Package ‘texter’

October 14, 2022

Title An Easy Text and Sentiment Analysis Library
Version 0.1.9
Maintainer Simi Kafaru <kafarusimileoluwa@gmail.com>
Description Implement text and sentiment analysis with ‘texter’.
Generate sentiment scores on text data and also visualize sentiments.
‘texter’ allows you to quickly generate insights on your data.
It includes support for lexicons such as ‘NRC’ and ‘Bing’.
License MIT + file LICENSE
URL https://github.com/simmieyungie/texter
BugReports https://github.com/simmieyungie/texter/issues
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1
Imports dplyr, plyr, ggplot2, magrittr, stringr, purrr, stopwords,
textdata, tidytext, tidyr
Depends R (>= 2.14)
Suggests testthat (>= 3.0.0)
Config/testthat/edition 3
NeedsCompilation no
Author Simi Kafaru [aut, cre]
Repository CRAN
Date/Publication 2021-09-20 14:20:02 UTC

R topics documented:

brexit ................................................................. 2
counter ............................................................ 2
doge ................................................................. 3
nrc ................................................................. 3
**Description**

This is the first data to be included in my package

**Author(s)**

SimiKafaru <kafarusimileoluwa@gmail.com>

---

**counter**

Get the number of times a vector of words occurs

**Description**

This function retrieves the number of times each word in a corpus occurs. It returns a dataframe containing the word and the corresponding counts

**Usage**

```
counter(word_vec, words)
```

**Arguments**

- `word_vec` This is the corpus you want to the word frequency extracted from
- `words` This is a vector of words you want to retrieve their frequency counts

**Value**

a data frame object. A data frame object of strings and their corresponding count
This is the first data to be included in my package

Description

It contains tweets on doge coin collected using twitter API

Author(s)

SimiKafaru <kafarusimileoluwa@gmail.com>

This data was saved NRC word-emotion association lexicon

Description

The dataset is saved from the textdata [https://github.com/EmilHvitfeldt/textdata/blob/master/R/lexicon_nrc.R](https://github.com/EmilHvitfeldt/textdata/blob/master/R/lexicon_nrc.R) for easier access

Value

A tibble with 13,901 rows and 4 variables:

- word  An English word
- sentiment Indicator for sentiment or emotion: "negative", "positive", "anger", "anticipation", "disgust", "fear", "joy", "sadness", "surprise", or "trust"

Source

[http://saifmohammad.com/WebPages/lexicons.html](http://saifmohammad.com/WebPages/lexicons.html)

Easily remove Punctuation from Text

Description

This function will help you remove punctuation and numbers from your text easily

Usage

`removeNumPunct(x)`

Arguments

- `x`  is the text column you want the punctuation and texts removed from
sentimentAnalyzer

Value

a character vector.

Examples

{
  removeNumPunct("is this your number? 01234")
}

removeURL

A function to help you remove URLs from text

Description

This function helps remove URLs from text, particularly designed for tweets

Usage

removeURL(x)

Arguments

x

is the text value you want to extract the texts from

Value

a character vector.

sentimentAnalyzer

Get the overall weight of emotions conveyed in a corpus

Description

This function will help you extract the weight of emotions conveyed in a tweet

Usage

sentimentAnalyzer(word_vec, details)

Arguments

word_vec

This is the corpus you want to extract the sentiments from

details

(A TRUE/FALSE value): If TRUE you get a more robust distribution of these emotions. FALSE is summarised as Positive or Negative
Value

a data frame object. A data frame of each emotions and their corresponding weight in text

Examples

    sentimentAnalyzer(doge$text, details = TRUE)

---

stop_words

Saved stop_word dataframe from tidytext

Description

it contains stop_words from tidytext package. It is saved for easier access from the tidytext package

Author(s)

tidytext

top_bigrams

Get the top bigrams from text Get the top n bigrams from vector of text

Description

This function is used to get the top N bigrams from a corpus. It will retrieve the most occurring two combinations based on frequency

Usage

top_bigrams(word_vec, remove_these, bigram_size)

Arguments

word_vec This is the corpus you want to extract the sentiments from
remove_these This is a vector of characters you want cleaned out of the text
bigram_size This is the Top N number of rows to be retrieved as an integer value

Value

a data frame object.

Examples

    {
        top_bigrams(brexit[, c("content")], remove_these = c("rt"), bigram_size = 20)
    }
### top_Sentiments

**Get the top 10 negative and positive words**

**Description**

This function returns the top 10 positive and negative words expressed in a text. By default, a data frame of words classified as positive or negative based on weights.

**Usage**

```r
top_Sentiments(word_vec, plot)
```

**Arguments**

- **word_vec**: This is the corpus you want to extract the sentiments from.
- **plot**: (TRUE/FALSE) TRUE means you want to return a plot which you can further customize. FALSE means a dataframe will be returned.

**Value**

- a data frame object if plot is FALSE.
- a ggplot object if plot = TRUE.

**Examples**

```r
top_Sentiments(doge$text, plot = TRUE)
```

### top_words

**Get the top n words from vector of text**

**Description**

This function is used to get the top N words from a corpus. It will retrieve the most occurring words based on frequency.

**Usage**

```r
top_words(word_vec, remove_these, size)
```

**Arguments**

- **word_vec**: This is the corpus you want to extract the sentiments from.
- **remove_these**: This is a vector of characters you want cleaned out of the text.
- **size**: This is the Top N number of rows to be retrieved as an integer value.
top_words_Retriever

Value

A data frame object.

Examples

```
{  
    top_words(brexit$content, remove_these = c("news","uk"), size = 10)  
}
```

Description

This function helps to search for the top n words but only based texts or rows containing a key word. It is particularly useful when you want to search the top n words revolving around a certain keyword.

Usage

```
top_words_Retriever(word_vec, word_ret, remove_these, size)  
```

Arguments

- **word_vec**: This is the corpus you want to extract the sentiments from.
- **word_ret**: Is the key word you want searched.
- **remove_these**: Is a vector of characters you want cleaned out of the text.
- **size**: Is the N number of rows to be retrieved as an integer value.

Value

A data frame object.

Examples

```
{  
    top_words_Retriever(brexit$content, word_ret = "brexit", remove_these = c("news","uk"), size = 10)  
}
```
users Extract Usernames and tagged handles from tweets

Description

The function will extract any tagged handles from text.

Usage

users(x, ...)

Arguments

x This is the corpus you want to extract the mentions from
...

More inputs

Value

a character vector.

Examples

{
  users("Come See this @simmie_kafaru")
}

Index

* data
  brexit, 2
  doge, 3
  stop_words, 5
brexit, 2
counter, 2
doge, 3
nrc, 3
removeNumPunct, 3
removeURL, 4
sentimentAnalyzer, 4
stop_words, 5
top_bigrams, 5
top_Sentiments, 6
top_words, 6
top_words_Retriever, 7
users, 8