

Package: readtextgrid (via r-universe)

August 24, 2024

Type Package

Title Read in a 'Praat' 'TextGrid' File

Version 0.1.2.9000

Description 'Praat' <<https://www.fon.hum.uva.nl/praat/>> is a widely used tool for manipulating, annotating and analyzing speech and acoustic data. It stores annotation data in a format called a 'TextGrid'. This package provides a way to read these files into R.

License GPL-3

Encoding UTF-8

Depends R (>= 4.1.0)

Suggests testthat (>= 2.1.0)

RoxygenNote 7.3.1

Imports stats, utils, tibble, purrr, readr, stringr, plyr, dplyr, rlang

URL <https://github.com/tjmahr/readtextgrid>

BugReports <https://github.com/tjmahr/readtextgrid/issues>

Roxygen list(markdown = TRUE)

Repository <https://tjmahr.r-universe.dev>

RemoteUrl <https://github.com/tjmahr/readtextgrid>

RemoteRef HEAD

RemoteSha 501f28ca94da91747ce97b29872613002a3b8e40

Contents

example_textgrid	2
pivot_textgrid_tiers	2
read_textgrid	4

Index	5
--------------	----------

example_textgrid *Locate the path of an example textgrid file*

Description

Locate the path of an example textgrid file

Usage

```
example_textgrid(which = 1)
```

Arguments

which index of the textgrid to load

Details

This function is a wrapper over `system.file()` to locate the paths to bundled textgrids. These files are used to test or demonstrate functionality of the package.

Two files are included:

1. "Mary_John_bell.TextGrid" - the default TextGrid created by Praat's Create TextGrid command. This file is saved as UTF-8 encoding.
2. "utf_16_be.TextGrid" - a TextGrid with some IPA characters entered using Praat's IPA character selector. This file is saved with UTF-16 encoding.
3. "nested-intervals.TextGrid" - A textgrid containing an "utterance" tier, a "words" tier, and a "phones" tier. This file is typical of forced alignment textgrids where utterances contain words which contain speech segments. In this case, alignment was made by hand so that word and phone boundaries do not correspond exactly.

Value

Path of "Mary_John_bell.TextGrid" bundled with the readtextgrid package.

pivot_textgrid_tiers *Pivot a textgrid into wide format, respecting nested tiers*

Description

Pivot a textgrid into wide format, respecting nested tiers

Usage

```
pivot_textgrid_tiers(data, tiers, join_cols = "file")
```

Arguments

<code>data</code>	a textgrid dataframe created with <code>read_textgrid()</code>
<code>tiers</code>	character vector of tiers to pivot into wide format. When <code>tiers</code> has more than 1 element, the tiers are treated as nested. For example, if <code>tiers</code> is <code>c("utterance", "word", "phone")</code> , where "utterance" intervals contain "word" intervals which in turn contain "phone" intervals, the output will have one row per "phone" interval and include <code>utterance_*</code> and <code>word_*</code> columns for the utterance and word intervals that contain each phone interval. <code>tiers</code> should be ordered from broadest to narrowest (e.g, "word" preceding "phone").
<code>join_cols</code>	character vector of the columns that will uniquely identify a textgrid file. Defaults to "file" because these columns have identical values for tiers read from the same textgrid file.

Details

For the joining nested intervals, two intervals a and b are combined into the same row if they match on the values in the `join_cols` columns and if the $a\$x_{min} \leq b\x_{mid} and $b\$x_{mid} \leq a\x_{max} . That is, if the midpoint of b is contained inside the interval a .

Value

a dataframe with just the intervals from tiers named in `tiers` converted into a wide format. Columns are renamed so that the `text` column is pivot into columns named after the tier names. For example, the `text` column in a `words` tier is renamed to `words`. The `xmax`, `xmin`, `annotation_num`, `tier_num`, `tier_type` are also prefixed with the tier name. For example, the `xmax` column in a `words` tier is renamed to `words_xmax`. An additional helper column `xmid` is added and prefixed appropriately. See examples below.

Examples

```
data <- example_textgrid(3) |>
  read_textgrid()
data

# With a single tier, we get just that tier with the columns prefixed with
# the tier_name
pivot_textgrid_tiers(data, "utterance")
pivot_textgrid_tiers(data, "words")

# With multiple tiers, intervals in one tier that contain intervals in
# another tier are combined into the same row.
a <- pivot_textgrid_tiers(data, c("utterance", "words"))
cols <- c(
  "utterance", "utterance_xmin", "utterance_xmax",
  "words", "words_xmin", "words_xmax"
)
a[cols]

a <- pivot_textgrid_tiers(data, c("utterance", "words", "phones"))
cols <- c(cols, "phones", "phones_xmin", "phones_xmax")
```

a[cols]

read_textgrid *Read a textgrid file into a tibble*

Description

Read a textgrid file into a tibble

Usage

```
read_textgrid(path, file = NULL, encoding = NULL)
```

```
read_textgrid_lines(lines, file = NULL)
```

Arguments

path	a path to a textgrid
file	an optional value to use for the file column. For <code>read_textgrid()</code> , the default is the base filename of the input file. For <code>read_textgrid_lines()</code> , the default is NA.
encoding	the encoding of the textgrid. The default value NULL uses <code>readr::guess_encoding()</code> to guess the encoding of the textgrid. If an encoding is provided, it is forwarded to <code>[readr::locale()]</code> and <code>[readr::read_lines()]</code> .
lines	alternatively, the lines of a textgrid file

Value

a tibble with one row per textgrid annotation

Examples

```
tg <- system.file("Mary_John_bell.TextGrid", package = "readtextgrid")
read_textgrid(tg)
```

Index

`example_textgrid`, 2

`pivot_textgrid_tiers`, 2

`read_textgrid`, 4

`read_textgrid()`, 3

`read_textgrid_lines(read_textgrid)`, 4

`readr::guess_encoding()`, 4

`system.file()`, 2