Package 'RSentiment'

July 27, 2018

Type Package

Title Analyse Sentiment of English Sentences

Version 2.2.2

Imports plyr,stringr,openNLP,NLP

Date 2018-07-27

Author Subhasree Bose <subhasree10.7@gmail.com> with contributons from Saptarsi Goswami.

Maintainer Subhasree Bose <subhasree10.7@gmail.com>

Description Analyses sentiment of a sentence in English and assigns score to it. It can classify sentences to the following categories of sentiments:- Positive, Negative, very Positive, very negative, Neutral. For a vector of sentences, it counts the number of sentences in each category of sentiment.In calculating the score, negation and various degrees of adjectives are taken into consideration. It deals only with English sentences.

Depends R (>= 2.1.0)

License GPL-2

LazyData true

RoxygenNote 5.0.1

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2018-07-27 15:20:02 UTC

R topics documented:

`calculate_custom_score` *Calculate the score of sentences*

Description

This function loads text and calculates score of each sentence on basis of presence of words of positive and negative sentiment supplied externally as paramater, presence of negation,and checking for sarcasm. 0 indicates neutral sentiment. Positive value indicates positive sentiment. Negative value indicates negative sentiment. 99 indicates sarcasm.

Usage

```
calculate_custom_score(text, positivewords, negativewords, check = 0)
```

Arguments

`text`	A vector of sentences or a sentence (English).
`positivewords`	A vector of words of positive sentiment.
`negativewords`	A vector of words of negative sentiment.
`check`	A flag variable whose value if 1 denotes appending of passed positive and negative words with existing words.By default, this method will work with supplied positive and negative words only.

Value

A vector containing polarity of each sentence.

Examples

```
calculate_custom_score("This is good",c("good"),c("bad"),1)
calculate_custom_score(c("This is good","This is bad"),c("good"),c("bad"),0)
```

`calculate_custom_sentiment` *Predicts the sentiment of sentences*

Description

This function loads text and words of positive and negative sentiment supplied externally as paramater and calculates sentiment of each sentence. It classifies sentences into 6 categories: Positive, Negative, Very Positive, Very Negative Sarcasm and Neutral.

Usage

```
calculate_custom_sentiment(text, positivewords, negativewords, check = 0)
```

Arguments

text	A vector of sentences or a sentence (English).
positivewords	A vector of words of positive sentiment.
negativewords	A vector of words of negative sentiment.
check	A flag variable whose value if 1 denotes appending of passed positive and negative words with existing words.By default, this method will work with supplied positive and negative words only.

Value

A vector containing sentiment of each sentence.

Examples

```
calculate_custom_sentiment("This is good",c("good"),c("bad"),1)
calculate_custom_sentiment(c("This is good","This is bad"),c("good"),c("bad"),0)
```

---

calculate_custom_total_presence_sentiment

Calculate the number of sentences in each category of sentiment.

Description

This function loads text and words of positive and negative sentiment supplied externally as paramater, and calculates number of sentences which are positive, negative, very positive, very negative, neutral and sarcasm.

Usage

```
calculate_custom_total_presence_sentiment(text, positivewords, negativewords,
  check = 0)
```

Arguments

text	A vector of sentences or a sentence (English).
positivewords	A vector of words of positive sentiment.
negativewords	A vector of words of negative sentiment.
check	A flag variable whose value if 1 denotes appending of passed positive and negative words with existing words.By default, this method will work with supplied positive and negative words only.

Value

A 2-D matrix with two rows and 6 columns where first row contains the name of sentiment category and the second row contains the number in each category in string format.

Examples

```
calculate_custom_total_presence_sentiment(c("This is good","This is bad"),c("good"),c("bad"),0)
```

`calculate_score` *Calculate the score of sentences*

Description

This function loads text and calculates score of each sentence on basis of presence of words of positive and negative sentiment, presence of negation, and checking for sarcasm. 0 indicates neutral sentiment. Positive value indicates positive sentiment. Negative value indicates negative sentiment. 99 indicates sarcasm.

Usage

```
calculate_score(text)
```

Arguments

`text`	A vector of sentences or a sentence (English).

Value

A vector containing polarity of each sentence.

Examples

```
calculate_score("This is good")
calculate_score(c("This is good","This is bad"))
```

calculate_sentiment *Predicts the sentiment of sentences*

Description

This function loads text and calculates sentiment of each sentence. It classifies sentences into 6 categories: Positive, Negative, Very Positive, Very Negative Sarcasm and Neutral.

Usage

```
calculate_sentiment(text)
```

Arguments

text	A vector of sentences or a sentence (English).

Value

A vector containing sentiment of each sentence.

Examples

```
calculate_sentiment("This is good")
calculate_sentiment(c("This is good","This is bad"))
```

calculate_total_presence_sentiment *Calculate the number of sentences in each category of sentiment.*

Description

This function loads text and calculates number of sentences which are positive, negative, very positive, very negative, neutral and sarcasm.

Usage

```
calculate_total_presence_sentiment(text)
```

Arguments

text	A vector of sentences or a sentence (English).

Value

A 2-D matrix with two rows and 6 columns where first row contains the name of sentiment category and the second row contains the number in each category in string format.

Examples

```
calculate_total_presence_sentiment(c("This is good","This is bad"))
```

Index