library(tokenizers)

options(max.print = 25)

###############tokenizacion por oraciones#######################

james <- paste0(

"The question thus becomes a verbal one\n",

"again; and our knowledge of all these early stages of thought and feeling\n",

"is in any case so conjectural and imperfect that farther discussion would\n",

"not be worth while.\n",

"\n",

"Religion, therefore, as I now ask you arbitrarily to take it, shall mean\n",

"for us \_the feelings, acts, and experiences of individual men in their\n",

"solitude, so far as they apprehend themselves to stand in relation to\n",

"whatever they may consider the divine\_. Since the relation may be either\n",

"moral, physical, or ritual, it is evident that out of religion in the\n",

"sense in which we take it, theologies, philosophies, and ecclesiastical\n",

"organizations may secondarily grow.\n"

)

tokenize\_sentences(james)

coronavirus <- readLines("noticia.txt",encoding="UTF-8")

tokenize\_sentences(coronavirus)

##############tokenizacion por palarbas#################

tokenize\_words(james)

tokenize\_words(coronavirus)

tokenize\_words(coronavirus[[2]])

##################limpieza##########

####forma 1#######

library(tm)

coronavirus <- readLines("noticia.txt",encoding="UTF-8")

#coronavirus = iconv(coronavirus, to="ASCII//TRANSLIT")

corpus <- VCorpus(VectorSource(coronavirus))

d <- tm\_map(corpus, content\_transformer(tolower))

d <- tm\_map(d, stripWhitespace)

d <- tm\_map(d, removePunctuation)

d <- tm\_map(d,removeNumbers)

d[["3"]][["content"]]

coronavirus[[1]]

######forma 2###########

library(tidyverse)

library(dplyr)

limpiar\_tokenizar <- function(texto){

# El orden de la limpieza no es arbitrario

# Se convierte todo el texto a minúsculas

nuevo\_texto <- tolower(texto)

# Eliminación de páginas web (palabras que empiezan por "http." seguidas

# de cualquier cosa que no sea un espacio)

nuevo\_texto <- str\_replace\_all(nuevo\_texto,"http\\S\*", "")

# Eliminación de signos de puntuación

nuevo\_texto <- str\_replace\_all(nuevo\_texto,"[[:punct:]]", " ")

# Eliminación de números

nuevo\_texto <- str\_replace\_all(nuevo\_texto,"[[:digit:]]", " ")

# Eliminación de espacios en blanco múltiples

nuevo\_texto <- str\_replace\_all(nuevo\_texto,"[\\s]+", " ")

# Eliminacion de tildes

nuevo\_texto <- chartr('áéíóúñ','aeioun',nuevo\_texto)

# Tokenización por palabras individuales

nuevo\_texto <- str\_split(nuevo\_texto, " ")[[1]]

# Eliminación de tokens con una longitud < 2

nuevo\_texto <- keep(.x = nuevo\_texto, .p = function(x){str\_length(x) > 1})

return(nuevo\_texto)

}

test = "El presidente de Perú, Martin VIZCARRA nació en 1990 en cañete a las 12:00 del mediodia

según e'l portal l'limpieza de6 TEXTO https://t.co/rnHPgyhx4Z usuario tRodrigo #text"

limpiar\_tokenizar(texto = test)

coronavirus=as.data.frame(coronavirus)

coronavirus <- coronavirus %>% mutate(texto\_tokenizado = map(.x = coronavirus,

.f = limpiar\_tokenizar))

coronavirus %>% select(texto\_tokenizado) %>% head()

coronavirus %>% slice(1) %>% select(texto\_tokenizado) %>% pull()

coronavirus$ID <- seq.int(nrow(coronavirus))

####################Stop Words#######################

##############ingles#######################

library(stopwords)

tokenize\_words(james)

tokenize\_words(james, stopwords = stopwords::stopwords("en"))

length(stopwords(source = "smart"))

length(stopwords(source = "snowball"))

length(stopwords(source = "stopwords-iso"))

stopwords(language = "en", source = "smart")

###########español#####################

####palabras por defecto de la libreria###

d <- tm\_map(d, removeWords, stopwords("spanish"))

d[["1"]][["content"]]

####lista personalizada 1###############

sw <- readLines("D:/URP/MATERIAL\_CLASES/CURSO\_TEXT\_MINING/stopwordses.txt",encoding="UTF-8")

sw = iconv(sw, to="ASCII//TRANSLIT")

d <- tm\_map(d, removeWords, sw)

d[["1"]][["content"]]

####lista personalizada 2############

coronavirus\_tidy <- coronavirus %>% select(-coronavirus) %>% unnest()

head(coronavirus\_tidy)

# Se filtran las stopwords

coronavirus\_tidy <- coronavirus\_tidy %>% filter(!(texto\_tokenizado %in% sw))

head(coronavirus\_tidy)

#######stemming################

library(pacman)

library(textstem)

dw <- c('driver', 'drive', 'drove', 'driven', 'drives', 'driving')

stem\_words(dw)

y <- c(

'the dirtier dog has eaten the pies',

'that shameful pooch is tricky and sneaky',

"He opened and then reopened the food bag",

'There are skies of blue and red roses too!',

NA,

"The doggies, well they aren't joyfully running.",

"The daddies are coming over...",

"This is 34.546 above"

)

stem\_strings(y)

#######lematization################

dw <- c('driver', 'drive', 'drove', 'driven', 'drives', 'driving')

lemmatize\_words(dw)

y <- c(

'the dirtier dog has eaten the pies',

'that shameful pooch is tricky and sneaky',

"He opened and then reopened the food bag",

'There are skies of blue and red roses too!',

NA,

"The doggies, well they aren't joyfully running.",

"The daddies are coming over...",

"This is 34.546 above"

)

lemmatize\_strings(y)

####lemtizacion\_español########

#############stemming y lemmantizacion########

library(SnowballC)

library(corpus)

tabes=read.delim("D:/URP/MATERIAL\_CLASES/CURSO\_TEXT\_MINING/lemmatizationesp.txt",header = FALSE, sep = "",

stringsAsFactors = FALSE)

names(tabes) <- c("stem", "term")

head(tabes,10)

stem\_liste <- function(term) {

i <- match(term, tabes$term)

if (is.na(i)) {

stem <- term

} else {

stem <- tabes$stem[[i]]

}

stem

}

text <- "los presidentes de las regiones tuvieron que aumentar los dias de cuarentena debido al incremento de nuevos casos"

text\_tokens(text, stemmer = stem\_liste)

names(coronavirus\_tidy)[1] <- "term"

coronavirus\_tidy=coronavirus\_tidy %>%left\_join(tabes, by = "term")

coronavirus\_tidy$lemma=ifelse(is.na(coronavirus\_tidy$stem ), coronavirus\_tidy$term, coronavirus\_tidy$stem)