**Ejercicios Sesión 02**

**Nombres: Luz Victoria Márquez Fernández**

1. **Carga de archivo ‘noticias.txt’**

library(tokenizers)

options(max.print=250)

noticia2 <- readLines("noticia.txt",encoding="UTF-8")

1. **Tokenización**

tokenize\_sentences(noticia2)

tokenize\_words(noticia2)

tokenize\_words(noticia2[[1]])

library(tm)

corpus<-VCorpus(VectorSource(noticia2))

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

d<- tm\_map(d,stripWhitespace)

d<- tm\_map(d,removePunctuation)

d<- tm\_map(d,removeNumbers)

texto <- noticia2

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)

}

nueva\_noticia <- limpiar\_tokenizar(noticia2)

nueva\_noticia

noticia2=as.data.frame(noticia2)

noticia2 <- noticia2 %>% mutate(texto\_tokenizado = map(.x = noticia2,

.f = limpiar\_tokenizar))

noticia2 %>% select(texto\_tokenizado) %>% head()

noticia2 %>% slice(1) %>% select(texto\_tokenizado) %>% pull()

noticia2$ID <- seq.int(nrow(noticia2))

1. **Stopwords**

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

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

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

sw <- readLines("D:/Maestría/Ciclo IV/Cursos/MCD-402\_Análisis EstadisticoTexto/Sesion2/stopwordses.txt",encoding="UTF-8")

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

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

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

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

noticia2\_tidy <- noticia2 %>% select(-noticia2) %>% unnest()

head(noticia2\_tidy)

# Se filtran las stopwords

noticia2\_tidy <- noticia2\_tidy %>% filter(!(texto\_tokenizado %in% sw))

head(noticia2\_tidy)

1. **Lematización**

tabes=read.delim("D:/Maestría/Ciclo IV/Cursos/MCD-402\_Análisis EstadisticoTexto/Sesion2/lemmatization-es.txt",encoding="UTF-8",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

}

names(noticia2\_tidy)[1] <- "term"

noticia2\_tidy=noticia2\_tidy %>%left\_join(tabes, by = "term")

noticia2\_tidy$lemma=ifelse(is.na(noticia2\_tidy$stem ), noticia2\_tidy$term, noticia2\_tidy$stem)

1. **Tagging**

ud\_model <- udpipe\_download\_model(language = "spanish")

data(brussels\_reviews)

comments <- subset(brussels\_reviews, language %in% "es")

ud\_model <- udpipe\_load\_model(ud\_model$file\_model)

x <- udpipe\_annotate(ud\_model, x = comments$feedback, doc\_id = comments$id)

x <- as.data.frame(x)

head(cbind(x$token,x$upos,x$lemma))

library(lattice)

stats <- txt\_freq(x$upos)

stats$key <- factor(stats$key, levels = rev(stats$key))

barchart(key ~ freq, data = stats, col = "cadetblue",

main = "UPOS (Universal Parts of Speech)\n frequency of occurrence",

xlab = "Freq")

### TAGGING ejemplo Noticias ###

x\_noticia <- udpipe\_annotate(ud\_model, x = noticia2$noticia2, doc\_id = noticia2$ID)

x\_noticia <- as.data.frame(x\_noticia)

stats <- txt\_freq(x\_noticia$upos)

stats$key <- factor(stats$key, levels = rev(stats$key))

barchart(key ~ freq, data = stats, col = "cadetblue",

main = "UPOS (Universal Parts of Speech)\n Frecuencia de Ocurrencia",

xlab = "Freq")

Gráfico, Gráfico de barras

Descripción generada automáticamente

1. **Bag of Words**

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

noticia\_bow=as.data.frame(noticia\_bow)

noticia\_bow$ID <- seq.int(nrow(noticia\_bow))

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

.f = limpiar\_tokenizar))

noticia\_bow\_tidy <- noticia\_bow %>% select(-noticia\_bow) %>% unnest()

head(noticia\_bow\_tidy)

####frecuencia#######

noticia\_bow\_tidy %>% group\_by(texto\_tokenizado) %>% count(texto\_tokenizado)%>%

arrange(desc(n))

####frecuencia sin stop words##########

noticia\_bow\_tidy %>% group\_by(texto\_tokenizado) %>% filter(!(texto\_tokenizado %in% sw))%>%

count(texto\_tokenizado)%>% arrange(desc(n))

***Gráfico 1:***

noticia\_bow\_tidy %>% group\_by(texto\_tokenizado) %>% count(texto\_tokenizado)%>%

filter(n >= 10) %>%arrange(desc(n))%>%

ggplot(aes(x = reorder(texto\_tokenizado,n), y = n)) +

ggtitle("frecuencia de palabras")+

geom\_col(fill="blue") +

theme\_bw() +

labs(y = "", x = "")

Gráfico, Histograma

Descripción generada automáticamente

***Gráfico 2:***

noticia\_bow\_tidy %>% group\_by(texto\_tokenizado) %>% filter(!(texto\_tokenizado %in% sw))%>%

count(texto\_tokenizado)%>% filter(n >= 5) %>%arrange(desc(n))%>%

ggplot(aes(x = reorder(texto\_tokenizado,n), y = n)) +

ggtitle("frecuencia de palabras sin stop words")+

geom\_col(fill="red") +

theme\_bw() +

labs(y = "", x = "")

Gráfico, Gráfico de barras, Histograma

Descripción generada automáticamente