Procesamiento del Lenguaje Natural

Práctica: Grupo 2

Alejandro Escorial Aparicio (alejandro.escorial@alumnos.upm.es)
Alex Sanchez Perez (alex.sanchezperez@alumnos.upm.es)
Yi Chen Chen (yi.chenchen@alumnos.upm.es)
Pablo Gil Gil (p.ggil@alumnos.upm.es) Channa Pan (channa.pan@alumnos.upm.es)

Documento creado el 14/01/2026

Tabla de contenidos

Realización de la práctica

Descarga el fichero CONLL-U que se encuentra aquí. Este fichero tiene 58.198 líneas y ocupa 3.44 MB en disco. Uno de las dificultades es que este fichero no es puro CoNLL-U, por lo que no puede ser leído por udpipe_read_conllu(). Entonces para solucionarlo, se lee como un fichero tsv (tab separated values) con la función read.table(file = 'el_fichero', sep ='', header = FALSE).

```
fichero <- "es_ancora-up-test.conllu"
datos <- read.table(file = fichero,
                        sep = '\t',
                        header = FALSE,
                        quote = "",
                        comment.char = "#",
                        blank.lines.skip = TRUE,
                        flush = TRUE,
                        fill=TRUE)
```

Pregunta 1

¿Cuántas frases se analizan en este fichero?

Código

```
#Identifico aquellas filas con valor 1 en V1
frases <- datos[datos$V1==1,]
#Cuento cuantas filas coinciden
num_frases <- nrow(frases)

cat("Numero de frases analizadas en fichero:", num_frases)
```

```
## Numero de frases analizadas en fichero: 1721
```

```
#Me salen 1721 frases
```

Pruebas

```
#PRUEBA 1: ver que cuenta bien el numero de frases
#Dataframe sencillo
df_prueba <- data.frame(
  V1=c("1","2","1","2","3","1"),
  V2=c("La", "tienda", "La", "ciudad", "vieja", "Él")
)

#Pruebo mi codigo
frases_prueba <- df_prueba[df_prueba$V1==1,]
num_frases_prueba <- nrow(frases_prueba)

if (num_frases_prueba == 3) {
```

```
  cat("TEST PASADO: Numero de frases correcto.\n")
} else {
  cat("TEST FALLADO: Numero de frases incorrecto.\n")
}
```

```
## TEST PASADO: Numero de frases correcto.
```

```
#PRUEBA 2: Excepciones
#Compruebo que en los casos en que hay "1-2" y luego "1" y "2"
#(ocurre cuando la palabra es una contracción) cuenta el numero
#de frases de manera correcta
df_prueba2 <- data.frame(
  V1=c("1","2","1-2","1","2","3"),
  V2=c("La", "tienda", "Al", "A", "el", "niño")
)
#Pruebo mi codigo
frases_prueba2 <- df_prueba2[df_prueba2$V1==1,]
num_frases_prueba2 <- nrow(frases_prueba2)

if (num_frases_prueba2 == 2) {
  cat("TEST PASADO: Numero de frases correcto.\n")
} else {
  cat("TEST FALLADO: Numero de frases incorrecto.\n")
}
```

```
## TEST PASADO: Numero de frases correcto.
```

Explicación

En la primera columna sale el numero de cada palabra dentro de cada frase. Por tanto, al inicio de cada frase, la primera palabra tiene valor 1 en la columna V1, entonces contar las filas con este valor permite obtener el número total de frases del fichero.

Pregunta 2

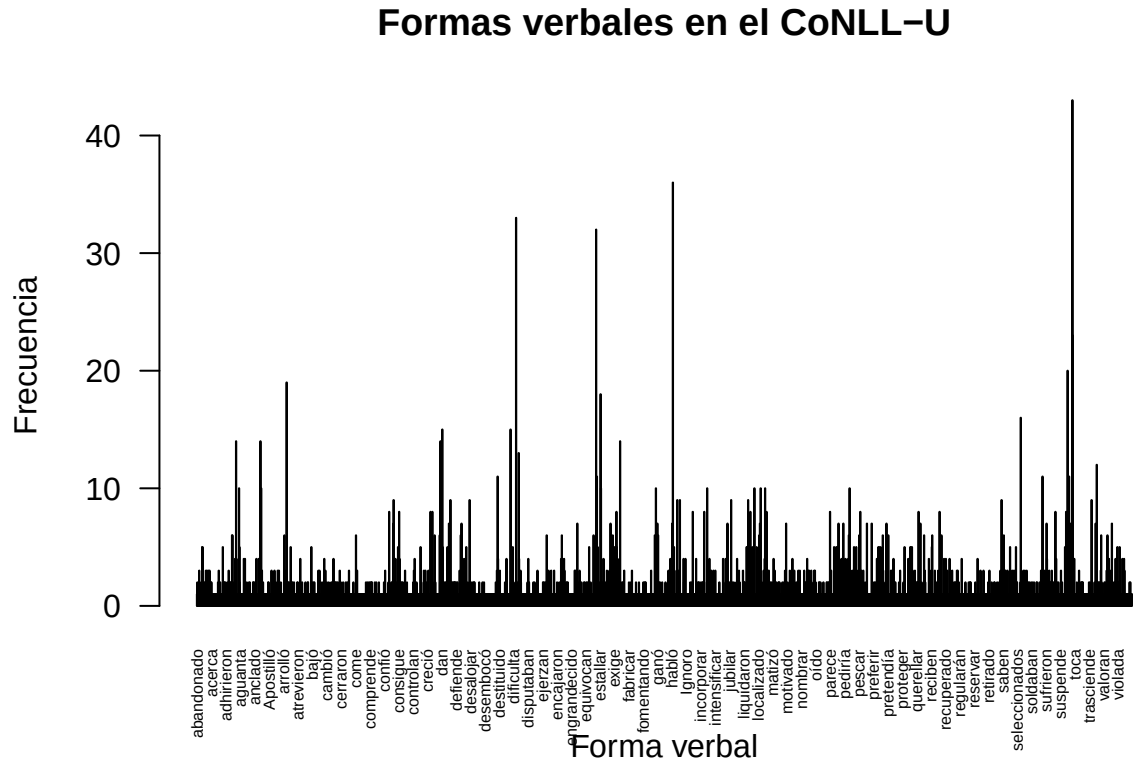
¿Cuántos verbos distintos se utilizan? Haz un histograma con las formas verbales encontradas.

Código

```
#Filtro las filas con verbos
filas_verbos <- datos[datos$V4 == "VERB", ]
#Guardo todos los verbos
verbos <- filas_verbos$V3
#Cuento los verbos distintos
num_verbos_distintos <- length(unique(verbos))

cat("Numero de verbos distintos analizados en fichero:", num_verbos_distintos)
```

```
## Numero de verbos distintos analizados en fichero: 987
```

```r
#Me salen 987 verbos distintos

#Guardo todas las formas verbales
formas <- filas_verbos$V2
#Saco las frecuencias de las formas
freq_formas <- table(formas)

barplot(freq_formas,
        las = 2,
        cex.names = 0.5,
        main = "Formas verbales en el CoNLL-U",
        xlab = "Forma verbal",
        ylab = "Frecuencia")
```



Pruebas

```r
#PRUEBA 1: Ver que cuenta bien el numero de verbos distintos.
#Dataframe sencillo
df_prueba <- data.frame(
  V3=c("entrar","de", "acusar","destituir","entrar", "1989"),
  V4=c("VERB", "DET", "VERB", "VERB", "VERB", "NOUN")
)
```

```r
#Pruebo mi código
filas_verbos_prueba <- df_prueba[df_prueba$V4 == "VERB", ]
verbos_prueba <- filas_verbos_prueba$V3
num_verbos_distintos_prueba <- length(unique(verbos_prueba))

if (num_verbos_distintos_prueba == 3) {
  cat("TEST PASADO: Numero de verbos distintos correcto.\n")
} else {
  cat("TEST FALLADO: Numero de verbos distintos incorrecto.\n")
}
```

```
## TEST PASADO: Numero de verbos distintos correcto.
```

```r
#PRUEBA 2: Ver que guarda bien la frecuencia de las formas verbales.
#Dataframe sencillo
df_prueba2 <- data.frame(
  V2=c("entró","la", "acusaba","destituir","entró", "1989"),
  V4=c("VERB", "DET", "VERB", "VERB", "VERB", "NOUN")
)

#Pruebo mi código
filas_verbos_prueba2 <- df_prueba2[df_prueba2$V4 == "VERB", ]
formas_prueba2 <- filas_verbos_prueba2$V2
freq_formas_prueba2 <- table(formas_prueba2)

if (freq_formas_prueba2["acusaba"]==1 && freq_formas_prueba2["destituir"]==1 && freq_formas_prueba2["en
  cat("TEST PASADO: Frecuencias de formas verbales correctas.\n")
} else {
  cat("TEST FALLADO: Frecuencias de formas verbales incorrectas.\n")
}
```

```
## TEST PASADO: Frecuencias de formas verbales correctas.
```

Explicación

En la cuarta columna sale el tipo de cada palabra. Por tanto, las palabras que en V4 tengan "VERB", son verbos. Una vez filtrados los verbos, se utiliza la tercera fila, que indica el infinitivo de los verbos, para ver si son distintos o nó, y se calcula cuantos verbos distintos aparecen. Para el histograma, se emplea la segunda columna, donde aparecen las formas verbales originales. Y a partir de esto, se saca su frecuencia para crear el histograma.

Pregunta 3

Haz un programa connlu_nlp.R que reciba por argumentos el nombre del fichero collu de entrada y el nombre de un fichero donde se escribirán las respuestas a las preguntas anteriores.

Código

```r
# Leer argumentos introducidos
args <- commandArgs(trailingOnly = TRUE)

#El programa se para si faltan argumentos
if (length(args) != 2) {
  stop("INTRODUCE EN EL TERMINAL: Rscript conllu_nlp.R nombre_fichero_entrada.conllu nombre_fichero_sali
}

#Guardar los argumentos
fichero_entrada <- args[1]
fichero_salida <- args[2]

#El programa se para cuando las extensiones de los argumentos introducidos
#no son .conllu y .pdf, respectivamente
if (!grepl("\\.conllu$", fichero_entrada)) {
  stop("EL FICHERO DE ENTRADA DEBE TENER EXTENSIÓN .conllu")
}

if (!grepl("\\.pdf$", fichero_salida)) {
  stop("EL FICHERO DE SALIDA DEBE TENER EXTENSIÓN .pdf")
}

# Leer el fichero CoNLL-U
datos <- read.table(file = fichero_entrada,
                    sep = '\t',
                    header = FALSE,
                    quote = "",
                    comment.char = "#",
                    blank.lines.skip = TRUE,
                    flush = TRUE,
                    fill=TRUE)

# Resolver Pregunta 1
frases <- datos[datos$V1==1,]
num_frases <- nrow(frases)

# Resolver Pregunta 2
filas_verbos <- datos[datos$V4 == "VERB", ]
verbos <- filas_verbos$V3
num_verbos_distintos <- length(unique(verbos))

formas <- filas_verbos$V2
freq_formas <- table(formas)

# Generar el PDF con los resultados
pdf(fichero_salida)

plot.new()
title("Análisis del fichero CoNLL-U")

text(0, 0.8, paste("Fichero analizado:", fichero_entrada), adj = 0)
text(0, 0.7, paste("Número de frases:", num_frases), adj = 0)
text(0, 0.6, paste("Número de verbos distintos:", num_verbos_distintos), adj = 0)
```

```r
#Gráfico de barras
barplot(freq_formas,
        las = 2,
        cex.names = 0.5,
        main = "Formas verbales en el CoNLL-U",
        xlab = "Forma verbal",
        ylab = "Frecuencia")

dev.off()

cat("Análisis completado.\nResultados guradados en:", fichero_salida, "\n")
```

Pruebas

```r
#PRUEBA para comprobar conllu_nlp.R

#Inicializar la prueba
script <- "conllu_nlp.R"
entrada <- "es_ancora-up-test.conllu"
salida <- "test_salida.pdf"

#PRUEBA 1: Ejecución con argumentos correctos
cmd <- paste("Rscript", script, entrada, salida)
status <- system(cmd)

if (status != 0) {
  cat("TEST FALLADO: El programa no se ejecutó correctamente.\n")
}

if (!file.exists(salida)) {
  cat("TEST FALLADO: El PDF no se ha generado.\n")
}

if (file.info(salida)$size == 0) {
  cat("TEST FALLADO: El PDF está vacío.\n")
}

cat("TEST PASADO: El programa se ha ejecutado correctamente, con resultados deseados.\n")
```

```
## TEST PASADO: El programa se ha ejecutado correctamente, con resultados deseados.
```

```r
#PRUEBA 2: Falta de argumentos
cmd <- paste("Rscript", script, entrada)
status <- system(cmd)

if (status == 0) {
  cat("TEST FALLADO: El programa no se paró cuando faltan argumentos.\n")
} else{
  cat("TEST PASADO: El programa sí se paró cuando faltan argumentos.\n")
}
```

```
## TEST PASADO: El programa sí se paró cuando faltan argumentos.
```

```
#PRUEBA 3: Fichero de entrada con extensión incorrecta
cmd <- paste("Rscript", script, "entrada.txt", salida)
status <- system(cmd)

if (status == 0) {
  cat("TEST FALLADO: El programa no se paró cuando el fichero de entrada tiene extensión incorrecta.\n")
} else{
  cat("TEST PASADO: El programa sí se paró cuando el fichero de entrada tiene extensión incorrecta.\n")
}
```

```
## TEST PASADO: El programa sí se paró cuando el fichero de entrada tiene extensión incorrecta.
```

```
#PRUEBA 4: Fichero de salida con extensión incorrexta
cmd <- paste("Rscript", script, entrada, "salida.txt")
status <- system(cmd)

if (status == 0) {
  cat("TEST FALLADO: El programa no se paró cuando el fichero de salida tiene extensión incorrecta.\n")
} else{
  cat("TEST PASADO: El programa sí se paró cuando el fichero de salida tiene extensión incorrecta.\n")
}
```

```
## TEST PASADO: El programa sí se paró cuando el fichero de salida tiene extensión incorrecta.
```

Explicación

Hemos juntado la lógica de las dos primeras preguntas en un solo script: el filtro de V1 == 1 para contar las frases y el de V4 == "VERB" para los verbos. Los resultados los sacamos directamente a un PDF usando las funciones pdf() y text(), así queda todo en un informe más visual con su gráfico de barras incluido.

El mayor problema al que nos enfeantamos fue configurar bien el read.table para que leyera el fichero con los mismos ajustes que usamos antes (los separadores, las comillas, etc.). Era fundamental que fuera idéntico porque, si no, los datos no coincidían y los resultados finales salían mal.

Pregunta 4

Ejecuta el programa connlu_nlp.R con el fichero CONLL-U que se encuentra aquí y adjunta el fichero resultado.

Código

```
fichero_entrada <- "es_ancora-up-dev.conllu"
fichero_salida <- "resultado.pdf"
system(paste("Rscript conllu_nlp.R", fichero_entrada, fichero_salida))
```

```
## [1] 0
```

Distribución de tareas y porcentajes de autoría

- Alejandro Escorial Aparicio: pregunta 2, 100%
- Alex Sanchez Perez: fichero .Rmd, 50%
- Yi Chen Chen: fichero .Rmd, 50%
- Pablo Gil Gil: lectura correcta del fichero y pregunta 1, 100%
- Channa Pan: pregunta 3, 100%