Package 'lingglosses'

December 19, 2022

Type Package

Title Interlinear Glossed Linguistic Examples and Abbreviation Lists Generation

Version 0.0.6

Depends R (>= 3.5.0)

Maintainer George Moroz <agricolamz@gmail.com>

Description Helps to render interlinear glossed linguistic examples in html 'rmarkdown' documents and then semi-automatically compiles the list of glosses at the end of the document. It also provides a database of linguistic glosses.

Language en-US

License GPL (>= 3)

Encoding UTF-8

LazyData true

URL https://CRAN.R-project.org/package=phonfieldwork, https://agricolamz.github.io/lingglosses/

BugReports https://github.com/agricolamz/lingglosses/issues

Imports kableExtra, knitr, rmarkdown, utils, htmltools, methods

RoxygenNote 7.2.3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author George Moroz [aut, cre] (<https://orcid.org/0000-0003-1990-6083>)

Repository CRAN

Date/Publication 2022-12-19 11:40:02 UTC

R topics documented:

add_gloss 2
convert_to_df 3
get_examples_db 4
glosses_df 4
gloss_example 5
make_gloss_list 6

Index **8**

`add_gloss` *Gloss an example*

Description

Adds glosses to the glosses list and adds small capitals to glosses. Escapes strings that begins and ends with curly brackets.

Usage

```
add_gloss(glosses)
```

Arguments

`glosses`	character vector with glosses in upper case.

Value

vector of small capitalized glosses (if string is in the upper case) and not glosses (if string is not in the upper case)

Author(s)

George Moroz <agricolamz@gmail.com>

Examples

```
add_gloss(c("ABS", "ERG"))
```

convert_to_df *Converts example to a data.frame*

Description

Converts example to a data.frame and adds it to the database of Interlinear-Glossed examples.

Usage

```
convert_to_df(
  transliteration,
  glosses,
  free_translation = "",
  comment = "",
  annotation = NULL,
  drop_transliteration = FALSE,
  write_to_db = TRUE,
  counter = getOption("lingglosses.example_counter")
)
```

Arguments

transliteration	character vector of the length one for the transliteration line.
glosses	character vector of the length one for the glosses line.
free_translation	character vector of the length one for the free translation line.
comment	character vector of the length one for the comment line (under the free translation line).
annotation	character vector of the length one for the annotation line (above translation).
drop_transliteration	logical variable that denotes, whether user wants to have an example without transliteration.
write_to_db	logical variable that denotes, whether example should be added to the example database.
counter	double, value that denotes example id. By default gathered automatically through hidden variables in the Rmd document.

Value

dataframe with

Author(s)

George Moroz <agricolamz@gmail.com>

get_examples_db *Get database of interlinear examples*

Description

Reads database of interlinear examples collected through the whole document.

Usage

```
get_examples_db()
```

Value

a dataframe with all interlinear examples from rmarkdown document.

Author(s)

George Moroz <agricolamz@gmail.com>

glosses_df *Catalog of glosses*

Description

A dataset contains the list of glosses from the Leipzig Glossing Rules by Comrie, Haspelmath, and Bickel and other glosses automatically gathered from Glossa Journal articles.

Usage

```
glosses_df
```

Format

A data frame with 1341 rows and 4 variables:

gloss the gloss abbreviation

definition_en the gloss definition

source the gloss source. Three possible values: Leipzig Glossing Rules, Wikipedia or lingglosses (this means parsed from Glossa).

weight glossa weight used for the choice in case of multiple definitions per gloss.

gloss_example *Gloss an example*

Description

Creates an interlinear glossed example for linguistics.

Usage

```
gloss_example(
  transliteration,
  glosses,
  free_translation = "",
  comment = "",
  annotation = NULL,
  grammaticality = NULL,
  audio_path = NULL,
  audio_label = "",
  video_path = NULL,
  video_width = 320,
  video_height = 240,
  line_length = 70,
  italic_transliteration = getOption("lingglosses.italic_transliteration"),
  drop_transliteration = FALSE,
  intext = FALSE,
  write_to_db = TRUE
)
```

Arguments

transliteration	character vector of the length one for the transliteration line.
glosses	character vector of the length one for the glosses line.
free_translation	character vector of the length one for the free translation line.
comment	character vector of the length one for the comment line (under the free translation line).
annotation	character vector of the length one for the annotation line (above translation).
grammaticality	character vector with the grammaticality value.
audio_path	character string with the path to the sound in .wav format.
audio_label	character string for the label to display.
video_path	character string with the path to the video.
video_width	width argument for the video in px.
video_height	height argument for the video in px.

`line_length` integer vector of the length one that denotes maximum number of characters per one line.

`italic_transliteration` logical variable that denotes, whether user wants to italicize your example.

`drop_transliteration` logical variable that denotes, whether user wants to have an example without transliteration.

`intext` logical variable that denotes, whether example should be considered as part of the text (`TRUE`) or as a standalone paragraph (`FALSE`)

`write_to_db` logical variable that denotes, whether example should be added to the example database.

Value

html/latex output(s) with glossed examples.

Author(s)

George Moroz <agricolamz@gmail.com>

Examples

```
gloss_example("bur-e-**ri** c'in-ne-s:u",
              "fly-NPST-**INF** know-HAB-NEG",
              "I cannot fly. (Zilo Andi, East Caucasian)",
              grammaticality = "*",
              comment = "(lit. do not know how to)")

gloss_example("bur-e-**ri** c'in-ne-s:u",
              "fly-NPST-**INF** know-HAB-NEG",
              "I cannot fly.",
              intext = TRUE)
```

`make_gloss_list` *Make a gloss list*

Description

Creates a gloss list based on glosses used in `gloss_example`. This function tries to guess the meaning of used glosses based on some internal database or database provided by user. You shouldn't treat result as carved in stone: you can copy, modify and paste in your markdown document. If you want your glossing list to be created automatically with `make_gloss_list` you can compile your own table in the `definition_source` argument.

Usage

```
make_gloss_list(
  definition_source = lingglosses::glosses_df,
  remove_glosses = "",
  all_possible_variants = FALSE,
  annotate_problematic = TRUE
)
```

Arguments

definition_source	dataframe with the columns `gloss` and `definition` that helps to automatic search for the gloss definitions.
remove_glosses	character vector that contains glosses that should be removed from the abbreviation list.
all_possible_variants	logical. Some glosses have multiple definitions.
annotate_problematic	logical. Whether emphasize duplicated and definitionless glosses

Value

a string with glosses and their definitions gathered from `definition_source` table.

Author(s)

George Moroz <agricolamz@gmail.com>

Index

∗ **datasets**
glosses_df, 4

add_gloss, 2

convert_to_df, 3

get_examples_db, 4
gloss_example, 5, *6*
glosses_df, 4

make_gloss_list, 6