

# Package ‘ngramrr’

May 9, 2026

**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

## Contents

dtmwrappers . . . . .	2
ngramrr . . . . .	3
<b>Index</b>	<b>4</b>

---

dtmwrappers

*Wrappers to DocumentTermMatrix and DocumentTermMatrix to use n-gram tokenizaion*


---

## 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

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

## Value

`DocumentTermMatrix` or `DocumentTermMatrix`

## See Also

`ngramrr`, [DocumentTermMatrix](#), [TermDocumentMatrix](#)

## Examples

```
nirvana <- c("hello hello hello how low", "hello hello hello how low",
"hello hello hello how low", "hello hello hello",
"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**

```
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**

```
require(tm)

nirvana <- c("hello hello hello how low", "hello hello hello how low",
"hello hello hello how low", "hello hello hello",
"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](#)