Package 'acroname'

May 7, 2026

Type Package

Title Engine for Acronyms and Initialisms

Version 0.1.0

Maintainer VP Nagraj <nagraj@nagraj.net>

Description A tool for generating acronyms and initialisms from arbitrary text input.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports magrittr, R.utils, dplyr, purrr, readr, stringr, hunspell

Suggests testthat

NeedsCompilation no

Author VP Nagraj [aut, cre]

Repository CRAN

Date/Publication 2021-01-25 08:40:11 UTC

Contents

engines *acroname engines*

Description

The acroname engines include methods to generate acronyms and initialisms. acronym() searches for candidates by constructing words from characters provided. Each word constructed is compared to the terms in the dictionary specified, and once a match is found the acronym is returned. initialism() takes the first characters from each word in the string. Both functions can optionally return a tibble, ignore articles, and/or use a "bag of words" approach (for more see mince).

Usage

```
acronym(
  input,
  dictionary = NULL,
  acronym_length = 3,
  ignore_articles = TRUE,
  alnum_only = TRUE,
  timeout = 60,
  bow = FALSE,
  bow_prop = 0.5,
  to_tibble = FALSE
)

initialism(
  input,
  ignore_articles = TRUE,
  alnum_only = TRUE,
  bow = FALSE,
  bow_prop = 0.5,
  to_tibble = FALSE
)
```

Arguments

input	Character vector with text to use as the input for the candidate
dictionary	Character vector containing dictionary of terms from which acronym should be created; default is NULL and hunspell "en_us" dictionary will be used
acronym_length	Number of characters in acronym; default is 3
ignore_articles	Logical indicating whether or not articles should be ignored ; default is TRUE
alnum_only	Logical that specifes whether only alphanumeric should be used; default is TRUE
timeout	Maximum seconds to spend searching for an acronym; default is 60
bow	Logical for whether or not a "bag of words" approach should be used for "input" vector; default is FALSE

bow_prop	Given bow = TRUE this specifies the proportion of words to sample; ignored if bow = FALSE; default is 0.5
to_tibble	Logical as to whether or not the result should be a tibble; default is FALSE

Value

If to_tibble = FALSE (default), then a character vector containing the name capitalized followed by the original string with letters used in the name capitalized.

If to_tibble = TRUE, then a tibble with the following columns:

- **formatted**: The candidate name and string with letters used capitalized
- **prefix**: The candidate name
- **suffix**: Words used with letters in name capitalized
- **original**: The original string used to construct the name

---

find_articles *Helper to find articles*

Description

This function will check if an input word is an article in the English language ('a', 'an', 'the').

Usage

```
find_articles(word)
```

Arguments

word	Character vector of length 1 with word to check

Value

Logical vector of length one, TRUE if the word is an article and FALSE if not.

Examples

```
find_articles("the")
find_articles("then")
find_articles("whatever")
```

find_candidate *Find candidate*

Description

This is an unexported helper for acronym. The function is used wrapped in a `tryCatch()` that uses withTimeout to manage maximum wait time for the candidate acronym search.

Usage

```
find_candidate(collapsed, acronym_length, probs, dictionary, words_len)
```

Arguments

`collapsed`	The collapsed string of characters generated by mince
`acronym_length`	Number of characters in acronym; default is `3`
`probs`	Vector of probabilities for selecting each character while generating candidate
`dictionary`	Character vector containing dictionary of terms from which acronym should be created; default is `NULL` and `hunspell` "en_us" dictionary will be used
`words_len`	Vector of the length of each word in the input

Value

Named list with three elements:

- **formatted**: The candidate acronym and string with letters used capitalized
- **prefix**: The candidate acronym
- **suffix**: Words used with letters in acronym capitalized #'

first_char *Extract the first character from a string*

Description

This helper function will extract the first character from a string. The element may be a letter, number, or special character but will be coerced to a character vector in the output.

Usage

```
first_char(string)
```

Arguments

`string`	Character vector from which the first character will be extracted

Value

Character vector with the first character from each element in the vector passed to the input "string" argument. This value will be the same length as the original vector.

Examples

```
first_char("purple")
first_char(c("purple","rain"))
first_char(c("nothing","compares","2u"))
```

`mince` *Prepare input string*

Description

This helper is used by both acronym and initialism to extract elements needed from the input string.

If the function is used with `bow = TRUE` the input will be processed with a "bag of words" approach, by which words will be shuffled and sampled without replacement. In this case, the number of characters used will be determined by the proportion passed to "bow_prop".

Usage

```
mince(
  input,
  ignore_articles = TRUE,
  alnum_only = TRUE,
  bow = FALSE,
  bow_prop = 0.5
)
```

Arguments

`input`	Character vector with text to use as the input for the candidate
`ignore_articles`	Logical indicating whether or not articles should be ignored ; default is `TRUE`
`alnum_only`	Logical that specifes whether only alphanumeric should be used; default is `TRUE`
`bow`	Logical for whether or not a "bag of words" approach should be used for "input" vector; default is `FALSE`
`bow_prop`	Given `bow = TRUE` this specifies the proportion of words to sample; ignored if `bow = FALSE`; default is `0.5`

Value

Named list with the following elements:

- **words**: Vector with one element per word to be used in the acronym or initialism
- **collapsed**: Vector of length 1 containing all characters from words collapsed
- **words_len**: Vector containing length of each word
- **first_chars**: Vector containing first character from each word

Index