Package: quanteda.textplots (via r-universe)

August 29, 2024

Title Plots for the Quantitative Analysis of Textual Data

Version 0.95

Description Plotting functions for visualising textual data. Extends 'quanteda' and related packages with plot methods designed specifically for text data, textual statistics, and models fit to textual data. Plot types include word clouds, lexical dispersion plots, scaling plots, network visualisations, and word 'keyness' plots.

License GPL-3

Imports quanteda, extrafont, ggplot2, ggrepel, grid, sna, igraph, Matrix, methods, network, RColorBrewer, Rcpp (>= 0.12.12), stringi

LinkingTo Rcpp

Suggests knitr, quanteda.textmodels, quanteda.textstats, rmarkdown, spelling, testthat, wordcloud

Encoding UTF-8

BugReports https://github.com/quanteda/quanteda.textplots/issues

Language en-GB

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

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

RemoteUrl https://github.com/quanteda/quanteda.textplots

RemoteRef HEAD

RemoteSha 66fa7c1807c8800f0d6e764aba9b29a86cbfc1e0

Contents

`textplot_keyness` *Plot word keyness*

Description

Plot the results of a "keyword" of features comparing their differential associations with a target and a reference group, after calculating keyness using `quanteda.textstats::textstat_keyness()`.

Usage

```
textplot_keyness(
  x,
  show_reference = TRUE,
  show_legend = TRUE,
  n = 20L,
  min_count = 2L,
  margin = 0.05,
  color = c("darkblue", "gray"),
  labelcolor = "gray30",
  labelsize = 4,
  font = NULL
)
```

Arguments

`x`	a return object from `quanteda.textstats::textstat_keyness()`
`show_reference`	logical; if `TRUE`, show key reference features in addition to key target features
`show_legend`	logical; if `TRUE`, show legend
`n`	integer; number of features to plot
`min_count`	numeric; minimum total count of feature across the target and reference categories, for a feature to be included in the plot
`margin`	numeric; size of margin where feature labels are shown
`color`	character or integer; colours of bars for target and reference documents. `color` must have two elements when `show_reference = TRUE`. See ggplot2::color.
`labelcolor`	character; color of feature labels.
`labelsize`	numeric; size of feature labels and bars. See ggplot2::size.
`font`	character; font-family of texts. Use default font if `NULL`.

Value

a **ggplot2** object

Author(s)

Haiyan Wang and Kohei Watanabe

See Also

quanteda.textstats::textstat_keyness()

Examples

```
## Not run:
library("quanteda")
# compare Trump speeches to other Presidents by chi^2
dfmat1 <- data_corpus_inaugural |>
     corpus_subset(Year > 1980) |>
     tokens(remove_punct = TRUE) |>
     tokens_remove(stopwords("en")) |>
     dfm()
dfmat1 <- dfm_group(dfmat1, groups = dfmat1$President)
tstat1 <- quanteda.textstats::textstat_keyness(dfmat1, target = "Trump")
textplot_keyness(tstat1, margin = 0.2, n = 10)
tstat1 <- quanteda.textstats::textstat_keyness(dfmat1, target = "Trump")
textplot_keyness(tstat1, margin = 0.2, n = 10)

# compare contemporary Democrats v. Republicans
corp <- data_corpus_inaugural |>
    corpus_subset(Year > 1960)
corp$party <- ifelse(docvars(corp, "President") %in% c("Nixon", "Reagan", "Bush", "Trump"),
                     "Republican", "Democrat")
dfmat2 <- corp |>
    tokens(remove_punct = TRUE) |>
    tokens_remove(stopwords("en")) |>
    dfm()
tstat2 <- quanteda.textstats::textstat_keyness(dfm_group(dfmat2, groups = dfmat2$party),
                                               target = "Democrat", measure = "lr")
textplot_keyness(tstat2, color = c("blue", "red"), n = 10)

## End(Not run)
```

---

textplot_network *Plot a network of feature co-occurrences*

Description

Plot an fcm object as a network, where edges show co-occurrences of features.

Usage

```
textplot_network(
  x,
  min_freq = 0.5,
  omit_isolated = TRUE,
  edge_color = "#1F78B4",
  edge_alpha = 0.5,
  edge_size = 2,
  vertex_color = "#4D4D4D",
  vertex_size = 2,
  vertex_labelcolor = NULL,
  vertex_labelfont = NULL,
  vertex_labelsize = 5,
  offset = NULL,
  ...
)

## S3 method for class 'fcm'
as.network(x, min_freq = 0.5, omit_isolated = TRUE, ...)

## S3 method for class 'fcm'
as.igraph(x, min_freq = 0.5, omit_isolated = TRUE, ...)
```

Arguments

x	a fcm or dfm object
min_freq	a frequency count threshold or proportion for co-occurrence frequencies of features to be included.
omit_isolated	if TRUE, features do not occur more frequent than min_freq will be omitted.
edge_color	colour of edges that connect vertices.
edge_alpha	opacity of edges ranging from 0 to 1.0.
edge_size	size of edges for most frequent co-occurrence The size of other edges are determined proportionally to the 99th percentile frequency instead of the maximum to reduce the impact of outliers.
vertex_color	colour of vertices.
vertex_size	size of vertices
vertex_labelcolor	colour of texts. Defaults to the same as vertex_color. If NA is given, texts are not rendered.
vertex_labelfont	font-family of texts. Use default font if NULL.
vertex_labelsize	size of vertex labels in mm. Defaults to size 5. Supports both integer values and vector values.
offset	if NULL, the distance between vertices and texts are determined automatically.
...	additional arguments passed to network or graph_from_adjacency_matrix. Not used for as.igraph.

Details

Currently the size of the network is limited to 1000, because of the computationally intensive nature of network formation for larger matrices. When the fcm is large, users should select features using fcm_select(), set the threshold using `min_freq`, or implement own plotting function using `as.network()`.

Author(s)

Kohei Watanabe and Stefan Müller

See Also

fcm

`network::network()`

`igraph::graph_from_adjacency_matrix()`

Examples

```
set.seed(100)
library("quanteda")
toks <- data_char_ukimmig2010 |>
    tokens(remove_punct = TRUE) |>
    tokens_tolower() |>
    tokens_remove(pattern = stopwords("english"), padding = FALSE)
fcmat <- fcm(toks, context = "window", tri = FALSE)
feat <- colSums(fcmat) |>
  sort(decreasing = TRUE) |>
  head(30) |>
  names()
fcm_select(fcmat, pattern = feat) |>
    textplot_network(min_freq = 0.5)
fcm_select(fcmat, pattern = feat) |>
    textplot_network(min_freq = 0.8)
fcm_select(fcmat, pattern = feat) |>
    textplot_network(min_freq = 0.8, vertex_labelcolor = rep(c('gray40', NA), 15))
fcm_select(fcmat, pattern = feat) |>
    textplot_network(vertex_labelsize = 10)
fcm_30 <- fcm_select(fcmat, pattern = feat)
textplot_network(fcm_30,
              vertex_labelsize = Matrix::rowSums(fcm_30) / min(Matrix::rowSums(fcm_30)))
# Vector inputs to vertex_labelsize can be scaled if too small / large
textplot_network(fcm_30,
                 vertex_labelsize = 1.5 * Matrix::rowSums(fcm_30) /
                                    min(Matrix::rowSums(fcm_30)))

# as.igraph
if (requireNamespace("igraph", quietly = TRUE)) {
    txt <- c("a a a b b c", "a a c e", "a c e f g")
    mat <- fcm(tokens(txt))
    as.igraph(mat, min_freq = 1, omit_isolated = FALSE)
}
```

`textplot_scale1d` *Plot a fitted scaling model*

Description

Plot the results of a fitted scaling model, from (e.g.) a predicted quanteda.textmodels::textmodel_wordscores model or a fitted quanteda.textmodels::textmodel_wordfish or quanteda.textmodels::textmodel_ca model. Either document or feature parameters may be plotted: an ideal point-style plot (estimated document position plus confidence interval on the x-axis, document labels on the y-axis) with optional renaming and sorting, or as a plot of estimated feature-level parameters (estimated feature positions on the x-axis, and a measure of relative frequency or influence on the y-axis, with feature names replacing plotting points with some being chosen by the user to be highlighted).

Usage

```
textplot_scale1d(
  x,
  margin = c("documents", "features"),
  doclabels = NULL,
  sort = TRUE,
  groups = NULL,
  highlighted = NULL,
  alpha = 0.7,
  highlighted_color = "black"
)
```

Arguments

`x`	the fitted or predicted scaling model object to be plotted
`margin`	`"documents"` to plot estimated document scores (the default) or `"features"` to plot estimated feature scores by a measure of relative frequency
`doclabels`	a vector of names for document; if left NULL (the default), docnames will be used
`sort`	if `TRUE` (the default), order points from low to high score. If a vector, order according to these values from low to high. Only applies when `margin = "documents"`.
`groups`	grouping variable for sampling, equal in length to the number of documents. This will be evaluated in the docvars data.frame, so that docvars may be referred to by name without quoting. This also changes previous behaviours for `groups`. See `news(Version >= "3.0", package = "quanteda")` for details.
`highlighted`	a vector of feature names to draw attention to in a feature plot; only applies if `margin = "features"`
`alpha`	A number between 0 and 1 (default 0.5) representing the level of alpha transparency used to overplot feature names in a feature plot; only applies if `margin = "features"`
`highlighted_color`	colour for highlighted terms in `highlighted`

Value

a **ggplot2** object

Note

The `groups` argument only applies when `margin = "documents"`.

Author(s)

Kenneth Benoit, Stefan Müller, and Adam Obeng

See Also

`quanteda.textmodels::textmodel_wordfish()`, `quanteda.textmodels::textmodel_wordscores()`, `quanteda.textmodels::textmodel_ca()`

Examples

```
library("quanteda")
if (require("quanteda.textmodels")) {
dfmat <- dfm(tokens(data_corpus_irishbudget2010))

## wordscores
refscores <- c(rep(NA, 4), 1, -1, rep(NA, 8))
tmod1 <- textmodel_wordscores(dfmat, y = refscores, smooth = 1)
# plot estimated document positions
textplot_scale1d(predict(tmod1, se.fit = TRUE),
                 groups = data_corpus_irishbudget2010$party)
# plot estimated word positions
textplot_scale1d(tmod1, margin = "features",
                 highlighted = c("minister", "have", "our", "budget"))

## wordfish
tmod2 <- quanteda.textmodels::textmodel_wordfish(dfmat, dir = c(6,5))
# plot estimated document positions
textplot_scale1d(tmod2)
textplot_scale1d(tmod2, groups = data_corpus_irishbudget2010$party)
# plot estimated word positions
textplot_scale1d(tmod2, margin = "features",
                 highlighted = c("government", "global", "children",
                                 "bank", "economy", "the", "citizenship",
                                 "productivity", "deficit"))

## correspondence analysis
tmod3 <- textmodel_ca(dfmat)
# plot estimated document positions
textplot_scale1d(tmod3, margin = "documents",
                 groups = docvars(data_corpus_irishbudget2010, "party"))
}
```

`textplot_wordcloud` *Plot features as a wordcloud*

Description

Plot a dfm or quanteda.textstats::textstat_keyness object as a wordcloud, where the feature labels are plotted with their sizes proportional to their numerical values in the dfm. When `comparison = TRUE`, it plots comparison word clouds by document (or by target and reference categories in the case of a keyness object).

Usage

```
textplot_wordcloud(
  x,
  min_size = 0.5,
  max_size = 4,
  min_count = 3,
  max_words = 500,
  color = "darkblue",
  font = NULL,
  adjust = 0,
  rotation = 0.1,
  random_order = FALSE,
  random_color = FALSE,
  ordered_color = FALSE,
  labelcolor = "gray20",
  labelsize = 1.5,
  labeloffset = 0,
  fixed_aspect = TRUE,
  ...,
  comparison = FALSE
)
```

Arguments

`x`	a dfm or quanteda.textstats::textstat_keyness object
`min_size`	size of the smallest word
`max_size`	size of the largest word
`min_count`	words with frequency below min_count will not be plotted
`max_words`	maximum number of words to be plotted. The least frequent terms dropped. The maximum frequency will be split evenly across categories when `comparison = TRUE`.
`color`	colour of words from least to most frequent
`font`	font-family of words and labels. Use default font if `NULL`.

adjust	adjust sizes of words by a constant. Useful for non-English words for which R fails to obtain correct sizes.
rotation	proportion of words with 90 degree rotation
random_order	plot words in random order. If FALSE, they will be plotted in decreasing frequency.
random_color	choose colours randomly from the colours. If FALSE, the colour is chosen based on the frequency
ordered_color	if TRUE, then colours are assigned to words in order.
labelcolor	colour of group labels. Only used when comparison = TRUE.
labelsize	size of group labels. Only used when comparison = TRUE.
labeloffset	position of group labels. Only used when comparison = TRUE.
fixed_aspect	logical; if TRUE, the aspect ratio is fixed. Variable aspect ratio only supported if rotation = 0.
...	additional parameters. Only used to make it compatible with **wordcloud**
comparison	logical; if TRUE, plot a wordcloud that compares documents in the same way as wordcloud::comparison.cloud(). If x is a quanteda.textstats::textstat_keyness object, then only the target category's key terms are plotted when comparison = FALSE, otherwise the top max_words / 2 terms are plotted from the target and reference categories.

Details

The default is to plot the word cloud of all features, summed across documents. To produce word cloud plots for specific document or set of documents, you need to slice out the document(s) from the dfm object.

Comparison wordcloud plots may be plotted by setting comparison = TRUE, which plots a separate grouping for *each document* in the dfm. This means that you will need to slice out just a few documents from the dfm, or to create a dfm where the "documents" represent a subset or a grouping of documents by some document variable.

Author(s)

Kohei Watanabe, building on code from Ian Fellows's **wordcloud** package.

Examples

```
# plot the features (without stopwords) from Obama's inaugural addresses
set.seed(10)
library("quanteda")
dfmat1 <- data_corpus_inaugural |>
    corpus_subset(President == "Obama") |>
    tokens(remove_punct = TRUE) |>
    tokens_remove(stopwords("en")) |>
    dfm() |>
    dfm_trim(min_termfreq = 3)

# basic wordcloud
```

```
textplot_wordcloud(dfmat1)

# plot in colours with some additional options
textplot_wordcloud(dfmat1, rotation = 0.25,
                   color = rev(RColorBrewer::brewer.pal(10, "RdBu")))

# other display options
col <- sapply(seq(0.1, 1, 0.1), function(x) adjustcolor("#1F78B4", x))
textplot_wordcloud(dfmat1, adjust = 0.5, random_order = FALSE,
                   color = col, rotation = FALSE)

# comparison plot of Obama v. Trump
dfmat2 <- data_corpus_inaugural |>
    corpus_subset(President %in% c("Obama", "Trump")) |>
    tokens(remove_punct = TRUE) |>
    tokens_remove(stopwords("en")) |>
    dfm()
dfmat2 <- dfm_group(dfmat2, dfmat2$President) |>
    dfm_trim(min_termfreq = 3)

textplot_wordcloud(dfmat2, comparison = TRUE, max_words = 100,
                   color = c("blue", "red"))

## Not run:
# for keyness
tstat <- data_corpus_inaugural[c(1, 3)] |>
    tokens(remove_punct = TRUE) |>
    tokens_remove(stopwords("en")) |>
    dfm() |>
    quanteda.textstats::textstat_keyness()
textplot_wordcloud(tstat, min_count = 2)
textplot_wordcloud(tstat, min_count = 2, comparison = FALSE)

## End(Not run)
```

`textplot_xray` *Plot the dispersion of key word(s)*

Description

Plots a dispersion or "x-ray" plot of selected word pattern(s) across one or more texts. The format of the plot depends on the number of kwic class objects passed: if there is only one document, keywords are plotted one below the other. If there are multiple documents the documents are plotted one below the other, with keywords shown side-by-side. Given that this returns a **ggplot2** object, you can modify the plot by adding **ggplot2** layers (see example).

Usage

```
textplot_xray(..., scale = c("absolute", "relative"), sort = FALSE)
```

Arguments

...	any number of kwic class objects
scale	whether to scale the token index axis by absolute position of the token in the document or by relative position. Defaults are absolute for single document and relative for multiple documents.
sort	whether to sort the rows of a multiple document plot by document name

Value

a **ggplot2** object

Known Issues

These are known issues on which we are working to solve in future versions:

- `textplot_xray()` will not display the patterns correctly when these are multi-token sequences.
- For dictionaries with keys that have overlapping value matches to tokens in the text, only the first match will be used in the plot. The way around this is to produce one kwic per dictionary key, and send them as a list to `textplot_xray`.

Examples

```
library("quanteda")
toks <- data_corpus_inaugural |>
  corpus_subset(Year > 1970) |>
  tokens()
# compare multiple documents
textplot_xray(kwic(toks, pattern = "american"))
textplot_xray(kwic(toks, pattern = "american"), scale = "absolute")

# compare multiple terms across multiple documents
textplot_xray(kwic(toks, pattern = "america*"),
              kwic(toks, pattern = "people"))

## Not run:
# how to modify the ggplot with different options
library("ggplot2")
tplot <- textplot_xray(kwic(toks, pattern = "american"),
                       kwic(toks, pattern = "people"))
tplot + aes(color = keyword) + scale_color_manual(values = c('red', 'blue'))

# adjust the names of the document names
docnames(toks) <- apply(docvars(toks, c("Year", "President")), 1, paste, collapse = ", ")
textplot_xray(kwic(toks, pattern = "america*"),
              kwic(toks, pattern = "people"))

## End(Not run)
```

Index