Sampler for Character Strings

Chris Bentz

14/20/2023

Description

This file contains code to sample strings of predefined lengths from the respective corpora (i.e. the data in the NaLaFi/data folder as well as TeDDi_dumps). The file needs to be run with "char.num" set to 10, 100, 1000 (or any other number of characters) separately.

Load libraries

If the libraries are not installed yet, you need to install them using, for example, the command: install.packages("ggplot2"). For the Hrate package this is different, since it comes from github. The devtools library needs to be installed, and then the install_github() function is used.

```
library(stringr)
```

List files

Create list with all the files in the directory "data".

```
# give file paths to the files to be processed
file.list <- list.files(path = "~/Github/NaLaFi/data",
                        recursive = T, full.names = T)
head(file.list)
```

```
## [1] "/home/chris/Github/NaLaFi/data/non-writing/animal/animal_bhg_0001.txt"
## [2] "/home/chris/Github/NaLaFi/data/non-writing/animal/animal_bhg_0002.txt"
## [3] "/home/chris/Github/NaLaFi/data/non-writing/animal/animal_bhg_0003.txt"
## [4] "/home/chris/Github/NaLaFi/data/non-writing/animal/animal_bhg_0004.txt"
## [5] "/home/chris/Github/NaLaFi/data/non-writing/animal/animal_bhg_0005.txt"
## [6] "/home/chris/Github/NaLaFi/data/non-writing/animal/animal_bhg_0006.txt"
```

```
length(file.list)
```

```
## [1] 291
```

```
#same for the teddi sample (downloaded from https://drive.switch.ch/index.php/s/MJv7xFkzqlzFn0y)
file.list.teddi <- list.files(path = "~/Data/TeDDi_dumps/Teddi_unifiedformat",
                        recursive = T, full.names = T)
head(file.list.teddi)
```

```
## [1] "/home/chris/Data/TeDDi_dumps/Teddi_unifiedformat/abk_pro_1.txt"
## [2] "/home/chris/Data/TeDDi_dumps/Teddi_unifiedformat/aey_nfi_1.txt"
## [3] "/home/chris/Data/TeDDi_dumps/Teddi_unifiedformat/amp_nfi_1.txt"
## [4] "/home/chris/Data/TeDDi_dumps/Teddi_unifiedformat/ape_nfi_1.txt"
## [5] "/home/chris/Data/TeDDi_dumps/Teddi_unifiedformat/apu_nfi_1.txt"
## [6] "/home/chris/Data/TeDDi_dumps/Teddi_unifiedformat/arn_nfi_1.txt"
```

```
length(file.list.teddi)
```

```
## [1] 23326
```

```
# downsample the number of teddi files
# (using all 23K files would yield an extremely unbalanced sample)
set.seed(09012020) # set seed to get same result when re-run
file.list.teddi <- sample(file.list.teddi, 100)

#concatenate the two lists
file.list.combined <- c(file.list, file.list.teddi)
length(file.list.combined)
```

```
## [1] 391
```

Sampler for Character Strings

```
# choose the length of chunks in number of characters
char.num <- 1000

# start time
start_time <- Sys.time()
for (file in file.list.combined){
  try({ # if the processing fails for a certain file, there will be no output for this file,
  # but the try() function allows the loop to keep running

  # basic processing
  # loading textfile
  textfile <- scan(file, what = "char", quote = "", comment.char = "",
                   encoding = "UTF-8", sep = "\n" , skip = 0)
  # remove the header lines beginning with '#'
  textfile <- textfile[!grepl('^#.*$', textfile)]
  # remove annotations marked by '<>'
  textfile <- gsub("<.*>","",textfile)
  # print(head(textfile))

  # Split into individual characters/signs
  # remove tabs and parentheses, as well as star signs `*' and plus signs `+´
  # note that this might have to be tuned according to the text files included
  textfile <- str_replace_all(textfile, c("\\\t" = "", "\\(" = "", "\\)" = "",
                                       "\\]" = "", "\\[" = "",  "\\}" = "",
                                       "\\{" = "", "\\*" = "", "\\+" = ""))
  # split the textfile into individual utf-8 characters. Note that white spaces are
  # counted as utf-8 characters here and not removed (to remove them uncomment line below).
  chars <- unlist(strsplit(textfile, ""))
  # chars <- chars[chars != " "] # remove white spaces from character vector
  # split into list of chunks of size char.num (i.e. 10, 100, 1000)
  chunks.list <- split(chars, ceiling(seq_along(chars)/char.num))
  # remove chunks from list which are shorter than char.num (the last chunk likely is)
  chunks.list <- Filter(function(x) length(x) == char.num, chunks.list)
  # use "next" statement to exclude empty chunks list
  if (length(chunks.list) == 0) {
    next
  }
  # limit number of text chunks to a defined maximum (since some text files are much larger than others
  max = 10
  chunks.list <- chunks.list[1:max]

  # prepare writing to file
  # get original filename
  filename <- basename(file)
  # create new file name
  # note: this is dependent on the exact file extensions (!)
  if (grepl('non-writing', file)){
    new.filename <- paste('non-writing_', filename, sep = "")
  } else if (grepl('writing', file)){
    new.filename <- paste('writing_', filename, sep = "")
  } else {
    new.filename <- paste('writing_teddi_', filename, sep = "")
  }

  # write to file
  lapply(chunks.list, write, paste("~/Github/NaLaFi/samples/",
                          paste(char.num, new.filename, sep = "_"), sep = ""),
         append = TRUE, ncolumns = char.num, sep = "")
  })
}
end_time <- Sys.time()
end_time - start_time
```

```
## Time difference of 5.72587 secs
```