

Package ‘SemNetCleaner’

May 7, 2026

Title An Automated Cleaning Tool for Semantic and Linguistic Data

Version 1.3.7

Date 2025-05-08

Maintainer Alexander P. Christensen <alexpaulchristensen@gmail.com>

Description Implements several functions that automates the cleaning and spell-checking of text data. Also converges, finalizes, removes plurals and continuous strings, and puts text data in binary format for semantic network analysis. Uses the 'SemNet-Dictionaries' package to make the cleaning process more accurate, efficient, and reproducible.

License GPL (>= 3.0)

URL <https://github.com/AlexChristensen/SemNetCleaner>

BugReports <https://github.com/AlexChristensen/SemNetCleaner/issues>

NeedsCompilation no

Encoding UTF-8

LazyData true

Depends R (>= 3.6.0), SemNetDictionaries (>= 0.1.8)

Imports foreign, parallel, pbapply, R.matlab, readxl, rstudioapi, searcher, shiny, stringi, stringdist, tcltk

Suggests DT, hunspell, easycsv, htmlTable, knitr, markdown, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.3.2

Author Alexander P. Christensen [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-9798-7037>>)

Repository CRAN

Date/Publication 2025-05-08 15:10:02 UTC

Contents

SemNetCleaner-package	2
bad.response	3

best.guess	4
bin2resp	5
convert2snafu	6
letter.freq	7
open.animals	7
open.clean	8
open.preprocess	9
pluralize	9
qwerty.dist	10
read.data	11
resp2bin	13
singularize	14
textcleaner	15

Index	19
--------------	-----------

SemNetCleaner-package *SemNetCleaner-package*

Description

Implements several functions that automates the cleaning and spell-checking of text data. Also converges, finalizes, removes plurals and continuous strings, and puts text data in binary format for semantic network analysis. Uses the SemNetDictionaries package to make the cleaning process more accurate, efficient, and reproducible.

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

See Also

Useful links:

- <https://github.com/AlexChristensen/SemNetCleaner>
- Report bugs at <https://github.com/AlexChristensen/SemNetCleaner/issues>

bad.response	<i>Bad Responses to NA</i>
--------------	----------------------------

Description

A wrapper function to determine whether responses are good or bad. Bad responses are replaced with missing (NA). Good responses are returned.

Usage

```
bad.response(word, ...)
```

Arguments

word	Character. A word to be tested for whether it is bad
...	Vector. Additional responses to be considered bad

Value

If response is bad, then returns NA. If response is valid, then returns the response

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

Examples

```
# Bad response
bad.response(word = " ")

# Good response
bad.response(word = "hello")

# Make a good response bad
bad.response(word = "hello", "hello")

# Add additional bad responses
bad.response(word = "hello", c("hello", "world"))
```

 best.guess

Makes Best Guess for Spelling Correction

Description

A wrapper function for the best guess of a spelling mistake based on the letters, the ordering of those letters, and the potential for letters to be interchanged. The **Damerau-Levenshtein distance** is used to guide inferences into what word the participant was trying to spell from a dictionary (see SemNetDictionaries)

Usage

```
best.guess(word, full.dictionary, dictionary = NULL, tolerance = 1)
```

Arguments

word	Character. A word to get best guess spelling options from dictionary
full.dictionary	Character vector. The dictionary to search for best guesses in. See SemNetDictionaries
dictionary	Character. A dictionary from SemNetDictionaries for monikers (enhances guessing)
tolerance	Numeric. The distance tolerance set for automatic spell-correction purposes. This function uses the function <code>stringdist</code> to compute the Damerau-Levenshtein distance, which is used to determine potential best guesses Unique words (i.e., $n = 1$) that are within the (distance) tolerance are automatically output as best guess responses. This default is based on Damerau's (1964) proclamation that more than 80% of all human misspellings can be expressed by a single error (e.g., insertion, deletion, substitution, and transposition). If there is more than one word that is within or below the distance tolerance, then these will be provided as potential options. The recommended and default distance tolerance is <code>tolerance = 1</code> , which only spell corrects a word if there is only one word with a DL distance of 1.

Value

The best guess(es) of the word

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

References

Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7, 171-176.

Examples

```
# Misspelled "bombay"
best.guess("bomba", full.dictionary = SemNetDictionaries::animals.dictionary)
```

bin2resp

Binary Responses to Character Responses

Description

Converts the binary response matrix into characters for each participant

Usage

```
bin2resp(rmat, to.data.frame = FALSE)
```

Arguments

rmat	Binary matrix. A binarized response matrix of verbal fluency or linguistic data
to.data.frame	Boolean. Should output be a data frame where participants are columns? Defaults to FALSE. Set to TRUE to convert output to data frame

Value

A list containing objects for each participant and their responses

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

Examples

```
# Toy example
raw <- open.animals[c(1:10),-c(1:3)]

if(interactive())
{
  # Clean and preprocess data
  clean <- textcleaner(open.animals[, -c(1:2)], partBY = "row", dictionary = "animals")

  # Change binary response matrix to word response matrix
  charmat <- bin2resp(clean$responses$binary)
}
```

convert2snafu	<i>Converts <code>textcleaner</code> object to a SNAFU GUI format</i>
---------------	---

Description

Converts `textcleaner` object to a SNAFU GUI format (only works for fluency data)

Usage

```
convert2snafu(..., category)
```

Arguments

<code>...</code>	Matrix or data frame. A clean response matrices
<code>category</code>	Character. Category of verbal fluency data

Details

The format of the file has 7 columns:

- `id` — Defaults to the row names of the inputted data
- `listnum` — The list number for the fluency category. Defaults to 0. Future implementations will allow more lists
- `category` — The verbal fluency category that is input into the `category` argument
- `item` — The verbal fluency responses for every participant
- `RT` — Response time. Currently not implemented. Defaults to 0
- `RTstart` — Start of response time. Currently not implemented. Defaults to 0
- `group` — Names of groups. Defaults to the names of the objects input into the function (`...`)

Value

A .csv file formatted for SNAFU

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

References

For SNAFU, see: Zemla, J. C., Cao, K., Mueller, K. D., & Austerweil, J. L. (2020). SNAFU: The Semantic Network and Fluency Utility. *Behavior Research Methods*, 1-19. <https://doi.org/10.3758/s13428-019-01343-w>

Examples

```
# Convert data to SNAFU
if(interactive())
{convert2snafu(open.clean, category = "animals")}
```

`letter.freq`*Letter Frequencies Based on 40,000 Words*

Description

A vector corresponding the frequency of letters across 40,000 words. Retrieved from: <http://pi.math.cornell.edu/~mec/2003-2004/cryptography/subs/frequencies.html>

Usage

```
data(letter.freq)
```

Format

letter.freq (26-element numeric vector)

Examples

```
data("letter.freq")
```

`open.animals`*Openness and Verbal Fluency*

Description

Raw Animals verbal fluency data ($n = 516$) from Christensen et al. (2018).

Usage

```
data(open.animals)
```

Format

open.animals (matrix 516 x 38)

Details

First column is a grouping variable ("Group") with 1 corresponding to low openness to experience and 2 to high openness to experience

Second column is the latent variable of openness to experience with Intellect items removed (see Christensen et al., 2018 for more details).

Third column is the ID variable for each participant.

Columns 4-38 are raw fluency data.

References

Christensen, A. P., Kenett, Y. N., Cotter, K. N., Beaty, R. E., & Silvia, P. J. (2018). Remotely close associations: Openness to experience and semantic memory structure. *European Journal of Personality*, 32, 480-492.

Examples

```
data("open.animals")
```

open.clean

Cleaned Response Matrices (Openness and Verbal Fluency)

Description

Cleaned response matrices for the Animals verbal fluency data ($n = 516$) from Christensen et al. (2018).

Usage

```
data(open.clean)
```

Format

```
open.clean (matrix, 516 x 35)
```

References

Christensen, A. P., Kenett, Y. N., Cotter, K. N., Beaty, R. E., & Silvia, P. J. (2018). Remotely close associations: Openness to experience and semantic memory structure. *European Journal of Personality*, 32, 480-492.

Examples

```
data("open.clean")
```

open.preprocess	<i>Preprocessed <code>textcleaner</code> Object (Openness and Verbal Fluency)</i>
-----------------	---

Description

Preprocessed `textcleaner` object for the Animals verbal fluency data ($n = 516$) from Christensen and Kenett (2020).

Usage

```
data(open.preprocess)
```

Format

```
open.preprocess (list, length = 4)
```

References

Christensen, A. P., & Kenett, Y. N. (2020). Semantic network analysis (SemNA): A tutorial on preprocessing, estimating, and analyzing semantic networks. *PsyArxiv*.

Examples

```
data("open.preprocess")
```

pluralize	<i>Converts Words to their Plural Form</i>
-----------	--

Description

A function to change words to their plural form. The rules for converting words to their plural forms are based on the grammar rules.

This function handles most special cases and some irregular cases (see examples) but caution is necessary. If no plural form is identified, then the original word is returned.

Usage

```
pluralize(word)
```

Arguments

word	A word
------	--------

Value

Returns the word in singular form, unless a plural form could not be found (then the original word is returned)

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

Examples

```
# Handles any prototypical cases
"dogs"
pluralize("dog")

"foxes"
pluralize("fox")

"wolves"
pluralize("wolf")

"octopi"
pluralize("octopus")

"taxa"
pluralize("taxon")

# And most special cases:
"wives"
pluralize("wife")

"roofs"
pluralize("roof")

"photos"
pluralize("photo")

# And some irregular cases:
"children"
pluralize("child")

"teeth"
pluralize("tooth")

"mice"
pluralize("mouse")
```

qwerty.dist

QWERTY Distance for Same Length Words

Description

Computes QWERTY Distance for words that have the same number of characters. Distance is computed based on the number of keys a character is away from another character on a QWERTY keyboard

Usage

```
qwerty.dist(wordA, wordB)
```

Arguments

wordA	Character vector. Word to be compared
wordB	Character vector. Word to be compared

Value

Numeric value for distance between wordA and wordB

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

Examples

```
#Identical values for Damerau-Levenshtein
stringdist::stringdist("big", "pig", method="dl")

stringdist::stringdist("big", "bug", method="dl")

#Different distances for QWERTY
qwerty.dist("big", "pig")

qwerty.dist("big", "bug") # Probably meant to type "bug"
```

read.data

Read in Common Data File Extensions

Description

A single function to read in common data file extensions. Note that this function is specialized for reading in text data in the format necessary for functions in SemNetCleaner

File extensions supported:

- .Rdata
- .rds
- .csv
- .xlsx
- .xls
- .sav
- .txt
- .mat
- .dat

Usage

```
read.data(file = file.choose(), header = TRUE, sep = ",", ...)
```

Arguments

file	Character. A path to the file to load. Defaults to interactive file selection using file.choose
header	Boolean. A logical value indicating whether the file contains the names of the variables as its first line. If missing, the value is determined from the file format: header is set to TRUE if and only if the first row contains one fewer field than the number of columns
sep	Character. The field separator character. Values on each line of the file are separated by this character. If sep = "" (the default for read.table) the separator is a 'white space', that is one or more spaces, tabs, newlines or carriage returns
...	Additional arguments. Allows for additional arguments to be passed onto the respective read functions. See documentation in the list below: <ul style="list-style-type: none">• .Rdata load• .rds readRDS• .csv read.table• .xlsx read_excel• .xls read_excel• .sav read.spss• .txt read.table• .mat readMat• .dat read.table

Value

A data frame containing a representation of the data in the file. If file extension is ".Rdata", then data will be read to the global environment

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

References

R Core Team

R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

readxl

Hadley Wickham and Jennifer Bryan (2019). readxl: Read Excel Files. R package version 1.3.1. <https://CRAN.R-project.org/package=readxl>

R.matlab

Henrik Bengtsson (2018). R.matlab: Read and Write MAT Files and Call MATLAB from Within R. R package version 3.6.2. <https://CRAN.R-project.org/package=R.matlab>

Examples

```
# Use this example for your data
if(interactive())
{read.data()}

# Example for CRAN tests
## Create test data
test1 <- c(1:5, "6,7", "8,9,10")

## Path to temporary file
tf <- tempfile()

## Create test file
writeLines(test1, tf)

## Read in data
read.data(tf)

# See documentation of respective R functions for specific examples
```

resp2bin

Responses to binary matrix

Description

Converts the response matrix to binary response matrix

Usage

```
resp2bin(resp)
```

Arguments

resp Response matrix. A response matrix of verbal fluency or linguistic data

Value

A list containing objects for each participant and their responses

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

Examples

```
# Toy example
raw <- open.animals[c(1:10),-c(1:3)]

if(interactive())
{
  # Clean and preprocess data
  clean <- textcleaner(open.animals[, -c(1:2)], partBY = "row", dictionary = "animals")

  # Change response matrix to binary response matrix
  binmat <- resp2bin(clean$responses$corrected)
}
```

singularize

Converts Words to their Singular Form

Description

A function to change words to their singular form. The rules for converting words to their singular forms are based on the *inverse* of the grammar rules. This function handles most special cases and some irregular cases (see examples) but caution is necessary. If no singular form is identified, then the original word is returned.

Usage

```
singularize(word, dictionary = TRUE)
```

Arguments

word	Character. A word
dictionary	Boolean. Should dictionary be used to verify word exists? Default to TRUE

Value

Returns the word in singular form, unless a singular form could not be found (then the original word is returned)

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

Examples

```
# Handles any prototypical cases
# "dog"
singularize("dogs")

# "fox"
singularize("foxes")

# "wolf"
singularize("wolves")

# "octopus"
singularize("octopi")

# "taxon"
singularize("taxa")

# And most special cases:
# "wife"
singularize("wives")

# "fez"
singularize("fezzes")

# "roof"
singularize("roofs")

# "photo"
singularize("photos")

# And some irregular cases:
# "child"
singularize("children")

# "tooth"
singularize("teeth")

# "mouse"
singularize("mice")
```

textcleaner

Text Cleaner

Description

An automated cleaning function for spell-checking, de-pluralizing, removing duplicates, and binarizing text data

Usage

```
textcleaner(
  data = NULL,
  type = c("fluency", "free"),
  miss = 99,
  partBY = c("row", "col"),
  dictionary = NULL,
  spelling = c("UK", "US"),
  add.path = NULL,
  keepStrings = FALSE,
  allowPunctuations,
  allowNumbers = FALSE,
  lowercase = TRUE,
  keepLength = NULL,
  keepCue = FALSE,
  continue = NULL
)
```

Arguments

data Matrix or data frame.
 For task = "fluency", data are expected to follow wide formatting (IDs are the row names and are **not** a column in the matrix or data frame):

row.names	Response 1	Response 2	Response n
ID_1	1	2	n
ID_2	1	2	n
ID_n	1	2	n

For task = "free", data are expected to follow long formatting:

ID	Cue	Response
1	1	1
1	1	2
1	1	n
1	2	1
1	2	2
1	2	n
1	n	1
1	n	2
1	n	n
2	1	1
2	1	2
2	1	n
2	2	1
2	2	2
2	2	n
2	n	1

2	n	2
2	n	n
n	1	1
n	1	2
n	1	n
n	2	1
n	2	2
n	2	n
n	n	1
n	n	2
n	n	n

type	Character vector. Type of task to be preprocessed. <ul style="list-style-type: none"> • "fluency" — Verbal fluency data (e.g., categories, phonological, synonyms) • "free" — Free association data (e.g., cue terms or words)
miss	Numeric or character. Value for missing data. Defaults to 99
partBY	Character. Are participants by row or column? Set to "row" for by row. Set to "col" for by column
dictionary	Character vector. Can be a vector of a corpus or any text for comparison. Dictionary to be used for more efficient text cleaning. Defaults to NULL, which will use general.dictionary Use <code>dictionaries()</code> or <code>find.dictionaries()</code> for more options (See <code>SemNetDictionaries</code> for more details)
spelling	Character vector. English spelling to be used. <ul style="list-style-type: none"> • "UK" — For British spelling (e.g., colour, grey, programme, theatre) • "US" — For American spelling (e.g., color, gray, program, theater)
add.path	Character. Path to additional dictionaries to be found. DOES NOT search recursively (through all folders in path) to avoid time intensive search. Set to "choose" to open an interactive directory explorer
keepStrings	Boolean. Should strings be retained or separated? Defaults to FALSE. Set to TRUE to retain strings as strings
allowPunctuations	Character vector. Allows punctuation characters to be included in responses. Defaults to "-". Set to "all" to keep all punctuation characters
allowNumbers	Boolean. Defaults to FALSE. Set to TRUE to keep numbers in text
lowercase	Boolean. Should words be converted to lowercase? Defaults to TRUE. Set to FALSE to keep words as they are
keepLength	Numeric. Maximum number of words allowed in a response. Defaults to NULL. Set a number to keep responses with words less than or equal to the number (e.g., 3 will keep responses with three or less words)
keepCue	Boolean. Should cue words be retained in the responses? Defaults to FALSE. Set to TRUE to allow cue words to be retained
continue	List. A result previously unfinished that still needs to be completed. Allows you to continue to manually spell-check their data after you've closed or errored out. Defaults to NULL

Value

This function returns a list containing the following objects:

binary	A matrix of responses where each row represents a participant and each column represents a unique response. A response that a participant has provided is a '1' and a response that a participant has not provided is a '0'
responses	A list containing two objects: <ul style="list-style-type: none"> • <code>clean</code> — A response matrix that has been spell-checked and de-pluralized with duplicates removed. This can be used as a final dataset for analyses (e.g., fluency of responses) • <code>original</code> — The original response matrix that has had white spaces before and after words response. Also converts all upper-case letters to lower case
spellcheck	A list containing three objects: <ul style="list-style-type: none"> • <code>full</code> — All responses regardless of spell-checking changes • <code>auto</code> — Only the incorrect responses that were changed during spell-check
removed	A list containing two objects: <ul style="list-style-type: none"> • <code>rows</code> — Identifies removed participants by their row (or column) location in the original data file • <code>ids</code> — Identifies removed participants by their ID (see argument <code>data</code>)
partChanges	A list where each participant is a list index with each response that was been changed. Participants are identified by their ID (see argument <code>data</code>). This can be used to replicate the cleaning process and to keep track of changes more generally. Participants with NA did not have any changes from their original data and participants with missing data are removed (see <code>removed\$ids</code>)

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

References

- Christensen, A. P., & Kenett, Y. N. (in press). Semantic network analysis (SemNA): A tutorial on preprocessing, estimating, and analyzing semantic networks. *Psychological Methods*.
- Hornik, K., & Murdoch, D. (2010). Watch Your Spelling!. *The R Journal*, 3, 22-28.

Examples

```
# Toy example
raw <- open.animals[c(1:10),-c(1:3)]

if(interactive())
{
  #Full test
  clean <- textcleaner(open.animals[,-c(1,2)], partBY = "row", dictionary = "animals")
}
```

Index

* datasets

- letter.freq, [7](#)
- open.animals, [7](#)
- open.clean, [8](#)
- open.preprocess, [9](#)

bad.response, [3](#)

best.guess, [4](#)

bin2resp, [5](#)

convert2snafu, [6](#)

file.choose, [12](#)

general.dictionary, [17](#)

letter.freq, [7](#)

load, [12](#)

open.animals, [7](#)

open.clean, [8](#)

open.preprocess, [9](#)

pluralize, [9](#)

qwerty.dist, [10](#)

read.data, [11](#)

read.spss, [12](#)

read.table, [12](#)

read_excel, [12](#)

readMat, [12](#)

readRDS, [12](#)

resp2bin, [13](#)

SemNetCleaner (SemNetCleaner-package), [2](#)

SemNetCleaner-package, [2](#)

singularize, [14](#)

stringdist, [4](#)

textcleaner, [6](#), [9](#), [15](#)