

Package: spelling (via r-universe)

July 31, 2026

Title Tools for Spell Checking in R

Version 2.3.2

Description Spell checking common document formats including latex, markdown, manual pages, and description files. Includes utilities to automate checking of documentation and vignettes as a unit test during 'R CMD check'. Both British and American English are supported out of the box and other languages can be added. In addition, packages may define a 'wordlist' to allow custom terminology without having to abuse punctuation.

License MIT + file LICENSE

Encoding UTF-8

URL <https://ropensci.r-universe.dev/spelling>
<https://docs.ropensci.org/spelling/>

BugReports <https://github.com/ropensci/spelling/issues>

Imports commonmark, xml2, hunspell (>= 3.0), knitr

Suggests pdftools

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Language en-US

Config/pak/sysreqs libxml2-dev

Repository <https://ropensci.r-universe.dev>

Date/Publication 2025-10-17 12:12:37 UTC

RemoteUrl <https://github.com/ropensci/spelling>

RemoteRef master

RemoteSha 4fdb6c350ae962c142d4e1d9b5261cb14c251b52

Contents

spell_check_files	2
spell_check_package	3
wordlist	4

spell_check_files	<i>Spell Check</i>
-------------------	--------------------

Description

Perform a spell check on document files or plain text.

Usage

```
spell_check_files(path, ignore = character(), lang = "en_US")
```

```
spell_check_text(text, ignore = character(), lang = "en_US")
```

Arguments

path	path to file or to spell check
ignore	character vector with words which will be added to the hunspell::dictionary
lang	set Language field in DESCRIPTION e.g. "en-US" or "en-GB". For supporting other languages, see the hunspell vignette .
text	character vector with plain text

Details

This function parses a file based on the file extension, and checks only text fields while ignoring code chunks and meta data. It works particularly well for markdown, but also latex, html, xml, pdf, and plain text are supported.

For more information about the underlying spelling engine, see the [hunspell package](#).

See Also

Other spelling: [spell_check_package\(\)](#), [wordlist](#)

Examples

```
# Example files
files <- list.files(system.file("examples", package = "knitr"),
  pattern = "\\.(Rnw|Rmd|html)$", full.names = TRUE)
spell_check_files(files)
```

spell_check_package *Package Spell Checking*

Description

Automatically spell-check package description, documentation, and vignettes.

Usage

```
spell_check_package(pkg = ".", vignettes = TRUE, use_wordlist = TRUE)
```

```
spell_check_setup(pkg = ".", vignettes = TRUE, lang = "en-US", error = FALSE)
```

Arguments

pkg	path to package root directory containing the DESCRIPTION file
vignettes	check all rmd and rnw files in the pkg root directory (e.g. readme.md) and package vignettes folder.
use_wordlist	ignore words in the package WORDLIST file
lang	set Language field in DESCRIPTION e.g. "en-US" or "en-GB". For supporting other languages, see the hunspell vignette .
error	should CMD check fail if spelling errors are found? Default only prints results.

Details

Parses and checks R manual pages, rmd/rnw vignettes, and text fields in the package DESCRIPTION file.

The preferred spelling language (typically en-GB or en-US) should be specified in the Language field from your package DESCRIPTION. To allow custom words, use the package [WORDLIST](#) file which will be added to the dictionary when spell checking. See [update_wordlist](#) to automatically populate and update this file.

The [spell_check_setup](#) function adds a unit test to your package which automatically runs a spell check on documentation and vignettes during R CMD check if the environment variable NOT_CRAN is set to TRUE. By default this unit test never fails; it merely prints potential spelling errors to the console. If not already done, the [spell_check_setup](#) function will add spelling as a Suggests dependency, and a Language field to DESCRIPTION.

Hunspell includes dictionaries for en_US and en_GB by default. Other languages require installation of a custom dictionary, see [hunspell](#) for details.

See Also

Other spelling: [spell_check_files\(\)](#), [wordlist](#)

wordlist*The WORDLIST file*

Description

The package wordlist file is used to allow custom words which will be added to the dictionary when spell checking. It is stored in `inst/WORDLIST` in the source package and must contain one word per line in UTF-8 encoded text.

Usage

```
update_wordlist(pkg = ".", vignettes = TRUE, confirm = TRUE)
```

```
get_wordlist(pkg = ".")
```

Arguments

<code>pkg</code>	path to package root directory containing the DESCRIPTION file
<code>vignettes</code>	check all rmd and rnw files in the pkg root directory (e.g. <code>readme.md</code>) and package vignettes folder.
<code>confirm</code>	show changes and ask confirmation before adding new words to the list

Details

The [update_wordlist](#) function runs a full spell check on a package, shows the results, and then prompts to add the found words to the package wordlist. Obviously you should check closely that these legitimate words and not actual spelling errors. It also removes words from the wordlist that no longer appear as spelling errors, either because they have been removed from the documentation or added to the lang dictionary.

See Also

Other spelling: [spell_check_files\(\)](#), [spell_check_package\(\)](#)

Index

* **spelling**

- spell_check_files, 2
- spell_check_package, 3
- wordlist, 4

get_wordlist (wordlist), 4

hunspell, 3

hunspell::dictionary, 2

spell_check_files, 2, 3, 4

spell_check_package, 2, 3, 4

spell_check_setup, 3

spell_check_setup

- (spell_check_package), 3

spell_check_test (spell_check_package),
3

spell_check_text (spell_check_files), 2

spelling (spell_check_package), 3

update_wordlist, 3, 4

update_wordlist (wordlist), 4

WORDLIST, 3

wordlist, 2, 3, 4