Package ‘ngramrr’

August 29, 2016

Title  A Simple General Purpose N-Gram Tokenizer
Version  0.2.0
Date  2016-03-10
Author  Chung-hong Chan <chainsawtiney@gmail.com>
Maintainer  Chung-hong Chan <chainsawtiney@gmail.com>
Description  A simple n-gram (contiguous sequences of n items from a
given sequence of text) tokenizer to be used with the ’tm’ package with no
’rJava’/’RWeka’ dependency.
URL  https://github.com/chainsawriot/ngramrr
Depends  R (>= 3.0.0)
License  GPL-2
LazyData  true
Imports  tm, tau
Suggests  testthat, magrittr
RoxygenNote  5.0.1
NeedsCompilation  no
Repository  CRAN
Date/Publication  2016-03-10 23:44:11

R topics documented:

  dtmwrappers ................................................................. 2
  ngramrr ................................................................. 3

Index  4
dtmwrappers

Wrappers to DocumentTermMatrix and DocumentTermMatrix to use n-gram tokenization

Description

Wrappers to DocumentTermMatrix and DocumentTermMatrix to use n-gram tokenization provided by ngramrr.

Usage

dtm2(x, char = FALSE, ngmin = 1, ngmax = 2, rmEOL = TRUE, ...)
tdm2(x, char = FALSE, ngmin = 1, ngmax = 2, rmEOL = TRUE, ...)

Arguments

x character vector, Source or Corpus to be converted
char logical, using character n-gram. char = FALSE denotes word n-gram.
ngmin integer, minimum order of n-gram
ngmax integer, maximum order of n-gram
rmEOL logical, remove nggrams with EOL character
... Additional options for DocumentTermMatrix or DocumentTermMatrix

Value

DocumentTermMatrix or DocumentTermMatrix

See Also

ngramrr, DocumentTermMatrix, TermDocumentMatrix

Examples

nirvana <- c("hello hello hello how low", "hello hello how low", "hello hello hello how low", "hello hello how low", "with the lights out", "it's less dangerous", "here we are now", "entertain us", "i feel stupid", "and contagious", "here we are now", "entertain us", "a mulatto", "an albino", "a mosquito", "my libido", "yeah", "hey yay")
dtm2(nirvana, ngmax = 3, removePunctuation = TRUE)
**ngramrr**

**General purpose n-gram tokenizer**

**Description**

A non-Java based n-gram tokenizer to be used with the tm package. Support both character and word n-gram.

**Usage**

```r
ngramrr(x, char = FALSE, ngmin = 1, ngmax = 2, rmEOL = TRUE)
```

**Arguments**

- `x` input string.
- `char` logical, using character n-gram. char = FALSE denotes word n-gram.
- `ngmin` integer, minimum order of n-gram
- `ngmax` integer, maximum order of n-gram
- `rmEOL` logical, remove ngrams with EOL character

**Value**

vector of n-grams

**Examples**

```r
require(tm)

nirvana <- c("hello hello hello how low", "hello hello hello how low", "hello hello hello", "hello hello how low", "with the lights out", "it's less dangerous", "here we are now", "entertain us", "i feel stupid", "and contagious", "here we are now", "entertain us", "a mulatto", "an albino", "a mosquito", "my libido", "yeah", "hey yay")

ngramrr(nirvana[1], ngmax = 3)
ngramrr(nirvana[1], ngmax = 3, char = TRUE)
nirvanacor <- Corpus(VectorSource(nirvana))
TermDocumentMatrix(nirvanacor, control = list(tokenize = function(x) ngramrr(x, ngmax =3)))

# Character ngram

TermDocumentMatrix(nirvanacor, control = list(tokenize = function(x) ngramrr(x, char = TRUE, ngmax =3), wordLengths = c(1, Inf)))
```
Index

DocumentTermMatrix, 2
dtm2 (dtmwrappers), 2
dtmwrappers, 2

ngramrr, 3

tdm2 (dtmwrappers), 2
TermDocumentMatrix, 2