

Package ‘gofastr’

October 13, 2022

Title Fast DocumentTermMatrix and TermDocumentMatrix Creation

Version 0.3.0

Maintainer Tyler Rinker <tyler.rinker@gmail.com>

Description Harness the power of 'quanteda', 'data.table' & 'stringi' to quickly generate 'tm' DocumentTermMatrix and TermDocumentMatrix data structures.

Depends R (>= 3.2.2)

Suggests testthat

Imports data.table (>= 1.9.5), quanteda, slam, SnowballC, stats, tm

Date 2017-12-16

License GPL-2

LazyData TRUE

RoxygenNote 6.0.1

URL <http://github.com/trinker/gofastr>

BugReports <http://github.com/trinker/gofastr/issues>

NeedsCompilation no

Author Tyler Rinker [aut, cre]

Repository CRAN

Date/Publication 2017-12-16 21:39:50 UTC

R topics documented:

as_dtm	2
filter_documents	3
filter_tf_idf	4
filter_words	5
gofastr	6
partial_republican_debates_2015	6
presidential_debates_2012	7
q_dtm	7

q_tdm	9
remove_stopwords	10
select_documents	11
sub_in_na	11

Index	13
--------------	-----------

as_dtm	<i>Coerce Various Object Into a DocumentTermMatrix/TermDocumentMatrix</i>
--------	---

Description

Convenience functions to convert a objects from different packages into either a `tm::DocumentTermMatrix` or `tm::TermDocumentMatrix` object. Grouping variables are used as the row/column names for the `DocumentTermMatrix/TermDocumentMatrix`.

Usage

```
as_dtm(x, weighting = tm::weightTf, docs = NULL, pos = TRUE, ...)
```

```
as_tdm(x, weighting = tm::weightTf, docs = NULL, pos = TRUE, ...)
```

Arguments

x	A data object.
weighting	A weighting function capable of handling a <code>tm::DocumentTermMatrix</code> . It defaults to <code>weightTf</code> for term frequency weighting. Available weighting functions shipped with the tm package are <code>weightTf</code> , <code>weightTfIdf</code> , <code>weightBin</code> , and <code>weightSMART</code> .
docs	The vector of integers or character strings denoting document columns.
pos	logical. If TRUE parts of speech will be used. If FALSE the corresponding tokens will be used.
...	ignored.

Value

Returns a `tm::DocumentTermMatrix` or `tm::TermDocumentMatrix` object.

Examples

```
with(partial_republican_debates_2015,
     as_dtm(dialogue, paste(location, element_id, sentence_id, sep = "_"))
)

as_dtm(mtcars)
as_dtm(CO2, docs = c('Plant', 'Type', 'Treatment'))
## Not run:
## termco object to DTM/TDM
```

```

library(termco)
as_dtm(markers)
as_dtm(markers,weighting = tm::weightTfIdf)
as_tdm(markers)

cosine_distance <- function (x, ...) {
  x <- t(slam::as.simple_triplet_matrix(x))
  stats::as.dist(1 - slam::crossprod_simple_triplet_matrix(x)/(sqrt(slam::col_sums(x^2) %*%
    t(slam::col_sums(x^2))))))
}

mod <- hclust(cosine_distance(as_dtm(markers)))
plot(mod)
rect.hclust(mod, k = 5, border = "red")

(clusters <- cutree(mod, 5))

## Parts of speech to DTM/TDM
library(tagger)
library(dplyr)
data(presidential_debates_2012_pos)

pos <- presidential_debates_2012_pos %>%
  select_tags(c("NN", "NNP", "NNPS", "NNS"))

as_dtm(pos_text)
as_dtm(pos_text, pos=FALSE)

as_tdm(pos_text)
as_tdm(pos_text, pos=FALSE)

presidential_debates_2012_pos %>%
  as_basic() %>%
  as_dtm()

## End(Not run)

```

filter_documents	<i>Remove Documents Below a Threshold from a TermDocumentMatrix/DocumentTermMatrix</i>
------------------	--

Description

Remove documents from a [TermDocumentMatrix](#) or [DocumentTermMatrix](#) not meeting a [rowSums/colSums](#) threshold. Useful for removing empty documents.

Usage

```
filter_documents(x, min = 1)
```

Arguments

`x` A [TermDocumentMatrix](#) or [DocumentTermMatrix](#).
`min` A minimal threshold that a documents row/column must sum to.

Value

Returns a [TermDocumentMatrix](#) or [DocumentTermMatrix](#).

Examples

```
(x <-with(presidential_debates_2012, q_dtm(dialogue, paste(time, tot, sep = "_"))))
filter_documents(x)
(y <- with(presidential_debates_2012, q_tdm(dialogue, paste(time, tot, sep = "_"))))
filter_documents(y)
```

filter_tf_idf	<i>Remove Words Below a TF-IDF Threshold from a TermDocumentMatrix/DocumentTermMatrix</i>
---------------	---

Description

Remove words from a [TermDocumentMatrix](#) or [DocumentTermMatrix](#) not meeting a tf-idf threshold. Code is based on Gruen & Hornik's (2011) code but allows for easier chaining and extends the filtering to a [TermDocumentMatrix](#). This can be used to remove words that appear too frequently in a corpus, therefore these words do not carry much information.

Usage

```
filter_tf_idf(x, min = NULL, verbose = FALSE)
```

Arguments

`x` A [TermDocumentMatrix](#) or [DocumentTermMatrix](#).
`min` A minimal threshold that a word tf-idf must exceed. If `min = NULL` the median of the tf-idf will be used.
`verbose` logical. If TRUE the summary stats from the tf-idf are printed. This can be useful for exploration and setting the min value.

Value

Returns a [TermDocumentMatrix](#) or [DocumentTermMatrix](#).

Author(s)

Bettina Gr"un, Kurt Hornik, and Tyler Rinker <tyler.rinker@gmail.com>.

References

Bettina Gruen & Kurt Hornik (2011). topicmodels: An R Package for Fitting Topic Models. Journal of Statistical Software, 40(13), 1-30. <http://www.jstatsoft.org/article/view/v40i13/v40i13.pdf>

Examples

```
(x <-with(presidential_debates_2012, q_dtm(dialogue, paste(person, time, sep = "_"))))
filter_tf_idf(x)
filter_tf_idf(x, .5)
filter_tf_idf(x, verbose=TRUE)
(y <- with(presidential_debates_2012, q_tdm(dialogue, paste(person, time, sep = "_"))))
filter_tf_idf(y)
```

filter_words	<i>Remove Words Below a Threshold from a TermDocumentMatrix/DocumentTermMatrix</i>
--------------	--

Description

Remove words from a [TermDocumentMatrix](#) or [DocumentTermMatrix](#) not meeting a [rowSums/colSums](#) threshold.

Usage

```
filter_words(x, min = 1)
```

Arguments

x	A TermDocumentMatrix or DocumentTermMatrix .
min	A minimal threshold that a words row/column must sum to.

Value

Returns a [TermDocumentMatrix](#) or [DocumentTermMatrix](#).

Examples

```
(x <-with(presidential_debates_2012, q_dtm(dialogue, paste(time, tot, sep = "_"))))
filter_words(x)
filter_words(x, 5)
(y <- with(presidential_debates_2012, q_tdm(dialogue, paste(time, tot, sep = "_"))))
filter_words(y, 6)
```

gofastr

Fast DocumentTermMatrix and TermDocumentMatrix Creation

Description

This package does one thing...It harness the power of **quanteda**, **data.table** & **stringi** to quickly generate **tm** [TermDocumentMatrix](#) & [DocumentTermMatrix](#) data structures without creating a [Corpus](#) first.

partial_republican_debates_2015

2015 U.S. Partial Republican Primary Presidential Debates

Description

A dataset containing a cleaned version of four primary presidential debates for the 2016 election.

Usage

```
data(partial_republican_debates_2015)
```

Format

A data frame with 7405 rows and 5 variables

Details

- location. Where debate took place
- person. The speaker
- dialogue. The words spoken
- element_id. Original line number (turn of talk) within location
- sentence_id. Sentence number within element_id

References

<http://www.presidency.ucsb.edu>

presidential_debates_2012

2012 U.S. Presidential Debates

Description

A dataset containing a cleaned version of all three presidential debates for the 2012 election.

Usage

```
data(presidential_debates_2012)
```

Format

A data frame with 2912 rows and 4 variables

Details

- person. The speaker
- tot. Turn of talk
- dialogue. The words spoken
- time. Variable indicating which of the three debates the dialogue is from

q_dtm

Quick DocumentTermMatrix

Description

Make a [DocumentTermMatrix](#) from a vector of text and an optional vector of documents. To stem a document as well use the `q_dtm_stem` version of `q_dtm` which uses **SnowballC**'s `wordStem`.

Usage

```
q_dtm(text, docs = seq_along(text), to = "tm", keep.hyphen = FALSE,  
      ngrams = NULL, ...)
```

```
q_dtm_stem(text, docs = seq_along(text), to = "tm", keep.hyphen = FALSE,  
           ngrams = NULL, ...)
```

Arguments

text	A vector of strings.
docs	A vector of document names.
to	target conversion format, consisting of the name of the package into whose document-term matrix representation the dfm will be converted: "lda" a list with components "documents" and "vocab" as needed by <code>lda.collapsed.gibbs.sampler</code> from the lda package "tm" a DocumentTermMatrix from the tm package "stm" the format for the stm package "austin" the wfm format from the austin package "topicmodels" the "dtm" format as used by the topicmodels package
keep.hyphen	logical. If TRUE hyphens are retained in the terms (e.g., "math-like" is kept as "math-like"), otherwise they become a split for terms (e.g., "math-like" is converted to "math" & "like").
ngrams	A vector of ngrams (multiple wrds with spaces). Using this option results in the ngrams that will be retained in the matrix.
...	Additional arguments passed to dfm .

Value

Returns a [DocumentTermMatrix](#).

See Also

[dfm](#), [convert](#)

Examples

```
(x <- with(presidential_debates_2012, q_dtm(dialogue, paste(time, tot, sep = "_"))))
tm::weightTfIdf(x)

(x2 <- with(presidential_debates_2012, q_dtm_stem(dialogue, paste(time, tot, sep = "_"))))
remove_stopwords(x2, stem=TRUE)

bigrams <- c('make sure', 'governor romney', 'mister president',
            'united states', 'middle class', 'middle east', 'health care',
            'american people', 'dodd frank', 'wall street', 'small business')

grep(" ", x$dimnames$Terms, value = TRUE) #no ngrams

(x3 <- with(presidential_debates_2012,
            q_dtm(dialogue, paste(time, tot, sep = "_"), ngrams = bigrams)
))

grep(" ", x3$dimnames$Terms, value = TRUE) #ngrams
```


q_tdm

*Quick TermDocumentMatrix***Description**

Make a [TermDocumentMatrix](#) from a vector of text and an optional vector of documents. To stem a document as well use the `q_tdm_stem` version of `q_tdm` which uses **SnowballC**'s `wordStem`.

Usage

```
q_tdm(text, docs = seq_along(text), to = "tm", keep.hyphen = FALSE,
      ngrams = NULL, ...)
```

```
q_tdm_stem(text, docs = seq_along(text), to = "tm", keep.hyphen = FALSE,
           ngrams = NULL, ...)
```

Arguments

<code>text</code>	A vector of strings.
<code>docs</code>	A vector of document names.
<code>to</code>	target conversion format, consisting of the name of the package into whose document-term matrix representation the <code>dfm</code> will be converted: " <code>lda</code> " a list with components "documents" and "vocab" as needed by <code>lda.collapsed.gibbs.sampler</code> from the lda package " <code>tm</code> " a DocumentTermMatrix from the tm package " <code>stm</code> " the format for the stm package " <code>austin</code> " the <code>wfm</code> format from the austin package " <code>topicmodels</code> " the " <code>dtm</code> " format as used by the topicmodels package
<code>keep.hyphen</code>	logical. If TRUE hyphens are retained in the terms (e.g., "math-like" is kept as "math-like"), otherwise they become a split for terms (e.g., "math-like" is converted to "math" & "like").
<code>ngrams</code>	A vector of ngrams (multiple wrds with spaces). Using this option results in the ngrams that will be retained in the matrix.
<code>...</code>	Additional arguments passed to dfm

Examples

```
(x <- with(presidential_debates_2012, q_tdm(dialogue, paste(time, tot, sep = "_"))))
tm::weightTfIdf(x)
```

```
(x2 <- with(presidential_debates_2012, q_tdm_stem(dialogue, paste(time, tot, sep = "_"))))
remove_stopwords(x2, stem=TRUE)
```

remove_stopwords	<i>Remove Stopwords from a TermDocumentMatrix/DocumentTermMatrix</i>
------------------	--

Description

remove_stopwords - Remove stopwords and <nchar words from a [TermDocumentMatrix](#) or [DocumentTermMatrix](#).

prep_stopwords - Join multiple vectors of words, convert to lower case, and return sorted unique words.

Usage

```
remove_stopwords(x, stopwords = tm::stopwords("english"), min.char = 3,
  max.char = NULL, stem = FALSE, denumber = TRUE)
```

```
prep_stopwords(...)
```

Arguments

x	A TermDocumentMatrix or DocumentTermMatrix .
stopwords	A vector of stopwords to remove.
min.char	The minimal length character for retained words.
max.char	The maximum length character for retained words.
stem	Logical. If TRUE the stopwords will be stemmed.
denumber	Logical. If TRUE numbers will be excluded.
...	vectors of words.

Value

Returns a [TermDocumentMatrix](#) or [DocumentTermMatrix](#).

Examples

```
(x <-with(presidential_debates_2012, q_dtm(dialogue, paste(time, tot, sep = "_"))))
remove_stopwords(x)
(y <- with(presidential_debates_2012, q_tdm(dialogue, paste(time, tot, sep = "_"))))
remove_stopwords(y)

prep_stopwords("the", "ChIcken", "Hello", tm::stopwords("english"), c("John", "Josh"))
```

select_documents	<i>Select Documents rom a TermDocumentMatrix/DocumentTermMatrix</i>
------------------	---

Description

Select documents from a [TermDocumentMatrix](#) or [DocumentTermMatrix](#) matching a regular expression.

Usage

```
select_documents(x, pattern, invert = FALSE, ...)
```

Arguments

x	A TermDocumentMatrix or DocumentTermMatrix .
pattern	A regex pattern used to select documents.
invert	logical. If TRUE the pattern is inverted to exclude these documents.
...	Other arguments passed to grepl (<code>perl = TRUE</code> is hard coded).

Value

Returns a [TermDocumentMatrix](#) or [DocumentTermMatrix](#).

Examples

```
(x <-with(presidential_debates_2012, q_dtm(dialogue, paste(time, person, sep = "_"))))
select_documents(x, 'romney', ignore.case=TRUE)
select_documents(x, '^(?!.*romney).*$', ignore.case = TRUE)      # regex way to invert
select_documents(x, 'romney', ignore.case = TRUE, invert = TRUE) # easier way to invert
(y <- with(presidential_debates_2012, q_tdm(dialogue, paste(time, person, sep = "_"))))
select_documents(y, '[2-3]')
```

sub_in_na	<i>Regex Sub to Missing</i>
-----------	-----------------------------

Description

Use a regex to identify elements to sub out for missing NA. Useful within a **magrittr** pipeline before producing the [TermDocumentMatrix](#) or [DocumentTermMatrix](#).

Usage

```
sub_in_na(x, regex = "[^A-Za-z]*$", ...)
```

Arguments

x A vector of text strings.
regex A regex to match strings in a vector.
... Other arguments passed to [grepl](#)

Value

Returns a vector with NAs inserted.

Examples

```
x <- c("45", "..", "", " ", "dog")
sub_in_na(x)
sub_in_na(x, "\\s*$")

## Not run:
library(tidyverse)
x %>%
  q_dtm() %>%
  as.matrix()

x %>%
  sub_in_na() %>%
  q_dtm() %>%
  as.matrix()

## End(Not run)
```

Index

- * **DocumentTermMatrix**
 - q_dtm, 7
 - * **TermDocumentMatrix**
 - q_tdm, 9
 - * **datasets**
 - partial_republican_debates_2015, 6
 - presidential_debates_2012, 7
 - * **documenttermmatrix,**
 - as_dtm, 2
 - * **dtm**
 - q_dtm, 7
 - * **stopwords**
 - remove_stopwords, 10
 - * **tdm**
 - q_tdm, 9
 - * **termdocumentmatrix**
 - as_dtm, 2
- as_dtm, 2
as_tdm (as_dtm), 2
- colSums, 3, 5
convert, 8
Corpus, 6
- dfm, 8, 9
DocumentTermMatrix, 3–11
- filter_documents, 3
filter_tf_idf, 4
filter_words, 5
- gofastr, 6
gofastr-package (gofastr), 6
grepl, 11, 12
- package-gofastr (gofastr), 6
partial_republican_debates_2015, 6
prep_stopwords (remove_stopwords), 10
presidential_debates_2012, 7
- q_dtm, 7
q_dtm_stem (q_dtm), 7
q_tdm, 9
q_tdm_stem (q_tdm), 9
- remove_stopwords, 10
rowSums, 3, 5
- select_documents, 11
sub_in_na, 11
- TermDocumentMatrix, 3–6, 9–11
- vector, 10
- wordStem, 7, 9