
```

Arguments

x	a list of data.frames, each containing the columns token and probability corresponding to how good a token is emitted by a topic. The list index is assumed to be the topic number
...	not used
biterms	a data.frame with columns term1, term2, topic with all biterms and the topic these were assigned to
which	integer vector indicating to display only these topics. See the examples.
labels	a character vector of names. Should be of the same length as the number of topics in the data.
title	character string with the title to use in the plot
subtitle	character string with the subtitle to use in the plot

Value

an object of class ggplot

Examples

```
library(igraph)
library(ggraph)
library(concaveman)
library(ggplot2)
library(BTM)
data(example_btm, package = 'textplot')
group_terms   <- terms(example_btm, top_n = 3)
group_biterms <- example_btm$biterms$biterms


textplot_bitermclusters(x = group_terms, biterms = group_biterms)
textplot_bitermclusters(x = group_terms, biterms = group_biterms,
                        title = "BTM model", subtitle = "Topics 7-15",
                        which = 7:15, labels = seq_len(example_btm$K))

group_terms   <- terms(example_btm, top_n = 10)
textplot_bitermclusters(x = group_terms, biterms = group_biterms,
                        title = "BTM model", subtitle = "Topics 1-5",
                        which = 1:5, labels = seq_len(example_btm$K))

group_terms   <- terms(example_btm, top_n = 7)
topiclabels <- c("Garbage",
  "Data Mining", "Gradient descent", "API's",
  "Random Forests", "Stat models", "Text Mining / NLP",
  "GLM / GAM / Bayesian", "Machine learning", "Variable selection",
  "Regularisation techniques", "Optimisation", "Fuzzy logic",
  "Classification/Regression trees", "Text frequencies",
  "Neural / Deep learning", "Variable selection",
  "Text file handling", "Text matching", "Topic modelling")
textplot_bitermclusters(x = group_terms, biterms = group_biterms,
```

```
                        title = "Biterm topic model", subtitle = "some topics",
                        which = c(3, 4, 5, 6, 7, 9, 12, 16, 20),
                        labels = topiclabels)
```

`textplot_cooccurrence` *Plot term cooccurrences as a network*

Description

Plot term cooccurrences in a graph structure

Usage

```
textplot_cooccurrence(x, ...)

## Default S3 method:
textplot_cooccurrence(
  x,
  terms,
  top_n = 50,
  title = "Term cooccurrences",
  subtitle = list(),
  vertex_color = "darkgreen",
  edge_color = "grey",
  base_family = "",
  ...
)
```

Arguments

`x`	a data.frame with columns term1, term2 and cooc indicating how many times 2 terms are occurring together
`...`	other parameters passed on to `ggraph::geom_node_text`
`terms`	a character vector with terms to only plot. Prevails compared to using `top_n`
`top_n`	integer indicating to show only the top n occurrences as in `head(x, n = top_n)`
`title`	character string with the title to use in the plot
`subtitle`	character string with the subtitle to use in the plot
`vertex_color`	character with the color of the label of each node. Defaults to darkgreen.
`edge_color`	character with the color of the edges between the nodes. Defaults to grey.
`base_family`	character passed on to `theme_void` setting the base font family

Value

an object of class ggplot

Examples

```
library(udpipe)
library(igraph)
library(ggraph)
library(ggplot2)
data(brussels_reviews_anno, package = 'udpipe')
x <- subset(brussels_reviews_anno, xpos %in% "JJ" & language %in% "fr")
x <- cooccurrence(x, group = "doc_id", term = "lemma")

textplot_cooccurrence(x, top_n = 25, subtitle = "showing only top 25")
textplot_cooccurrence(x, top_n = 25, title = "Adjectives",
                      vertex_color = "orange", edge_color = "black",
                      fontface = "bold")
```

---

textplot_correlation_glasso

Plot sparse term correlations as a graph structure

Description

Plot sparse term correlations as a graph structure. Uses the glasso procedure (`glasso::glassopath`) to reduce the correlation matrix to retain only the relevant correlations and next visualises these sparse correlations.

Usage

```
textplot_correlation_glasso(x, ...)

## Default S3 method:
textplot_correlation_glasso(
  x,
  n = 1000,
  exclude_zero = TRUE,
  label.cex = 1,
  node.width = 0.5,
  ...
)
```

Arguments

x	a correlation matrix
...	further arguments passed on to `qgraph::qgraph`, except layout which is set to 'spring', labels (taken from the colnames of `x`), and borders which is set to FALSE.
n	sample size used in computing the sparse correlation matrix. Defaults to 1000.

exclude_zero	logical indicating to exclude zero-correlations from the graph
label.cex	passed on to qgraph::qgraph
node.width	passed on to qgraph::qgraph

Value

an object of class ggplot

Examples

```
library(udpipe)
library(qgraph)
library(glasso)
data(brussels_reviews_anno, package = 'udpipe')
x <- subset(brussels_reviews_anno, xpos %in% "NN" & language %in% "fr" & !is.na(lemma))
x <- document_term_frequencies(x, document = "doc_id", term = "lemma")
dtm <- document_term_matrix(x)
dtm <- dtm_remove_lowfreq(dtm, maxterms = 60)

m <- dtm_cor(dtm)
textplot_correlation_glasso(m, exclude_zero = TRUE)

textplot_correlation_glasso(m, exclude_zero = FALSE)
```

---

textplot_correlation_lines

Document/Term Correlation Plot

Description

Plots the highest occurring correlations among terms.
This is done by plotting the terms into nodes and the correlations between the terms as lines between the nodes. Lines of the edges are proportional to the correlation height. This uses the plot function for graphNEL objects (using the Rgraphviz package)

Usage

```
textplot_correlation_lines(x, ...)

## Default S3 method:
textplot_correlation_lines(
  x,
  terms = colnames(x),
  threshold = 0.05,
  top_n,
```

```
  attrs = textplot_correlation_lines_attrs(),
  terms_highlight,
  label = FALSE,
  cex.label = 1,
  col.highlight = "red",
  lwd = 1,
  ...
)
```

Arguments

x	a document-term matrix of class dgCMatrix
...	other arguments passed on to plot
terms	a character vector with terms present in the columns of x indicating terms to focus on
threshold	a threshold to show only correlations between the terms with absolute values above this threshold. Defaults to 0.05.
top_n	an integer indicating to show only the top top_n correlations. This can be set to plot only the top correlations. E.g. set it to 20 to show only the top 20 correlations with the highest absolute value.
attrs	a list of attributes with graph visualisation elements passed on to the plot function of an object of class graphNEL. Defaults to textplot_correlation_lines_attrs.
terms_highlight	a vector of character terms to highlight or a vector of numeric values in the 0-1 range indicating how much (in percentage) to increase the node font size. See the examples.
label	logical indicating to draw the label with the correlation size between the nodes
cex.label	cex of the label of the correlation size
col.highlight	color to use for highlighted terms specified in terms_highlight. Defaults to red.
lwd	numeric value - graphical parameter used to increase the edge thickness which indicates the correlation strength. Defaults to 1.

Value

invisibly the plot

Examples

```
## Construct document/frequency/matrix
library(graph)
library(Rgraphviz)
library(udpipe)
data(brussels_reviews_anno, package = 'udpipe')
exclude <- c(32337682L, 27210436L, 26820445L, 37658826L, 33661134L, 48756422L,
  23454554L, 30461127L, 23292176L, 32850277L, 30566303L, 21595142L,
  20441279L, 38097066L, 28651065L, 29011387L, 37316020L, 22135291L,
```

```
  40169379L, 38627667L, 29470172L, 24071827L, 40478869L, 36825304L,
  21597085L, 21427658L, 7890178L, 32322472L, 39874379L, 32581310L,
  43865675L, 31586937L, 32454912L, 34861703L, 31403168L, 35997324L,
  29002317L, 33546304L, 47677695L)
dtm <- brussels_reviews_anno
dtm <- subset(dtm, !doc_id %in% exclude)
dtm <- subset(dtm, xpos %in% c("NN") & language == "nl" & !is.na(lemma))
dtm <- document_term_frequencies(dtm, document = "doc_id", term = "lemma")
dtm <- document_term_matrix(dtm)
dtm <- dtm_remove_lowfreq(dtm, minfreq = 5)
dtm <- dtm_remove_tfidf(dtm, top = 500)

## Plot top 20 correlations, having at least a correlation of 0.01
textplot_correlation_lines(dtm, top_n = 25, threshold = 0.01)

## Plot top 20 correlations
textplot_correlation_lines(dtm, top_n = 25, label = TRUE, lwd = 5)

## Plot top 20 correlations and highlight some terms
textplot_correlation_lines(dtm, top_n = 25, label = TRUE, lwd = 5,
                           terms_highlight = c("prijs", "privacy"),
                           main = "Top correlations in topic xyz")

## Plot top 20 correlations and highlight + increase some terms
textplot_correlation_lines(dtm, top_n = 25, label = TRUE, lwd=5,
                           terms_highlight = c(prijs = 0.8, privacy = 0.1),
                           col.highlight = "red")

## Plot correlations between specific terms
w <- dtm_colsums(dtm)
w <- head(sort(w, decreasing = TRUE), 100)
textplot_correlation_lines(dtm, terms = names(w), top_n = 20, label = TRUE)

attrs <- textplot_correlation_lines_attrs()
attrs$node$shape <- "rectangle"
attrs$edge$color <- "steelblue"
textplot_correlation_lines(dtm, top_n = 20, label = TRUE,
                           attrs = attrs)
```

`textplot_correlation_lines_attrs`
Document/Term Correlation Plot graphical attributes

Description

Document/Term Correlation Plot graphical attributes

Usage

```
textplot_correlation_lines_attrs(fontsize = 25)
```

Arguments

fontsize	size of the font. Defaults to 25

Value

a list with graph visualisation elements used by textplot_correlation_lines

Examples

```
textplot_correlation_lines_attrs()
```

---

textplot_dependencyparser

Plot output of a dependency parser

Description

Plot output of a dependency parser. This plot takes one sentence and shows for the sentence, the words, the parts of speech tag and the dependency relationship between the words.

Usage

```
textplot_dependencyparser(x, ...)

## Default S3 method:
textplot_dependencyparser(
  x,
  title = "Dependency Parser",
  subtitle = "tokenisation, parts of speech tagging & dependency relations",
  vertex_color = "darkgreen",
  edge_color = "red",
  size = 3,
  base_family = "",
  layout = "linear",
  ...
)
```

Arguments

x	a data.frame as returned by a call to udpipe containing 1 sentence
...	not used yet
title	character string with the title to use in the plot
subtitle	character string with the title to use in the plot
vertex_color	character with the color of the label of each node. Defaults to darkgreen.
edge_color	character with the color of the edges between the nodes. Defaults to red.
size	size of the labels in the plot. Defaults to 3.
base_family	character passed on to theme_void setting the base font family
layout	the type of layout, defaults to 'linear', passed on to ggraph

Value

an object of class ggplot

See Also

udpipe

Examples

```
library(udpipe)
library(ggraph)
library(ggplot2)
library(igraph)

x <- udpipe("The economy is weak but the outlook is bright", "english")
textplot_dependencyparser(x)

x <- udpipe("His speech about marshmallows in New York is utter bullshit", "english")
textplot_dependencyparser(x, size = 4)

x <- udpipe("UDPipe provides tokenization, tagging, lemmatization and
             dependency parsing of raw text", "english")
textplot_dependencyparser(x, size = 4)


data("example_udpipe", package = "textplot")
textplot_dependencyparser(example_udpipe, size = 4)
```

`textplot_embedding_2d` *Plot word embeddings in 2D*

Description

This plot displays words in 2 dimensions, optionally grouped by cluster.
This allows to visualise embeddings which are reduced by dimensionality reduction techniques like UMAP, t-SNE, PCA or similar techniques. It allows to highlight the words by groups and is a good way to visualise a small sets of word or topic embeddings.

Usage

```
textplot_embedding_2d(x, ...)

## Default S3 method:
textplot_embedding_2d(
  x,
  title = "Embedding plot in 2D",
  subtitle = list(),
  encircle = FALSE,
  points = FALSE,
  alpha = 0.4,
  ...
)
```

Arguments

`x`	a data.frame with columns 'x', 'y', 'term' and optionally 'group' (color by group), 'weight' (size of the text / point shown), 'type' (pch used for the type of point)
`...`	not used yet
`title`	character string with the title to use in the plot
`subtitle`	character string with the subtitle to use in the plot
`encircle`	logical indicating to encircle all the points belonging to a group using `geom_encircle`
`points`	logical indicating to add points. Defaults to `FALSE`.
`alpha`	transparancy level passed on to `geom_encircle` in case `encircle` is set to `TRUE`

Value

an object of class ggplot

Examples

```
library(ggplot2)
library(ggrepel)
library(ggalt)
##
## Generate some fake embeddings
##   probably you want to use word2vec::word2vec(...) + uwot::umap(...)
embeddings <- matrix(runif(26 * 2), nrow = 26, ncol = 2, dimnames = list(letters))
x <- data.frame(term = rownames(embeddings), x = embeddings[, 1], y = embeddings[, 2])

## 2D plot
textplot_embedding_2d(x)

## 2D plot with groups
x$group <- sample(c("clustera", "clusterb", "clusterc"), size = 26, replace = TRUE)
textplot_embedding_2d(x)

## 2D plot with groups and weights for each word
x$weight <- runif(nrow(x))
textplot_embedding_2d(x)
textplot_embedding_2d(x, points = TRUE)

## 2D plot with groups and weights for each word and different types of points
x$type <- sample(c("word", "center"), size = 26, replace = TRUE)
x$type <- factor(x$type, levels = c("word", "center"))
textplot_embedding_2d(x, points = TRUE)
textplot_embedding_2d(x, title = "Embedding plot in 2D", subtitle = "example")

## Encircle the words belonging to each group
textplot_embedding_2d(x, title = "Embedding plot in 2D", subtitle = "example",
                      encircle = TRUE, alpha = 0.2)
```

Index

BTM, *4*

example_btm, 2
example_embedding, 2
example_embedding_clusters, 3
example_udpipe, 3

geom_encircle, *16*
ggraph, *15*

plot.BTM, 3

textplot_bar, 5
textplot_bitermclusters, 7
textplot_bitermclusters.default, *4*
textplot_cooccurrence, 9
textplot_correlation_glasso, 10
textplot_correlation_lines, 11, *14*
textplot_correlation_lines_attrs, *12*, 13
textplot_dependencyparser, 14
textplot_embedding_2d, 16

udpipe, *15*