Trabajo 2

Thelma

2023-12-27

# Trabajo Final

*Introducción*

El trabajo final del módulo 5 se basará en utilizar el corpus provisto (cuyas características se describen a continuación) y recorrer todo el flujo de trabajo visto en clase: preprocesamiento, generación de distribuciones de frecuencias de palabras y modelado de tópicos. Prestaremos especial atención en la corrección a la explicación y justificación de las decisiones tomadas, en tanto muestren manejo de los tópicos vistos en clase. Dataset

El archivo que se adjunta consiste en un corpus de unas 7.000 noticias scrapeadas entre julio y septiembre de 2019 de los siguientes medios de circulación nacional: - Télam - La Nación - Clarín - Perfil - Infobae - MinutoUno - Página 12

Constituye una muestra aleatoria del corpus construido por Florencia Piñeyrúa para su tesina de grado “Procesamiento del lenguaje natural aplicado al estudio de tópicos de noticias de seguridad en Argentina: julio a septiembre 2019”.

El corpus contiene, las siguientes variables: - id : identificador de cada documento - url : link a la noticia original - fecha : fecha de publicación - anio : año de publicación - mes : mes de publicación - dia : dia de publicación - medio : medio en el que fue publicado - orientacion: clasificación -provisoria- de los medios según su línea editorial predominante (más conservador, más progresista, neutral) - titulo - texto

*A partir del corpus deberán:* - Cargar los datos - Preprocesarlos (normalizar texto, eliminar stopwords) - Generar una matriz token-por-fila para cada documento

Abrir librerías

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(tidytext)

### 1. Cargamos los datos

noticias <- read\_csv('C:/Users/USUARIO/Documents/DCSCyHD/Trabajo 2/trabajo2\_DCSCyHD/M5\_corpus\_medios.csv')

## Rows: 7000 Columns: 9  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (4): url, medio, titulo, texto  
## dbl (4): id, anio, mes, dia  
## date (1): fecha  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

print (noticias)

## # A tibble: 7,000 × 9  
## id url fecha anio mes dia medio titulo texto  
## <dbl> <chr> <date> <dbl> <dbl> <dbl> <chr> <chr> <chr>  
## 1 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… "Copa… "Arg…  
## 2 52024 https://www.clarin.com… 2019-07-01 2019 7 1 clar… "Robo… "La …  
## 3 52023 https://www.clarin.com… 2019-07-01 2019 7 1 clar… "Inve… "La …  
## 4 52095 http://luz.perfil.com/… 2019-07-01 2019 7 1 perf… "Arti… "Art…  
## 5 51863 https://www.perfil.com… 2019-07-01 2019 7 1 perf… "Cier… "mar…  
## 6 52087 https://exitoina.perfi… 2019-07-01 2019 7 1 perf… "Tele… "Con…  
## 7 52084 https://www.clarin.com… 2019-07-01 2019 7 1 clar… "Vene… "Car…  
## 8 51945 https://www.minutouno.… 2019-07-01 2019 7 1 minu… "La a… "El …  
## 9 51928 https://www.clarin.com… 2019-07-01 2019 7 1 clar… "¿Mit… "Alg…  
## 10 52375 https://www.clarin.com… 2019-07-01 2019 7 1 clar… "Los … "Hay…  
## # ℹ 6,990 more rows

### 2. Preprocesamos los datos

#### 2.a. Construimos la tabla de tokens del texto

noticias\_tidy <- noticias %>%  
 unnest\_tokens(output=word, input=texto, token = "words") %>%   
print()

## # A tibble: 4,006,051 × 9  
## id url fecha anio mes dia medio titulo word   
## <dbl> <chr> <date> <dbl> <dbl> <dbl> <chr> <chr> <chr>  
## 1 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … arge…  
## 2 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … arra…  
## 3 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … a   
## 4 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … paso   
## 5 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … flojo  
## 6 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … en   
## 7 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … esta   
## 8 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … copa   
## 9 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … amér…  
## 10 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … 2019   
## # ℹ 4,006,041 more rows

#### 2.b Eliminamos stopwords

Cargamos el diccionario de stopwords

stop\_words <- read\_csv("https://raw.githubusercontent.com/Alir3z4/stop-words/master/spanish.txt", col\_names=FALSE) %>%  
 rename(word = X1) %>%  
 mutate(word = stringi::stri\_trans\_general(word, "Latin-ASCII")) %>%   
 print()

## Rows: 608 Columns: 1  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (1): X1  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

## # A tibble: 608 × 1  
## word   
## <chr>   
## 1 a   
## 2 actualmente  
## 3 adelante   
## 4 ademas   
## 5 afirmo   
## 6 agrego   
## 7 ahora   
## 8 ahi   
## 9 al   
## 10 algo   
## # ℹ 598 more rows

stop\_words <- stop\_words %>%  
 bind\_rows( tibble(word=c('embed', 'leé', 'cronica.com.ar', 'minutouno.com', 'comentar', 'jpg', 'loading', 'pristupluk', ' minutouno.com', 'guardar', ' páginai12', '01',   
 'l.l', 'loading', 'jpe', 'más', 'también', 'está', 'había', 'qué', 'así', 'están', 'además', 'según', 'frente', 'día', '1', 'gusta', 'twitter', 'fuente', 'whatsapp', 'compartir', 'facebook', 'mail', 'él')))

#### 2.c. Procedemos a la eliminación:

corpus\_noticias <- noticias\_tidy %>%  
 anti\_join (stop\_words, by = c("word" = "word")) %>%   
 print()

## # A tibble: 1,846,749 × 9  
## id url fecha anio mes dia medio titulo word   
## <dbl> <chr> <date> <dbl> <dbl> <dbl> <chr> <chr> <chr>  
## 1 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … arge…  
## 2 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … arra…  
## 3 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … paso   
## 4 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … flojo  
## 5 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … copa   
## 6 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … amér…  
## 7 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … 2019   
## 8 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … supe…  
## 9 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … obst…  
## 10 51960 https://www.clarin.com… 2019-07-01 2019 7 1 clar… Copa … riva…  
## # ℹ 1,846,739 more rows

### 3. Generar una matriz token-por-fila para cada documento

# Consigna 1)

1. ¿Cuáles son las palabras más utilizadas en cada uno de los medios?
2. ¿Pueden verse diferencias? (Tener en cuenta las diferentes métricas trabajadas en el curso: tf, tf-idf, etc.)
3. Generar las visualizaciones que considere más pertinentes para responder la pregunta

corpus\_tf\_idf <- corpus\_noticias %>%  
 count(medio, word, sort = TRUE) %>%  
 bind\_tf\_idf (word, medio, n) %>%   
 arrange(desc(tf)) %>%   
 print()

## # A tibble: 276,739 × 6  
## medio word n tf idf tf\_idf  
## <chr> <chr> <int> <dbl> <dbl> <dbl>  
## 1 cronica años 636 0.00599 0 0  
## 2 infobae años 1330 0.00482 0 0  
## 3 lanacion años 1195 0.00479 0 0  
## 4 clarin años 1877 0.00425 0 0  
## 5 pagina12 años 1244 0.00419 0 0  
## 6 perfil años 1592 0.00416 0 0  
## 7 lanacion 2019 996 0.00400 0 0  
## 8 telam presidente 178 0.00393 0 0  
## 9 telam gobierno 176 0.00389 0 0  
## 10 minutouno años 182 0.00377 0 0  
## # ℹ 276,729 more rows

Las palabras más utilizadas por cada medio

unique(corpus\_noticias$medio)

## [1] "clarin" "perfil" "minutouno" "infobae" "lanacion" "pagina12"   
## [7] "cronica" "telam"

#### Clarín

Las palabras más utilizadas de Clarín

top\_words\_by\_medio <- corpus\_tf\_idf %>%  
 filter(medio == "clarin") %>%   
 group\_by(medio) %>%  
 top\_n(20, wt = tf) %>%  
 arrange(desc(n)) %>%   
 print()

## # A tibble: 20 × 6  
## # Groups: medio [1]  
## medio word n tf idf tf\_idf  
## <chr> <chr> <int> <dbl> <dbl> <dbl>  
## 1 clarin años 1877 0.00425 0 0  
## 2 clarin argentina 1409 0.00319 0 0  
## 3 clarin gobierno 1315 0.00298 0 0  
## 4 clarin presidente 1119 0.00253 0 0  
## 5 clarin país 1051 0.00238 0 0  
## 6 clarin año 981 0.00222 0 0  
## 7 clarin después 881 0.00200 0 0  
## 8 clarin millones 876 0.00198 0 0  
## 9 clarin fernández 742 0.00168 0 0  
## 10 clarin nacional 735 0.00166 0 0  
## 11 clarin macri 713 0.00161 0 0  
## 12 clarin días 652 0.00148 0 0  
## 13 clarin semana 614 0.00139 0 0  
## 14 clarin ciudad 611 0.00138 0 0  
## 15 clarin mundo 598 0.00135 0 0  
## 16 clarin casa 590 0.00134 0 0  
## 17 clarin personas 588 0.00133 0 0  
## 18 clarin foto 576 0.00130 0 0  
## 19 clarin partido 574 0.00130 0 0  
## 20 clarin caso 572 0.00130 0 0

#### Perfil

Las palabras más utilizadas de Perfil

top\_words\_by\_medio <- corpus\_tf\_idf %>%  
 filter(medio == "perfil") %>%   
 group\_by(medio) %>%  
 top\_n(20, wt = tf) %>%  
 arrange(desc(n)) %>%   
 print()

## # A tibble: 20 × 6  
## # Groups: medio [1]  
## medio word n tf idf tf\_idf  
## <chr> <chr> <int> <dbl> <dbl> <dbl>  
## 1 perfil años 1592 0.00416 0 0  
## 2 perfil 2019 1359 0.00355 0 0  
## 3 perfil argentina 1254 0.00328 0 0  
## 4 perfil gobierno 1134 0.00296 0 0  
## 5 perfil país 999 0.00261 0 0  
## 6 perfil macri 918 0.00240 0 0  
## 7 perfil año 887 0.00232 0 0  
## 8 perfil fernández 874 0.00228 0 0  
## 9 perfil presidente 871 0.00227 0 0  
## 10 perfil después 792 0.00207 0 0  
## 11 perfil millones 704 0.00184 0 0  
## 12 perfil agosto 660 0.00172 0 0  
## 13 perfil nacional 618 0.00161 0 0  
## 14 perfil alberto 617 0.00161 0 0  
## 15 perfil mundo 614 0.00160 0 0  
## 16 perfil julio 612 0.00160 0 0  
## 17 perfil política 609 0.00159 0 0  
## 18 perfil días 586 0.00153 0 0  
## 19 perfil vida 586 0.00153 0 0  
## 20 perfil personas 548 0.00143 0 0

#### Minuto Uno

Las palabras más utilizadas de Minutouno

top\_words\_by\_medio <- corpus\_tf\_idf %>%  
 filter(medio == "minutouno") %>%   
 group\_by(medio) %>%  
 top\_n(20, wt = tf) %>%  
 arrange(desc(n)) %>%   
 print()

## # A tibble: 20 × 6  
## # Groups: medio [1]  
## medio word n tf idf tf\_idf  
## <chr> <chr> <int> <dbl> <dbl> <dbl>  
## 1 minutouno años 182 0.00377 0 0  
## 2 minutouno gobierno 153 0.00317 0 0  
## 3 minutouno macri 146 0.00302 0 0  
## 4 minutouno argentina 137 0.00284 0 0  
## 5 minutouno presidente 104 0.00215 0 0  
## 6 minutouno año 103 0.00213 0 0  
## 7 minutouno 2019 98 0.00203 0 0  
## 8 minutouno acuerdo 90 0.00186 0 0  
## 9 minutouno después 88 0.00182 0 0  
## 10 minutouno nacional 88 0.00182 0 0  
## 11 minutouno país 86 0.00178 0 0  
## 12 minutouno millones 81 0.00168 0 0  
## 13 minutouno aseguró 78 0.00161 0 0  
## 14 minutouno personas 77 0.00159 0 0  
## 15 minutouno aires 74 0.00153 0 0  
## 16 minutouno partido 73 0.00151 0 0  
## 17 minutouno ciudad 72 0.00149 0 0  
## 18 minutouno días 69 0.00143 0 0  
## 19 minutouno meses 68 0.00141 0 0  
## 20 minutouno paso 68 0.00141 0 0

#### Infobae

Las palabras más utilizadas de Infobae

top\_words\_by\_medio <- corpus\_tf\_idf %>%  
 filter(medio == "infobae") %>%   
 group\_by(medio) %>%  
 top\_n(20, wt = tf) %>%  
 arrange(desc(n)) %>%   
 print()

## # A tibble: 20 × 6  
## # Groups: medio [1]  
## medio word n tf idf tf\_idf  
## <chr> <chr> <int> <dbl> <dbl> <dbl>  
## 1 infobae años 1330 0.00482 0 0  
## 2 infobae después 728 0.00264 0 0  
## 3 infobae país 704 0.00255 0 0  
## 4 infobae gobierno 699 0.00253 0 0  
## 5 infobae presidente 645 0.00234 0 0  
## 6 infobae personas 622 0.00225 0 0  
## 7 infobae año 527 0.00191 0 0  
## 8 infobae mundo 527 0.00191 0 0  
## 9 infobae vida 515 0.00187 0 0  
## 10 infobae millones 470 0.00170 0 0  
## 11 infobae argentina 445 0.00161 0 0  
## 12 infobae unidos 445 0.00161 0 0  
## 13 infobae nacional 419 0.00152 0 0  
## 14 infobae cómo 392 0.00142 0 0  
## 15 infobae días 388 0.00141 0 0  
## 16 infobae méxico 368 0.00133 0 0  
## 17 infobae caso 360 0.00130 0 0  
## 18 infobae acuerdo 357 0.00129 0 0  
## 19 infobae tenía 340 0.00123 0 0  
## 20 infobae venezuela 340 0.00123 0 0

#### La Nación

Las palabras más utilizadas de La Nación

top\_words\_by\_medio <- corpus\_tf\_idf %>%  
 filter(medio == "lanacion") %>%   
 group\_by(medio) %>%  
 top\_n(20, wt = tf) %>%  
 arrange(desc(n)) %>%   
 print()

## # A tibble: 20 × 6  
## # Groups: medio [1]  
## medio word n tf idf tf\_idf  
## <chr> <chr> <int> <dbl> <dbl> <dbl>  
## 1 lanacion años 1195 0.00479 0 0   
## 2 lanacion 2019 996 0.00400 0 0   
## 3 lanacion crédito 806 0.00323 0 0   
## 4 lanacion argentina 725 0.00291 0 0   
## 5 lanacion después 633 0.00254 0 0   
## 6 lanacion nacion 577 0.00231 1.39 0.00321  
## 7 lanacion fernández 540 0.00217 0 0   
## 8 lanacion gobierno 526 0.00211 0 0   
## 9 lanacion país 510 0.00205 0 0   
## 10 lanacion año 493 0.00198 0 0   
## 11 lanacion presidente 486 0.00195 0 0   
## 12 lanacion vida 476 0.00191 0 0   
## 13 lanacion video 475 0.00191 0 0   
## 14 lanacion macri 470 0.00189 0 0   
## 15 lanacion cómo 449 0.00180 0 0   
## 16 lanacion mundo 433 0.00174 0 0   
## 17 lanacion julio 415 0.00167 0 0   
## 18 lanacion agosto 411 0.00165 0 0   
## 19 lanacion septiembre 386 0.00155 0 0   
## 20 lanacion días 383 0.00154 0 0

#### Página 12

Las palabras más utilizadas de Página 12

top\_words\_by\_medio <- corpus\_tf\_idf %>%  
 filter(medio == "pagina12") %>%   
 group\_by(medio) %>%  
 top\_n(20, wt = tf) %>%  
 arrange(desc(n)) %>%   
 print()

## # A tibble: 20 × 6  
## # Groups: medio [1]  
## medio word n tf idf tf\_idf  
## <chr> <chr> <int> <dbl> <dbl> <dbl>  
## 1 pagina12 años 1244 0.00419 0 0  
## 2 pagina12 gobierno 1030 0.00347 0 0  
## 3 pagina12 ciento 994 0.00335 0 0  
## 4 pagina12 argentina 727 0.00245 0 0  
## 5 pagina12 país 650 0.00219 0 0  
## 6 pagina12 macri 627 0.00211 0 0  
## 7 pagina12 presidente 609 0.00205 0 0  
## 8 pagina12 año 608 0.00205 0 0  
## 9 pagina12 nacional 555 0.00187 0 0  
## 10 pagina12 después 531 0.00179 0 0  
## 11 pagina12 millones 527 0.00177 0 0  
## 12 pagina12 política 499 0.00168 0 0  
## 13 pagina12 cómo 438 0.00147 0 0  
## 14 pagina12 personas 436 0.00147 0 0  
## 15 pagina12 pesos 436 0.00147 0 0  
## 16 pagina12 fernández 406 0.00137 0 0  
## 17 pagina12 días 393 0.00132 0 0  
## 18 pagina12 vida 387 0.00130 0 0  
## 19 pagina12 mundo 384 0.00129 0 0  
## 20 pagina12 social 381 0.00128 0 0

#### Crónica

Las palabras más utilizadas de Crónica

top\_words\_by\_medio <- corpus\_tf\_idf %>%  
 filter(medio == "cronica") %>%   
 group\_by(medio) %>%  
 top\_n(20, wt = tf) %>%  
 arrange(desc(n)) %>%   
 print()

## # A tibble: 20 × 6  
## # Groups: medio [1]  
## medio word n tf idf tf\_idf  
## <chr> <chr> <int> <dbl> <dbl> <dbl>  
## 1 cronica años 636 0.00599 0 0  
## 2 cronica policía 286 0.00269 0 0  
## 3 cronica ciudad 282 0.00265 0 0  
## 4 cronica mujer 281 0.00264 0 0  
## 5 cronica hombre 254 0.00239 0 0  
## 6 cronica casa 250 0.00235 0 0  
## 7 cronica gobierno 250 0.00235 0 0  
## 8 cronica presidente 232 0.00218 0 0  
## 9 cronica país 224 0.00211 0 0  
## 10 cronica personas 216 0.00203 0 0  
## 11 cronica víctima 212 0.00200 0 0  
## 12 cronica argentina 202 0.00190 0 0  
## 13 cronica san 191 0.00180 0 0  
## 14 cronica joven 190 0.00179 0 0  
## 15 cronica días 183 0.00172 0 0  
## 16 cronica nacional 181 0.00170 0 0  
## 17 cronica después 179 0.00168 0 0  
## 18 cronica acuerdo 168 0.00158 0 0  
## 19 cronica seguridad 166 0.00156 0 0  
## 20 cronica hospital 164 0.00154 0 0

#### Telam

Las palabras más utilizadas de Telam

top\_words\_by\_medio <- corpus\_tf\_idf %>%  
 filter(medio == "telam") %>%   
 group\_by(medio) %>%  
 top\_n(20, wt = tf) %>%  
 arrange(desc(n)) %>%   
 print()

## # A tibble: 20 × 6  
## # Groups: medio [1]  
## medio word n tf idf tf\_idf  
## <chr> <chr> <int> <dbl> <dbl> <dbl>  
## 1 telam presidente 178 0.00393 0 0  
## 2 telam gobierno 176 0.00389 0 0  
## 3 telam argentina 169 0.00373 0 0  
## 4 telam país 145 0.00320 0 0  
## 5 telam nacional 144 0.00318 0 0  
## 6 telam años 139 0.00307 0 0  
## 7 telam diputados 117 0.00259 0 0  
## 8 telam cambio 99 0.00219 0 0  
## 9 telam fernández 99 0.00219 0 0  
## 10 telam 2019 97 0.00214 0 0  
## 11 telam acuerdo 95 0.00210 0 0  
## 12 telam macri 86 0.00190 0 0  
## 13 telam partido 83 0.00183 0 0  
## 14 telam ciudad 75 0.00166 0 0  
## 15 telam año 74 0.00164 0 0  
## 16 telam millones 74 0.00164 0 0  
## 17 telam federal 73 0.00161 0 0  
## 18 telam candidato 72 0.00159 0 0  
## 19 telam aires 70 0.00155 0 0  
## 20 telam ministro 70 0.00155 0 0

### Gráfico

# Consigna 2)

¿Cuáles son los tópicos principales en el corpus? ¿Pueden evidenciar diferencias en cada uno de los medios? Explicar qué método se utilizó para responder la pregunta, cuáles son los supuestos del mismo. Generar las visualizaciones más adecuadas para responder a las preguntas

# Consigna 3)

A continuación, seleccionar las noticias vinculadas a algún tópico relevante (por ejemplo, “Elecciones”) y construir un clasificador para predecir la orientación del diario1. Utilizar alguno de los modelos de clasificación vistos a lo largo de al Diplomatura (regresión logística, random forest, etc.). Utilizar como features el “Spanish Billion Word Corpus and Embeddings”, analizado en clase (pueden descargar el embedding en formato .bin del link).

¿Qué resultados arroja el modelo? ¿Es posible mediante el texto de las noticias conocer la línea editorial del diario? Generar las visualizaciones y tablas correspondientes para una correcta evaluación del modelo.