

# Package ‘SemNetCleaner’

September 16, 2021

**Title** An Automated Cleaning Tool for Semantic and Linguistic Data

**Version** 1.3.4

**Date** 2021-09-14

**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** stringdist, searcher, tcltk, foreign, readxl, R.matlab, stringi, rstudioapi, easycsv, shiny, editData, miniUI

**Suggests** knitr, htmlTable, markdown, rmarkdown

**VignetteBuilder** knitr

**RoxygenNote** 7.1.2

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

**Repository** CRAN

**Date/Publication** 2021-09-16 14:00:02 UTC

## R topics documented:

SemNetCleaner-package . . . . .	2
bad.response . . . . .	3

best.guess . . . . .	4
bin2resp . . . . .	5
convert2snafu . . . . .	6
correct.changes . . . . .	7
letter.freq . . . . .	8
open.animals . . . . .	8
open.clean . . . . .	9
open.preprocess . . . . .	10
pluralize . . . . .	10
qwerty.dist . . . . .	11
read.data . . . . .	12
resp2bin . . . . .	14
singularize . . . . .	15
textcleaner . . . . .	16

<b>Index</b>	<b>19</b>
--------------	-----------

---

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](mailto: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

<code>word</code>	Character. A word to get best guess spelling options from dictionary
<code>full.dictionary</code>	Character vector. The dictionary to search for best guesses in. See <a href="#">SemNetDictionaries</a>
<code>dictionary</code>	Character. A dictionary from <a href="#">SemNetDictionaries</a> for monikers (enhances guessing)
<code>tolerance</code>	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](mailto: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>Pathfinder Network</i>
---------------	---------------------------

---

### Description

Estimates a pathfinder network using the MST-Pathfinder Network method from Quirin et al. (2008; see also Schvaneveldt, 1990)

### Usage

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

### Arguments

...	Matrix or data frame. A clean response matrices
category	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")}
```

---

correct.changes      *Correct Changes from [textcleaner](#)*

---

**Description**

A function that corrects changes that were made automatically by [textcleaner](#)

**Usage**

```
correct.changes(textcleaner.obj)
```

**Arguments**

textcleaner.obj  
Object from [textcleaner](#)

**Value**

This function returns the corrected lists from [textcleaners](#):

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"> <li>• clean 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)</li> <li>• original The original response matrix that has had white spaces before and after words response. Also converts all upper-case letters to lower case</li> </ul>
spellcheck	A list containing three objects: <ul style="list-style-type: none"> <li>• full All responses regardless of spell-checking changes</li> <li>• auto Only the incorrect responses that were changed during spell-check</li> <li>• changes Only the changes made within the function <a href="#">correct.changes</a></li> </ul>
removed	A list containing two objects: <ul style="list-style-type: none"> <li>• rows Identifies removed participants by their row (or column) location in the original data file</li> <li>• ids Identifies removed participants by their ID (see argument data)</li> </ul>
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 data). 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 removed\$ids)

**Author(s)**

Alexander Christensen <alexpaulchristensen@gmail.com>

**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")
}
```

---

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 found here: <https://www.grammarly.com/blog/plural-nouns/>. 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 <a href="#">file.choose</a>
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 <a href="#">read.table</a> ) 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"><li>• .Rdata <a href="#">load</a></li><li>• .rds <a href="#">readRDS</a></li><li>• .csv <a href="#">read.table</a></li><li>• .xlsx <a href="#">read_excel</a></li><li>• .xls <a href="#">read_excel</a></li><li>• .sav <a href="#">read.spss</a></li><li>• .txt <a href="#">read.table</a></li><li>• .mat <a href="#">readMat</a></li><li>• .dat <a href="#">read.table</a></li></ul>

**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](mailto: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 found here: <https://www.grammarly.com/blog/plural-nouns/>. 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,
  miss = 99,
  partBY = c("row", "col"),
  dictionary = NULL,
  spelling = c("UK", "US"),
  add.path = NULL,
  keepStrings = FALSE,
  allowPunctuations = c("-", "all"),
  allowNumbers = FALSE,
  lowercase = TRUE,
  continue = NULL
)
```

**Arguments**

data	Matrix or data frame. A dataset of text data. Participant IDs will be automatically identified if they are included. If no IDs are provided, then their order in the corresponding row (or column is used). A message will notify the user how IDs were assigned
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 <a href="#">general.dictionary</a> Use <code>dictionaries()</code> or <code>find.dictionaries()</code> for more options (See <a href="#">SemNetDictionaries</a> for more details)
spelling	Character vector. English spelling to be used. <ul style="list-style-type: none"> <li>• "UK" For British spelling (e.g., colour, grey, programme, theatre)</li> <li>• "US" For American spelling (e.g., color, gray, program, theater)</li> </ul>
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
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"> <li>• <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)</li> <li>• <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</li> </ul>
spellcheck	A list containing three objects: <ul style="list-style-type: none"> <li>• <code>full</code> All responses regardless of spell-checking changes</li> <li>• <code>auto</code> Only the incorrect responses that were changed during spell-check</li> </ul>
removed	A list containing two objects: <ul style="list-style-type: none"> <li>• <code>rows</code> Identifies removed participants by their row (or column) location in the original data file</li> <li>• <code>ids</code> Identifies removed participants by their ID (see argument <code>data</code>)</li> </ul>
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**

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, 8
- open.animals, 8
- open.clean, 9
- open.preprocess, 10

bad.response, 3

best.guess, 4

bin2resp, 5

convert2snafu, 6

correct.changes, 7, 7

file.choose, 13

general.dictionary, 17

letter.freq, 8

load, 13

open.animals, 8

open.clean, 9

open.preprocess, 10

pluralize, 10

qwerty.dist, 11

read.data, 12

read.spss, 13

read.table, 13

read\_excel, 13

readMat, 13

readRDS, 13

resp2bin, 14

SemNetCleaner (SemNetCleaner-package), 2

SemNetCleaner-package, 2

SemNetDictionaries, 2, 4, 17

singularize, 15

stringdist, 4

textcleaner, 7, 10, 16