Package ‘syuzhet’

February 22, 2015

Type Package

Title Extracts Sentiment and Sentiment-Derived Plot Arcs from Text

Version 0.2.0

Date 2015-01-20

Maintainer Matthew Jockers <mjockers@gmail.com>

Description Extracts sentiment and sentiment-derived plot arcs
from text using three sentiment dictionaries conveniently
packaged for consumption by R users. Implemented dictionaries include
``afinn'' developed by Finn \{AA}rup Nielsen, ``bing'' developed by Mingqing Hu
and Bing Liu, and ``nrc'' developed by Mohammad, Saif M. and Turney, Peter D.
Applicable references are available in README.md and in the documentation
for the ``get_sentiment'' function. The package also provides a method for
implementing Stanford's coreNLP sentiment parser. The package provides
several methods for plot arc normalization.

URL https://github.com/mjockers/syuzhet

License GPL-3

Imports openNLP, NLP

LazyData true

Suggests knitr, pander, testthat (>= 0.9.1)

NeedsCompilation no

VignetteBuilder knitr

Author Matthew Jockers [aut, cre]

Repository CRAN

Date/Publication 2015-02-22 00:24:40

R topics documented:

  get_nrc_sentiment ............................................................. 2
  get_nrc_values ................................................................. 3
  get_percentage_values ....................................................... 3
get_nrc_sentiment

Get Emotions and Valence from NRC Dictionary

Description

Calls the NRC sentiment dictionary to calculate the presence of eight different emotions and their corresponding valence in a text file.

Usage

get_nrc_sentiment(char_v)

Arguments

char_v A character vector

Value

A data frame where each row represents a sentence from the original file. The columns include one for each emotion type as well as a positive or negative valence. The ten columns are as follows: "anger", "anticipation", "disgust", "fear", "joy", "sadness", "surprise", "trust", "negative", "positive."

References

**get_nrc_values**

**Summarize NRC Values**

**Description**
Access the NRC dictionary to compute emotion types and valence for a set of words in the input vector.

**Usage**

get_nrc_values(word_vector)

**Arguments**

word_vector A character vector.

**Value**
A vector of values for the emotions and valence detected in the input vector.

**get_percentage_values**

**Chunk a Text and Get Means**

**Description**
Chunks text into 100 Percentage based segments and calculates means.

**Usage**

get_percentage_values(raw_values)

**Arguments**

raw_values Raw sentiment values

**Value**
A vector of mean values from each chunk
get_sentences | Sentence Tokenization
---|---

**Description**

Parses a string into a vector of sentences.

**Usage**

```r
get_sentences(text_of_file)
```

**Arguments**

- `text_of_file` A Text String

**Value**

A Character Vector of Sentences

---

get_sentiment | Get Sentiment Values for a String
---|---

**Description**

Iterates over a vector of strings and returns sentiment values based on user supplied method.

**Usage**

```r
get_sentiment(char_v, method = c("afinn", "bing", "nrc", "stanford"),
               path_to_tagger = NULL)
```

**Arguments**

- `char_v` A vector of strings for evaluation.
- `method` A string indicating which sentiment method to use. Options include "bing", "afinn", "nrc" and "stanford." See references for more detail on methods.
- `path_to_tagger` local path to location of Stanford CoreNLP package

**Value**

Return value is a numeric vector of sentiment values, one value for each input sentence.
References


get_sent_values

Assigns Sentiment Values

description

Assigns sentiment values to words based on preloaded dictionary

Usage

get_sent_values(char_v, method = "bing")

Arguments

char_v A string
method A string indicating which sentiment dictionary to use

Value

A single numerical value (positive or negative) based on the assessed sentiment in the string
get_stanford_sentiment

*Get Sentiment from the Stanford Tagger*

**Description**

Call the Stanford Sentiment tagger with a vector of strings. The Stanford tagger automatically detects sentence boundaries and treats each sentence as a distinct instance to measure. As a result, the vector that gets returned will not be the same length as the input vector.

**Usage**

```r
get_stanford_sentiment(text_vector, path_to_stanford_tagger)
```

**Arguments**

- `text_vector`: A vector of strings
- `path_to_stanford_tagger`: a local file path indicating where the coreNLP package is installed.

get_text_as_string

*Load Text from a File*

**Description**

Loads a file as a single text string.

**Usage**

```r
get_text_as_string(path_to_file)
```

**Arguments**

- `path_to_file`: file path

**Value**

A character vector of length 1 containing the text of the file in the `path_to_file` argument.
**get_transformed_values**

*Fourier Transform and Reverse Transform Values*

**Description**

Converts input values into a standardized set of filtered and reverse transformed values for easy plotting and/or comparison.

**Usage**

```r
get_transformed_values(raw_values, low_pass_size = 3, x_reverse_len = 100, scale_vals = FALSE, scale_range = FALSE)
```

**Arguments**

- `raw_values`: the raw sentiment values calculated for each sentence
- `low_pass_size`: The number of components to retain in the low pass filtering. Default = 3
- `x_reverse_len`: the number of values to return. Default = 100
- `scale_vals`: Logical determines whether or not to normalize the values using the scale function. Default = FALSE. If TRUE, values will be scaled by subtracting the means and scaled by dividing by their standard deviations. See ?scale
- `scale_range`: Logical determines whether or not to scale the values from -1 to +1. Default = FALSE. If set to TRUE, the lowest value in the vector will be set to -1 and the highest values set to +1 and all the values scaled accordingly in between.

**Value**

The transformed values

**Examples**

```r
s_v <- get_sentences("I begin this story with a neutral statement. Now I add a statement about how much I despise cats. I am allergic to them. Basically this is a very silly test.")
raw_values <- get_sentiment(s_v, method = "bing")
get_transformed_values(raw_values)
```
rescale  Vector Value Rescaling

Description
Rescale Transformed values from -1 to 1

Usage
rescale(x)

Arguments
x  A vector of values
Index

get_nrc_sentiment, 2
get_nrc_values, 3
get_percentage_values, 3
get_sent_values, 5
get_sentences, 4
get_sentiment, 4
get_stanford_sentiment, 6
get_text_as_string, 6
get_transformed_values, 7

rescale, 8