Universidad De Chile

Facultad de Ciencias Físicas y Matemáticas

LDA con datos de Twitter

Javier Castro Medina

Abril 2021

1 Análisis Exploratorio

Para la obtención de tweets se usa la librería `twint`[1]. Esta librería es simple de usar, permite obtener una gran cantidad de datos y almacenarlos en formato `csv` u otros. En las Figuras 1a y 1b se muestra el código que se necesita para obtener los datos utilizados en este trabajo.

```
# Define c object
c = twint.Config()
# Parameters
c.Search = word
c.Near = 'Santiago, Chile'
c.Since = since
c.Until = until
c.Store_csv = True
c.Output = 'data/'+ file_name +'.csv'
# Run search
twint.run.Search(c)
```

(a)

```
# Define c object
c = twint.Config()
# Parameters
c.Username = id
c.Store_csv = True
c.Output = 'data/'+ id +'.csv'
# Run search
twint.run.Search(c)
```

(b)

Figure 1: A la izquierda se buscan y guardan tweets creados al rededor de Santiago entre las fechas `since` y `until` que además contengan la palabra `word`. A la derecha se obtienen y guardan los tweets de la cuenta `id`.

En la primera parte del proyecto se realiza un proceso de adaptación a la librería `twint` y el formato de los datos que permite obtener además de un análisis superficial a los datos que obtenemos en esta fase inicial. Se decide trabajar sólo con tweets en Santiago, esto porque en esta región hay una población más grande y por lo tanto una probabilidad más alta de encontrar cuentas de Twitter con respecto a otros lugares. Lo que queda de esta sección se divide en dos partes, en la primera se mostrarán worclouds y en la segunda gráficos de la frecuencia de tweets diarios para distintos set de datos.

Remark 1.1. *Otra razón por la cual fijar el lugar geográfico de la obtención de tweets es para disminuir la complejidad del proyecto, puede ser una variable a considerar en una tarea de mayor escala que además tome en cuenta la variable espacial de nuestros datos.*

1.1 Wordclouds

Se muestran las wordclouds asociadas a tres set de tweets (o datos) distintos. Por un lado tenemos dos hitos importantes en Chile, el primero es el estallido social del 18 de octubre del 2019 y el segundo es el día de la mujer del 2020. El tercero contiene los tweets que contienen la palabra "piñera" y fueron creados durante el 2020.

[1] github.com/twintproject/twint

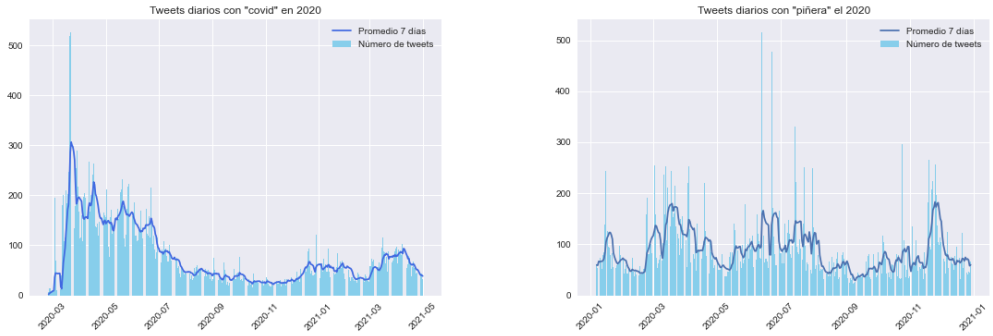
(a) (b) (c)

Figure 2: De izquierda a derecha: estallido social, día de la mujera y tweets que contienen la palabra "piñera".

Para encontrar los tweets del estallido y el día de la mujer sólo se especificaron las fechas y el lugar, el campo `c.Search` que aparece en la Figura 1a se deja vacío. En cambio, para los tweets con "piñera", se debe usar `c.Search = 'piñera'`. Las palabras **metrodesantiago** y **evasionmasivatodoeldía** que aparecen en 2a corresponden a hashtags que estuvieron presentes en Twitter durante los días previos al estallido. En la Figura 2b observamos las palabras **mujeres** y **marcha**, esto sucede porque en esa fecha ocurre la marcha en conmemoración del día de la mujer que, como vemos, tiene impacto en redes sociales como Twitter. En la Figura 2c se destacan las palabras **gobierno**, **Chile** y **derecha** que se relacionan bastante con lo que representa la palabra "piñera". En las tres wordclouds se encuentran palabras bastante esperables y que guardan estrecha relación con el contexto social en el cual fueron creados los tweets, esta es una señal que confirma que nuestros datos tienen sentido con su contexto.

1.2 Series de tiempo

Se exponen series de tiempo relativas a la cantidad de tweets diarios que contengan cierta palabra clave. Acá trabajamos con dos set de tweets, por un lado tenemos el mismo set de datos de la sección anterior asociado a la palabra "piñera" y por otro contamos con los tweets que contienen la palabra "covid" y que fueron creados durante el año 2020.



(a) (b)

Figure 3: Cantidad de tweets diarios con la palabra "covid" (izq) y "piñera" (der).

En la Figura 3a vemos que la curva que se genera al suavizar tomando media móvil de siete días es similar a la curva de casos diarios de covid en Chile (también suavizada), pero adelantada en tiempo. El primer peak de los casos diarios ocurrió al rededor de junio [2], mientras que en 3a esto ocurre en marzo cuando todo estaba comenzando. Aproximadamente, en ambas curvas, durante los siguientes seis meses se ve una tendencia a la baja. Esta especie de caracter predictivo de la curva 3a no es sostenible en el tiempo porque si bien los casos aumentaron hasta superar el primer peak, en Twitter sólo se ve un aumento relativamente pequeño en la cantidad de tweets asociados al covid durante enero. Por otra parte, el peak más alto en 3a corresponde al día 23 de marzo del 2020, en esta fecha se informa la primera muerte por covid en Chile.

Con respecto a la Figura 3b, observamos que la cantidad de tweets que hablan de Piñera es relativamente constante en el tiempo, con un promedio de 86 tweets diarios, y sin una tendencia marcada.

[2]Link a los casos diarios covid en Chile según Google.

2 Latent Dirichlet Allocation aplicado a datos de Twitter

Este algoritmo cae en la categoría del **topic modelling** en la cual se busca, a partir de datos de texto, encontrar tópicos latentes en este. Para este modelo se necesita un set de palabras V y una cantidad de tópicos K. Los tópicos se modelan como distribuciones de probabilidad sobre el conjunto de palabras V. Si bien este algoritmo tiene una fase generativa, su objetivo es calibrar parámetros $\alpha \in [0,1]^K$ y $\beta^{K\times V}$ optimizando una cota variacional. La idea es, dados α y β, generar palabras con el siguiente algoritmo:

- Se samplea una distribución sobre tópicos $\theta \sim Dir(\alpha, K)$. Notar que $\sum \theta_i = 1$.
- Dado θ, se samplea un tópico $t \sim \theta$.
- Dado el tópico t, se samplea una palabra $w \sim \beta_t$..

Lo anterior permite samplear una (1) palabra, la idea es tomar estos sampleos de forma independiente. Tomemos una palabra w_n, luego, la probabilidad de generar esta palabra se puede calcular como,

$$\begin{aligned} p(w_n|\alpha,\beta) &= \sum_{t_n} p(w_n|\alpha,\beta,t_n)p(t_n|\alpha,\beta) = \sum_{t_n} p(w_n|\beta,t_n)\int_\theta p(t_n|\alpha,\beta,\theta)p(\theta|\alpha,\beta)d\theta \\ &= \sum_{t_n} p(w_n|\beta,t_n)\int_\theta p(t_n|\theta)p(\theta|\alpha)d\theta. \end{aligned}$$

Donde usamos probabilidades totales y que, según nuestro modelo, $p(w_n|\alpha,\beta,t_n) = p(w_n|\beta,t_n)$, $p(t_n|\alpha,\beta,\theta) = p(t_n|\theta)$ y que $p(\theta|\alpha,\beta) = p(\theta|\alpha)$. Usando la independencia vemos que,

$$p(w|\alpha,\beta) = \int p(\theta|\alpha)\left(\prod_{n=1}^{N}\sum_{t_n} p(w_n|t_n,\beta)p(t_n|\theta)\right)d\theta.$$

Primero se hace un review a los datos mostrando distintas estadísticas asociadas a estos. Luego, se expone la limpieza y preprocesamiento que se realiza sobre el texto para dejarlo en condiciones de ser usado por el modelo y los métodos de la librería `gensim`[3].

2.1 Datos

Podríamos aplicar el modelo a todos los tweets de cierto espacio y tiempo pero se inclina por otro approach que es trabajar con los tweets de una cuenta en específico dado que el rango de temas o tópicos tocados por una cuenta es menor o más controlado que al trabajar con tweets de muchas cuentas. La cuenta que se escoge para partir es `@gabrielboric`, obtenemos sus tweets hasta junio del 2021. Las características de este data set se pueden apreciar en las Figuras 4a y 4b

[3] Lda con Gensim (link)

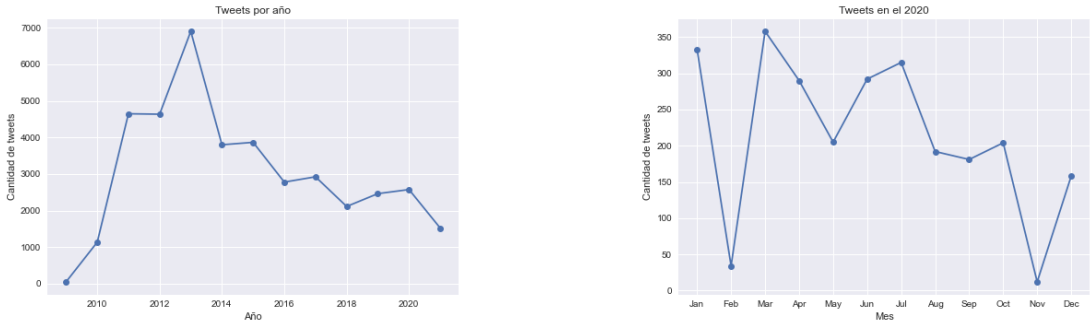


Figure 4: Tweets por año (izq) y por mes en el 2020 (der).

	Media	**Min**
Total	cell5	cell6
7 días	cell8	cell9

2.2 Limpieza y preprocesamiento de datos

Los datos que se tienen disponible corresponden a tweets en formato de texto. El procesamiento de base o inicial que se realiza sobre este corpus corresponde a eliminar **stop words**, links, palabras de largo menor o igual a 3, caracteres indeseables[4] y finalmente pasar todas las palabras a minúscula. El procesamiento anterior se realizará siempre y al inicio de todo código. Lo siguiente que se puede hacer es quitar palabras de baja frecuencia, sin quitar estas palabras el diccionario[5] resultante tiene un tamaño de 35892 palabras. En el gráfico a continuación muestra la cantidad de palabras resultantes al variar este parámetro.

[4]Se eliminaron: #, comas, signos de exclamación y pregunta, paréntesis y el signo igual.
[5]Entendemos por diccionario al conjunto de todas las palabras utilizadas en el corpus

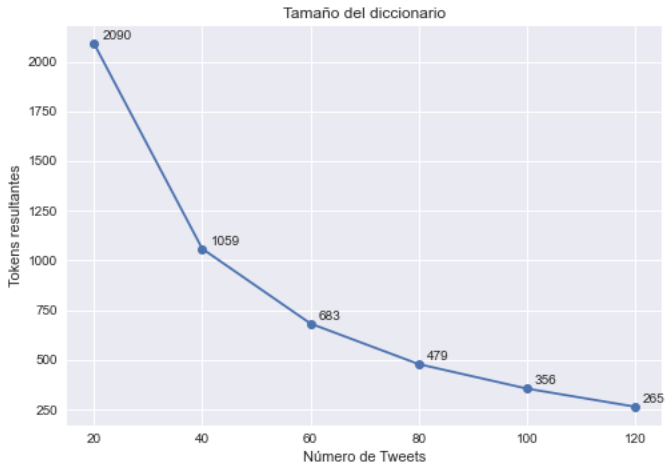


Figure 5: Tamaño del diccionario en función de la cantidad de tweets mínima a la que las palabras deben permanecer. Es decir, y corresponde a las palabras que resultan al filtrar las que aparezcan en menos de x tweets.

La siguiente etapa en el preprocesamiento es transformar los tweets en vectores. Para esto, a cada palabra del diccionario, digamos V, se le asigna un único índice de forma que el conjunto se palabras (o diccionario) se modela como $\{1, ..., |V|\}$. Luego, cada tweet, entendido como una colección de palabras en V de la forma $t = \{v_1, ..., v_n\}$, se transforma en $\{(v_1, r_1), ..., (v_n, r_n)\}$ donde v_i es el índice de la palabra en V y r_i la cantidad de veces que v_i aparece en t. A continuación, un ejemplo de esto.

```
bow tweet 304:
 [(256, 1), (343, 1), (653, 1), (662, 1), (755, 1), (1078, 1), (1340, 1), (1457, 1), (1646, 1),
(1650, 1), (1805, 1), (1862, 1), (1870, 1), (2263, 1), (2264, 1), (2295, 1), (2434, 1), (2435, 1),
(2438, 1), (2454, 1), (2455, 1), (2456, 1)]
bow tweet 666:
 [(4, 1), (10, 1), (273, 1), (445, 1), (492, 1), (733, 1), (752, 1), (769, 2), (1008, 1), (1011, 1),
(1471, 1), (2274, 1), (4151, 1), (4152, 1), (4153, 1), (4154, 1)]
```

Figure 6: Estado final de dos tweets escogidos al azar. Esta representación se conoce como "bow" (**bag of words**).

Remark 2.1. *Notar que en los ejemplos que se muestran en la Figura 6 la segunda componente de las tuplas es mayoritariamente* 1. *Esto pasa porque el largo de cada tweet en el corpus es relativamente pequeño en comparación a otros set de datos como por ejemplo una colección de libros o notas periodísticas. Esto sugiere que esta variable no es muy decisiva en este contexto.*

Remark 2.2. *Se decide dejar las palabras que hacen referencia a cuentas de Twitter, por ejemplo* `@javier`. *La razón de esta decisión es que la aparición de una cuenta en un tópico nos indica que tal persona es relevante para la cuenta sobre la que se está trabajando.*

2.3 Aplicando el modelo

Primera Iteración: En la Figura 7 se muestra el output de pedirle los tópicos al modelo.

```
TOPIC: 0
WORDS: 0.010*"educación" + 0.009*"política" + 0.008*"chile" + 0.007*"acuerdo" +
0.006*"izquierda" + 0.006*"derecha" + 0.005*"gobierno" + 0.005*"puede" + 0.005*"mejor" +
0.005*"recomiendo"
TOPIC: 1
WORDS: 0.010*"saludos" + 0.008*"@jschaulsohn" + 0.008*"@valenzuelalevi" + 0.007*"alguien"
+ 0.006*"@jparedesgodoy" + 0.005*"quiere" + 0.005*"marcha" + 0.004*"bueno" +
0.004*"mercurio" + 0.004*"trata"
TOPIC: 2
WORDS: 0.014*"abrazo" + 0.013*"ahora" + 0.012*"magallanes" + 0.011*"buena" +
0.011*"mañana" + 0.010*"vamos" + 0.010*"fech" + 0.009*"arenas" + 0.008*"aguante" +
0.008*"punta"
TOPIC: 3
WORDS: 0.013*"muchas" + 0.012*"gracias" + 0.010*"@jen_abate" + 0.008*"@matiasdelrio" +
0.008*"compa" + 0.007*"feliz" + 0.006*"concerta" + 0.006*"@saladehistoria" +
0.005*"concertación" + 0.005*"buenos"
TOPIC: 4
WORDS: 0.012*"@cbellolio" + 0.010*"casa" + 0.008*"@lboric" + 0.007*"terrible" +
0.006*"facultad" + 0.006*"rector" + 0.006*"@donmatas" + 0.005*"universidades" +
0.005*"stgo" + 0.005*"superior"
```

Figure 7: Los tópicos están ordenados por coherencia y los números que aparecen al lado de cada palabra corresponde a la probabilidad de tal palabra en el tópico.