Introducción y enunciado de la práctica análisis de chistes

16/05/2022

Contenidos

1 Análisis de un conjunto de chistes con metadatos

Algunas ayudas y ejemplos en "Data_model_chistes2.Rmd", se ha cambiado a la librería "word2vec" más reciente pero menos comentada.

El fichero "data/chistes_con_metadatos_curado.csv" contiene unos 7170 chistes de la web 1000chistes.com y de pintamania.

1.1 Carga de datos

Los datos están en un fichero separado por ";" contiene 5 variables

- origen: la web de origen del chiste; 1000chistes o pintamanía `factor`.
- titulo: EL título del chiste `character`.
- categoria: cortos|malos|Jaimito; son una variable `character` de categorías separadas por "|"
- palabra_clave: políticos|argentinos; son una una variable `character` de palabras clave separadas por "|".
- votos: Número de votos `integer`; solo para pintamania.

- texto: tipo character; es el texto del chiste en UTF-8 separado por " " `character`.

```
data_raw=read_csv("data/chistes_con_metadatos_curado.csv",col_names=TRUE)
```

```
## Rows: 7169 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): origen, titulo, categorias, palabra_clave, texto
## dbl (1): votos
##
## 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.
```

```
glimpse(data_raw)
```

```
## Rows: 7,169
## Columns: 6
## $ origen        <chr> "1000 chistes", "1000 chistes", "1000 chistes", "1000 ch~
## $ titulo        <chr> "Dime con quién andas...", "Luz automática", "Política a~
## $ categorias    <chr> "cortos|malos", "cortos|malos|borrachos|matrimonios", "c~
## $ palabra_clave <chr> "feos", "neveras", "políticos|argentinos", "sangre", "fu~
## $ votos         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ texto         <chr> "- Dime con quién andas y te diré quién eres.  - No ando~
```

```
knitr::kable(head(data_raw,20))
```

origen	titulo	categorias	palabra_clave	votos	texto
1000 chistes	Dime con quién andas...	cortos|malos	feos	NA	- Dime con quién andas y te diré quién eres. - No ando con nadie... - Eres feo.
1000 chistes	Luz automática	cortos|malos|borrachos|matrimonios	neveras	NA	Va un marido completamente borracho y le dice a su mujer al irse para cama: - Me ha pasado algo increíble. He ido al baño y al abrir la puerta se ha encendido la luz automáticamente, sin hacer nada. - ¡La madre que te parió!, ¡Te mato!, ya te has vuelto a mear en la nevera.
1000 chistes	Política argentina	cortos|malos	políticos|argentinos	NA	Un diputado argentino se encuentra en la calle con un amigo de la infancia y éste le pregunta: - ¿Cómo estás llevando esta crisis? - ¡La verdad que duermo como un bebé! - ¡Dormís como un bebé! ¿Pero cómo hacés? - ¡Me despierto cada 3 horas llorando!
1000 chistes	0 positivo	cortos|malos	sangre	NA	- ¡Rápido, necesitamos sangre! - Yo soy 0 positivo. - Pues muy mal, necesitamos una mentalidad optimista.
1000 chistes	Mejor portero	cortos|malos	futbol|porteros	NA	- ¿Cuál es el mejor portero del mundial? - Evidente ¡el de Para-guay!
1000 chistes	Donación para la piscina	cortos|malos	dinero|agua	NA	El otro día unas chicas llamarón a mi puerta y me pidieron una pequeña donación para una piscina local. Les di un garrafa de agua.

origen	titulo	categorias	palabra_clave	votos	texto
1000 chistes	Clase de astrología	cortos|malos	profesores|planetas	NA	- Andresito, ¿qué planeta va después de Marte? - Miércole, señorita.
1000 chistes	Bob Esponja	cortos|malos	esponja|gimnasio	NA	- ¿Por qué Bob Esponja no va al gimnasio? - Porque ya está cuadrado.
1000 chistes	Ojalá lloviera	cortos|malos	ciegos	NA	Van dos ciegos y le dice uno al otro: - Ojalá lloviera... - Ojalá yo también...
1000 chistes	En Canarias	cortos|suegras	canarias|coches	NA	Noticias de última hora!! Muere una suegra atropellada en Canarias. Y esto es todo, las 8 en España y UNA menos en Canarias...
1000 chistes	Dicen que estoy loco	cortos|malos	locos|sillas	NA	– Mamá, mamá, en el colegio dicen que estoy loco. – ¿Y quién dice eso de ti? – ... Me lo dicen las sillas...
1000 chistes	Bocadillo de jamón	cortos|malos	madres|jamón	NA	– Mamá, mamá, ¿me haces un bocata de jamón? – ¿York? – Sí, túrk.
1000 chistes	Te echan de varias universidades	malos|cortos	universitarios	NA	- Qué pasa si te expulsan de cuatro univerdades? - .... - Que estás perdiendo facultades
1000 chistes	Un pelo en la cama	cortos|malos	cuentos|pelos	NA	- Qué es un pelo en una cama? - ... - El bello durmiente
1000 chistes	Entre techos	cortos|malos	casas	NA	- Qué le dice el techo del comedor al techo de la cocina? - .... - Te hecho de menos!
1000 chistes	Se va la luz	cortos|malos	pijos|escuelas	NA	- Qué pasa si se va la luz en una escuela privada? - .... - No se ve ni un pijo!
1000 chistes	País sin tacos	cortos|malos	país	NA	- En qué se convierte un país en el que se prohíben los tacos? - .... - En un país destacado!
1000 chistes	Messi de aquí a 45 días	cortos|malos	deportistas|futbol	NA	- Qué es Messi en 45 días? - ..... - Mes y medio!
1000 chistes	Mundo son forma cubica	cortos|malos	cubanos|planetas	NA	- Qué pasaría si el mundo en lugar de ser una esfera fuera un cubo? - .... - Pues que todos seríamos cubanos
1000 chistes	Saludable	cortos|malos	amigos|deporte	NA	- Soy una persona muy saludable. - ¿Haces mucho deporte y comes sano? - No. Es que la gente me saluda por la calle y yo... pues les devuelvo el saludo.

1.2 Extracción del diccionario raw empírico desde los chistes

Extraemos al dic_raw_1 todas las palabras que aparecen con separación espacio.

Criterios iniciales:

- Decidimos enconding a UTF-8 columna text_utf8 si hay que depurar por enconding habrá que ver cómo.
- Hay que decidir qué se hace con los CARACTERES SPECIALES:{,:; () ¿?!!}. De momento los voy a eliminar
- Todas las MAYÚSCULAS a MINÚSCULAS
- De momento NO SE ELIMINAN DIGITOS: se quedan tal cual, hay que distinguir los de los dígitos de años.
- No catalogamos idiomas.... se supone que todo está en castellano o términos técnicos que añadiremos
- Castellano es toda palabra o derivado de palabra que se encuentre en un spelling estándar de castellano que podemos ir adaptando.

```
library(tidytext)
library(stringr)
texto_df = data_raw
glimpse(texto_df)
```

```
## Rows: 7,169
## Columns: 6
## $ origen        <chr> "1000 chistes", "1000 chistes", "1000 chistes", "1000 ch~
## $ titulo        <chr> "Dime con quién andas...", "Luz automática", "Política a~
## $ categorias    <chr> "cortos|malos", "cortos|malos|borrachos|matrimonios", "c~
## $ palabra_clave <chr> "feos", "neveras", "políticos|argentinos", "sangre", "fu~
## $ votos         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ texto         <chr> "- Dime con quién andas y te diré quién eres.  - No ando~
```

```
#arreglo categorias a columnas distintas se podrían pasar a arrays.
texto_df = texto_df %>% separate(
  col = c("categorias"),
  sep = "\\|",
  into = paste0("C", 1:5),
  fill = "right"
)
```

```
## Warning: Expected 5 pieces. Additional pieces discarded in 4 rows [1015, 1039,
## 1529, 1669].
```

```
texto_df = texto_df %>% separate(
  col = c("palabra_clave"),
  sep = "\\|",
  into = paste0("palabra", 1:5),
  fill = "right"
)
```

```
## Warning: Expected 5 pieces. Additional pieces discarded in 10 rows [167, 1587,
## 1589, 1657, 1988, 2072, 2190, 2233, 2363, 2376].
```

```
texto_df = texto_df %>% mutate(texto_curado = str_squish(str_replace_all(texto, "\\:|-|#|_", " ")))
glimpse(texto_df)
```

```
## Rows: 7,169
## Columns: 15
```

```
## $ origen       <chr> "1000 chistes", "1000 chistes", "1000 chistes", "1000 chi~
## $ titulo       <chr> "Dime con quién andas...", "Luz automática", "Política ar~
## $ C1           <chr> "cortos", "cortos", "cortos", "cortos", "cortos", "cortos~
## $ C2           <chr> "malos", "malos", "malos", "malos", "malos", "malos", "ma~
## $ C3           <chr> NA, "borrachos", NA, NA, NA, NA, "profesores", NA, NA, NA~
## $ C4           <chr> NA, "matrimonios", NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ C5           <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ palabra1     <chr> "feos", "neveras", "políticos", "sangre", "futbol", "dine~
## $ palabra2     <chr> NA, NA, "argentinos", NA, "porteros", "agua", NA, "gimnas~
## $ palabra3     <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, "noticias", NA, NA, N~
## $ palabra4     <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ palabra5     <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ votos        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ texto        <chr> "- Dime con quién andas y te diré quién eres.  - No ando ~
## $ texto_curado <chr> "Dime con quién andas y te diré quién eres. No ando con n~
```

```
## str_replace_all(text, "\\:|-|#", " ") reemplazo ":" o "-" o "#" por espacio
# esto es necesario para arreglar "hola:Pepe" que quedaría cómo una palabra si elimino:
## str_squish quita espacios  repetidos

texto_tokens = texto_df %>%  unnest_tokens(word, texto_curado)
glimpse(texto_tokens)
```

```
## Rows: 295,503
## Columns: 15
## $ origen   <chr> "1000 chistes", "1000 chistes", "1000 chistes", "1000 chistes~
## $ titulo   <chr> "Dime con quién andas...", "Dime con quién andas...", "Dime c~
## $ C1       <chr> "cortos", "cortos", "cortos", "cortos", "cortos", "cortos", "~
## $ C2       <chr> "malos", "malos", "malos", "malos", "malos", "malos", "malos"~
## $ C3       <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "~
## $ C4       <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "~
## $ C5       <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ palabra1 <chr> "feos", "feos", "feos", "feos", "feos", "feos", "feos", "feos~
## $ palabra2 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ palabra3 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ palabra4 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ palabra5 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ votos    <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ texto    <chr> "- Dime con quién andas y te diré quién eres.  - No ando con ~
## $ word     <chr> "dime", "con", "quién", "andas", "y", "te", "diré", "quién", ~
```

```
knitr::kable(head(texto_tokens, 20))
```

origen	titulo	C1	C2	C3	C4	C5	palabra1	palabra2	palabra3	palabra4	palabra5	votos	texto	word
1000 chistes	Dime con quién andas...	cortos	malos	NA	NA	NA	feos	NA	NA	NA	NA	NA	- Dime con quién andas y te diré quién eres. - No ando con nadie... - Eres feo.	dime

origen	titulo	C1	C2	C3	C4	C5	palabra	palabra1	palabra2	palabra3	palabra4	palabra5	texto	word
1000chistes	Dime con quién andas...	cortos	salados	NA	NA	NA	feos	NA	NA	NA	NA	NA	- Dime con quién andas y te diré quién eres. - No ando con nadie... - Eres feo.	con
1000chistes	Dime con quién andas...	cortos	salados	NA	NA	NA	feos	NA	NA	NA	NA	NA	- Dime con quién andas y te diré quién eres. - No ando con nadie... - Eres feo.	quién
1000chistes	Dime con quién andas...	cortos	salados	NA	NA	NA	feos	NA	NA	NA	NA	NA	- Dime con quién andas y te diré quién eres. - No ando con nadie... - Eres feo.	andas
1000chistes	Dime con quién andas...	cortos	salados	NA	NA	NA	feos	NA	NA	NA	NA	NA	- Dime con quién andas y te diré quién eres. - No ando con nadie... - Eres feo.	y
1000chistes	Dime con quién andas...	cortos	salados	NA	NA	NA	feos	NA	NA	NA	NA	NA	- Dime con quién andas y te diré quién eres. - No ando con nadie... - Eres feo.	te
1000chistes	Dime con quién andas...	cortos	salados	NA	NA	NA	feos	NA	NA	NA	NA	NA	- Dime con quién andas y te diré quién eres. - No ando con nadie... - Eres feo.	diré
1000chistes	Dime con quién andas...	cortos	salados	NA	NA	NA	feos	NA	NA	NA	NA	NA	- Dime con quién andas y te diré quién eres. - No ando con nadie... - Eres feo.	quién
1000chistes	Dime con quién andas...	cortos	salados	NA	NA	NA	feos	NA	NA	NA	NA	NA	- Dime con quién andas y te diré quién eres. - No ando con nadie... - Eres feo.	eres
1000chistes	Dime con quién andas...	cortos	salados	NA	NA	NA	feos	NA	NA	NA	NA	NA	- Dime con quién andas y te diré quién eres. - No ando con nadie... - Eres feo.	no
1000chistes	Dime con quién andas...	cortos	salados	NA	NA	NA	feos	NA	NA	NA	NA	NA	- Dime con quién andas y te diré quién eres. - No ando con nadie... - Eres feo.	ando

origen	titulo	C1	C2	C3	C4	C5	palabra	palabra1	palabra2	palabra3	palabra4	palabra5	texto	word
1000chistes	Dime con quién andas...	cortos	salón	NA	NA	NA	feos	NA	NA	NA	NA	NA	- Dime con quién andas y te diré quién eres. - No ando con nadie... - Eres feo.	con
1000chistes	Dime con quién andas...	cortos	salón	NA	NA	NA	feos	NA	NA	NA	NA	NA	- Dime con quién andas y te diré quién eres. - No ando con nadie... - Eres feo.	nadie
1000chistes	Dime con quién andas...	cortos	salón	NA	NA	NA	feos	NA	NA	NA	NA	NA	- Dime con quién andas y te diré quién eres. - No ando con nadie... - Eres feo.	eres
1000chistes	Dime con quién andas...	cortos	salón	NA	NA	NA	feos	NA	NA	NA	NA	NA	- Dime con quién andas y te diré quién eres. - No ando con nadie... - Eres feo.	feo
1000chistes	Luz automática	cortos	salón	borracho	actual	NA	nevera	NA	NA	NA	NA	NA	Va el marido completamente borracho y le dice a su mujer al irse para cama: - Me ha pasado algo increíble. He ido al baño y al abrir la puerta se ha encendido la luz automáticamente, sin hacer nada. - ¡La madre que te parió!, ¡Te mato!, ya te has vuelto a mear en la nevera.	va
1000chistes	Luz automática	cortos	salón	borracho	actual	NA	nevera	NA	NA	NA	NA	NA	Va el marido completamente borracho y le dice a su mujer al irse para cama: - Me ha pasado algo increíble. He ido al baño y al abrir la puerta se ha encendido la luz automáticamente, sin hacer nada. - ¡La madre que te parió!, ¡Te mato!, ya te has vuelto a mear en la nevera.	el
1000chistes	Luz automática	cortos	salón	borracho	actual	NA	nevera	NA	NA	NA	NA	NA	Va el marido completamente borracho y le dice a su mujer al irse para cama: - Me ha pasado algo increíble. He ido al baño y al abrir la puerta se ha encendido la luz automáticamente, sin hacer nada. - ¡La madre que te parió!, ¡Te mato!, ya te has vuelto a mear en la nevera.	marido
1000chistes	Luz automática	cortos	salón	borracho	actual	NA	nevera	NA	NA	NA	NA	NA	Va el marido completamente borracho y le dice a su mujer al irse para cama: - Me ha pasado algo increíble. He ido al baño y al abrir la puerta se ha encendido la luz automáticamente, sin hacer nada. - ¡La madre que te parió!, ¡Te mato!, ya te has vuelto a mear en la nevera.	completamente

origen	titulo	C1	C2	C3	C4	C5	palabra1	palabra2	palabra3	palabra4	palabra5	texto	word
1000 chistes	Luz automática	corta	salón	borracho	matrimonio	NA	NA	NA	NA	NA	NA	Va el marido completamente borracho y le dice a su mujer al irse para cama: - Me ha pasado algo increíble. He ido al baño y al abrir la puerta se ha encendido la luz automáticamente, sin hacer nada. - ¡La madre que te parió!, ¡Te mato!, ya te has vuelto a mear en la nevera.	borracho

```
dic_raw_1 = sort(unique(texto_tokens$word))
nw = length(dic_raw_1)#
nw # número de palabras distintas
```

```
## [1] 23456
```

1.3 Construcción del modelo de diccionario

Construiremos una tabla de modelado del corpus de palabras de los chistes:

- Como primary key la word (las nw words) (desde el text_raw en utf8)
- Su frecuencia: número de veces que aparece en los chistes
- Si es correcta según un spelling de español de España (hay que buscar... qué hay mejor)

```
count_freq=texto_tokens %>%
  group_by(word) %>% summarise(N=n())
dic_raw_1 = tibble(word=dic_raw_1) %>% left_join(count_freq,by="word")
```

Ahora vemos claramente cómo podemos mejorar las words para UNIFICARLAS en un único "léxico" que nos permita un tratamiento unificado, auqnue las variantes escritas podrían tener significado humorístico.

Ejemplos

Palabras que contienen "zq"

```
dic_raw_1[grep("zq",dic_raw_1$word),]
```

```
## # A tibble: 4 x 2
##   word          N
##   <chr>     <int>
## 1 izquierda    20
## 2 izquierdo     7
## 3 vazquezy      1
## 4 vezque        1
```

Palabras que contienen "ch"

```
dic_raw_1[grep("(ch)",dic_raw_1$word),]
```

```
## # A tibble: 832 x 2
##    word                  N
##    <chr>             <int>
##  1 2ºchiste              1
##  2 abolladuras.dicho     1
##  3 abrochados            1
##  4 acha                  1
##  5 achedo                1
##  6 achica                1
##  7 achíííííííííííís      1
##  8 achillar              1
##  9 achina                1
## 10 achiqué               1
## # ... with 822 more rows
```

Palabras (dos palabras) con :

```
dic_raw_1[grep(":",dic_raw_1$word),]
```

```
## # A tibble: 0 x 2
## # ... with 2 variables: word <chr>, N <int>
```

1.3.1 Añadimos columna de spelling al diccionario

Primero veamos algunos ejemplos de las sugerencias: ver manual en de hunspell. Github diccionarios open office

```
library("spelling")
library("hunspell")
#https://github.com/titoBouzout/Dictionaries #
es_ES<- dictionary("diccionarios/es_ES.dic")
#print(es_ES)
list_dictionaries()# estos son los que  vienen por defecto
```

```
## [1] "en_AU" "en_CA" "en_GB" "en_US"
```

```
hunspell_check(c("bieja","colon","colón"),dic= es_ES)
```

```
## [1] FALSE  TRUE FALSE
```

```
hunspell_suggest(c("bieja","colon","colón"),dic=es_ES)
```

```
## [[1]]
## [1] "vieja" "biela"
##
## [[2]]
## [1] "colon"  "clono"  "colo"   "colona" "colono" "colan"  "colen"  "color"
##
## [[3]]
## [1] "colon" "clonó" "coló"  "colan" "colen"
```

```
palabras=c("amor", "amoroso", "amorosamente", "amado", "amante", "amador")
hunspell_analyze(palabras,dic=es_ES)
```

```
## [[1]]
## [1] " st:amor"      "a st:mor fl:a"
##
## [[2]]
## [1] "a st:moroso fl:a"
##
## [[3]]
## [1] "a st:morosamente fl:a"
##
## [[4]]
## [1] " st:amar fl:D"
##
## [[5]]
## [1] " st:amante"       " st:amantar fl:E"
##
## [[6]]
## [1] " st:amador"      "a st:mador fl:a"
```

Eliminaremos las palabras que aparezcan menos de $K_{min} = 3$ o $K_{max} = 500$ veces y números y tomaremos la primera sugerencia para las palabras que den incorrectas y solo la primera sugerencia.

```
K_min = 3
K_max = 500
dic_raw_1 = dic_raw_1 %>% filter(N > K_min & N < K_max)
dim(dic_raw_1)
```

```
## [1] 5332    2
```

```
dic_raw_1 = dic_raw_1[-grep("\\w*[0-9]+\\w*\\s*", dic_raw_1$word), ]
dim(dic_raw_1)
```

```
## [1] 5259    2
```

```
palabras_incorrectas = sapply(
  dic_raw_1$word,
  FUN = function(x)
    hunspell_check(x, dic = es_ES)
)
table(palabras_incorrectas)
```

```
## palabras_incorrectas
## FALSE  TRUE
##  1485  3774
```

```
lista_sugerencias = sapply(
  dic_raw_1$word,
  FUN = function(x)
```

```
    hunspell_suggest(x, dic = es_ES)
)

# nos quedamos con la primera tanto para correctas como para incorrectas

dic_raw_1$word_curada = sapply(
  lista_sugerencias,
  FUN = function(x)
    x[1]
)
dic_raw_1$lista_sugerencias = sapply(
  lista_sugerencias,
  FUN = function(x) {
    if (length(x) >= 1) {
      return(paste(x, collapse = ","))
    }
    if (length(x) == 0) {
      return(NA)
    }
  }
)
# eliminamos NA

dic_raw_1 = dic_raw_1[!is.na(dic_raw_1$word_curada), ]
dim(dic_raw_1)
```

```
## [1] 5238    4
```

2 Primer modelo de curado de los chistes

```
knitr::kable(head(dic_raw_1, 20))
```

word	N	word_curada	lista_sugerencias
â	5	a	a,e,o,d,u,y
aa	5	as	as,a,ara,asa,ata,ala,ama,aja,aya,ea,ar,na,ca,ta,al
aaa	10	asa	asa,ara,ata,ala,ama,aja,aya,a
aaaa	7	bezaar	bezaar
abajo	76	abajo	abajo,abajó,bajo,abaja,abaje,abano,abato,atajo,abalo,ahajo,abajá,abajé,a bajo
abanico	4	abanico	abanico,abanicó,abanicos,abanica,abanicá
abecedario	8	abecedario	abecedario,abecedarios
abeja	7	abeja	abeja,abaje,aneja,abejar,abejas,abaja,aleja
aber	12	abre	abre,saber,caber,haber,abey,ayer,aberra,rabera
abeto	4	abeto	abeto,aneto,abetos,beato,abato,abete,abito,ateto,aleto
abia	66	abiar	abiar,rabia,abina,sabia,abita,labia,abra,aria,amia,babi
abian	10	abina	abina,rabian,abinan,abitan,abiar,abran,babiano
abienta	8	avienta	avienta,ablienta,ambienta,abierta,asienta,alienta,habiente,enrabieta,tienta,entablen
abierta	8	abierta	abierta,abiertas,rabieta,abierto,acierta
abiertas	5	abiertas	abiertas,abierta,rabietas,abiertos,aciertas

word	N	word_curadalista_sugerencias	
abierto	5	abierto	abierto,abiertos,abierta,acierto
abiertos	4	abiertos	abiertos,abierto,abiertas,aciertos
abion	5	abino	abino,sabiondo
abitacion	6	habitaciÃ³n	habitaciÃ³n
abla	7	bala	bala,alba,ala,abala,nabla,tabla,ambla,fabla,habla,abra,arla,aula,aballa

```
texto_tokens = texto_tokens %>%
  right_join(dic_raw_1, word_curada, by = "word")
```

2.1 Siguiente paso tratamiento de los datos curados y generación de las Document Term Matrix

Primera aproximación generación dela DTM del corpus de peticiones curadas. Cruzar estos datos con los tópicos/key words de losa chistes. Podéis hacerlo con tidytext o con tm (o con quanteda).

```
library(tm)
library(tidytext)
texto_tokens$N = 1
DTM = cast_dtm(texto_tokens,
               document = "titulo",
               term = "word_curada",
               value = N)
MM = as.matrix(DTM)
titulos = row.names(MM)
MM = as_tibble(MM)
MM$titulo = titulos
```

2.2 Generación de tópicos 4 tópicos

```
library(topicmodels)
set.seed(22)
chistes_2 = LDA(
  DTM,
  k = 2,
  method = "Gibbs",
  control = NULL,
  model = NULL
)

chistes_documentos <- tidy(chistes_2, matrix = "gamma")
chistes_documentos %>% arrange(document)
```

```
## # A tibble: 11,960 x 3
##    document                       topic gamma
##    <chr>                          <int> <dbl>
##  1 --DAA---NI YO SE                   1 0.508
##  2 --DAA---NI YO SE                   2 0.492
```

```
##  3 -¿A TI QUÉ ES LO QUE MÁS TE MO     1 0.419
##  4 -¿A TI QUÉ ES LO QUE MÁS TE MO     2 0.581
##  5 -NO ME CORRIJAS                    1 0.540
##  6 -NO ME CORRIJAS                    2 0.460
##  7 !!QUE LOCO!!                       1 0.45
##  8 !!QUE LOCO!!                       2 0.55
##  9 !REPITINEDO TODO!!!!!!             1 0.608
## 10 !REPITINEDO TODO!!!!!!             2 0.392
## # ... with 11,950 more rows
```

```
tabla_topicos = chistes_documentos %>% pivot_wider(id_cols = document,
                                                   names_from = topic,
                                                   values_from = gamma)
names(tabla_topicos)[2:3] = paste0("Topico_", names(tabla_topicos)[2:3])
names(tabla_topicos)
```

```
## [1] "document" "Topico_1" "Topico_2"
```

```
Topico = apply(
  tabla_topicos[, 2:3],
  1,
  FUN = function(x) {
    if (x[1] > x[2]) {
      topico = 1
    }
    if (x[1] < x[2]) {
      topico = 2
    }
    if (x[1] == x[2]) {
      topico = 0
    }
    return(topico)
  }
)


tabla_topicos = tabla_topicos %>% mutate(Clase = Topico)
tabla_topicos
```

```
## # A tibble: 5,980 x 4
##    document                 Topico_1 Topico_2 Clase
##    <chr>                       <dbl>    <dbl> <dbl>
##  1 Dime con quién andas...     0.561    0.439     1
##  2 Luz automática              0.429    0.571     2
##  3 Política argentina          0.463    0.537     2
##  4 0 positivo                  0.518    0.482     1
##  5 Mejor portero               0.491    0.509     2
##  6 Donación para la piscina    0.492    0.508     2
##  7 Clase de astrología         0.5      0.5       0
##  8 Bob Esponja                 0.526    0.474     1
##  9 Ojalá lloviera              0.491    0.509     2
## 10 En Canarias                 0.483    0.517     2
## # ... with 5,970 more rows
```

Podemos extraer también las categoría o palabras clave pero son demasiadas.

```
C1 = texto_df %>% select(titulo, C1)

df = C1 %>% right_join(MM, by = "titulo")
names(df)[1:10]
```

```
##  [1] "titulo" "C1"     "dime"   "quien"  "andas"  "eres"   "ando"   "nadie"
##  [9] "feo"    "marido"
```

```
library(naivebayes)
set.seed(1)
nrow(df)
```

```
## [1] 7133
```

```
Ntraining = floor(0.8 * nrow(df))
Ntraining
```

```
## [1] 5706
```

```
Ntesting = nrow(df) - Ntraining
Ntesting
```

```
## [1] 1427
```

```
training = sample(1:nrow(df), size = Ntraining, replace = FALSE)
testing = setdiff(1:row(df), training)
```

```
## Warning in 1:row(df): numerical expression has 34110006 elements: only the first
## used
```

```
train_data = df[training, -1]
testing_data = df[testing, -c(1:2)]
```

Quizá demasiadas categorías mejor topic models a 2 , 3 o 4 ,categorías.

3 Word to vect NUEVA librería word2vec

https://github.com/bnosac/word2vec

```
#install.packages("devtools","Rtools")
#install.packages("word2vec")

library(word2vec)
txt_clean = txt_clean_word2vec(
  x = data_raw$texto,
  ascii = FALSE,
```

```
  alpha = TRUE,
  tolower = TRUE,
  trim = TRUE
)
str(txt_clean)
```

```
##  chr [1:7169] "dime con quién andas y te diré quién eres no ando con nadie eres feo" ...
```

```
model = word2vec(
  x = txt_clean,
  type = "skip-gram",
  dim = 50,
  window = 10,
  iter = 5L,
  lr = 0.05,
  hs = FALSE,
  negative = 5L,
  sample = 0.001,
  min_count = 5L,
  split = c(" \n,.-!?:;/\"#$%&'()*+<=>@[]\\^_`{|}~\t\v\f\r", ".\n?!"),
  stopwords = character(),
  threads = 1L,
  encoding = "UTF-8"
)
```

```
embeding=as.matrix(model)
emb <- predict(model, c("autobus", "jaimito", "mujer"), type = "embedding")
emb
```

```
##               [,1]       [,2]        [,3]       [,4]      [,5]      [,6]
## autobus  0.4614497 -0.7015693 -0.08625729 -0.6368105 0.8653237 1.3184415
## jaimito -0.1230050  0.5476866 -0.54544300 -0.6536254 0.8905313 1.1177982
## mujer    0.5482492 -0.1097108 -0.75671971 -1.6972733 1.2018762 0.8578544
##              [,7]        [,8]       [,9]       [,10]       [,11]      [,12]
## autobus 1.0124913 -0.81226635 -1.8247426 -1.35571420  0.07717469  0.3323330
## jaimito 1.0840741  0.68837571 -0.3871457 -0.01122863 -0.15046863  0.6557256
## mujer   0.7220466 -0.09512452 -2.9089534  1.24603081  0.97407484 -1.5102351
##               [,13]      [,14]      [,15]      [,16]      [,17]     [,18]
## autobus  0.43101633 -0.3058873 -1.2772893 -1.1673007  0.5094044 0.9245596
## jaimito  0.06743942  0.8230564  0.1129118  2.1305294  0.2897556 0.7926894
## mujer   -0.87425512 -0.3123395  0.4964284 -0.4350386 -0.7200098 0.5253875
##             [,19]     [,20]      [,21]       [,22]     [,23]      [,24]
## autobus -1.643215 0.3629346 -0.1819946 -1.62984753 -1.103383 -1.4924392
## jaimito -2.197766 0.4734361 -1.0086843  0.06615321 -1.549496 -1.1303518
## mujer   -1.658968 0.1606160 -0.1121685 -1.24519408 -1.481628 -0.4543702
##               [,25]     [,26]      [,27]     [,28]      [,29]    [,30]
## autobus -0.07794574 0.5979277  1.5199399 2.0707569  0.5411263 2.117563
## jaimito -0.25519410 1.1837728  1.7212716 1.2693154 -1.4515793 1.179400
## mujer   -1.32938445 0.1488088 -0.3832394 0.6257533  1.1875179 2.174679
##                [,31]       [,32]      [,33]      [,34]      [,35]       [,36]
## autobus  0.379304826  0.18808720  0.3152350  0.6377318  0.1932275  1.89423800
## jaimito  0.434991628 -0.04600479  0.7614939  1.0820260 -0.8154675 -0.05148405
```

```
## mujer   -0.004361928  0.18420744 -0.2912782 -1.0610646  0.2476150  0.38918146
##              [,37]      [,38]      [,39]     [,40]     [,41]      [,42]
## autobus -0.1384565  0.4885756 -0.1177438 0.8971664 0.5343782 -0.6950428
## jaimito -0.1825012  0.6427447 -0.3123657 1.2637273 1.0866156  0.1153000
## mujer   -0.7534750 -0.3838530 -0.1966227 0.9228778 1.4414974  1.2608608
##              [,43]      [,44]      [,45]       [,46]      [,47]      [,48]
## autobus -0.4098321  0.7007095 -2.0941243  0.12383435 -0.1705630  0.4475664
## jaimito -0.6172656 -0.5942466 -0.6749417  2.74237251 -0.4259383  1.2990482
## mujer   -1.1789026 -0.8782324  0.4719648 -0.08341617  0.9798093 -0.5113722
##             [,49]      [,50]
## autobus -1.199504  1.2038382
## jaimito -1.685879 -0.2458823
## mujer    1.192020  1.1014234
```

```
nn  <- predict(model, c("jaimito", "profesor"), type = "nearest", top_n = 5)
nn
```

```
## $jaimito
##     term1   term2 similarity rank
## 1 jaimito   anota  0.8845856    1
## 2 jaimito aleluya  0.8741408    2
## 3 jaimito   jaimi  0.8687846    3
## 4 jaimito  lleben  0.8600357    4
## 5 jaimito  decime  0.8597242    5
##
## $profesor
##      term1     term2 similarity rank
## 1 profesor     memin  0.8942836    1
## 2 profesor     frase  0.8934199    2
## 3 profesor    frutas  0.8802133    3
## 4 profesor profesora  0.8791907    4
## 5 profesor    alumno  0.8734357    5
```

```
doc2vec(model,c("padre","madre","hijo"))
```

```
##              [,1]        [,2]        [,3]       [,4]      [,5]      [,6]
## [1,] -0.008430397 -0.02774017  0.42351053 -1.5585219 1.1941671 0.4167079
## [2,] -0.272798863  1.31413628 -0.04796652 -1.9333440 0.5908909 0.3706485
## [3,] -0.069426555  0.53999826  0.41430284 -0.7614521 0.9652834 0.1818042
##           [,7]       [,8]       [,9]       [,10]      [,11]      [,12]
## [1,] 0.7261723  0.5418362 -0.7659065 1.280058669  0.7332981 -0.2991854
## [2,] 2.1123039  0.5543399 -0.5527552 1.181207417  0.3325732 -0.1627728
## [3,] 0.7014146 -0.2596275 -1.0985138 0.003610637 -0.7138980  0.2104440
##           [,13]        [,14]      [,15]    [,16]      [,17]      [,18]
## [1,]  0.5728250 -0.445520614 -1.7587444 0.925433  0.3761573  0.4242116
## [2,] -0.5121179 -0.001967118 -0.9688074 0.899889 -1.3580434 -0.2903111
## [3,]  0.3548024  0.369000750 -1.1144382 1.191136 -0.8359121 -0.3610345
##          [,19]      [,20]      [,21]      [,22]      [,23]      [,24]     [,25]
## [1,] -2.918011  0.3841107 -1.7157562 -0.2006897  0.8552871 -1.0966499 -1.350121
## [2,] -2.462864 -0.6332163 -0.4048422  0.2684772 -0.1316036 -0.6331256 -1.550270
## [3,] -1.723401 -0.2266302 -1.3403578 -0.1245633  0.3470832 -0.9637611 -1.798917
##          [,26]    [,27]     [,28]      [,29]    [,30]     [,31]     [,32]
## [1,] 1.7831205 1.316556 1.4862621  0.3094210 0.527542 1.4652790 0.1733522
```

```
## [2,] 0.9135450 1.370884 1.1514993 -0.6892326 1.785975 0.4207382 0.8749534
## [3,] 0.9693741 2.127361 0.8929945  0.1375768 0.884964 0.3988609 0.4135168
##           [,33]     [,34]      [,35]     [,36]      [,37]      [,38]
## [1,] -1.1295121 0.6878630 -0.6611583 0.4584027 -1.2385837 -1.2188151
## [2,] -0.9414576 0.8278263 -1.2421320 0.7295561 -0.3089577 -0.1186221
## [3,] -1.0602015 0.4983946 -1.6116085 1.9866433 -1.6127036 -1.3355423
##            [,39]      [,40]     [,41]        [,42]      [,43]      [,44]
## [1,] -0.36093963 -0.9146458 0.1176836 -1.000713888 -0.5510713  0.2149831
## [2,] -0.75478138  0.5684960 2.1036823 -0.003413682  0.4187725 -0.8580225
## [3,] -0.06120688 -0.0362366 1.2340657 -2.183687786 -0.1056057 -0.6571932
##           [,45]     [,46]       [,47]      [,48]      [,49]      [,50]
## [1,] -0.2040070 0.1123397 -0.08846012  0.3566116 -2.0542892  0.2942753
## [2,] -1.4639237 1.5538071  0.24262612 -0.3630036 -0.2305186 -0.6833944
## [3,] -0.6808598 1.6112599  0.41439839 -0.1501099 -0.9484317  1.4857369
```

```
M=as.matrix(model)
dim(M)
```

```
## [1] 4375   50
```

```
#Simi=word2vec_similarity(M,M,top_n=+Inf, type="cosine")
cosine <- function(x,y) sum(x * y)/sqrt(sum(x^2)*sum(y^2))
# install.packages("proxy")
library(proxy)
```

```
##
## Attaching package: 'proxy'
```

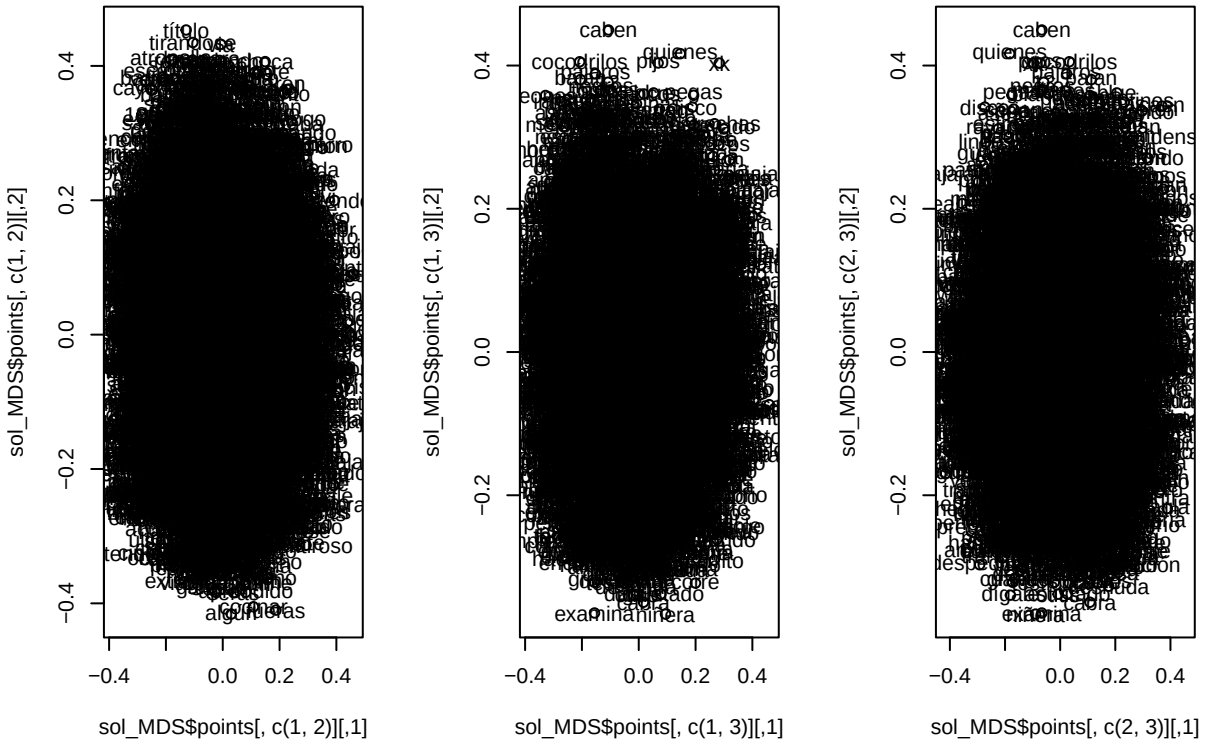
```
## The following objects are masked from 'package:stats':
##
##     as.dist, dist
```

```
## The following object is masked from 'package:base':
##
##     as.matrix
```

```
SS=as.matrix(simil(M,method=cosine))
diag(SS)=1
D=sqrt(1-SS)
dimnames(D)=list(dimnames(M)[[1]],dimnames(M)[[1]])
sol_MDS=cmdscale(D,k = 3,list=TRUE)
str(sol_MDS)
```

```
## List of 5
##  $ points: num [1:4375, 1:3] -0.3845 -0.0437 0.0641 0.1511 0.1144 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4375] "usd" "tocas" "ria" "caducado" ...
##   .. ..$ : NULL
##  $ eig   : NULL
##  $ x     : NULL
##  $ ac    : num 0
##  $ GOF   : num [1:2] 0.244 0.244
```

```
par(mfrow=c(1,3))
plot(sol_MDS$points[,c(1,2)])
text(sol_MDS$points[,c(1,2)],dimnames(M)[[1]])
plot(sol_MDS$points[,c(1,3)])
text(sol_MDS$points[,c(1,3)],dimnames(M)[[1]])
plot(sol_MDS$points[,c(2,3)])
text(sol_MDS$points[,c(2,3)],dimnames(M)[[1]])
```




```
par(mfrow=c(1,1))
```

4 Naive bayes

Podéis utilizar algún algoritmo de naivebayes con los metadatos de los chistes (fichero que se explica abajo) o con topic models.

5 Enunciado del taller

Basándonos en las ayudas de Enunciado_taller2_chistes_con_metadatos.Rmd" lo anterior generar un modelo de datos con 4 tópicos (de topic models o combinado con categorías o palabras clave. Asignar cada tópico a su γ más alto) y un diccionario de palabras curadas por chistes.

Entregad un informe contestando estas dos cuestiones (enviad Rmd y html).

5.1 Cuestión 1

Naive Bayes para predecir las 4 categorías de chistes a partir de las variables de presencia ausencia de las palabras. Evaluar el modelo.

5.2 Cuestión 2

Calcular para cada palabra de word2vec (si existe) el β de cada tópico y asignarlo cada palabra al tópico del β más alto.

A partir de la librería `word2vec` generar una proyección y estudiar si las palabras se agrupan según los tópicos.