Package 'striprtf'

December 4, 2017

Type Package

Title Extract Text from RTF File

Version 0.5.1

Description Extracts plain text from RTF (Rich Text Format) file.

License MIT + file LICENSE

LazyData TRUE

Depends R (>= 3.0)

Imports magrittr, Rcpp, stringr, utils

Suggests testthat

RoxygenNote 6.0.1

LinkingTo Rcpp

URL https://github.com/kota7/striprtf

BugReports https://github.com/kota7/striprtf/issues

NeedsCompilation yes

Author Kota Mori [aut, cre]

Maintainer Kota Mori <kmori05@gmail.com>

Repository CRAN

Date/Publication 2017-12-04 15:55:49 UTC

R topics documented:

read_rtf *Extract Text from RTF (Rich Text Format) File*

Description

Parses an RTF file and extracts plain text as character vector.

Usage

```
read_rtf(file, verbose = FALSE, row_start = "*| ", row_end = "",
  cell_end = " | ", ignore_tables = FALSE, ...)

strip_rtf(text, verbose = FALSE, row_start = "*| ", row_end = "",
  cell_end = " | ", ignore_tables = FALSE)
```

Arguments

file	Path to an RTF file. Must be character of length 1.
verbose	Logical. If TRUE, progress report is printed on console. While it can be informative when parsing a large file, this option itself makes the process slow.
row_start, row_end	strings to be added at the beginning and end of table rows
cell_end	string to be put at the end of table cells
ignore_tables	if TRUE, no special treatment for tables
...	Addional arguments passed to readLines
text	Character of length 1. Expected to be contents of an RTF file.

Details

Rich text format (RTF) files are written as a text file consisting of ASCII characters. The specification has been developed by Microsoft. This function interprets the character strings and extracts plain texts of the file. Major part of the algorithm of this function comes from a stack overflow thread (http://stackoverflow.com/a/188877) and the implemented by Gilson Filho for python 3 (https://gist.github.com/gilsondev/7c1d2d753ddb522e7bc22511cfb08676). The function is a translation of the above codes to R language, associated with C++ codes for enhancement.

An advance from the preceding implementation is that the function accomodates with various ANSI code pages. For example, RTF files created by Japanese version of Microsoft Word marks \ansicpg932, which indicates the code page 932 is used for letter-code conversion. The function detects the code page indication and convert the characters to UTF-8 where possible. Conversion tables are retrieved from here: (http://www.unicode.org/Public/MAPPINGS/VENDORS/MICSFT/).

Value

Character vector of extracted text

References

- Python 3 implementation by Gilson Filho: https://gist.github.com/gilsondev/7c1d2d753ddb522e7bc22511cfb
- Original discussion thread: http://stackoverflow.com/a/188877
- Code page table: http://www.unicode.org/Public/MAPPINGS/VENDORS/MICSFT/

Examples

```
read_rtf(system.file("extdata/king.rtf", package = "striprtf"))
```

---

striprtf-deprecated *Renamed Functions*

Description

From ver 0.3.1, the functions are renamed as follows:

- striprtf –> read_rtf
- rtf2text –> strip_rtf

Usage

```
striprtf(file, verbose = FALSE, ...)

rtf2text(text, verbose = FALSE)
```

Arguments

file	Path to an RTF file. Must be character of length 1.
verbose	Logical. If TRUE, progress report is printed on console. While it can be informative when parsing a large file, this option itself makes the process slow.
...	Addional arguments passed to readLines
text	Character of length 1. Expected to be contents of an RTF file.

Value

Character vector of extracted text

`unused_letters` *Find letters not used in strings*

Description

Returns letters not used in strings

Usage

```
unused_letters(s, n = 1, avoid_strifrtf_internal = TRUE,
  as_number = FALSE, as_vector = FALSE)
```

Arguments

`s`	character vector
`n`	number of letters to return
`avoid_strifrtf_internal`	If `TRUE`, letters used in the package's internal process are also regarded as "used".
`as_number`	if `TRUE`, return unicode numbers instead of letters itself
`as_vector`	if `FALSE` (and `as_number` is `FALSE`), return a single concatenated character, otherwise returns a character vector

Details

This function can be useful when some special characters must be temporarily converted to another letter without being confused with the same letters used elsewhere.

Letters are first searched from `\u0001` upto `\uffff`. Do not specify too large `n`; An error is raised if a sufficient number of unsed letters are not found.

Value

unsed characters, format depends on `as_number` and `as_vector` arguments

Index