Estimations of Quantitative Measures: StringBase Corpus

Chris Bentz

20/12/2022

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)
library(ggplot2)
library(ggrepel)
library(plyr)
library(entropy)
library(ggExtra)
library(gsubfn)
```

```
## Loading required package: proto
```

```
#library(devtools)
#install_github("dimalik/Hrate")
library(Hrate)
```

List files

Create list with all the files in the directory “corpus”.

```
# file list for stringBase texts
file.list <- list.files(path = "~/Github/NaLaFi/data/",
                        recursive = T, full.names = T)
head(file.list)
```

```
## [1] "/home/chris/Github/NaLaFi/data//generated/random/random_ran_10"
## [2] "/home/chris/Github/NaLaFi/data//generated/random/random_ran_11"
## [3] "/home/chris/Github/NaLaFi/data//generated/random/random_ran_12"
## [4] "/home/chris/Github/NaLaFi/data//generated/random/random_ran_13"
## [5] "/home/chris/Github/NaLaFi/data//generated/random/random_ran_14"
## [6] "/home/chris/Github/NaLaFi/data//generated/random/random_ran_15"
```

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

```
## [1] 329
```

Character entropy calculation per file

```
# set counter
counter = 0
# set the maximal number of units (n) to be used for analysis
n = 100
# initialize dataframe to append results to
estimations.df <- data.frame(filename = character(0), subcorpus = character(0),
                         code = character(0), huni.chars = numeric (0),
                         huni.strings = numeric(0), hrate.chars = numeric(0),
                         hrate.strings = numeric(0), ttr.chars = numeric(0),
                         ttr.strings = numeric (0))

# start time
start_time <- Sys.time()
for (file in file.list)
{
  # basic processing
  # loading textfile
  textfile <- scan(file, what = "char", quote = "", comment.char = "",
                   encoding = "UTF-8", sep = "\n" , skip = 7)
  # remove annotations marked by '<>'
  textfile <- gsub("<.*>","",textfile)
  # print(head(textfile))
  # get filename
  filename <- basename(file)
  print(filename) # for visual inspection
  # get subcorpus category
  subcorpus <- sub("_.*", "", filename)
  # print(subcorpus) # for visual inspection
  # get the three letter identification code + the running number
  code <- substring(substring(filename, regexpr("_", filename) + 1), 1, 8)
  # print(code) # for visual inspection

  # 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.
  chars <- unlist(strsplit(textfile, ""))
  chars <- chars[1:n] # use only maximally n units
  chars <- chars[!is.na(chars)] # remove NAs for vectors which are already shorter
  # than n
  # chars <- chars[chars != " "] # remove white spaces from character vector
  # Split the textfile into strings delimited by white spaces. The output of strsplit()
  # is a list, so it needs to be "unlisted"" to get a vector.
  strings <- unlist(strsplit(textfile, " "))
  strings <- strings[1:n] # use only maximally n units
  strings <- strings[!is.na(strings)] # remove NAs for vectors which are already
  # shorter than n
  # Unigram entropy estimation
  # calculate unigram entropy for characters
  chars.df <- as.data.frame(table(chars))
  # print(chars.df)
  huni.chars <- entropy(chars.df$Freq, method = "ML", unit = "log2")
  # calculate unigram entropy for strings (Maximum Likelihood method)
  strings.df <- as.data.frame(table(strings))
  # print(strings.df)
  huni.strings <- entropy(strings.df$Freq, method = "ML", unit = "log2")

  # entropy rate estimation
  # the values chosen for max.length and every.word will crucially
  # impact processing time. In case of "max.length = NULL" the length is n units
  hrate.chars <- get.estimate(text = chars, every.word = 1, max.length = NULL)
  hrate.strings <- get.estimate(text = strings, every.word = 1, max.length = NULL)

  # calculate type-token ratio (ttr)
  ttr.chars <- nrow(chars.df)/sum(chars.df$Freq)
  ttr.strings <- nrow(strings.df)/sum(strings.df$Freq)

  # calculate repetition measure according to Sproat (2014)
  # the overall number of repetitions is the sum of frequency counts minus 1.
  R <- sum(chars.df$Freq-1)
  # calculate the number of adjacent repetitions
  r = 0
  if (length(chars) > 1){
    for (i in 1:(length(chars)-1)){
      if (chars[i] == chars[i+1]){
        r = r + 1
      } else {
        r = r + 0
      }
    }
    # calculate the repetition measure
    rm.chars <- r/R
  } else {
    rm.chars <- "NA"
  }

  # append results to dataframe
  local.df <- data.frame(filename, subcorpus, code, huni.chars,
                         hrate.chars, huni.strings, hrate.strings,
                         ttr.chars, ttr.strings, rm.chars)
  estimations.df <- rbind(estimations.df, local.df)
  # counter
  counter <- counter + 1
  # print(counter)
}
```

```
## [1] "random_ran_10"
## [1] "random_ran_11"
## [1] "random_ran_12"
## [1] "random_ran_13"
## [1] "random_ran_14"
## [1] "random_ran_15"
## [1] "random_ran_16"
## [1] "random_ran_17"
## [1] "random_ran_18"
## [1] "random_ran_19"
## [1] "random_ran_2"
## [1] "random_ran_20"
## [1] "random_ran_21"
## [1] "random_ran_22"
## [1] "random_ran_23"
## [1] "random_ran_24"
## [1] "random_ran_25"
## [1] "random_ran_26"
## [1] "random_ran_27"
## [1] "random_ran_28"
## [1] "random_ran_29"
## [1] "random_ran_3"
## [1] "random_ran_30"
## [1] "random_ran_31"
## [1] "random_ran_32"
## [1] "random_ran_33"
## [1] "random_ran_34"
## [1] "random_ran_35"
## [1] "random_ran_36"
## [1] "random_ran_37"
## [1] "random_ran_38"
## [1] "random_ran_39"
## [1] "random_ran_4"
## [1] "random_ran_40"
## [1] "random_ran_41"
## [1] "random_ran_42"
## [1] "random_ran_43"
## [1] "random_ran_44"
## [1] "random_ran_45"
## [1] "random_ran_46"
## [1] "random_ran_47"
## [1] "random_ran_48"
## [1] "random_ran_5"
## [1] "random_ran_6"
## [1] "random_ran_7"
## [1] "random_ran_8"
## [1] "random_ran_9"
## [1] "shuffled_aii_0001"
## [1] "shuffled_akk_0001"
## [1] "shuffled_akk_0002"
## [1] "shuffled_arb_0001"
## [1] "shuffled_azj_0001"
## [1] "shuffled_azj_0002"
## [1] "shuffled_ben_0001"
## [1] "shuffled_bhg_0001"
## [1] "shuffled_bhg_0002"
## [1] "shuffled_bhg_0003"
## [1] "shuffled_bhg_0004"
## [1] "shuffled_bhg_0005"
## [1] "shuffled_bhg_0006"
## [1] "shuffled_bhg_0007"
## [1] "shuffled_bhg_0008"
## [1] "shuffled_bhg_0009"
## [1] "shuffled_bhg_0010"
## [1] "shuffled_bla_0001"
## [1] "shuffled_bla_0002"
## [1] "shuffled_blt_0001"
## [1] "shuffled_bod_0001"
## [1] "shuffled_bos_0001"
## [1] "shuffled_bos_0002"
## [1] "shuffled_cad_0001"
## [1] "shuffled_cad_0002"
## [1] "shuffled_cav_0001"
## [1] "shuffled_cav_0002"
## [1] "shuffled_cav_0003"
## [1] "shuffled_cav_0004"
## [1] "shuffled_cav_0005"
## [1] "shuffled_cav_0006"
## [1] "shuffled_cav_0007"
## [1] "shuffled_cav_0008"
## [1] "shuffled_cav_0009"
## [1] "shuffled_chr_0001"
## [1] "shuffled_cmn_0001"
## [1] "shuffled_cmn_0002"
## [1] "shuffled_cre_0001"
## [1] "shuffled_cre_0002"
## [1] "shuffled_csw_0001"
## [1] "shuffled_cth_0001"
## [1] "shuffled_cth_0002"
## [1] "shuffled_cth_0003"
## [1] "shuffled_cth_0004"
## [1] "shuffled_cth_0005"
## [1] "shuffled_cth_0006"
## [1] "shuffled_cth_0007"
## [1] "shuffled_cth_0008"
## [1] "shuffled_cth_0009"
## [1] "shuffled_cth_0010"
## [1] "shuffled_cth_0011"
## [1] "shuffled_div_0001"
## [1] "shuffled_dna_0001"
## [1] "shuffled_dna_0002"
## [1] "shuffled_dna_0003"
## [1] "shuffled_dna_0004"
## [1] "shuffled_dna_0005"
## [1] "shuffled_dna_0006"
## [1] "shuffled_dna_0007"
## [1] "shuffled_dna_0008"
## [1] "shuffled_dna_0009"
## [1] "shuffled_dna_0010"
## [1] "shuffled_dna_0011"
## [1] "shuffled_dna_0012"
## [1] "shuffled_dna_0013"
## [1] "shuffled_dna_0014"
## [1] "shuffled_dna_0015"
## [1] "shuffled_dna_0016"
## [1] "shuffled_dna_0017"
## [1] "shuffled_dna_0018"
## [1] "shuffled_dna_0019"
## [1] "shuffled_dna_0020"
## [1] "shuffled_dna_0021"
## [1] "shuffled_dna_0022"
## [1] "shuffled_dna_0023"
## [1] "shuffled_dna_0024"
## [1] "shuffled_dna_0025"
## [1] "shuffled_dna_0026"
## [1] "shuffled_dna_0027"
## [1] "shuffled_dna_0028"
## [1] "shuffled_dna_0029"
## [1] "shuffled_dna_0030"
## [1] "shuffled_ela_0001"
## [1] "shuffled_ela_0002"
## [1] "shuffled_ell_0001"
## [1] "shuffled_eng_0001"
## [1] "shuffled_epo_0001"
## [1] "shuffled_eus_0001"
## [1] "shuffled_gaz_0001"
## [1] "shuffled_guj_0001"
## [1] "shuffled_heb_0001"
## [1] "shuffled_hin_0001"
## [1] "shuffled_hye_0001"
## [1] "shuffled_ibb_0001"
## [1] "shuffled_iii_0001"
## [1] "shuffled_ike_0001"
## [1] "shuffled_jav_0001"
## [1] "shuffled_jav_0002"
## [1] "shuffled_jpn_0001"
## [1] "shuffled_kal_0001"
## [1] "shuffled_kan_0001"
## [1] "shuffled_kat_0001"
## [1] "shuffled_khm_0001"
## [1] "shuffled_kkh_0001"
## [1] "shuffled_kor_0001"
## [1] "shuffled_lao_0001"
## [1] "shuffled_lug_0001"
## [1] "shuffled_mal_0001"
## [1] "shuffled_moc_0001"
## [1] "shuffled_moc_0002"
## [1] "shuffled_moc_0003"
## [1] "shuffled_mya_0001"
## [1] "shuffled_pan_0001"
## [1] "shuffled_pra_0001"
## [1] "shuffled_prc_0001"
## [1] "shuffled_prc_0002"
## [1] "shuffled_prc_0003"
## [1] "shuffled_prc_0004"
## [1] "shuffled_rus_0001"
## [1] "shuffled_sin_0001"
## [1] "shuffled_sum_0001"
## [1] "shuffled_sum_0002"
## [1] "shuffled_sum_0003"
## [1] "shuffled_sum_0004"
## [1] "shuffled_sum_0005"
## [1] "shuffled_sum_0006"
## [1] "shuffled_sum_0007"
## [1] "shuffled_sum_0008"
## [1] "shuffled_sum_0009"
## [1] "shuffled_sum_0010"
## [1] "shuffled_tam_0001"
## [1] "shuffled_tel_0001"
## [1] "shuffled_tgl_0001"
## [1] "shuffled_tha_0001"
## [1] "shuffled_tir_0001"
## [1] "shuffled_tsl_0001"
## [1] "shuffled_vai_0001"
## [1] "shuffled_wsy_0001"
## [1] "shuffled_zfi_0001"
## [1] "shuffled_zgh_0001"
## [1] "shuffled_zul_0001"
## [1] "animal_bhg_0001.txt"
## [1] "animal_bhg_0002.txt"
## [1] "animal_bhg_0003.txt"
## [1] "animal_bhg_0004.txt"
## [1] "animal_bhg_0005.txt"
## [1] "animal_bhg_0006.txt"
## [1] "animal_bhg_0007.txt"
## [1] "animal_bhg_0008.txt"
## [1] "animal_bhg_0009.txt"
## [1] "animal_bhg_0010.txt"
## [1] "animal_cad_0001.txt"
## [1] "animal_cad_0002.txt"
## [1] "animal_cav_0001.txt"
## [1] "animal_cav_0002.txt"
## [1] "animal_cav_0003.txt"
## [1] "animal_cav_0004.txt"
## [1] "animal_cav_0005.txt"
## [1] "animal_cav_0006.txt"
## [1] "animal_cav_0007.txt"
## [1] "animal_cav_0008.txt"
## [1] "animal_cav_0009.txt"
## [1] "animal_cth_0001.txt"
## [1] "animal_cth_0002.txt"
## [1] "animal_cth_0003.txt"
## [1] "animal_cth_0004.txt"
## [1] "animal_cth_0005.txt"
## [1] "animal_cth_0006.txt"
## [1] "animal_cth_0007.txt"
## [1] "animal_cth_0008.txt"
## [1] "animal_cth_0009.txt"
## [1] "animal_cth_0010.txt"
## [1] "animal_cth_0011.txt"
## [1] "animal_zfi_0001.txt"
## [1] "morse_moc_0001.txt"
## [1] "morse_moc_0002.txt"
## [1] "morse_moc_0003.txt"
## [1] "natural_dna_0001.txt"
## [1] "natural_dna_0002.txt"
## [1] "natural_dna_0003.txt"
## [1] "natural_dna_0004.txt"
## [1] "natural_dna_0005.txt"
## [1] "natural_dna_0006.txt"
## [1] "natural_dna_0007.txt"
## [1] "natural_dna_0008.txt"
## [1] "natural_dna_0009.txt"
## [1] "natural_dna_0010.txt"
## [1] "natural_dna_0011.txt"
## [1] "natural_dna_0012.txt"
## [1] "natural_dna_0013.txt"
## [1] "natural_dna_0014.txt"
## [1] "natural_dna_0015.txt"
## [1] "natural_dna_0016.txt"
## [1] "natural_dna_0017.txt"
## [1] "natural_dna_0018.txt"
## [1] "natural_dna_0019.txt"
## [1] "natural_dna_0020.txt"
## [1] "natural_dna_0021.txt"
## [1] "natural_dna_0022.txt"
## [1] "natural_dna_0023.txt"
## [1] "natural_dna_0024.txt"
## [1] "natural_dna_0025.txt"
## [1] "natural_dna_0026.txt"
## [1] "natural_dna_0027.txt"
## [1] "natural_dna_0028.txt"
## [1] "natural_dna_0029.txt"
## [1] "natural_dna_0030.txt"
## [1] "signlang_tsl_0001.txt"
## [1] "weather_wsy_0001.txt"
## [1] "ancient_akk_0001.txt"
## [1] "ancient_akk_0002.txt"
## [1] "ancient_cre_0001.txt"
## [1] "ancient_cre_0002.txt"
## [1] "ancient_ela_0001.txt"
## [1] "ancient_ela_0002.txt"
## [1] "ancient_pra_0001.txt"
## [1] "ancient_prc_0001.txt"
## [1] "ancient_prc_0002.txt"
## [1] "ancient_prc_0003.txt"
## [1] "ancient_prc_0004.txt"
## [1] "ancient_sum_0001.txt"
## [1] "ancient_sum_0002.txt"
## [1] "ancient_sum_0003.txt"
## [1] "ancient_sum_0004.txt"
## [1] "ancient_sum_0005.txt"
## [1] "ancient_sum_0006.txt"
## [1] "ancient_sum_0007.txt"
## [1] "ancient_sum_0008.txt"
## [1] "ancient_sum_0009.txt"
## [1] "ancient_sum_0010.txt"
## [1] "heraldics_bla_0001.txt"
## [1] "heraldics_bla_0002.txt"
## [1] "writing_aii_0001.txt"
## [1] "writing_arb_0001.txt"
## [1] "writing_azj_0001.txt"
## [1] "writing_azj_0002.txt"
## [1] "writing_ben_0001.txt"
## [1] "writing_blt_0001.txt"
## [1] "writing_bod_0001.txt"
## [1] "writing_bos_0001.txt"
## [1] "writing_bos_0002.txt"
## [1] "writing_chr_0001.txt"
## [1] "writing_cmn_0001.txt"
## [1] "writing_cmn_0002.txt"
## [1] "writing_csw_0001.txt"
## [1] "writing_div_0001.txt"
## [1] "writing_ell_0001.txt"
## [1] "writing_eng_0001.txt"
## [1] "writing_epo_0001.txt"
## [1] "writing_eus_0001.txt"
## [1] "writing_gaz_0001.txt"
## [1] "writing_guj_0001.txt"
## [1] "writing_heb_0001.txt"
## [1] "writing_hin_0001.txt"
## [1] "writing_hye_0001.txt"
## [1] "writing_ibb_0001.txt"
## [1] "writing_iii_0001.txt"
## [1] "writing_ike_0001.txt"
## [1] "writing_jav_0001.txt"
## [1] "writing_jav_0002.txt"
## [1] "writing_jpn_0001.txt"
## [1] "writing_kal_0001.txt"
## [1] "writing_kan_0001.txt"
## [1] "writing_kat_0001.txt"
## [1] "writing_khm_0001.txt"
## [1] "writing_kkh_0001.txt"
## [1] "writing_kor_0001.txt"
## [1] "writing_lao_0001.txt"
## [1] "writing_lug_0001.txt"
## [1] "writing_mal_0001.txt"
## [1] "writing_mya_0001.txt"
## [1] "writing_pan_0001.txt"
## [1] "writing_rus_0001.txt"
## [1] "writing_sin_0001.txt"
## [1] "writing_tam_0001.txt"
## [1] "writing_tel_0001.txt"
## [1] "writing_tgl_0001.txt"
## [1] "writing_tha_0001.txt"
## [1] "writing_tir_0001.txt"
## [1] "writing_vai_0001.txt"
## [1] "writing_zgh_0001.txt"
## [1] "writing_zul_0001.txt"
```

```
end_time <- Sys.time()
end_time - start_time
```

```
## Time difference of 5.057698 secs
```

```
#estimations.df
```

Safe outputs to file

```
write.csv(estimations.df, "~/Github/NaLaFi/results/tables/estimations.csv", row.names = F)
```