Package 'textfeatures'

October 14, 2022

Type Package

Title Extracts Features from Text

Version 0.3.3

Description A tool for extracting some generic features (e.g., number of words, line breaks, characters per word, URLs, lower case, upper case, commas, periods, exclamation points, etc.) from strings of text.

License MIT + file LICENSE

URL https://github.com/mkearney/textfeatures

BugReports https://github.com/mkearney/textfeatures/issues

Depends R (>= 3.1.0)

Imports dplyr, purrr, rlang, text2vec, tfse, tibble, tokenizers, utils, stats

Suggests knitr, roxygen2, testthat, covr

Encoding UTF-8

LazyData yes

RoxygenNote 6.1.1

NeedsCompilation no

Author Michael W. Kearney [aut, cre] (<https://orcid.org/0000-0002-0730-4694>), Emil Hvitfeldt [ctb] (<https://orcid.org/0000-0002-0679-1945>)

Maintainer Michael W. Kearney <kearneymw@missouri.edu>

Repository CRAN

Date/Publication 2019-09-03 21:10:02 UTC

R topics documented:

count_functions *List of all feature counting functions*

Description

List of all feature counting functions

Usage

```
count_functions
```

Format

Named list of all ferature counting functions

`n_words` Number of words.

`n_uq_words` Number of unique words.

`n_charS` Number of characters. Not counting urls, hashtags, mentions or white spaces.

`n_uq_charS` Number of unique characters. Not counting urls, hashtags, mentions or white spaces.

`n_digits` Number of digits.

`n_hashtags` Number of hashtags, word preceded by a '#'.

`n_uq_hashtags` Number of unique hashtags, word preceded by a '#'.

`n_mentions` Number of mentions, word preceded by a '@'.

`n_uq_mentions` Number of unique mentions, word preceded by a '@'.

`n_commas` Number of commas.

`n_periods` Number of periods.

`n_exclaims` Number of exclamation points.

`n_extraspaces` Number of times more then 1 consecutive space have been used.

`n_caps` Number of upper case characters.

`n_lowers` Number of lower case characters.

`n_urls` Number of urls.

`n_uq_urls` Number of unique urls.

`n_nonasciis` Number of non ascii characters.

`n_puncts` Number of punctuations characters, not including exclamation points, periods and commas.

`politeness` Summed sentiment value calculated using politeness_dict.

`first_person` Number of "first person" words.

`first_personp` Number of "first person plural" words.

`second_person` Number of "second person" words.

`second_personp` Number of "second person plural" words.

`third_person` Number of "third person" words.

`to_be` Number of "to be" words.

`prepositions` Number of preposition words.

Details

In this function we refer to "first person", "first person plural" and so on. This list describes what words are contained in each group.

first person I, me, myself, my, mine, this.

first person plural we, us, our, ours, these.

second person you, yours, your, yourself.

second person plural he, she, it, its, his, hers.

third person they, them, theirs, their, they're, their's, those, that.

to be am, is, are, was, were, being, been, be, were, be.

prepositions about, below, excepting, off, toward, above, beneath, on, under, across, from, onto, underneath, after, between, in, out, until, against, beyond, outside, up, along, but, inside, over, upon, among, by, past, around, concerning, regarding, with, at, despite, into, since, within, down, like, through, without, before, during, near, throughout, behind, except, of, to, for.

`scale_count` *Apply various transformations to numeric (and non-id) data*

Description

scale_count: Transforms integer and integerlike columns using log

scale_log: Transforms numeric columns using log

scale_normal: Transforms numeric columns using mean centering and dividing by standard deviation

scale_standard: Transforms numeric columns onto 0-1 scales with 0 and 1 set empirically

scale_sqrt: Transforms numeric columns using sqrt

Usage

```
scale_count(x)

scale_log(x)

scale_normal(x)

scale_standard(x)

scale_sqrt(x)
```

Arguments

`x`	Input data frame containing numeric columns.

Details

Scale transformations are applied only to numeric (or in the case of `scale_count` only integer or integerish) columns that are not named `"id"` or `"(\.|_)?id"`.

Value

A data frame with the same dimensions but with the numeric/relevant variables transformed.

`textfeatures` *textfeatures*

Description

Extracts features from text vector.

Usage

```
textfeatures(text, sentiment = TRUE, word_dims = NULL,
  normalize = TRUE, newdata = NULL, verbose = TRUE)
```

Arguments

`text`	Input data. Should be character vector or data frame with character variable of interest named "text". If a data frame then the first "id|*_id" variable, if found, is assumed to be an ID variable.
`sentiment`	Logical, indicating whether to return sentiment analysis features, the variables `sent_afinn` and `sent_bing`. Defaults to TRUE. Setting this to FALSE will speed things up a bit.
`word_dims`	Integer indicating the desired number of word2vec dimension estimates. When NULL, the default, this function will pick a reasonable number of dimensions (ranging from 2 to 200) based on size of input. To disable word2vec estimates, set this to 0 or FALSE.
`normalize`	Logical indicating whether to normalize (mean center, sd = 1) features. Defaults to TRUE.
`newdata`	If a textfeatures_model is supplied to text, supply this with new data to which you would like to apply the textfeatures_model.
`verbose`	A single logical for printing logging messages as work progresses.

Value

A tibble data frame with extracted features as columns.

Examples

```
## the text of five of Trump's most retweeted tweets
trump_tweets <- c(
  "#FraudNewsCNN #FNN https://t.co/WYUnHjjUjg",
  "TODAY WE MAKE AMERICA GREAT AGAIN!",
  paste("Why would Kim Jong-un insult me by calling me \"old,\" when I would",
    "NEVER call him \"short and fat?\" Oh well, I try so hard to be his",
    "friend - and maybe someday that will happen!"),
  paste("Such a beautiful and important evening! The forgotten man and woman",
    "will never be forgotten again. We will all come together as never before"),
  paste("North Korean Leader Kim Jong Un just stated that the \"Nuclear",
    "Button is on his desk at all times.\" Will someone from his depleted and",
    "food starved regime please inform him that I too have a Nuclear Button,",
    "but it is a much bigger &amp;amp; more powerful one than his, and my Button",
    "works!")
)

## get the text features of a character vector
textfeatures(trump_tweets)

## data frame with a character vector named "text"
df <- data.frame(
  id = c(1, 2, 3),
  text = c("this is A!\t sEntence https://github.com about #rstats @github",
    "and another sentence here",
    "The following list:\n- one\n- two\n- three\nOkay!?!"),
  stringsAsFactors = FALSE
)

## get text features of a data frame with "text" variable
textfeatures(df)
```

---

word_dims_newtext *Calculates word2vec dimension estimates*

Description

Calculates word2vec dimension estimates

Usage

```
word_dims_newtext(lda_model, text, n_iter = 20)

word_dims(text, n = 10, n_iter = 20)
```

Arguments

lda_model	A pretrained LDA model from **text2vec**.
text	Input data. Should be character vector.
n_iter	Integer, number of sampling iterations.
n	Integer, determines the number of latent topics.

Value

A tibble data frame

Examples

```
trump_tweets <- c(
"#FraudNewsCNN #FNN https://t.co/WYUnHjjUjg",
"TODAY WE MAKE AMERICA GREAT AGAIN!",
paste("Why would Kim Jong-un insult me by calling me \"old,\" when I would",
      "NEVER call him \"short and fat?\" Oh well, I try so hard to be his",
      "friend - and maybe someday that will happen!"),
paste("Such a beautiful and important evening! The forgotten man and woman",
      "will never be forgotten again. We will all come together as never before"),
paste("North Korean Leader Kim Jong Un just stated that the \"Nuclear",
      "Button is on his desk at all times.\" Will someone from his depleted and",
      "food starved regime please inform him that I too have a Nuclear Button,",
      "but it is a much bigger &amp;amp; more powerful one than his, and my Button",
      "works!")
)
word_dims(trump_tweets)
```

Index