

# Package ‘rflashtext’

June 30, 2022

**Title** FlashText Algorithm for Finding and Replacing Words

**Version** 0.1.0

**Description** Implementation of the FlashText algorithm, by Singh (2017) <[arXiv:1711.00046](#)>. It can be used to find and replace words in a given text with only one pass over the document.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**URL** <https://github.com/AbrJA/rflashtext>

**BugReports** <https://github.com/AbrJA/rflashtext/issues>

**Imports** R6

**NeedsCompilation** no

**Author** Abraham Jaimes [aut, cre]

**Maintainer** Abraham Jaimes <[ajaimes@occ.com.mx](mailto:ajaimes@occ.com.mx)>

**Repository** CRAN

**Date/Publication** 2022-06-30 11:40:02 UTC

## R topics documented:

keyword_processor . . . . .	2
<b>Index</b>	<b>7</b>

---

keyword\_processor      *FlashText algorithm to find and replace words*

---

## Description

Based on the python library [flashtext](#). To see more details about the algorithm visit: [FlashText](#)

## Methods

### Public methods:

- `keyword_processor$new()`
- `keyword_processor$show_attrs()`
- `keyword_processor$add_keys_words()`
- `keyword_processor$contain_keys()`
- `keyword_processor$get_words()`
- `keyword_processor$find_keys()`
- `keyword_processor$replace_keys()`

### Method `new()`:

*Usage:*

```
keyword_processor$new(  
  ignore_case = TRUE,  
  word_chars = c(letters, LETTERS, 0:9, "_"),  
  dict = NULL  
)
```

*Arguments:*

`ignore_case` logical. If FALSE the search is case sensitive. Default TRUE.

`word_chars` character vector. Used to validate if a word continues. Default `c(letters, LETTERS, 0:9, "_")` equivalent to `[a-zA-Z0-9_]`.

`dict` list. Internally built character by character and needed for the search. Recommended to let the default value NULL.

*Returns:* invisible. Assign to a variable to inspect the output. Logical. TRUE if all went good.

*Examples:*

```
library(rflashtext)
```

```
processor <- keyword_processor$new(ignore_case = FALSE, word_chars = letters)  
processor
```

### Method `show_attrs()`:

*Usage:*

```
keyword_processor$show_attrs(attrs = "all")
```

*Arguments:*

`attrs` character vector. Options are subsets of `c("all", "id", "word_chars", "dict", "ignore_case", "dict_size")`. Default "all".

*Returns:* list with the values of the `attrs`. Useful to save `dict` and reuse it or to check the `dict_size`.

*Examples:*

```
library(rflashtext)
```

```
processor <- keyword_processor$new()
processor$add_keys_words(keys = c("NY", "LA"), words = c("New York", "Los Angeles"))
processor$show_attrs(attrs = "dict_size")
processor$show_attrs(attrs = "dict")
```

#### **Method** `add_keys_words()`:

*Usage:*

```
keyword_processor$add_keys_words(keys, words = NULL)
```

*Arguments:*

`keys` character vector. Strings to identify (find/replace) in the text.

`words` character vector. Strings to be returned (find) or replaced (replace) when found the respective keys. Should have the same length as `keys`. If not provided, `words = keys`.

*Returns:* invisible. Assign to a variable to inspect the output. Logical vector. FALSE if keys are duplicated, the respective words will be updated.

*Examples:*

```
library(rflashtext)
```

```
processor <- keyword_processor$new()
processor$add_keys_words(keys = c("NY", "LA"), words = c("New York", "Los Angeles"))
correct <- processor$add_keys_words(keys = c("NY", "CA"), words = c("New York City", "California"))
# To check if there are duplicate keys
correct
```

#### **Method** `contain_keys()`:

*Usage:*

```
keyword_processor$contain_keys(keys)
```

*Arguments:*

`keys` character vector. Strings to check if already are on the search dictionary.

*Returns:* logical vector. TRUE if the keys are on the search dictionary.

*Examples:*

```
library(rflashtext)
```

```
processor <- keyword_processor$new()
processor$add_keys_words(keys = c("NY", "LA"), words = c("New York", "Los Angeles"))
processor$contain_keys(keys = c("NY", "LA", "TX"))
```

#### **Method** `get_words()`:

*Usage:*

```
keyword_processor$get_words(keys)
```

*Arguments:*

keys character vector. Strings to get back the respective words.

*Returns:* character vector. Respective words. If keys not found returns NA\_character\_.

*Examples:*

```
library(rflashtext)
```

```
processor <- keyword_processor$new()
processor$add_keys_words(keys = c("NY", "LA"), words = c("New York", "Los Angeles"))
processor$get_words(keys = c("NY", "LA", "TX"))
```

**Method** find\_keys():*Usage:*

```
keyword_processor$find_keys(sentence, span_info = TRUE)
```

*Arguments:*

sentence character. Text to find the keys previously defined. Not vectorized.

span\_info logical. TRUE to retrieve the words and the position of the matches. FALSE to only retrieve the words. Default TRUE.

*Returns:* list with the words corresponding to keys found in the sentence. Hint: Use `do.call(rbind, ...)` to transform the list to a matrix.

*Examples:*

```
library(rflashtext)
```

```
processor <- keyword_processor$new()
processor$add_keys_words(keys = c("NY", "LA"), words = c("New York", "Los Angeles"))
words_found <- processor$find_keys(sentence = "I live in LA but I like NY")
do.call(rbind, words_found)
```

**Method** replace\_keys():*Usage:*

```
keyword_processor$replace_keys(sentence)
```

*Arguments:*

sentence character. Text to replace the keys found by the corresponding words. Not vectorized.

*Returns:* character. Text with the keys replaced by the respective words.

*Examples:*

```
library(rflashtext)
```

```
processor <- keyword_processor$new()
processor$add_keys_words(keys = c("NY", "LA"), words = c("New York", "Los Angeles"))
new_sentence <- processor$replace_keys(sentence = "I live in LA but I like NY")
new_sentence
```

**Examples**

```

library(rflashtext)

processor <- keyword_processor$new()
processor$add_keys_words(keys = c("NY", "LA"), words = c("New York", "Los Angeles"))

processor$contain_keys(keys = "NY")
processor$get_words(keys = "LA")

processor$find_keys(sentence = "I live in LA but I like NY")
processor$replace_keys(sentence = "I live in LA but I like NY")

## -----
## Method `keyword_processor$new`
## -----

library(rflashtext)

processor <- keyword_processor$new(ignore_case = FALSE, word_chars = letters)
processor

## -----
## Method `keyword_processor$show_attrs`
## -----

library(rflashtext)

processor <- keyword_processor$new()
processor$add_keys_words(keys = c("NY", "LA"), words = c("New York", "Los Angeles"))
processor$show_attrs(attrs = "dict_size")
processor$show_attrs(attrs = "dict")

## -----
## Method `keyword_processor$add_keys_words`
## -----

library(rflashtext)

processor <- keyword_processor$new()
processor$add_keys_words(keys = c("NY", "LA"), words = c("New York", "Los Angeles"))
correct <- processor$add_keys_words(keys = c("NY", "CA"), words = c("New York City", "California"))
# To check if there are duplicate keys
correct

## -----
## Method `keyword_processor$contain_keys`
## -----

library(rflashtext)

processor <- keyword_processor$new()
processor$add_keys_words(keys = c("NY", "LA"), words = c("New York", "Los Angeles"))

```

```
processor$contain_keys(keys = c("NY", "LA", "TX"))

## -----
## Method `keyword_processor$get_words`
## -----

library(rflashtext)

processor <- keyword_processor$new()
processor$add_keys_words(keys = c("NY", "LA"), words = c("New York", "Los Angeles"))
processor$get_words(keys = c("NY", "LA", "TX"))

## -----
## Method `keyword_processor$find_keys`
## -----

library(rflashtext)

processor <- keyword_processor$new()
processor$add_keys_words(keys = c("NY", "LA"), words = c("New York", "Los Angeles"))
words_found <- processor$find_keys(sentence = "I live in LA but I like NY")
do.call(rbind, words_found)

## -----
## Method `keyword_processor$replace_keys`
## -----

library(rflashtext)

processor <- keyword_processor$new()
processor$add_keys_words(keys = c("NY", "LA"), words = c("New York", "Los Angeles"))
new_sentence <- processor$replace_keys(sentence = "I live in LA but I like NY")
new_sentence
```

# Index

keyword\_processor, [2](#)