Package ‘DeducerText’

February 19, 2015

Type Package
Title Deducer GUI for Text Data
Version 0.1-2
Date 2014-06-12
Author Alex Rickett and Ian Fellows, with contributions from Neal Fultz
Maintainer Ian Fellows <ian@fellstat.com>
Depends R (>= 2.10.0), Deducer (>= 0.7-0), tm (>= 0.6), wordcloud (>= 2.1), RColorBrewer
Imports SnowballC
Suggests
SystemRequirements Java (>= 1.5), JRI
License LGPL-2
Description A GUI for text mining
LazyLoad yes
http://research.cens.ucla.edu/
NeedsCompilation no
Repository CRAN
Date/Publication 2014-06-13 01:10:32

R topics documented:

DeducerText-package ........................................ 2
make.color.scale ........................................... 2
term.freq ................................................... 3

Index 5
DeducerText-package  

Deducer GUI for Text Data

Description

A Deducer GUI for performing some of basic text mining operations provided by the 'tm' package.

Details

<table>
<thead>
<tr>
<th>Package</th>
<th>DeducerText</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>Version</td>
<td>0.1-0</td>
</tr>
<tr>
<td>Date</td>
<td>2011-01-17</td>
</tr>
<tr>
<td>License</td>
<td>What license is it under?</td>
</tr>
<tr>
<td>LazyLoad</td>
<td>yes</td>
</tr>
</tbody>
</table>

There is no reason to directly invoke any of the R functions provided by the package. This package is meant to be loaded from within Deducer/JGR. You may then perform various text-mining operations from the 'Text' menu in JGR.

Author(s)

Neal Fultz, Ian Fellows, Alex Rickett

Maintainer: Neal Fultz <nfultz@stat.ucla.edu>

make.color.scale  

Make Color Scale

Description

Construct 2 color gradient for the function 'wordcloud'.

Usage

make.color.scale(aColor, bColor, steps, gradientExp=.5)

Arguments

- **aColor**: The starting color of the gradient
- **bColor**: The ending color of the gradient
- **steps**: The number of elements in the output gradient vector
- **gradientExp**: A parameter for controlling the interpolation between the 2 colors. a value of 1 will yield straight linear interpolation between the colors. A value less than 0 will cause the gradient to rapidly transition and plateau into the the second color,
Value

A vector of colors (represented by character strings) giving a smooth transition between the 2 input colors.

Examples

# Make a cyan to red gradient with 10 steps.
make.color.scale(c(0,1,1), c(1,0,0), 10, 1)

Description

Get a table of term frequencies.

Usage

term.freq( d, 
  topN = 0, 
  percent = 0, 
  sorted = c("none", "alpha", "freq"), 
  decreasing = FALSE, 
  useDocFreq = FALSE, 
  minFreq = 1 )

Arguments

d The corpus from which term frequencies are calculated.
topN If specified, only the 'topN' most frequent terms are returned. If more terms are requested than available, all terms are returned. If both 'topN' and 'percent' are zero, then all terms are returned.
percent If specified, only the top 'percent' % most frequent terms are returned. If more terms are requested than available, all terms are returned. If both 'topN' and 'percent' are zero, then all terms are returned.
sorted A string specifying how to sort the terms. 'none' for no sorting, 'alpha' for alphanumeric sorting, and 'freq' for sorting by frequency.
decreasing If TRUE, terms are sorted in decreasing order, if FALSE, sorted ascending order.
useDocFreq If TRUE, the returned frequencies are for the total number of documents in which the term occurs. If false, they are the total number of occurrences.
minFreq Terms with *TOTAL* frequencies below this threshold will not be included in the output.

Value

A name vector of the term frequencies.
Examples

```r
if(require(tm)){
  data(crude)
  term.freq(d=crude, percent=0, topN=10, minFreq=0, useDocFreq=FALSE, sorted="alpha", decreasing=TRUE)
}
```
Index

*Topic package
  DeducerText-package, 2

DeducerText (DeducerText-package), 2
DeducerText-package, 2

make.color.scale, 2

term.freq, 3