Estruturador de Notas Taquigráficas

Autor: Alisson Soares

26 de abril de 2022

Script em R que estrutura em dado tabular e salva em csv e Rds as Notas Taquigráficas do Senado. Testada com os arquivos da CPI da Pandemia.

Carregando os pacotes necessários

```
library(rvest)
library(stringr)
library(dplyr)
library(purrr)
```

Criando os subdiretórios para os arquivos que vamos criar

```
dir.create(paste0(getwd(),"/rds"), showWarnings = F)
dir.create(paste0(getwd(),"/csv"), showWarnings = F)
```

Criando a listagem com as notas taquigráficas e respectivos links

Cria um tibble com:

- data
- número da reunião
- Descrição das notas taquigráficas
- links

```
# URL de exemplo. Tal como está com este escopo de tempo, imprime todas as reuniões e links numa mesma p
url = "https://legis.senado.leg.br/comissoes/comissao?codcol=2441&data1=2021-04-05&data2=2021-10-26"

oitivas <- read_html(url)

oitivas.vetor <- oitivas %>% html_elements('.row:nth-child(2) .content .col-md-12')

datas.vetor <- oitivas.vetor %>%
        html_element('a:nth-child(1) span:nth-child(1)') %>% html_text() %>%
        grep("^$", .,value=T, invert = T) %>%
        gsub("([0-9]{2})/([0-9]{2})/([0-9]{4})", "\\3-\\2-\\1",.)
reuniao_dia <-  oitivas.vetor %>% html_element('span+ span') %>% html_text() %>%
        grep("Reunião", .,value = T) %>%
        gsub("([0-9]+).*Reunião.*", "\\1", .) %>% as.integer()
Depoente.tema <-  oitivas.vetor %>% html_element('.f2') %>% html_text()
link_notaTaquigrafica <- oitivas.vetor %>% html_element('.bgc-cpi:nth-child(4) a') %>%
        html_attr('href')
```

```
nt.lista <- tibble(data = datas.vetor, reuniao_dia, Depoente.tema, link_notaTaquigrafica)
head(nt.lista)
## # A tibble: 6 x 4
##   data       reuniao_dia Depoente.tema                    link_notaTaquig~
##   <chr>            <int> <chr>                            <chr>
## 1 2021-10-26          69 Discussão e Deliberação do Rela~ http://www25.se~
## 2 2021-10-26          69 Discussão e Deliberação do Rela~ http://www25.se~
## 3 2021-10-20          68 Apresentação do Relatório Final  http://www25.se~
## 4 2021-10-19          67 Oitiva - Elton da Silva Chaves   http://www25.se~
## 5 2021-10-19          68 Apresentação do Relatório Final  <NA>
## 6 2021-10-18          66 1ª PARTE - Audiência Pública In~ http://www25.se~
```

Salvando a listagem de links

Caso porventura queira salvar a listagem com as notas localmente (não é necessário)

```
write.csv(nt.lista,"csv/NotasTaq-CPI_Pandemia-listagem.csv")
saveRDS(nt.lista,"rds/NotasTaq-CPI_Pandemia-listagem.Rds")
```

Criando o tibble estruturado com as Notas Taquigráficas

Função que estrutura as notas taquigráficas e cria o tibble/data-frame

```
# Função que gera nome do arquivo a partir de nt.lista para ser usado ao salvar arquivo
nomeArqRds <- function(N.arq){
        paste0("/rds/NT_",nt.lista[N.arq,]$reuniao_dia, "-",nt.lista[N.arq,]$Depoente.tema, ".") %>%
                gsub("ª|,", "",.) %>% gsub(" - ", "-", .) %>% gsub(" ", "_",.)
}

# siglas dos Estados da federação
siglas.estados <- "\\b(AC|AL|AP|AM|BA|CE|DF|ES|GO|MA|MT|MS|MG|PA|PB|PR|PE|PI|RJ|RN|RS|RO|RR|SC|SP|SE|TO
# Todos os partidos políticos do Brasil devem estar nesta lista
TodosPartidos = "\\b(AVANTE|CIDADANIA|DC|DEM|MDB|NOVO|PATRIOTA|PC|PCB|PCdoB|PCO|PDT|PL|PMB|PMN|PODEMOS|
```

A função abaixo - `func_DB_NT()` - a partir da listagem de links geradas e salvas no objeto `nt.lista`, recebe de input o índice/número da linha, e a partir do html, estrutura os dados com regex em um tibble e o salva nos formatos csv e rds em seus respectivos diretórios.

```
func_DB_NT <- function(linha){
        message("Baixando: ", nt.lista[linha,2],", ", nt.lista[linha,3] ,"\".
                Analisando url: \"",nt.lista[linha,4],"\"")
        url_atual <- nt.lista[linha,4] %>% as.character()

# carregando o conteúdo da url numa variável, dentro do R
        NT_html <- url_atual %>% read_html(., encoding = "utf8")

# texto da pagina
        texto <- NT_html %>% html_nodes('.escriba-jq') %>% html_text()
        alerta <- NT_html %>% html_element('.alert') %>% html_text()
        message("Aviso no arquivo: ", alerta)

        texto_vetores0 = gsub('[0-9]{2}\\:[0-9]{2}  R|\\(Pausa\\.\\)', '', texto) %>%
                gsub('(O SR|A SRA)\\.', 'ZZZVECTOR_\\1\\.', .) %>%
                strsplit(. , "ZZZVECTOR_") %>% unlist()
```

```r
# limpando a linha 1 da tabela, se ela contiver o indesejável texto abaixo (sempre vem)
        reuniao = texto_vetores0[1] %>% gsub(".* ([0-9]+)ª.*", "\\1",.)
        regex.1linha = "\n\n+.*@import.*Texto com revisão +"

        if (grepl(regex.1linha, texto_vetores0[1])){
          texto_vetores = texto_vetores0[-1]
          } else {
            texto_vetores = texto_vetores0
          }

          ExpReg <- '(O SR|A SRA)\\. ([A-ZÀ-Ÿ \\.]+)(\\(.*?\\)| ?)([-   --]{3})(.*)'
          vetor_nomes = unlist(str_extract_all(texto_vetores, ExpReg))
          nome = gsub(ExpReg,'\\2', texto_vetores) %>% gsub(' $','',.)
          unique(nome)
          funcao_bloco = gsub(ExpReg,'\\3', texto_vetores)
          fala <- gsub(ExpReg,'\\5', texto_vetores)

          cargo_funcao = gsub(ExpReg,'\\3', texto_vetores)

          estado <- gsub(paste0(".*",siglas.estados,".*"), "\\1", cargo_funcao) %>%
                  unlist()

          regex.bloco = paste0(".*([Bb]loco.*[Pp]arlamentar.*)\\/(",
                               TodosPartidos,") - ", siglas.estados, "(.*)")
          BlocoParl <- gsub(regex.bloco,"\\1",cargo_funcao)

          partido1 <- cargo_funcao %>%
                  gsub(".*(\\/|\\. |\\()([A-Z]+) . ([A-Z]{2}).*).*","\\2",.)
          # adicionando NA para a lista preservar seu tamanho
          partido1[lengths(partido1) == 0] <- NA
          # pegando apenas o último elemento da lista
          partido <- sapply(partido1, tail,1) %>% unlist()

          complemento = gsub(paste0(".*", siglas.estados, "(.*)"),"\\2", funcao_bloco) %>%
                  gsub("\\.|)","",.) %>% str_trim()
  #criando um tibble vazio
  NotasTaq_db <- tibble(reuniao, data = nt.lista$data[linha], Nome = nome,
                        funcao_bloco, BlocoParl, partido, estado,
                        complemento, fala)
  # Trocando "nome" por nome - função
  regex.nome = "\\((.*?)\\..*"

# 'Presidente' aparece na coluna 'nome'. vamos colocá-lo na coluna 'funcao_bloco'
  NTDB <- NotasTaq_db %>%
      mutate(nome = ifelse(Nome == "PRESIDENTE", funcao_bloco, Nome), .before = Nome) %>%
      mutate(funcao_blocoPar = ifelse(Nome == "PRESIDENTE", Nome, funcao_bloco), .before = funcao_bloco
      select(!c(Nome, funcao_bloco)) %>%
       # limpando: pegando apenas o nome na var nome, deixando de fora partido, bloco parlamentar e est
      mutate(nome =   gsub(regex.nome, "\\1",nome))

# Criando o nome de arquivos que serão salvos
  nomearq = paste0("NT_",nt.lista[linha,]$reuniao_dia, "-",nt.lista[linha,]$Depoente.tema) %>%
          gsub("ª|,", "",.) %>% gsub(" - ", "-", .) %>% gsub(" ", "_",.)
```

```
## Salvando em arquivos csv rds rdata
  write.csv(NTDB, paste0("csv/", nomearq, ".csv"))
  saveRDS(NTDB, paste0("rds/", nomearq, ".Rds"))
}
```

Baixando apenas notas novas

Esta função é útil se o pesquisador(a) estiver acompanhando as Notas Taquigráficas à medida que são publicadas, para ir baixando somente os arquivos novos: A função abaixo checa se arquivo de Nota Taquigráfica que recebe de input já existe no computador. Se não existir, o baixa (chamando a função acima, a `func_DB_NT`).

```
# função para checar se arquivo já existe no diretório
arq_existente <- function(N.arq){
  nomearq = paste0(nomeArqRds(N.arq),"Rds")
  testeExisteArq = file.exists(paste0(getwd(),nomearq))

  if (testeExisteArq) {
    message("Arquivo \"", nomearq, "\" já existe.")
    }  else {
      message("Arquivo \"", nomearq, "\" NÃO existe localmente no diretório. Processando...")
              if (is.na(nt.lista[N.arq,]$link_notaTaquigrafica)) {
                message("Ops! Porém não há ainda link disponível para esta nota taquigráfica.")
              } else{
               func_DB_NT(N.arq)
                Sys.sleep(5.5)
                    }
      }
}
```

Baixando as notas

Baixar um arquivo específico. o numero refere-se ao numero da linha em `nt.lista`, não ao número da reunião.

```
arq_existente(51)
```

Baixar todos os arquivos da listagem em `nt.lista`.

```
purrr::map_dfr(1:length(nt.lista$link_notaTaquigrafica), arq_existente)
```

Criando um único dataframe/tibble

Unindo todos os rds em um único tibble

```
dirwd <- getwd()
dir <- paste0(dirwd, "/rds/")
arqs <- list.files(dir, pattern = "NT_")
arqs2 <- paste0(dir,arqs)

# criando o data frame vazio
NT_todas_df <- data.frame(matrix(ncol=9, nrow=0))

# iterando e juntando todos os tibbles em um só
for (file in arqs2){
  message(file)
```

```
  rdstemp <- readRDS(file)
  NT_todas_df <- rbind(NT_todas_df,rdstemp)
}
```

Salvando este dataframe completo

```
saveRDS(NT_todas_df, "NT_todas_df.rds")
```