August 8, 2022

The results below are generated from an R script.

```
##### libraries #####
library(stringr)

#### regex ####
regex_anthroponyme <- "(Mgr )?[:upper:][:lower:]+ (((l[aei']s?|d[euo']l?u?|au?)?){0,2} ?[:upper:][:lowe
regex_anthroponyme_caps <- "(MGR )?[:upper:]{2,} (((L[AEI']S?|D[EUO']L?U?|AU?)?){0,2} ?[:upper:]{2,}(-[

#### Methode pour capturer des EN ####
ren_extract <- function(text, first = FALSE){
  if (first){
    str_extract(text,regex_anthroponyme)
  } else {
    str_extract_all(text,regex_anthroponyme)[[1]]
  }
}

#### Methode pour capturer des EN en lettres capitales ####
ren_extract_caps <- function(text, first = FALSE){
  if (first){
    str_extract(text,regex_anthroponyme_caps)
  } else {
    str_extract_all(text,regex_anthroponyme_caps)[[1]]
  }
}

#### Methode derivee de Damerau-Levenshtein ####
DamerauLevenshtein_mod <- function(str1,str2){
    distance_modicateur <- 0
    if(str_ends(str1,"S") != str_ends(str2,"S")){
      distance_modicateur <- distance_modicateur - .75
   }
  if((str_detect(str1,"CE") && str_detect(str2,"CHE"))||
     (str_detect(str1,"CI") && str_detect(str2,"CHI"))){
    distance_modicateur <- distance_modicateur - .75

  } else if((str_detect(str1,"CHE") && str_detect(str2,"CE"))||
            (str_detect(str1,"CHE") && str_detect(str2,"CI"))){
    distance_modicateur <- distance_modicateur - .75
  }

  obj <- new("DamerauLevenshtein",
             deletion = 1,
             insertion = 1,
```

```
              substitution = 1.25,
              transposition = 1)
  return (obj(str1,str2)+distance_modicateur)
}

#### clustering ####
myClustering <- function(l_anthroponymes,clustering_lim,m_distance){
  l_cluster = list()
  for(i in 1:nrow(m_distance)){
    m_distance[i,1:i] <- NA #  pour eviter les doubles detections
    if(length(v_row <-l_anthroponymes[which(m_distance[i,] <= clustering_lim)])){
      l_cluster[[i]] <- c(l_anthroponymes[i],v_row)
    } else {
      l_cluster[[i]] <- NA
    }
  }
  l_cluster <- l_cluster[!is.na(l_cluster)]
  return(l_cluster)
}


#### Calcul de la distance Damerau-Levenshtein  ####
myDamereauLevenstheinDist <- function(v_string){
  distance <- NULL
  dim <- length(v_string)
  for(i in v_string[1:dim]){
    for(j in v_string[1:dim]){
      print(c(i,"  ", j))
      distance <- c(distance,DamerauLevenshtein_mod(i,j))
    }
  }
  m_distance = matrix(distance,nrow = dim,ncol = dim, byrow = TRUE)
  return(m_distance)
}
```

The R session information (including the OS info, R version and all packages used):

```
sessionInfo()

## R version 4.0.3 (2020-10-10)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS  12.3.1
##
## Matrix products: default
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib
##
## locale:
## [1] fr_BE.UTF-8/fr_BE.UTF-8/fr_BE.UTF-8/C/fr_BE.UTF-8/fr_BE.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices datasets  utils     methods   base
##
## other attached packages:
##  [1] RColorBrewer_1.1-3 concaveman_1.1.0   ggforce_0.3.3      scales_1.2.0
```

```
##  [5] ggrepel_0.9.1      readxl_1.3.1       tidygeocoder_1.0.5 ggraph_2.0.5.9000
##  [9] ggmap_3.0.0        igraph_1.3.0       comparator_0.1.2   forcats_0.5.1
## [13] dplyr_1.0.9        purrr_0.3.4        readr_2.1.2        tidyr_1.2.0
## [17] tibble_3.1.8       ggplot2_3.3.6      tidyverse_1.3.1    stringr_1.4.0.9000
##
## loaded via a namespace (and not attached):
##  [1] bitops_1.0-7        fs_1.5.2            lubridate_1.8.0     httr_1.4.2
##  [5] tools_4.0.3         backports_1.4.1     utf8_1.2.2          R6_2.5.1
##  [9] DBI_1.1.2           colorspace_2.0-3    withr_2.5.0         sp_1.5-0
## [13] tidyselect_1.1.2    gridExtra_2.3       curl_4.3.2          compiler_4.0.3
## [17] cli_3.3.0           rvest_1.0.2         xml2_1.3.3          proxy_0.4-26
## [21] digest_0.6.29       jpeg_0.1-9          pkgconfig_2.0.3     highr_0.9
## [25] dbplyr_2.1.1        rlang_1.0.4         rstudioapi_0.13     farver_2.1.1
## [29] generics_0.1.3      jsonlite_1.8.0      magrittr_2.0.3      Rcpp_1.0.9
## [33] munsell_0.5.0       fansi_1.0.3         viridis_0.6.2       lifecycle_1.0.1
## [37] stringi_1.7.6       MASS_7.3-53         plyr_1.8.7          grid_4.0.3
## [41] crayon_1.5.0        lattice_0.20-41     graphlayouts_0.8.0  haven_2.4.3
## [45] hms_1.1.1           knitr_1.37          pillar_1.8.0        rjson_0.2.21
## [49] reprex_2.0.1        glue_1.6.2          evaluate_0.15       renv_0.15.4
## [53] modelr_0.1.8        png_0.1-7           vctrs_0.4.1         tzdb_0.2.0
## [57] tweenr_1.0.2        RgoogleMaps_1.4.5.3 cellranger_1.1.0    gtable_0.3.0
## [61] polyclip_1.10-0     clue_0.3-60         assertthat_0.2.1    xfun_0.30
## [65] broom_0.7.12        tidygraph_1.2.1     viridisLite_0.4.0   tinytex_0.37
## [69] cluster_2.1.0       ellipsis_0.3.2
```

```
Sys.time()
```

```
## [1] "2022-08-08 07:41:02 CEST"
```