Package 'speech'

December 22, 2019

Type Package

Title Legislative Speeches

Version 0.1.0

Description Converts the floor speeches of Uruguayan legislators, extracted from the parliamentary minutes, to tidy data.frame where each observation is the intervention of a single legislator.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.6.0)

URL https://github.com/Nicolas-Schmidt/speech

Imports dplyr, lubridate, magrittr, purrr, tabulizer, stringr, tibble, tm, tidyr, pdftools

RoxygenNote 7.0.2

NeedsCompilation no

Author Nicolas Schmidt [aut, cre], Diego Lujan [aut], Juan Andres Moraes [aut]

Maintainer Nicolas Schmidt <nschmidt@cienciassociales.edu.uy>

Repository CRAN

Date/Publication 2019-12-22 16:10:02 UTC

R topics documented:

speech_build *Transform speeches in pdf to data.frame*

Description

It allows to extract the individual speeches of each legislator in a document and obtain a data.frame.

Usage

```
speech_build(
  file,
  add.error.sir = NULL,
  rm.error.leg = NULL,
  compiler = FALSE,
  quality = FALSE,
  param = list(char = 6500, drop.page = 2)
)
```

Arguments

file	list or character vector specifying the path or URL to a PDF file. It can be one or more files.
add.error.sir	character vector. It allows to specify different ways in which the term that orders the speeches could be miswritten: sir. By default it is NULL.
rm.error.leg	character vector. It allows to add legislator's names to be eliminated. By default it is NULL. By default, "PRESIDENTE", "SECRETARIO", "SUBSECRETARIO", and "MINISTRO" are eliminated.
compiler	logical. When the checking of the process of conversion from pdf to data frame is completed, it is necessary to compile the data frame. To compile implies to unite all the speeches of each of the legislators for each document. As it is an operation that must be carried out after making corrections, it is necessary to opt for it. By default it is FALSE.
quality	logical. If TRUE, two quality indicators are added about the process, according to the quality of the document. • index_1: Proportion of the text recovered according to the original document (param = list(char = 6500,drop.page = 2)) that must have the document. • index_2: Proportion of the final text as a function of the recovered text. It is the proportion of the document in which there are only interventions by legislators.
param	list of length 2 with magnitudes for arguments "character for page" and "drop page non evaluate" respectively. The default values are the median characters of 8500 documents that make up the speech datasets.

Details

This function converts PDF documents to data.frame. The conversion is made by seeking interventions of legislators from the word "SENOR". As the quality of PDF files is not always the best it is recommended to verify that no legislator is omitted in the data.frame construction process. To make corrections of the word "SENOR" is that the argument `add.error.sir` should be used. The function has a long list of different ways in which the word "SENOR" may be written in a document, but not all possible future problems are covered. When the PDF document is a scan that was treated with an OCR, it should be checked with greater caution to ensure that the operation was performed correctly.

Value

data.frame class puy with the following variables:

- `legislator`: name of the legislators
- `speech`: speeches by legislators
- `date`: session date
- `id`: name `file`
- `legislature`: legislature id (period of government)
- `chamber`: chamber to which the document belongs. It can be: Chamber of Representatives, Senate, General Assembly or Permanent Commission.

If quality is TRUE, the following are added:

- `index_1`: index_1
- `index_2`: index_2

Examples

```
url <- "http://bit.ly/35AUVF4"
out <- speech_build(file = url)

out <- speech_build(file = url, compiler = FALSE,
                    quality = TRUE,
                    add.error.sir = c("SEf'IOR"),
                    rm.error.leg = c("PRtSIDENTE", "SUB", "PRfSlENTE"),
                    param = list(char = 6000, drop.page = 3))

out <- list.files(pattern = "*.pdf") %>% speech_build()

out <- list.files(pattern = "*.pdf") %>%
    speech_build(., compiler = TRUE, param = list(char = 4500, drop.page = 3))
```

speech_check *Check the names of legislators*

Description

It allows to check that the names of the legislators are correctly written before compiling the documents in speech_build.

Usage

```
speech_check(tidy_speech, initial, expand = FALSE)
```

Arguments

tidy_speech	data.frame class puy
initial	character vector. Initial of the legislators' names. If no initial is entered, all will be checked.
expand	logical. If TRUE, the legislature to which the name of the legislator belongs is shown. By default By default is FALSE.

Value

list with a data.frame for each initial of legislators' names.

Examples

```
url <- "http://bit.ly/35AUVF4"
out <- speech_build(file = url)
speech_check(out, initial = c("A", "M"), expand = FALSE)
```

speech_legis_replace *Rename legislators*

Description

allows to modify the legislators' name prior to compiling the data.

Usage

```
speech_legis_replace(tidy_speech, old, new, id = NULL)
```

Arguments

tidy_speech	data.frame class puy.
old	old legislator's name.
new	new legislator's name.
id	id 'floor speech'.

Value

data.frame.

Examples

```
url <- "http://bit.ly/35AUVF4"
out <- speech_build(file = url)
speech_check(out, "G")
out <- speech_legis_replace(out, old = "GOI",  new = "GONI")
```

---

speech_recompiler *Speech recompiler*

Description

It allows to recompile an object of the puy class, the datasets speech or a data.frame built with speech_build to which the variable political party was added.

Usage

```
speech_recompiler(
  tidy_speech,
  compiler_by = c("legislator", "party", "legislature", "chamber")
)
```

Arguments

tidy_speech	data.frame.
compiler_by	character vector. Variables for which you may want to recompile the data frame.

Details

The default compilation is that of \ code speech_build (., compiler = TRUE). This function allows to recompile the data by different levels of aggregation: chamber, legislature or other variables.

Value

data.frame.

Examples

```
url <- "http://bit.ly/35AUVF4"
out <- speech_build(file = url)
out2 <- speech_recompiler(out)
out2 <- speech_recompiler(out, compiler_by = c("legislator", "legislature", "chamber"))
```

---

speech_view *View control speech*

Description

Allows to see the legislators' names with problems prior to compiling the data.

Usage

```
speech_view(tidy_speech, legis = character(), view = FALSE)
```

Arguments

tidy_speech	data.frame class puy.
legis	name of the legislator.
view	logical. If TRUE View displays datasets containing legislators' interventions (legis). By default is FALSE.

Value

data.frame.

Examples

```
url <- "http://bit.ly/35AUVF4"
out <- speech_build(file = url)
speech_view(tidy_speech = out, legis = c("ABDALA", "LAZO"), view = FALSE)
```

speech_word_count *Number of words*

Description

Word count.

Usage

```
speech_word_count(
  string,
  exclude = NULL,
  min.char = 0L,
  rm.long = Inf,
  rm.num = FALSE,
  replace.punct = ""
)
```

Arguments

string	character of length equal to or greater than one.
exclude	words that are to be excluded from counting.
min.char	integer that determines the words that have less than a certain number of characters.
rm.long	integer that determines the number of characters from which words have to be deleted from the count.
rm.num	logical. Indicates whether the numbers in the count will be eliminated.
replace.punct	logical. If TRUE punctuation marks within a single word will be replaced by a space. By default is TRUE.

Value

integer.

Examples

```
vec <- "Hello world!"
speech_word_count(vec)

vec2 <- "Hello.world!"
speech_word_count(vec2)
speech_word_count(vec2, replace.punct = " ")

vec3 <- "Hello.world!, HelloHelloHelloHelloHelloHello"
speech_word_count(vec3, replace.punct = " ", rm.long = 20)

speech_word_count("R version", min.char = 1)
```

```
r <- "R version 3.5.2 (2018-12-20) -- 'Eggshell Igloo'"
speech_word_count(r, rm.num = TRUE)

speech_word_count(NA)
```

Index