Package 'fuzzystring'

February 8, 2026

Title Fast Fuzzy String Joins for Data Frames

Version 0.0.1

Description Perform fuzzy joins on data frames using approximate string matching. Implements all standard join types (inner, left, right, full, semi, anti) with support for multiple string distance metrics from the 'stringdist' package including Levenshtein, Damerau-Levenshtein, Jaro-Winkler, and Soundex. Features a high-performance 'data.table' backend with 'C++' row binding for efficient processing of large datasets. Ideal for matching misspellings, inconsistent labels, messy user input, or reconciling datasets with slight variations in identifiers. Optionally returns distance metrics alongside matched records.

License MIT + file LICENSE

Depends R (>= 4.1)

Imports data.table, Rcpp, stringdist

LinkingTo Rcpp

Suggests dplyr, ggplot2, knitr, qdapDictionaries, readr, rmarkdown, rvest, stringr, testthat (>= 3.0.0), tidyr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

URL https://github.com/PaulESantos/fuzzystring, https://paulesantos.github.io/fuzzystring/

BugReports https://github.com/PaulESantos/fuzzystring/issues

VignetteBuilder knitr

NeedsCompilation yes

Author Paul E. Santos Andrade [aut, cre] (ORCID: <https://orcid.org/0000-0002-6635-0375>), David Robinson [ctb] (aut of fuzzyjoin)

Maintainer Paul E. Santos Andrade <paulefrens@gmail.com>

Repository CRAN

Date/Publication 2026-02-08 17:00:15 UTC

Contents

`fstring_anti_join` *Fuzzy anti join*

Description

Convenience wrapper for `fuzzystring_join_backend(mode = "anti")`.

Usage

```
fstring_anti_join(x, y, by = NULL, match_fun, ...)
```

Arguments

`x`	A `data.frame` or `data.table`.
`y`	A `data.frame` or `data.table`.
`by`	Columns by which to join the two tables. See `fuzzystring_join`.
`match_fun`	A function used to match values. It must return a logical vector (or a data.frame/data.table whose first column is logical) indicating which pairs match. For multi-column joins, you may pass a list of functions (one per column).
`...`	Additional arguments passed to the matching function(s).

Value

See `fuzzystring_join_backend`.

fstring_full_join *Fuzzy full join*

Description

Convenience wrapper for `fuzzystring_join_backend(mode = "full")`.

Usage

```
fstring_full_join(x, y, by = NULL, match_fun, ...)
```

Arguments

x	A `data.frame` or `data.table`.
y	A `data.frame` or `data.table`.
by	Columns by which to join the two tables. See `fuzzystring_join`.
match_fun	A function used to match values. It must return a logical vector (or a data.frame/data.table whose first column is logical) indicating which pairs match. For multi-column joins, you may pass a list of functions (one per column).
...	Additional arguments passed to the matching function(s).

Value

See `fuzzystring_join_backend`.

fstring_inner_join *Fuzzy inner join*

Description

Convenience wrapper for `fuzzystring_join_backend(mode = "inner")`.

Usage

```
fstring_inner_join(x, y, by = NULL, match_fun, ...)
```

Arguments

x	A `data.frame` or `data.table`.
y	A `data.frame` or `data.table`.
by	Columns by which to join the two tables. See `fuzzystring_join`.
match_fun	A function used to match values. It must return a logical vector (or a data.frame/data.table whose first column is logical) indicating which pairs match. For multi-column joins, you may pass a list of functions (one per column).
...	Additional arguments passed to the matching function(s).

Value

See fuzzystring_join_backend.

`fstring_left_join` *Fuzzy left join*

Description

Convenience wrapper for `fuzzystring_join_backend(mode = "left")`.

Usage

```
fstring_left_join(x, y, by = NULL, match_fun, ...)
```

Arguments

x	A data.frame or data.table.
y	A data.frame or data.table.
by	Columns by which to join the two tables. See fuzzystring_join.
match_fun	A function used to match values. It must return a logical vector (or a data.frame/data.table whose first column is logical) indicating which pairs match. For multi-column joins, you may pass a list of functions (one per column).
...	Additional arguments passed to the matching function(s).

Value

See fuzzystring_join_backend.

`fstring_right_join` *Fuzzy right join*

Description

Convenience wrapper for `fuzzystring_join_backend(mode = "right")`.

Usage

```
fstring_right_join(x, y, by = NULL, match_fun, ...)
```

Arguments

x	A data.frame or data.table.
y	A data.frame or data.table.
by	Columns by which to join the two tables. See fuzzystring_join.
match_fun	A function used to match values. It must return a logical vector (or a data.frame/data.table whose first column is logical) indicating which pairs match. For multi-column joins, you may pass a list of functions (one per column).
...	Additional arguments passed to the matching function(s).

Value

See fuzzystring_join_backend.

---

fstring_semi_join *Fuzzy semi join*

Description

Convenience wrapper for fuzzystring_join_backend(mode = "semi").

Usage

```
fstring_semi_join(x, y, by = NULL, match_fun, ...)
```

Arguments

x	A data.frame or data.table.
y	A data.frame or data.table.
by	Columns by which to join the two tables. See fuzzystring_join.
match_fun	A function used to match values. It must return a logical vector (or a data.frame/data.table whose first column is logical) indicating which pairs match. For multi-column joins, you may pass a list of functions (one per column).
...	Additional arguments passed to the matching function(s).

Value

See fuzzystring_join_backend.

`fuzzystring_join` *Join two tables based on fuzzy string matching*

Description

Uses `stringdist::stringdist()` to compute distances and a data.table-based backend to assemble the final result. This is the main user-facing entry point for fuzzy joins on strings.

Usage

```
fuzzystring_join(
  x,
  y,
  by = NULL,
  max_dist = 2,
 method = c("osa", "lv", "dl", "hamming", "lcs", "qgram", "cosine", "jaccard", "jw",
    "soundex"),
  mode = "inner",
  ignore_case = FALSE,
  distance_col = NULL,
  ...
)

fuzzystring_inner_join(x, y, by = NULL, distance_col = NULL, ...)

fuzzystring_left_join(x, y, by = NULL, distance_col = NULL, ...)

fuzzystring_right_join(x, y, by = NULL, distance_col = NULL, ...)

fuzzystring_full_join(x, y, by = NULL, distance_col = NULL, ...)

fuzzystring_semi_join(x, y, by = NULL, distance_col = NULL, ...)

fuzzystring_anti_join(x, y, by = NULL, distance_col = NULL, ...)
```

Arguments

`x`	A `data.frame` or `data.table`.
`y`	A `data.frame` or `data.table`.
`by`	Columns by which to join the two tables. You can supply a character vector of common names (e.g. `c("name")`), or a named vector mapping `x` to `y` (e.g. `c(name = "approx_name")`).
`max_dist`	Maximum distance to use for joining. Smaller values are stricter.
`method`	Method for computing string distance, see `?stringdist::stringdist` and the `stringdist` package vignettes.

mode	One of "inner", "left", "right", "full", "semi", or "anti".
ignore_case	Logical; if TRUE, comparisons are case-insensitive.
distance_col	If not NULL, adds a column with this name containing the computed distance for each matched pair (or NA for unmatched rows in outer joins).
...	Additional arguments passed to stringdist.

Details

If method = "soundex", max_dist is automatically set to 0.5, since Soundex distance is 0 (match) or 1 (no match).

For Levenshtein-like methods ("osa", "lv", "dl"), a fast prefilter is applied: if abs(nchar(v1) - nchar(v2)) > max_dist, the pair cannot match, so distance is not computed for that pair.

Value

A joined table (same container type as x). See fuzzystring_join_backend for details on output structure.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  d <- data.table::data.table(approximate_name = c("Idea", "Premiom"))
  # Match diamonds$cut to d$approximate_name
  res <- fuzzystring_inner_join(ggplot2::diamonds, d,
    by = c(cut = "approximate_name"),
    max_dist = 1
  )
  head(res)
}
```

---

misspellings *A corpus of common misspellings, for examples and practice*

Description

This is a tbl_df mapping misspellings of their words, compiled by Wikipedia, where it is licensed under the CC-BY SA license. (Three words with non-ASCII characters were filtered out). If you'd like to reproduce this dataset from Wikipedia, see the example code below.

Usage

```
misspellings
```

Format

An object of class tbl_df (inherits from tbl, data.frame) with 4505 rows and 2 columns.

Source

https://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings/For_machines

Examples

```
library(rvest)
library(readr)
library(dplyr)
library(stringr)
library(tidyr)

u <- "https://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings/For_machines"
h <- read_html(u)

misspellings <- h %>%
  html_nodes("pre") %>%
  html_text() %>%
  read_delim(col_names = c("misspelling", "correct"),
                   delim = ">",
                   skip = 1) %>%
  mutate(misspelling = str_sub(misspelling,
                                              1, -2)) |>
  separate_rows(correct, sep = ", ") |>
  filter(Encoding(correct) != "UTF-8")
```

Index