######préparation packages, wd, etc

setwd("C:/Users/lb3/OneDrive/sync/git/loubill/Domino/R/Domino/MTX")

install.packages(c('tm', 'SnowballC', 'wordcloud', 'topicmodels', "koRpus", "mclustcomp", "devtools", "installr", "RNewsflow", "koRpus","wordVectors"))

# installr::install.Rtools()

# devtools::install\_github("lbilliet/loubill")

# slam\_url <- "https://cran.r-project.org/src/contrib/Archive/slam/slam\_0.1-37.tar.gz"

# install\_url(slam\_url)

# urlSlam <- "https://cran.r-project.org/bin/windows/contrib/3.6/slam\_0.1-43.zip"

# install\_url(urlSlam)

# install\_github("bmschmidt/wordVectors")

#devtools::install\_github("bmschmidt/wordVectors")

library('devtools')

library(tm)

library(SnowballC)

library(wordcloud)

library(topicmodels)

library("koRpus")

library("wordVectors")

library("RNewsflow")

library("mclustcomp")

######import données

df.mtx<-read.csv2("data/mtx.csv", sep = "\n" ,header=F, quote = "", stringsAsFactors = F, fileEncoding = "UTF8")

df.alt<-read.csv2("data/alt.csv", sep = "\n" ,header=F, quote = "", stringsAsFactors = F, fileEncoding = "UTF8")

######conversion encodage suppression caractère speciaux

#définition fonctio suppression char speciaux

Unaccent <- function(text) {

text <- gsub("['`^~\"]", " ", text)

text <- iconv(text, to="ASCII//TRANSLIT//IGNORE")

text <- gsub("['`^~\"]", "", text)

return(text)

}

###application fonction suppr char speciaux

df.mtx.unaccent<-Unaccent(df.mtx)

df.alt.unaccent<-Unaccent(df.alt)

#constitution corpus

corpus.mtx = Corpus(VectorSource(df.mtx.unaccent))

corpus.alt = Corpus(VectorSource(df.alt.unaccent))

#####préparation texte

#retirer ponctuation

corpus.mtx<-tm\_map(corpus.mtx, removePunctuation)

corpus.alt<-tm\_map(corpus.alt, removePunctuation)

#retirer stopwords

corpus.mtx <- tm\_map(corpus.mtx, removeWords, stopwords("french"))

corpus.alt <- tm\_map(corpus.alt, removeWords, stopwords("french"))

#stemmatiser

corpus.mtx <- tm\_map(corpus.mtx, stemDocument)

corpus.alt <- tm\_map(corpus.alt, stemDocument)

#######consitution matrice documents

#matrice frequence termes

tdm.mtx <- TermDocumentMatrix(corpus.mtx)

tdm.alt <- TermDocumentMatrix(corpus.alt)

inspect(tdm.mtx)

inspect(tdm.alt)

#matrice frequence documents

dtm.mtx <- DocumentTermMatrix(corpus.mtx, control = list(weighting = weightTfIdf, stopwords = TRUE))

inspect(dtm.mtx)

dtm.alt <- DocumentTermMatrix(corpus.alt, control = list(weighting = weightTfIdf, stopwords = TRUE))

inspect(dtm.alt)

#démo rch termes retrouvés à une fréq de 2

freq.tdm.mtx<-findFreqTerms(tdm.mtx,lowfreq = 5, highfreq = Inf)

summary(freq.tdm.mtx)

freq.tdm.alt<-findFreqTerms(tdm.alt,lowfreq = 5, highfreq = Inf)

summary(freq.tdm.alt)

################################

#boucle constitution vecteur concaténation corpus mtx

txt.mtx<-NULL

for (i in 1 : nrow(df.mtx)){

txt.mtx<-paste0(result, df.mtx[i,])

}

#normalisation contenu textuel du vecteur

#suppression ponctuation

txt.mtx<-gsub("[[:punct:]]", " ", txt.mtx)###remove punct atxt cette méthode pour remplacer par des whitespaces plutôt que les suppr et risque de concaténer des mots ensembles et former de nouveaux mots

txt.mtx<-Unaccent(txt.mtx)

txt.mtx<-tolower(txt.mtx)

vec.mtx<-unlist(strsplit(txt.mtx, split = " "))

vec.mtx<-gsub(" ","",vec.mtx)

vec.mtx[vec.mtx != ""]

stem.vec.mtx<-wordStem(vec.mtx)

#boucle constitution vecteur concaténation corpus alt

txt.alt<-NULL

for (i in 1 : nrow(df.alt)){

txt.alt<-paste0(result, df.alt[i,])

}

#normalisation contenu textuel du txtteur

#suppression ponctuation

txt.alt<-gsub("[[:punct:]]", " ", txt.alt)###remove punct atxt cette méthode pour remplacer par des whitespaces plutôt que les suppr et risque de concaténer des mots ensembles et former de nouveaux mots

txt.alt<-Unaccent(txt.alt)

txt.alt<-tolower(txt.alt)

vec.alt<-unlist(strsplit(txt.alt, split = " "))

vec.alt<-gsub(" ","",vec.alt)

vec.alt[vec.alt != ""]

stem.vec.alt<-wordStem(vec.alt)

#mettre vecteurs même longueur

stem.vec.alt<-c(stem.vec.alt, rep(NA, length(stem.vec.mtx)-length(stem.vec.alt)))

write.table(stem.vec.alt, file = "data/stemVecAlt.txt", sep = "\t", row.names = F, quote = F)

write.table(stem.vec.mtx, file = "data/stemVecMtx.txt", sep = "\t", row.names = F, quote = F)

#comparaison via mclustcomp

mclustcomp(stem.vec.mtx,stem.vec.alt, types = "all")#pb ici il faut une boucle pour parcourir chaque vecteur de la df