
Factorización de matrices no negativas y aplicación en los textos

YAIR ANTONIO CASTILLO-CASTILLO¹

¹Centro de Investigación de Matemáticas (CIMAT), Monterrey, Nuevo León, México
yair.castillo@cimat.mx

La factorización de matrices es un tema importante en el procesamiento de información, el cual tiene numerosas aplicaciones en diversas áreas. La factorización de matrices no negativa (NMF, por sus siglas en inglés, Nonnegative Matrix Factorization) es nuestro caballo de batalla de este trabajo para representación de datos dispersos, o datos donde muchos de sus valores son ceros. Existe una gran variedad de algoritmos para resolver este problema, los algoritmos multiplicativos iterativos (un algoritmo para cada β -divergencia $\beta = \{0, 1, 2\}$), ADMM (Método de Direcciones Alternantes de Multiplicadores) y el algoritmo de *Scikit learn* serán estudiados. Se aplicará esta herramienta para la búsqueda y extracción de tópicos de un conjunto de textos o corpus. Se compararán los algoritmos, calificando cuáles si pueden ser relacionados con un tema en particular.

Keywords: Factorización de matrices no negativas; textos; tópicos

1. INTRODUCCIÓN

Un problema bastante extendido en diferentes técnicas de análisis de datos consiste en encontrar una representación adecuada de los datos. Muchos datos del mundo real no son negativos y los componentes ocultos correspondientes tienen un significado físico únicamente cuando no son negativos.

En la práctica, las descomposiciones de datos tanto no negativas como escasas (“sparse”) suelen ser deseables o necesarias cuando los componentes subyacentes tienen una interpretación física. Por ejemplo, en el procesamiento de imágenes y la visión por computadora, las variables y los parámetros involucrados pueden corresponder a píxeles, y la descomposición dispersa no negativa está relacionada con la extracción de partes relevantes de las imágenes. Una imagen en color se puede considerar como datos 3D no negativos, dos de las dimensiones (filas y columnas) son espaciales y la tercera es un plano de color (canal) dependiendo de su espacio de color. En la recuperación de información, los documentos generalmente se representan como frecuencias relativas de palabras en un vocabulario prescrito. En ciencias ambientales, los científicos investigan una proporción relativa de diferentes contaminantes en el agua o el aire.

En biología, cada eje de coordenadas puede corresponder a un gen específico y la escasez es necesaria para encontrar patrones locales ocultos en los datos, mientras que la no negatividad es necesaria para dar un significado físico o fisiológico [6].

Un tipo de representación de gran utilidad será aquella que permita reducir las dimensiones de los datos a la vez que muestre ciertas características del conjunto de éstos que permanezcan ocultas a priori. NMF es una técnica de reciente creación cuya principal utilidad consiste en encontrar una representación lineal de los datos, que deben de ser no negativos.

NMF es un modelo aditivo que no permite la resta; por lo tanto a menudo describe cuantitativamente las partes que componen la entidad completa. En otras palabras, NMF puede considerarse como una representación basada en partes en la que un valor cero representa la ausencia y un número positivo representa la presencia de algún evento o componente. Además, los métodos de factorización matricial que explotan las restricciones de no negatividad y dispersión generalmente conducen a la estimación de los componentes ocultos con estructuras específicas e interpretaciones físicas, en contraste con otros métodos de separación de fuentes ciegas [6].

ADMM es un algoritmo que pretende combinar la descomponibilidad de doble ascenso con las propiedades de convergencia superiores del método de los multiplicadores [3]. Recientemente, se demostró que el método de multiplicador de dirección alterna (ADMM) es más preciso y eficiente que los enfoques clásicos, como la regla de actualización multiplicativa (MUR). Cuya función de costo es una divergencia beta, una clase amplia de funciones de divergencia.

En este trabajo primero se definirá el problema NMF, luego métodos o algoritmos para resolverlo ya sea Multiplicadores aditivos o ADMM, la implementación de NMF en python para luego aplicarlo para en textos y así obtener sus tópicos.

2. FACTORIZACIÓN DE MATRICES NO NEGATIVAS

El problema básico de la Factorización de matrices no negativas se puede plantear de la siguiente manera: Dada un matriz de datos no negativa $\mathbf{V}_{n \times d}$ (con $v_{i,j} \geq 0$ ó equivalente $\mathbf{V} \geq 0$) y un rango reducido r ($r \leq \min(n, d)$), encontrar 2 matrices no negativas $\mathbf{W}_{n \times k}$ (base) y $\mathbf{H}_{k \times d}$ (pesos o coeficientes), con $w_{i,j} \geq 0$ ó equivalente $\mathbf{W} \geq 0$ y $h_{j,t} \geq 0$ ó equivalente $\mathbf{H} \geq 0$ respectivamente, tal que:

$$\mathbf{V} \approx \mathbf{W}\mathbf{H} \quad (1)$$

Los matrices \mathbf{W} y \mathbf{H} pueden tener diferentes significados físicos en diferentes aplicaciones. En un problema de BSS (Blind Signal Separation ó separación de señal ciega, que es la separación de un conjunto de señales fuente de un conjunto de señales mixtas, sin la ayuda de información o con muy poca información sobre las señales fuente o el proceso de mezcla), \mathbf{W} desempeña el papel de matriz de mezcla, mientras que \mathbf{H} expresa señales de fuente. En problemas de agrupamiento, \mathbf{W} es la matriz base, mientras que \mathbf{H} denota la matriz de ponderaciones. En el análisis acústico, \mathbf{W} representa los patrones básicos, mientras que cada fila de \mathbf{H} expresa puntos de tiempo (posiciones) cuando se activan los patrones de sonido. En el análisis de textos, si las filas \mathbf{V} son los textos y las columnas las palabras (en este punto ya se hizo en preprocesamiento de los textos y se quitaron las “stop words”, números, acentos y puntuación), entonces \mathbf{W} representa los tópicos en los documentos y \mathbf{H} representa las palabras en los tópicos.

Este trabajo se enfoca en este último ejemplo de análisis de textos para poder obtener los tópicos o los temas de un corpus. En la Figura 1 se puede observar una representación visual de cómo funciona la factorización para el caso mencionado, ahí representa cada uno los aspectos.

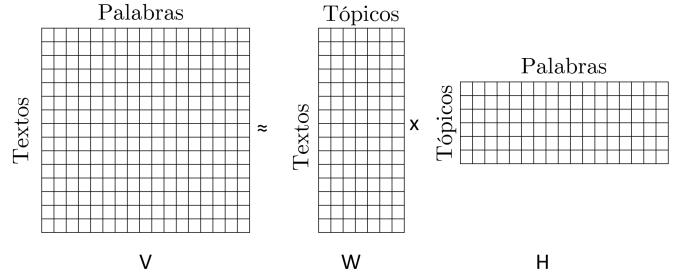


Figure 1. Representación visual de la factorización de matrices no negativas aplicado en textos

Hay casos especial NMF tales como Symmetric NMF, Semi-Orthogonal NMF, Semi-NMF and Nonnegative Factorization of Arbitrary Matrix, Three-factor NMF, NMF with Offset (Affine NMF), Multi-layer NMF, Simultaneous NMF, Projective and Convex NMF, Kernel NMF, Convulsive NMF, Overlapping NMF entre otros caso, que se puede encontrar en [6].

3. ALGORITMOS MULTIPLICATIVOS ITERATIVOS PARA NMF

Hay una amplia familia de algoritmos multiplicativos iterativos para la factorización matricial no negativa (NMF) y problemas relacionados, sujetos a restricciones adicionales como la dispersión y/o suavidad.

Por lo que consideramos la formulación de NMF como problema de optimización [12]:

$$\begin{aligned} & \min_{\mathbf{W}, \mathbf{H}} D(\mathbf{V} || \mathbf{W}\mathbf{H}) \\ & \text{sujeto a: } \mathbf{W}, \mathbf{H} \geq 0 \end{aligned} \quad (2)$$

Para encontrar una aproximación primero necesitamos definir una función de costo $D(\mathbf{V} || \mathbf{W}\mathbf{H})$ que cuantifique la calidad de nuestra aproximación. Tal función de costo puede ser construido usando alguna medida de distancia entre 2 matrices no negativas \mathbf{A} y \mathbf{B} , para esto existe un gama de funciones de costo que incluye un gran número de divergencias generalizadas, como la distancia euclidiana ponderada al cuadrado, la entropía relativa, la Kullback Leibler I-divergencia, α -& β -divergencias, la divergencia de Bregman y la Csiszár f -divergencia. Como casos especiales, también hay algoritmos multiplicativos para las distancias de Hellinger al cuadrado, Chi-cuadrado de Pearson e Itakura-Saito.

De la β -divergencias tenemos una familia, y están definidas como

$$D_\beta(X || Y) = \sum_{i,j} d_\beta(x_{i,j} || y_{i,j}) \quad (3)$$

donde $\beta \in \mathbb{R} \setminus \{0, 1\}$ y donde d_β está definido como

$$d_\beta(x||y) = \frac{x^\beta}{\beta(\beta - 1)} + \frac{y^\beta}{\beta} - \frac{xy^{\beta-1}}{\beta - 1} \quad (4)$$

La definición (4) la podemos extender tomando límites. Las tres funciones de divergencia más utilizadas con NMF son los casos especiales de la β -divergencia:

- $\beta = 2$ (Euclideana): $d(x||y) = \frac{1}{2}(x - y)^2$
- $\beta = 1$ (Kullback-Leibler): $d(x||y) = x \log \frac{x}{y} - x + y$
- $\beta = 0$ (Itakura-Saito): $d(x||y) = \frac{x}{y} - \log \frac{x}{y} - 1$

En la Figura 2 se observa una gráfica para varios valores de β . La distancia euclidiana es quizás la forma natural de medir la divergencia entre dos números, pero en muchas aplicaciones que involucran datos no negativos, no es natural, es decir, porque subyace a una distribución gaussiana de valor real para los datos. En el modelado de temas donde los datos son cuentas, es natural minimizar la divergencia Kullback-Leibler (KL). En audio, la divergencia Itakura-Saito (IS) es una elección natural. También es invarianta bajo la escala, por lo que le da a los pequeños coeficientes de tiempo-frecuencia la misma importancia que a los más grandes, al igual que el oído humano [18].

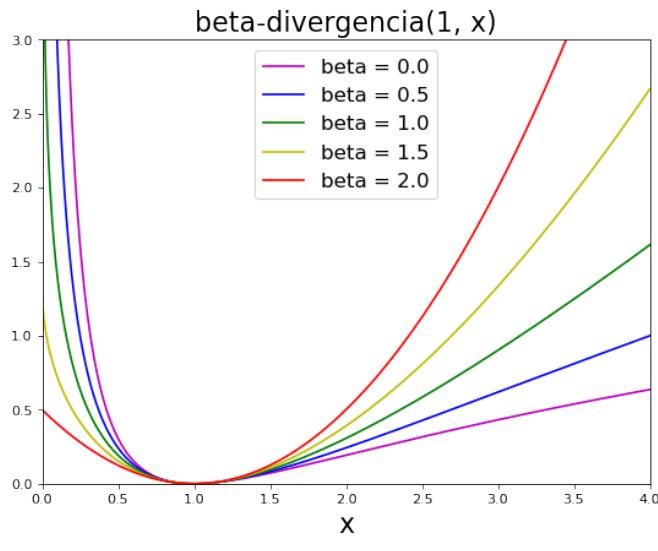


Figure 2. Comparación de varios β -divergencias

Tomar en cuenta que la derivada de $d_\beta(x||y)$ con respecto a y es también continua en β , y simplemente se escribe

$$\nabla d_\beta(x||y) = y^{\beta-2}(y - x) \quad (5)$$

La derivada (ecuación (5)) muestra que $d_\beta(x||y)$, en función de y , tiene un mínimo único en $y = x$ y que

aumenta con $|y - x|$, justificando su relevancia como medida de ajuste [9].

Por lo que consideraremos la nueva formulación de NMF como problema de optimización [12]:

$$\begin{aligned} & \min_{W,H} D_\beta(V||WH) \\ & \text{sujeto a: } W, H \geq 0 \end{aligned} \quad (6)$$

Donde $D_\beta(V||WH)$ puede ser una de las distancias o divergencias de (3), debido a que las funciones de distancias son convexas solo en W o solo en H , no son convexas en ambas variables juntas. Por lo tanto, no es realista esperar que un algoritmo resuelva todos los problemas con diferentes funciones en el sentido de encontrar mínimos globales. Sin embargo, existen muchas técnicas de optimización numéricas que se pueden aplicar para encontrar mínimos locales [12].

El descenso por gradientes es quizás la técnica más sencilla de implementar, pero la convergencia puede ser lenta. Otros métodos, como el gradiente conjugado, tienen una convergencia más rápida, al menos en las proximidades de los mínimos locales, pero son más complicados de implementar que el descenso del gradiente [14]. La convergencia de métodos basados en gradientes también tiene la desventaja de ser muy sensible a la elección del tamaño de paso, lo que puede ser muy inconveniente para aplicaciones grandes [12].

3.1. Distancia Euclideana

Podemos formular el problema de la estimación de matrices W y H como el de maximizar la función de verosimilitud:

$$p(V||W, H) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|V - WH\|_F^2}{2\sigma^2}\right) \quad (7)$$

sujeto a $W \geq 0$ y $H \geq 0$.

Maximizar la probabilidad es equivalente a minimizar la función de probabilidad logarítmica negativa correspondiente, o de manera equivalente, la norma de Frobenius al cuadrado que es una extensión obvia de la norma euclídea a matrices [1]:

$$\begin{aligned} D_\beta(V||WH) &= \frac{1}{2} \sum_{i,t} (v_{i,t} - [WH]_{i,t})^2 \\ & \min \quad \frac{1}{2} \sum_{i,t} (v_{i,t} - [WH]_{i,t})^2 \\ & \text{sujeto a: } w_{i,j}, h_{j,t} \geq 0 \end{aligned} \quad (8)$$

Usando el enfoque de descenso por gradiente y cambiando alternativamente entre los dos conjuntos de parámetros, obtenemos fórmulas de actualización multiplicativas simples:

$$\begin{aligned} w_{i,j} &= w_{i,j} \frac{[VH^T]_{i,j}}{[WH^T]_{i,j}} \\ h_{j,t} &= h_{j,t} \frac{[WV^T]_{i,j}}{[W^TWH]_{j,t}} \end{aligned} \quad (9)$$

La distancia euclídea es invariante bajo estas actualizaciones si y solo si \mathbf{W} y \mathbf{H} están en un punto estacionario de la distancia.

El algoritmo de la ecuación (9) es llamado a menudo el algoritmo de Lee-Seung NMF [12]. Por lo general, se agrega una pequeña constante positiva ϵ a los denominadores para evitar la división por cero.

3.2. Divergencia de Kullback-Leibler

Los algoritmos multiplicativos adaptativos más conocidos y más utilizados para NMF se basan en dos funciones de pérdida: distancia euclídea al cuadrado (Sección 3.1) y divergencia Kullback-Leibler generalizada, también llamada I-divergencia definida como:

$$D(V||WH) = \sum_{i,t} \left(v_{i,t} \log \left(\frac{v_{i,t}}{[WH]_{i,t}} \right) - v_{i,t} + [WH]_{i,t} \right) \quad (10)$$

Similar a la función de costo euclídeo, es convexa con respecto a \mathbf{W} o \mathbf{H} , pero no es convexa con respecto a \mathbf{W} y \mathbf{H} (cuando la función de costo se minimiza simultáneamente con respecto a ambos conjuntos de parámetros), por lo que la minimización de tal función de costo puede producir muchos mínimos locales [6].

Ahora mostraremos que la minimización de (10) es equivalente a la maximización de la probabilidad de Poisson:

$$L(W) = \prod_{i,t} \left(\frac{[WH]_{i,t}}{w_{i,t}} \exp(-w_{i,t}) \right) \quad (11)$$

Usando la función de Stirling $n! \approx n \log(n) - n$ para $n \gg 1$ la verosimilitud se convierte en:

$$\begin{aligned} -\log(L(W)) &= \sum_{i,t} \left([WH]_{i,t} - v_{i,t} + v_{i,t} \log \frac{v_{i,t}}{[WH]_{i,t}} \right. \\ &\quad \left. + \log \left(\sqrt{2\pi [WH]_{i,t}} \right) \right) \end{aligned} \quad (12)$$

Tomando en cuenta que $D(V||WH) = -\log(L(W))$. Entonces la función de costo se puede simplificar a

$$D(V||WH) = \sum_{i,t} ([WH]_{i,t} - v_{i,t} \log ([WH]_{i,t})) \quad (13)$$

Al minimizar la función de costo (10) sujeta a restricciones de no negatividad, podemos derivar la

siguiente regla de aprendizaje multiplicativo:

$$\begin{aligned} w_{i,j} &= w_{i,j} \frac{\sum_t h_{j,t} v_{i,t} / [WH]_{i,t}}{\sum_t H_{j,t}} \\ h_{j,t} &= h_{j,t} \frac{\sum_i w_{i,j} v_{i,t} / [WH]_{i,t}}{\sum_i w_{i,j}} \end{aligned} \quad (14)$$

La divergencia es invariante bajo estas actualizaciones si y solo si \mathbf{W} y \mathbf{H} están en un punto estacionario de la divergencia [6] [12].

3.3. Divergencia de Itakura-Saito

Divergencia de Itakura-Saito está dada por:

$$D(V||WH) = \sum_{i,t} \left(\frac{v_{i,t}}{[WH]_{i,t}} - \log \left(\frac{v_{i,t}}{[WH]_{i,t}} \right) - 1 \right) \quad (15)$$

Esta divergencia fue obtenida por Itakura y Saito [11] de la estimación de máxima verosimilitud de espectros de voz de corta duración bajo modelado auto-regresivo. Se presentó como “una medida de la bondad del ajuste entre dos espectro” y se hizo popular en la comunidad de habla durante los años setenta. En particular, fue elogiado por las buenas propiedades perceptivas de las señales reconstruidas a las que condujo [9].

También es invariante bajo escala, es decir, $d(\lambda x||\lambda y) = d(x||y)$, y es la única en la familia de β -divergencias que posee esta propiedad [10].

Para obtener las fórmulas de actualización multiplicativas lo haremos de manera general y luego haremos el caso especial de Itakura-Saito. Usando la ecuación (5), los gradientes de criterio para la ecuación (3) con respecto a \mathbf{W} y a \mathbf{H} se escriben simplemente:

$$\begin{aligned} \nabla_H D_\beta(V||WH) &= W^T \left((WH)^{-(\beta-2)} \cdot (WH - V) \right) \\ \nabla_W D_\beta(V||WH) &= \left((WH)^{-(\beta-2)} \cdot (WH - V) \right) H^T \end{aligned} \quad (16)$$

donde \cdot denota el producto Hadamard (entrada por entrada) y $A^{n \cdot}$ denota la matriz con entradas $[A]_{i,j}^n$. El enfoque de descenso por gradiente multiplicativo tomado en [12] equivale a actualizar cada parámetro por multiplicar su valor en la iteración anterior por la razón de la parte positiva y negativa de la derivada del criterio con respecto a este parámetro, es decir, $\theta \leftarrow \theta \cdot [\nabla f(\theta)]_- / [\nabla f(\theta)]_+$, donde $\nabla f(\theta) = [\nabla f(\theta)]_+ - [\nabla f(\theta)]_-$ y los sumandos sean no negativos [9]. Esto asegura la no negatividad en las actualizaciones de los parámetros, siempre que la inicialización tenga un valor no negativo. Un punto fijo θ^* del algoritmo implica una de las dos cosas, ya sea $\nabla f(\theta^*) = 0$ ó $\theta^* = 0$. Esto conduce

a las siguientes actualizaciones:

$$\begin{aligned} H &\leftarrow H \cdot \frac{W^T((WH)^{\beta-2} \cdot V)}{W^T(WH)^{\cdot(\beta-1)}} \\ W &\leftarrow W \cdot \frac{((WH)^{\beta-2} \cdot V)H^T}{(WH)^{\cdot(\beta-1)}H^T} \end{aligned} \quad (17)$$

donde $\frac{A}{B}$ denota la matriz $A \cdot B^{-1}$ [9]. En [13] mostró que $D(V||WH)$ no aumenta con las últimas actualizaciones para $\beta = 2$ (Distancia Euclídea) y $\beta = 1$ (Divergencia de Kullback-Leibler).

Para derivar las fórmulas de actualización para la divergencia de Itakura-Saito de la ecuación (17) se reemplaza $\beta = 0$ y se obtiene:

$$\begin{aligned} H &\leftarrow H \cdot \frac{W^T((WH)^{\cdot(-2)} \cdot V)}{W^T(WH)^{\cdot(-1)}} \\ W &\leftarrow W \cdot \frac{((WH)^{\cdot(-2)} \cdot V)H^T}{(WH)^{\cdot(-1)}H^T} \end{aligned} \quad (18)$$

4. MÉTODO DE DIRECCIONES ALTERNANTES DE MULTIPLICADORES

La división de variables es una técnica poderosa de optimización. La idea es dividir las ocurrencias múltiples de una sola variable en un problema como: $\min_x \sum_{i=1}^p f_i(x)$ en múltiples variables, con una restricción que une las variables:

$$\begin{aligned} \min_{x_1, \dots, x_p} \quad & \sum_{i=1}^p f_i(x_i) \\ \text{sujeto a: } \quad & \sum_{i=1}^p A_i x_i = b \end{aligned} \quad (19)$$

La ventaja es que $\sum_{i=1}^p f_i(x_i)$ se puede optimizar por coordenadas, aunque hay una restricción que une los problemas.

El método de direcciones alternantes de multiplicadores (ADMM) proporciona una forma elegante de manejar la restricción, mientras se mantiene la separabilidad del objetivo. ADMM optimiza alternativamente el Lagrangiano aumentado

$$\begin{aligned} \mathcal{L}_\rho(x_1, \dots, x_p, \lambda) = & \sum_{i=1}^p f_i(x_i) + \lambda^T \left(\sum_{i=1}^p A_i x_i - b \right) \\ & + \frac{\rho}{2} \left\| \sum_{i=1}^p A_i x_i - b \right\|_2^2 \end{aligned} \quad (20)$$

con respecto a cada x_i , seguido del “dual ascent” en λ [18], donde ρ es el parámetro de penalización de ADMM; notar que los primeros 2 términos de la ecuación (20)

forman el Lagrangiano estandar para el problema (19). La convergencia es conocida para el caso especial cuando $p = 2$ para f_1 y f_2 convexas.

4.1. Caso especial: $p = 2$

El problema es el siguiente:

$$\begin{aligned} \min_{x, z} \quad & f(x) + g(z) \\ \text{sujeto a: } \quad & Ax + Bz = c \end{aligned} \quad (21)$$

formamos el Lagrangiano aumentado del problema:

$$\begin{aligned} \mathcal{L}_\rho(x, z, \lambda) = & f(x) + g(z) + \lambda^T (Ax + Bz - c) \\ & + \frac{\rho}{2} \|Ax + Bz - c\|_2^2 \end{aligned} \quad (22)$$

ADMM consta de las iteraciones:

$$\begin{aligned} x^{k+1} &= \arg \min_x \mathcal{L}_\rho(x, z^k, \lambda^k) \\ z^{k+1} &= \arg \min_z \mathcal{L}_\rho(x^{k+1}, z, \lambda^k) \\ \lambda^{k+1} &= \lambda^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \end{aligned} \quad (23)$$

Consiste en actualizar alternativamente cada variable primal y después la variable dual [3].

4.2. ADMM para NMF

Hay algunas razones para usar ADMM en NMF:

- (i) Es simple de minimizar $D_\beta(V||X)$ con respecto a X , no es simple de minimizar $D_\beta(V||WH)$ con respecto a W ó H . En un contexto ADMM, una división natural sería minimizar $D_\beta(V||X)$ con la restricción $X = WH$.
- (ii) Las restricciones de no negatividad en W y H complican la optimización sobre W y H . Podemos introducir nuevas variables W_+ y H_+ a las que se aplican las restricciones de no negatividad, con las restricciones $W = W_+$ y $H = H_+$ [18]

El problema NMF en la ecuación (6) puede ser reescrito como:

$$\begin{aligned} \min_{W, H} \quad & D_\beta(V||X) \\ \text{sujeto a: } \quad & X = WH \\ & W = W_+, H = H_+ \\ & W_+ \geq 0, H_+ \geq 0 \end{aligned} \quad (24)$$

La notación anterior implica un Lagrangiano aumentado consta de ocho variables, cinco primarias y tres duales, aunque desde la perspectiva de ADMM, esta es solo una optimización de tres bloques: W , H y (X, W_+, H_+) . Esto se debe a que el objetivo se divide

en función de \mathbf{X} , \mathbf{W}_+ y \mathbf{H}_+ , por lo que optimizarlos por separado equivale a optimizarlos conjuntamente:

$$\begin{aligned} \mathcal{L}_\rho(X, W, H, W_+, H_+, \alpha_X, \alpha_W, \alpha_H) = \\ D_\beta(V||X) + \langle \alpha_X, X - WH \rangle + \frac{\rho}{2} \|X - WH\|_F^2 \\ + \langle \alpha_W, W - W_+ \rangle + \frac{\rho}{2} \|W - W_+\|_F^2 \\ + \langle \alpha_H, H - H_+ \rangle + \frac{\rho}{2} \|H - H_+\|_F^2 \end{aligned} \quad (25)$$

Donde $\langle ., . \rangle$ es el producto punto y ρ es la penalización de ADMM. Las actualizaciones optimizan alternativamente \mathcal{L}_ρ con respecto a cada una de las cinco variables primarias, seguido de un ascenso del gradiente en cada una de las tres variables duales. Esto se resume en el Algoritmo 1.

Algorithm 1 ADMM para NMF con la β -divergencia

```

input  $V$ : Matriz de Datos
output  $W_+, H_+$ 
1: initialize  $X, W, H, W_+, H_+, \alpha_X, \alpha_W, \alpha_H, \rho$ 
2: while convergencia do
3:    $W^T \leftarrow (HH^T + I) \backslash (HX^T + W_+^T + \frac{1}{\rho} (H\alpha_X^T \alpha_W^T))$ 
4:    $H \leftarrow (W^T W + I) \backslash (W^T X + H_+ + \frac{1}{\rho} (W^T \alpha_X \alpha_W))$ 
5:    $X \leftarrow \underset{X \geq 0}{\operatorname{argmin}} D_\beta(V||X) + \langle \alpha_X, X \rangle + \frac{\rho}{2} \|X - WH\|_F^2$ 
6:    $W_+ \leftarrow \max(W + \frac{1}{\rho} \alpha_W, 0)$ 
7:    $H_+ \leftarrow \max(H + \frac{1}{\rho} \alpha_H, 0)$ 
8:    $\alpha_X \leftarrow \alpha_X + \rho(X - WH)$ 
9:    $\alpha_H \leftarrow \alpha_H + \rho(H - H_+)$ 
10:   $\alpha_W \leftarrow \alpha_W + \rho(W - W_+)$ 
11: end while

```

En las actualizaciones para \mathbf{W} y \mathbf{H} en el Algoritmo 1, hemos utilizado la notación $A \backslash b$ para denotar la solución del problema de mínimos cuadrados $\arg \min_x \|Ax - b\|_2$.

Dado que las matrices A en estas actualizaciones son cuadradas y no singulares, $A \backslash b = A^{-1}b$. Aunque los problemas de mínimos cuadrados pueden ser inestables en general, la adición de la matriz de identidad I en estos casos estabiliza el problema [18].

La única actualización que no se proporciona en forma cerrada arriba es la de \mathbf{X} , que reafirmamos aquí por conveniencia:

$$X \leftarrow \underset{X \geq 0}{\arg \min} D_\beta(V||X) + \langle \alpha_X, X \rangle + \frac{\rho}{2} \|X - WH\|_F^2 \quad (26)$$

Tener en cuenta que esta es la única actualización que depende de β . Como veremos, la ecuación (26) se puede resolver en forma cerrada en los tres casos más importantes $\beta = 0, 1, 2$. En general, la ecuación (26) se puede resolver de manera eficiente utilizando el método

de Newton. Las actualizaciones para $\beta = 2$ se pueden derivar de manera similar, aunque en el caso euclíadiano, la división de \mathbf{X} y \mathbf{WH} es innecesaria. El algoritmo para $\beta = 2$ sin esta división se puede encontrar en la Sección 3.1.

Para la divergencia de Kullback-Leibler $\beta = 1$, la ecuación (26) está dada por:

$$X \leftarrow \frac{(\rho WH - \alpha_X - 1) \sqrt{(\rho WH - \alpha_X - 1)^2 + 4\rho V}}{2\rho} \quad (27)$$

donde todas las operaciones son por elementos. Para la demostración es, sustituyendo la expresión para D_β , $\beta = 1$ en la ecuación (26) e igualando a cero, obtenemos la condición:

$$-\frac{v_{i,t}}{x_{i,t}} + 1 + [\alpha_X]_{i,t} + \rho(x_{i,t} - [WH]_{i,t}) = 0 \quad (28)$$

Multiplicando por $x_{i,t}$, obtenemos una ecuación cuadrática; aplicando la fórmula cuadrática, obtenemos una raíz positiva y una negativa. La raíz positiva es la ecuación (27).

Para la divergencia Itakura-Saito ($\beta = 0$), la ecuación (26) viene dada por la siguiente serie de actualizaciones:

$$\begin{aligned} A &\leftarrow \alpha_X / \rho - WH \\ B_{i,t} &\leftarrow 1/(3\rho) - A_{i,t}^2/9 \\ C_{i,t} &\leftarrow -A_{i,t}^3/27 + A_{i,t}/(6\rho) + V_{i,t}/(2\rho) \\ D_{i,t} &\leftarrow B_{i,t}^3 + C_{i,t}^2 \\ Y_{i,t} &\leftarrow \begin{cases} (C_{i,t} + \sqrt{D_{i,t}})^{1/3} + (C_{i,t} - \sqrt{D_{i,t}})^{1/3}, & D_{i,t} \geq 0 \\ 2\sqrt{-B_{i,t}} \cos\left(\frac{1}{3} \cos^{-1} \frac{C_{i,t}}{\sqrt{-B_{i,t}^3}}\right), & D_{i,t} \leq 0 \end{cases} \\ X_{i,t} &\leftarrow Y_{i,t} - A_{i,t}/3 \end{aligned} \quad (29)$$

La demostración se encuentra en [18].

5. IMPLEMENTACIÓN DE NMF EN PYTHON USANDO SCIKIT LEARN

La implementación de python es resolver el problema de la ecuación (6) que es encontrar una descomposición de datos \mathbf{V} en dos matrices \mathbf{W} y \mathbf{H} de elementos no negativos, optimizando la distancia entre la matriz \mathbf{V} y \mathbf{WH} . La función de distancia es la ecuación (3) para cada una de $\beta = \{0, 1, 2\}$.

L1 y L2 prior pueden ser añadidos a la “loss function” (función de distancia) para regularizar el modelo. La L2 prior usa la norma de Frobenius, mientras que L1 prior usa la norma L1 por elementos. Usando la penalización de ElasticNet [17] [22] (que es una combinación convexa

de la penalización de Lasso y Ridge), controlamos la combinación de L1 y L2 con el parámetro `l1_ratio` (ρ), y la intensidad de la regularización con el parámetro `alpha` (α). Entonces los términos “priors” son:

$$\begin{aligned} \alpha\rho\|W\|_1 + \frac{\alpha(1-\rho)}{2}\|W\|_{\text{Fro}}^2 + \\ \alpha\rho\|H\|_1 + \frac{\alpha(1-\rho)}{2}\|H\|_{\text{Fro}}^2 \end{aligned} \quad (30)$$

y la función objetivo regularizada es

$$\begin{aligned} D_\beta(V\|WH) + \alpha\rho\|W\|_1 + \frac{\alpha(1-\rho)}{2}\|W\|_{\text{Fro}}^2 + \\ \alpha\rho\|H\|_1 + \frac{\alpha(1-\rho)}{2}\|H\|_{\text{Fro}}^2 \end{aligned} \quad (31)$$

Scikit learn regulariza tanto \mathbf{W} como \mathbf{H} por defecto. El parámetro `regularization` permite un control más preciso, con lo cual solo \mathbf{W} , solo \mathbf{H} , o ambos se pueden regularizar.

La función en Python implementa dos solucionadores `solver`, usando “Coordinate Decent” `cd` [5] y Multiplicative Update `mu` [10]. El solucionador `mu` puede optimizar todas las divergencias beta, incluida, por supuesto, la norma de Frobenius ($\beta = 2$), la Divergencia Kullback-Leibler ($\beta = 1$) y la divergencia Itakura-Saito ($\beta = 0$). Con $\beta \in (1; 2)$, el solucionador `mu` es significativamente más rápido que para otros valores de β . Con un β negativo (o 0, es decir, la divergencia de Itakura-Saito), la matriz de entrada no puede contener valores cero.

También está el atributo `init` que determina el método de inicialización aplicado. *Scikit learn* implementa el método de descomposición de valores singulares dobles no negativos. **NNDSVD** [2] se basa en dos procesos de SVD (descomposición de valores singulares por sus siglas en inglés), uno que se aproxima a la matriz de datos, y el otro que se aproxima a las secciones positivas de los factores de **SVD** parcial resultantes utilizando una propiedad algebraica de las matrices de rango unitario. El algoritmo básico **NNDSVD** se ajusta mejor a la factorización dispersa (la mayoría de los datos son ceros). Sus variantes **NNDSVDA** (en la que todos los ceros se igualan a la media de todos los elementos de los datos) y **NNDSVDar** (en la que los ceros se establecen en perturbaciones aleatorias menores que la media de los datos dividida por 100) se recomiendan en el denso caso [1].

NMF se utiliza mejor con el método `fit_transform`, que devuelve la matriz \mathbf{W} . La matriz \mathbf{H} se almacena en el modelo ajustado en el atributo `components_`; el método `transform` descompondrá una nueva matriz basada en estos componentes almacenados, para más detalles ver [15] [1].

6. APLICACIONES DE NMF EN TEXTOS

La descripción anterior de NMF se ha especializado para imágenes [13] [6]. Ilustramos esta versatilidad aplicando NMF al análisis semántico de documentos de texto. Para esta aplicación, un corpus de documentos de interés se someten preprocesamiento de palabras. Luego, para cada documento se pueden construir 2 cosas, una matriz \mathbf{V} donde $v_{i,t}$ es el número de veces que la i -ésima palabra del vocabulario aparece en el t -ésimo documento (“bag of words”), ó $\mathbf{V} = [\mathbf{v}_{i,t}]$ que se construye a partir de vectores de frecuencia de términos ponderados, es decir

$$v_{i,t} = F_{i,t} \log \left(\frac{d}{D_i} \right) \quad (32)$$

donde $F_{i,t}$ es la frecuencia de aparición del i -ésimo término en el t -ésimo documento, y D_i es el número de documentos que contienen el i -ésimo término. Las entradas de \mathbf{V} son siempre no negativas e iguales a cero cuando el i -ésimo término no aparece en el t -ésimo documento o aparece en todos los documentos [6]. Este se le llama Tf-idf (del inglés Term frequency–Inverse document frequency) (ver [4] [16]).

El objetivo es factorizar la matriz \mathbf{V} en la matriz de bases no negativa \mathbf{W} y la matriz del documento de temas no negativos \mathbf{H} , donde el número de columnas de \mathbf{W} o el números la filas de \mathbf{H} denota el número de temas. La posición del valor máximo en cada columna-vector en \mathbf{W} nos informa a qué tema se puede clasificar un documento determinado. Un ejemplo ilustrativo del agrupamiento término-documento se muestra en la Figura 1, donde la matriz Palabras-Textos \mathbf{V} se factoriza en la matriz Tópicos-Textos (Bases) \mathbf{W} y la matriz Palabras-Textos \mathbf{H} . Las columnas de \mathbf{W} se refieren a los centros del “cluster” (cuales son los grupos), y las columnas en \mathbf{H} están asociadas con los indicadores de “cluster” (cuáles documentos pertenecen a cada grupo) [6].

Estas herramientas se aplicarán a los siguientes corpora: Corpus de categorías, tweets de Elon Musk y Bill Gates y a las transcripciones de las mañaneras de AMLO. También se hará **clusterización** al Corpus de categorías para obtener estas mismas.

6.1. Ejemplo de textos de categorías

En este ejemplo ya se saben los tópicos de cada texto dentro del corpus, por el cual se analizará si este algoritmo NMF puede clasificar cada documento con los mismo tópicos. El corpus cuenta con 400 textos y 8 categorías que son: coches, hoteles, lavadoras, libros, móviles, música, ordenadores y películas. También se hizo procesamiento de palabras quitando “stop words”, número, acentos y puntuación, tomando solo 10,000 palabras. Por lo nuestra matriz \mathbf{V} es de tamaño 400×10000 , matriz \mathbf{W} es de

tamaño 400×8 y V es de tamaño 8×10000 , es 8 porque ya sabemos el número de tópicos.

En la Figura 3 se puede observar cada uno de los 8 tópicos utilizando la distancia de Frobenius y usando Tf-idf y se obtiene que clasifica los 8 tópicos de manera perfecta. Podemos relacionar cada una categorías a cada tópico de manera que no haya confusión. **Tópico 1:** películas, **Tópico 2:** lavadoras, **Tópico 3:** móviles, **Tópico 4:** hoteles, **Tópico 5:** coches, **Tópico 6:** música, **Tópico 7:** ordenadores y **Tópico 8:** libros.

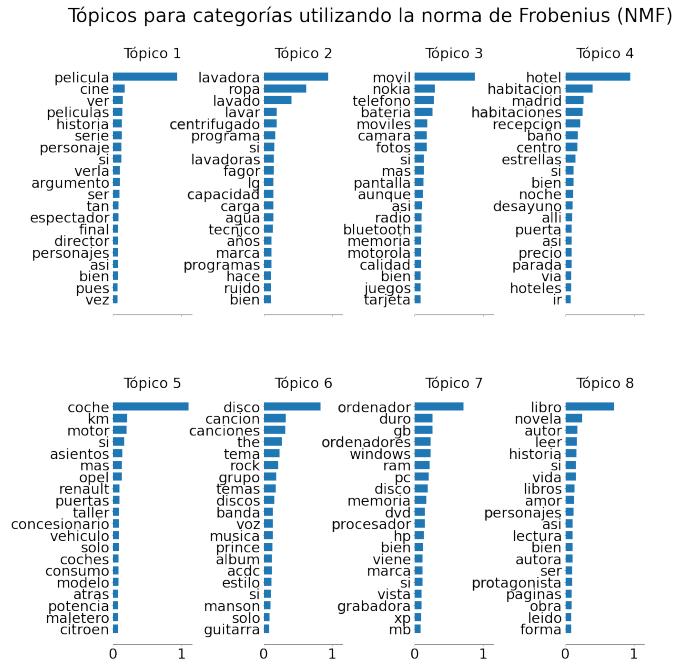


Figure 3. Extracción de categorías con las palabras más representativas para cada uno utilizando distancia de Frobenius y usando Tf-idf

De la Figura A3 (ver Apéndice A.1.1) se puede observar que se realizó lo mismo pero ahora utilizando “bag of words” y se puede notar que no clasifica bien los **Tópico 2** y **Tópico 6** y no podemos establecer algunas de las categorías ya mencionadas, por lo cual es una gran diferencia usar Tf-idf y “bag of words”.

En la Figura A1 se encuentra la extracción de tópicos utilizando la distancia de K-L (respecto al Frobenius no cambia considerablemente) y la Figura A2 que utiliza LDA no realiza buenas representaciones en tópicos, solo logró detectar 4 tópicos de los 8 tópicos, vemos que hay una mejoría con NMF que con LDA, en los dos casos anteriores se usó Tf-idf. En la Figura A3, Figura A4 y la Figura A5 es lo mismo con el anterior, pero tomando la datos como la frecuencia de palabras (“Bag of Words”) para LDA sí logró detectar 7 tópicos a excepción de

la móviles. (ver apéndice A.1, sección A.1.1). También se analizó para el sentimiento pero no se llegó a nada concreto y por eso no se muestra aquí en el documento.

6.2. Ejemplo con Tweets de Elon Musk y Bill Gates

Este ejemplo se tomaron 2678 tweets de Elon Musk del año 2013 al 2017 [20] y 2087 tweets de Bill Gates del año 2011 al 2017 [19]. En este caso no tenemos los tópicos de los tweets, por lo que ahora se elegirá la cantidad de tópicos y se tomó 12 tópicos para cada uno, esto se eligió después de hacer varias pruebas con diferentes número de tópicos.

6.2.1. Tweets de Elon Musk

En la Figura 4 podemos observar 12 tópicos, los tópico que sí podemos relacionar con algo en particular son el **Tópico 2:** que dice sobre los carros de Tesla y la energía solar, **Tópico 3:** que habla de SpaceX y misiones, **Tópico 4:** se relaciona con el cambio climático, **Tópico 5:** se relaciona con agradecimientos a SpaceX, **Tópico 8:** habla con el piloto automático de los carros y **Tópico 12:** habla sobre su compañía The Boring Company. Los demás podríamos hacer la relación pero estos que se mencionaron se consideraron los más importantes. Esta abstracción de tópico tienen mucha relación con Elon Musk ya que es dueño de Tesla y SpaceX, The Boring Company.

En la Figura A6 está lo tópicos utilizando la distancia de Frobenius (respecto al K-L no cambia mucho) y la Figura A7 que utiliza LDA(Latent Dirichlet Allocation) que ahí vemos que hay una mejoría con NMF que con LDA. En la Figura A8, Figura A9 y la Figura A10 es lo mismo con el anterior, pero tomando la datos como la frecuencia de palabras (“Bag of Words”).(ver apéndice A.1, sección A.1.2)

6.2.2. Tweets de Bill Gates

Igual que con Elon Musk (sección 6.2.1) en la Figura 5 se tomaron 12 tópicos los tópico que sí podemos relacionar con algo en particular son el **Tópico 1:** que dice vacunas que puede salvar vidas, **Tópico 2:** se relaciona con libros, **Tópico 3:** se relaciona con lucha contra la polio, **Tópico 6:** se relaciona con el cambio climático y energía limpia , **Tópico 9:** se relaciona con una nueva pandemia y **Tópico 11:** se relaciona con la escuela. Los demás podríamos hacer la relación pero eso que se mencionaron se consideraron los más importantes. Esta abstracción de tópico tienen mucha relación con Bill Gates ya que siempre ha estado interesado en el cambio climático y enfermedades.

En la Figura A11 está lo tópicos utilizando la distancia de Frobenius (respecto al K-L no cambia mucho) y la Figura A12 que utiliza LDA(Latent Dirichlet Allocation) que ahí vemos que hay una mejoría con NMF que con

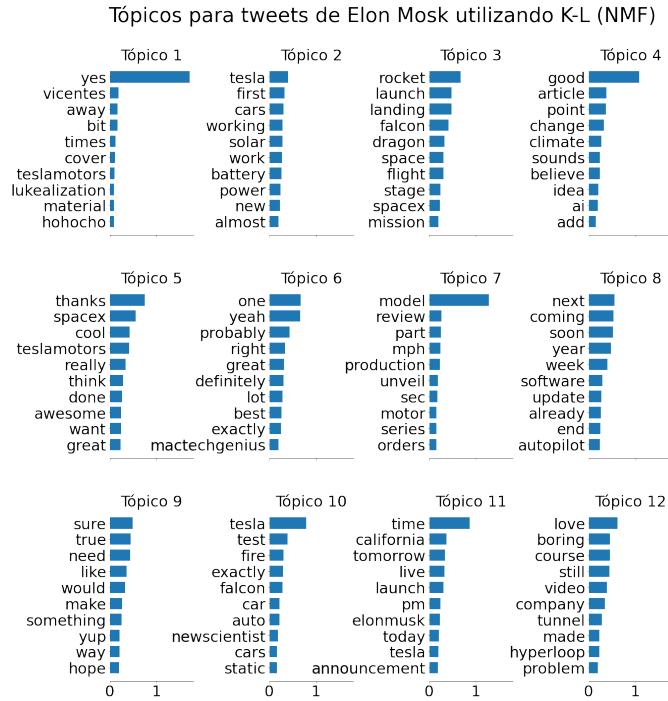


Figure 4. Extracción de tópicos con las palabras más representativas para cada uno utilizando la divergencia del Kullback-Leibler y usando Tf-idf de palabras para los tweets de Elon Musk

LDA. En la Figura A13, Figura A14 y la Figura A15 es lo mismo con el anterior, pero tomando los datos como la frecuencia de palabras (“Bag of Words”). (ver apéndice A.1, sección A.1.2)

6.3. Ejemplo de la transcripción de las mañanera de AMLO

Este ejemplo se tomaron las mañaneras de AMLO y las conferencias del Covid de Hugo Gatell desde Noviembre del 2018 hasta Mayo de 2021 y cuentan con 1557 textos [7]. En la Figura 6 se tomaron 20 tópicos, de los tópicos que sí podemos relacionar con algo en particular son el **Tópico 1**: se relaciona con Lopez Gatell y Salud, **Tópico 2**: se relaciona con Covid (general), **Tópico 4**: se relaciona con la mañanera, **Tópico 5**: se relaciona con programas de bienestar, **Tópico 6**: se relaciona con la vacunación, **Tópico 8**: se relaciona con la guardia nacional, **Tópico 11**: se relaciona con el tren maya, **Tópico 12**: se relaciona con Marcelo Ebrard, **Tópico 13**: se relaciona con datos de la pandemia de Covid, **Tópico 14**: se relaciona con pueblos indígenas, **Tópico 17**: se relaciona con salud y **Tópico 19**: que se relaciona con Oaxaca y siembra. Los demás podríamos hacer la relación pero eso que se mencionaron se consideraron los

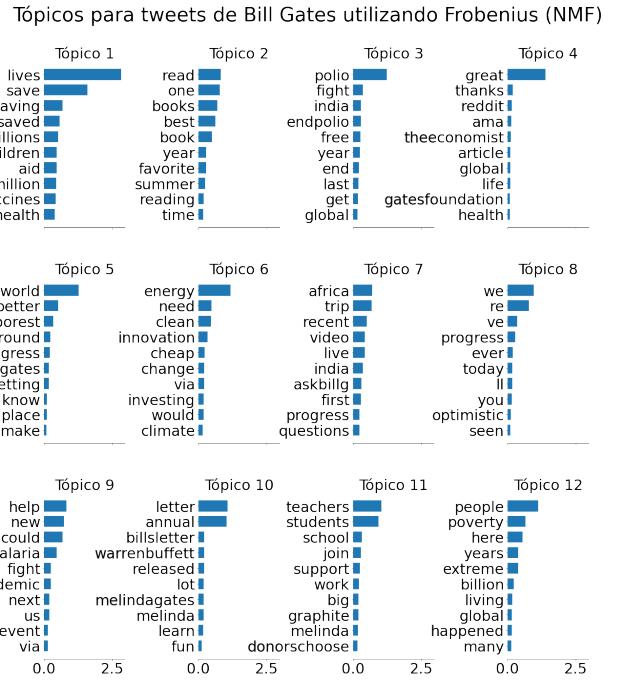


Figure 5. Extracción de tópicos con las palabras más representativas para cada uno utilizando la distancia de Frobenius y usando Tf-idf para palabras para los tweets de Bill Gates

más importantes. Esta abstracción de tópico tienen mucha relación con AMLO porque es el presidente de México y Gatell con la pandemia.

En la Figura A16 está los tópicos utilizando la distancia de Frobenius (respecto al K-L no cambia mucho) y la Figura A17 que utiliza LDA (Latent Dirichlet Allocation) que ahí vemos que hay una mejoría con NMF que con LDA. En la Figura A18, Figura A19 y Figura A20 es lo mismo con el anterior, pero tomando los datos como la frecuencia de palabras (“Bag of Words”). (ver apéndice A.1, sección A.1.3)

6.4. Clustering de Tópicos

Se tomaron los textos de la sección 6.1 para hacer clustering ya que con estos cuentan con etiquetado de los textos. También teniendo en cuenta que la matriz \mathbf{W} da los tópicos por textos, esto se puede ver en la Figura A21 (ver A.1.4). Obteniendo esto se tomó un tópico (Una columna de la matriz \mathbf{W}) y se obtuvieron todos los textos que fueran mayor a cierto valor muy pequeño, todo el conjunto se le asignó la categoría predominante, que se puede obtener de ver las palabras más importantes. Como resultado obtener la matriz de confusión de la Figura 7.

Se puede ver que clusteriza de manera casi perfecta las categorías a excepción de 6 elementos que los confunde ya

Tópicos para AMLO utilizando la divergencia de Kullback-Leiblers (NMF)

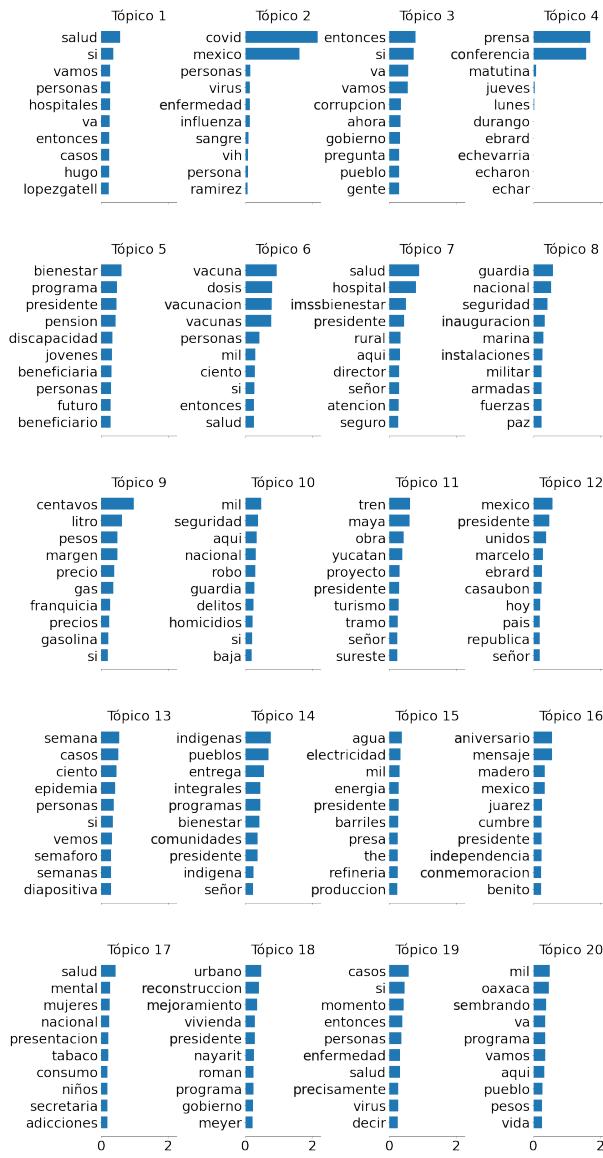


Figure 6. Extracción de tópicos de la mañana de AMLO con las palabras más representativas para cada uno utilizando la divergencia de Kullback-Leibler y usando usando Tf-idf de palabras para las mañana de AMLO

que son temas muy cercanos. Por lo que es buen método de clustering (ver [8] [21])

7. CONCLUSIONES

NMF es un problema ha sido eficiente en términos de clusterización, detección de tópicos, análisis de imágenes,

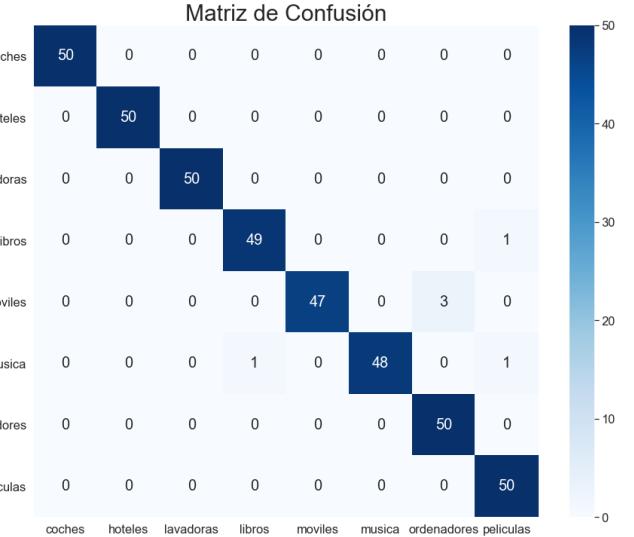


Figure 7. Matriz de Confusión para la clusterización de las categorías dadas en la sección A.1.1

análisis de sonido, entre otras cosas por su característica de la no negatividad y tener bastante interpretación de verlo como una de suma de partes. Hay una extensa gama de algoritmos con el cual se han resuelto este problema, aquí solo nos enfocamos en 5, que sería los algoritmos multiplicativos iterativos (uno para cada β -divergencia) y el ADMM, y también la implementación de python.

Los resultados de la aplicación a textos han sido eficientes para detectar tópicos de manera bien, para textos de categoría logró detectar los 8 tópicos de manera exacta solo cuando utilizando Tf-idt, pero utilizando “bag of word” solo logró detectar 6 categorías de los 8, por lo que Tf-idt tiene una ventaja sobre “bag of word”. LDA (Latent Dirichlet Allocation) no hace buenas representaciones en tópicos, solo logró detectar 4 tópicos de los 8 tópicos utilizando Tf-idt y utilizando “bag of word” sí logró detectar 7 tópicos a excepción de los móviles.

Para los tweets de Elon Musk y Bill Gates los 3 métodos los hace de manera bien a excepción de LDA utilizando Tf-idf para los tweets de Bill Gates. Y para la mañana de AMLO todos los métodos lo hacen de manera bien a excepción de LDA utilizando Tf-idf. Observamos que LDA no da buenos tópicos cuando se utiliza Tf-idf y ya da buenos con “bag of words”.

Cuando teníamos ya los tópicos establecidos como el caso de textos de categorías ya teníamos nuestra k ya definida, pero para los demás corpus se eligió un número de tópicos al azar por lo que faltaría hacer una métrica para obtener el número de tópicos óptimo.

La comparación de NMF con LDA, el NMF es mejor obteniendo tópicos y se puede ver en varias

Figuras, donde LDA mezcla algunos parámetros. Faltaría obtener buenos parámetros para estas aplicaciones a estos textos. También se pudo haber utilizado SVD pero por simplicidad se decidió utilizar LDA.

Para el ejemplo de la clusterización si detectó de manera casi perfecta cuáles textos pertenecen al mismo grupo, solo dejó 6 textos un poco confundido de lado que confundió por razón de podrían tener las mismas palabras. Por lo cual también puede ser un método de clustering

REFERENCES

- [1] 2.5. *Decomposing signals in components (matrix factorization problems)*. URL: <https://scikit-learn.org/stable/modules/decomposition.html#nmf>.
- [2] Christos Boutsidis and Efstratios Gallopoulos. “SVD based initialization: A head start for nonnegative matrix factorization”. In: *Pattern recognition* 41.4 (2008), pp. 1350–1362.
- [3] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.
- [4] Stefano Ceri et al. “An introduction to information retrieval”. In: *Web information retrieval*. Springer, 2013, pp. 3–11.
- [5] Andrzej Cichocki and Anh-Huy Phan. “Fast local algorithms for large scale nonnegative matrix and tensor factorizations”. In: *IEICE transactions on fundamentals of electronics, communications and computer sciences* 92.3 (2009), pp. 708–721.
- [6] Andrzej Cichocki et al. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.
- [7] *Conferencias Mañaneras: Versiones escritas de las conferencias de la presidencia de México*. URL: <https://www.kaggle.com/ioexception/mananeras>.
- [8] Chris Ding, Xiaofeng He, and Horst D Simon. “On the equivalence of nonnegative matrix factorization and spectral clustering”. In: *Proceedings of the 2005 SIAM international conference on data mining*. SIAM, 2005, pp. 606–610.
- [9] Cédric Févotte, Nancy Bertin, and Jean-Louis Durrieu. “Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis”. In: *Neural computation* 21.3 (2009), pp. 793–830.
- [10] Cédric Févotte and Jérôme Idier. “Algorithms for nonnegative matrix factorization with the β -divergence”. In: *Neural computation* 23.9 (2011), pp. 2421–2456.
- [11] Fumitada Itakura. “Analysis synthesis telephony based on the maximum likelihood method”. In: *The 6th international congress on acoustics, 1968*. 1968, pp. 280–292.
- [12] Daniel Lee and H. Sebastian Seung. “Algorithms for Non-negative Matrix Factorization”. In: *Advances in Neural Information Processing Systems* 13 (Feb. 2001). Ed. by T. Leen, T. Dietterich, and V. Tresp.
- [13] Daniel D Lee and H Sebastian Seung. “Learning the parts of objects by non-negative matrix factorization”. In: *Nature* 401.6755 (1999), pp. 788–791.
- [14] William H Press et al. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.
- [15] *sklearn.decomposition.NMF*. URL: <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html#sklearn-decomposition-nmf>.
- [16] *sklearn.feature_extraction.text.TfidfTransformer*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html#sklearn.feature_extraction.text.TfidfTransformer.
- [17] *sklearn.linear_model.ElasticNet*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html.
- [18] Dennis L Sun and Cedric Févotte. “Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence”. In: *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2014, pp. 6201–6205.
- [19] *Tweets de Bill Gates*. URL: https://raw.githubusercontent.com/JoaquinAmatRodrigo/Estadistica-con-R/master/datos/datos_tweets_BillGates.csv.
- [20] *Tweets de Elon Musk*. URL: https://raw.githubusercontent.com/JoaquinAmatRodrigo/Estadistica-con-R/master/datos/datos_tweets_ElonMusk.csv.
- [21] Ron Zass and Amnon Shashua. “A unifying approach to hard and probabilistic clustering”. In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. Vol. 1. IEEE, 2005, pp. 294–301.

- [22] Hui Zou and Trevor Hastie. “Regularization and variable selection via the elastic net”. In: *Journal of the royal statistical society: series B (statistical methodology)* 67.2 (2005), pp. 301–320.

APPENDIX

A.1. FIGURAS COMPLEMENTARIAS

A.1.1. Figuras para el ejemplo de categorías

Tópicos para categorías utilizando la divergencia Kullback-Leibler (NMF)

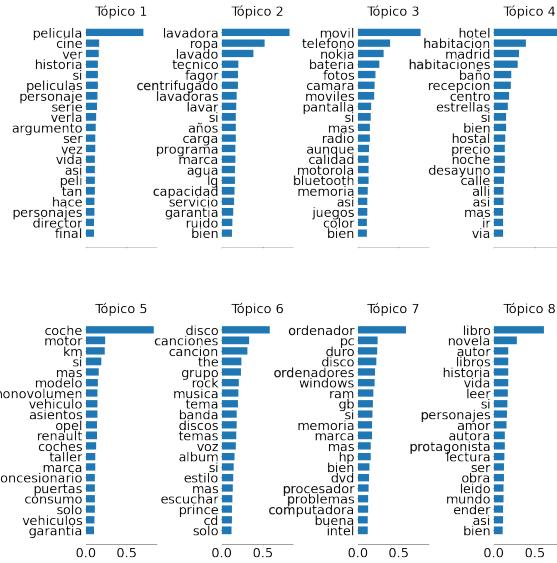


Figure A1. Extracción de categorías con las palabras más representativas para cada uno utilizando la la divergencia del Kullback-Leibler y usando usando Tf-idf

Tópicos para categorías utilizando la norma de Frobenius (NMF)

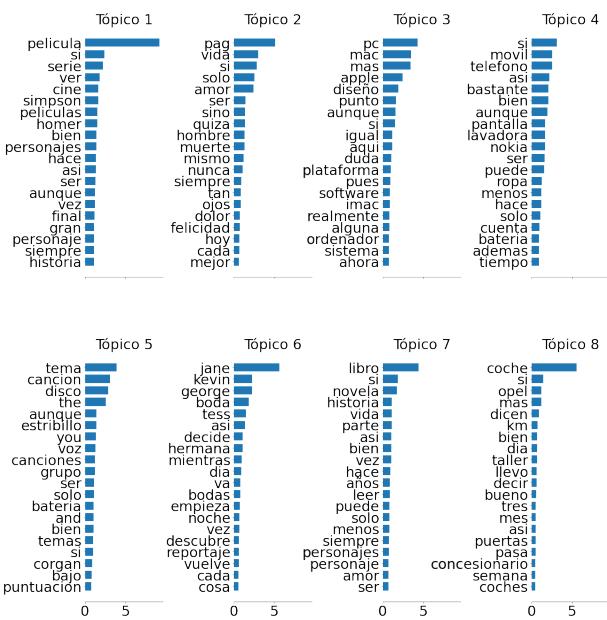


Figure A3. Extracción de categorías con las palabras más representativas para cada uno utilizando la distancia de Frobenius y usando “bag of words”

Tópicos utilizando LDA

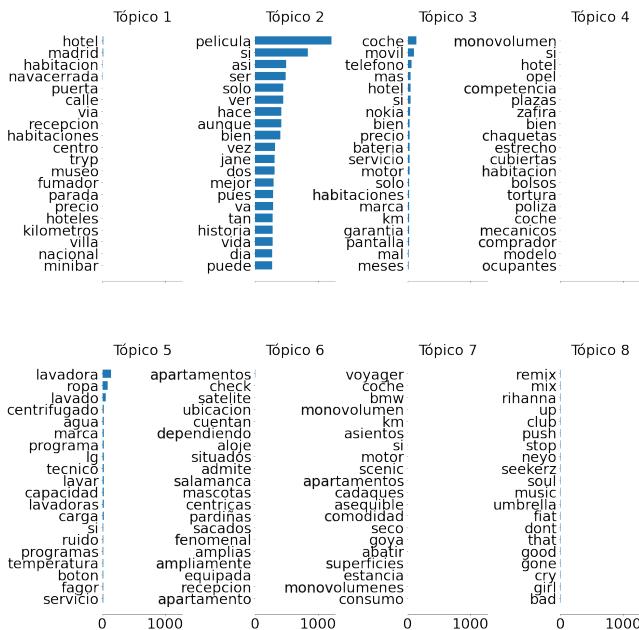


Figure A2. Extracción de categorías con las palabras más representativas para cada uno utilizado LDA (Latent Dirichlet Allocation) y usando Tf-idf palabras

Tópicos para categorías utilizando la divergencia Kullback-Leibler (NMF)

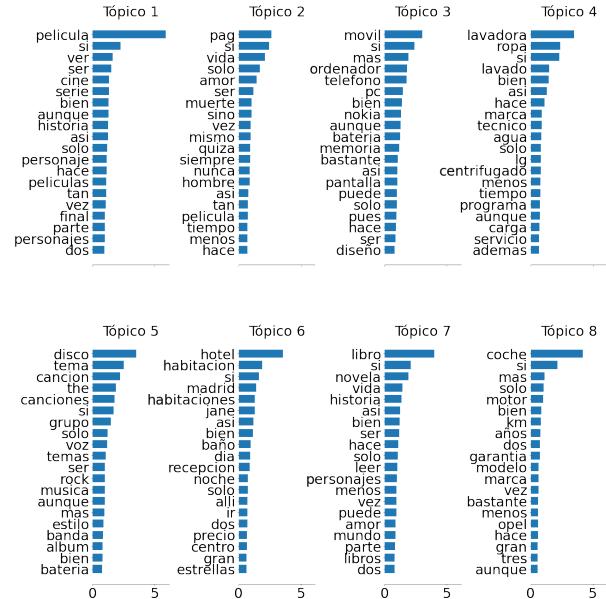


Figure A4. Extracción de categorías con las palabras más representativas para cada uno utilizado la divergencia del Kullback-Leibler y usando “bag of words”

Tópicos para categorías utilizando LDA

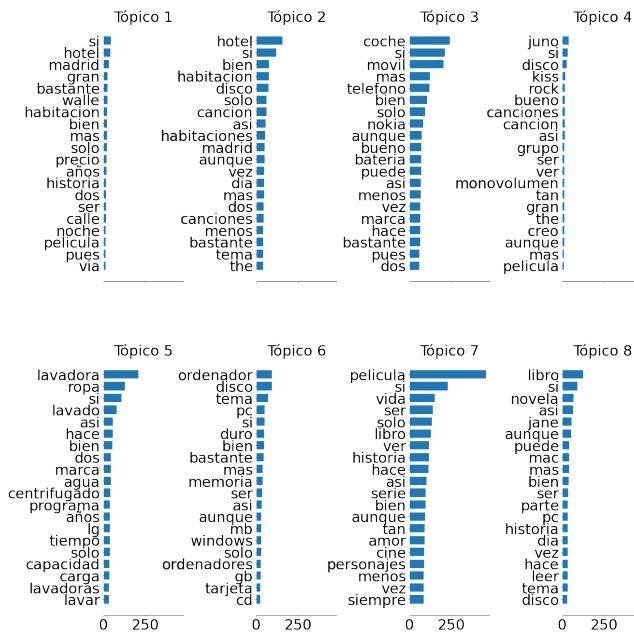


Figure A5. Extracción de categorías con las palabras más representativas para cada uno LDA(Latent Dirichlet Allocation) y usando “bag of words”

A.1.2. Figuras para el ejemplo de tweets

Tópicos para tweets de Elon Musk utilizando Frobenius (NMF)

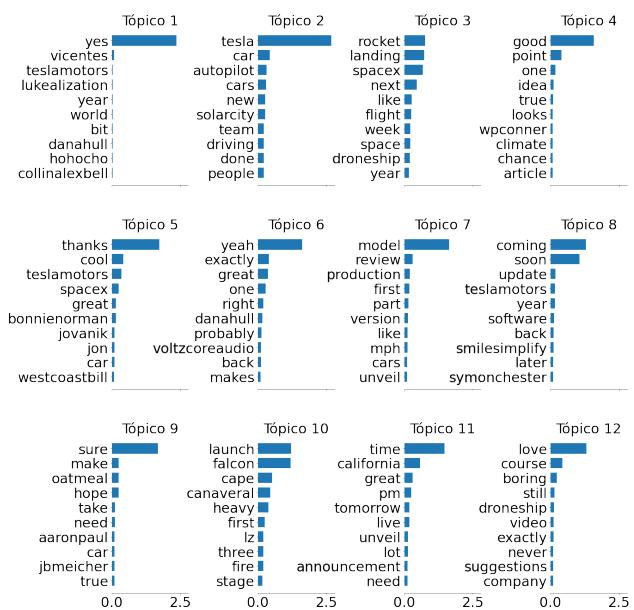


Figure A6. Extracción de tópicos de tweets de Elon Musk con las palabras más representativas para cada uno utilizado la divergencia del Kullback-Leibler y usando usando Tf-idf

Tópicos para tweets de Elon Mosk utilizando LDA

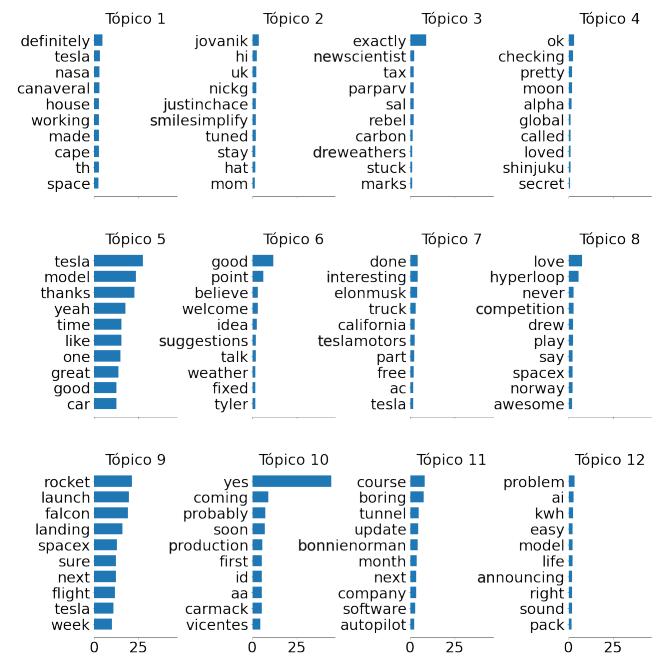


Figure A7. Extracción de tópicos de tweets de Elon Musk con las palabras más representativas para cada uno utilizado LDA (Latent Dirichlet Allocation) y usando Tf-idf

Tópicos para tweets de Elon Musk utilizando Frobenius (NMF)

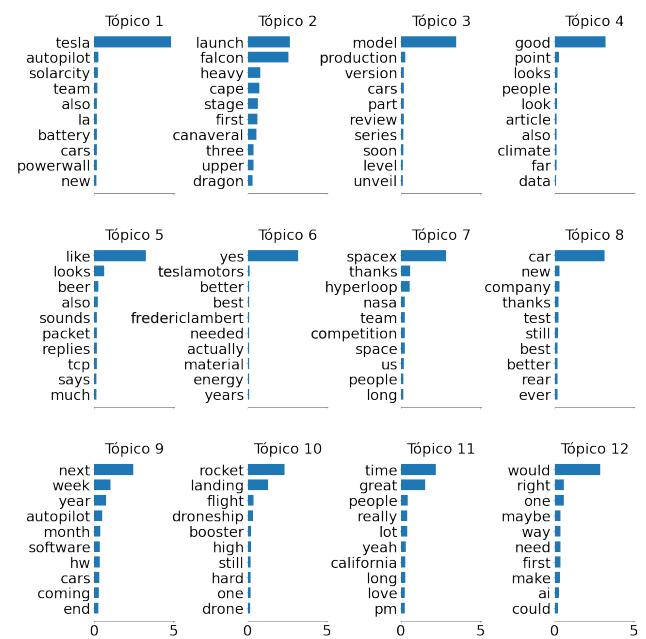


Figure A8. Extracción de tópicos de tweets de Elon Musk con las palabras más representativas para cada uno utilizado la distancia de Frobenius y usando usando “bag of word”

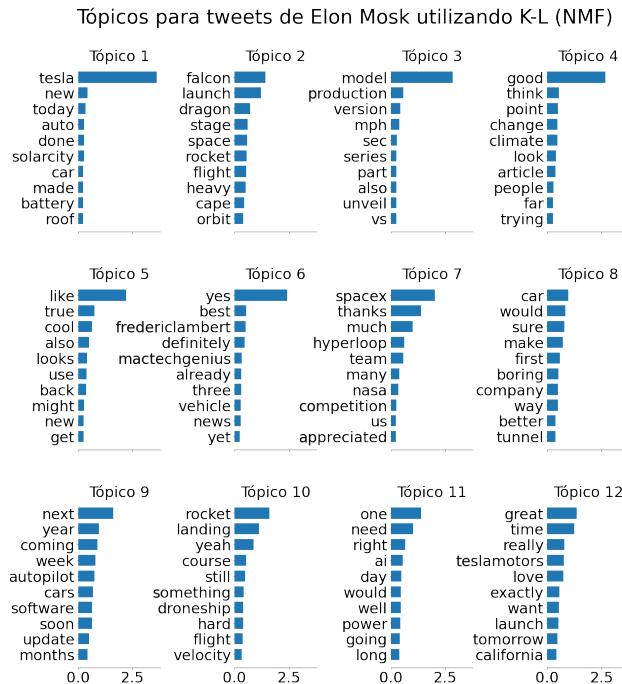


Figure A9. Extracción de tópicos de tweets de Elon Musk con las palabras más representativas para cada uno utilizado la la divergencia del Kullback-Leibler y usando usando “bag of word”

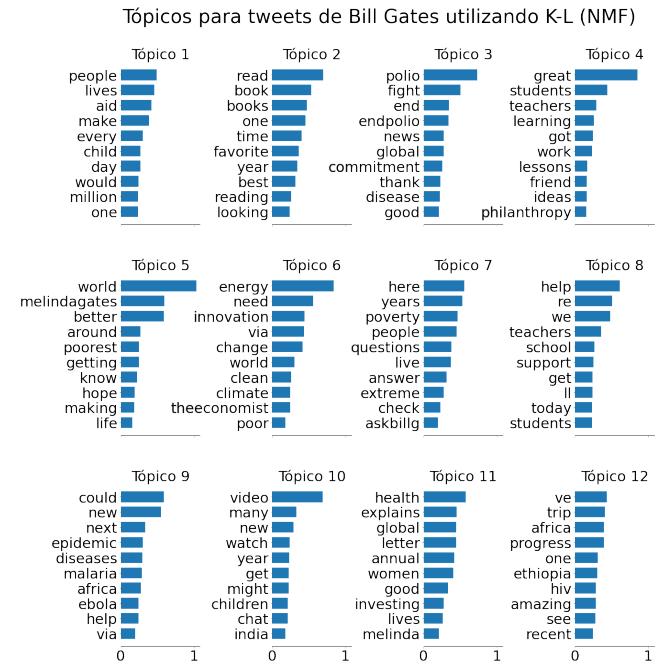


Figure A11. Extracción de tópicos de tweets de Bill Gates con las palabras más representativas para cada uno utilizado la la divergencia del Kullback-Leibler y usando Tf-idf

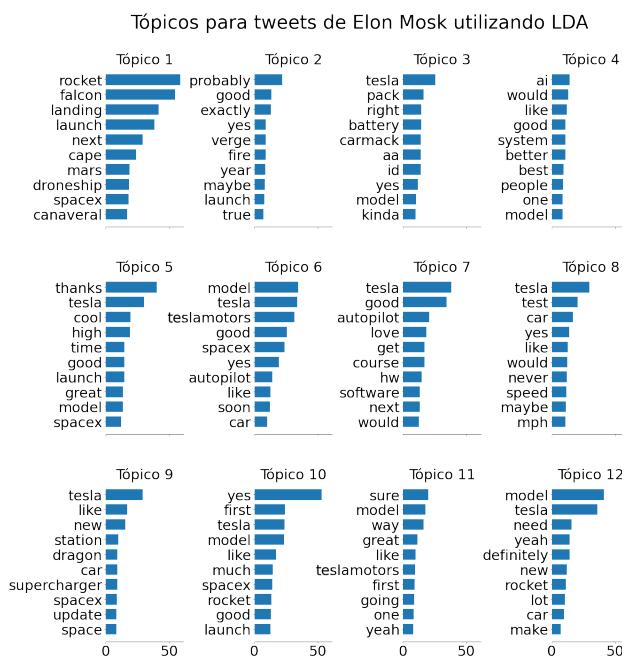


Figure A10. Extracción de tópicos de tweets de Elon Musk con las palabras más representativas para cada uno utilizado LDA (Latent Dirichlet Allocation) y usando usando “bag of word” palabras

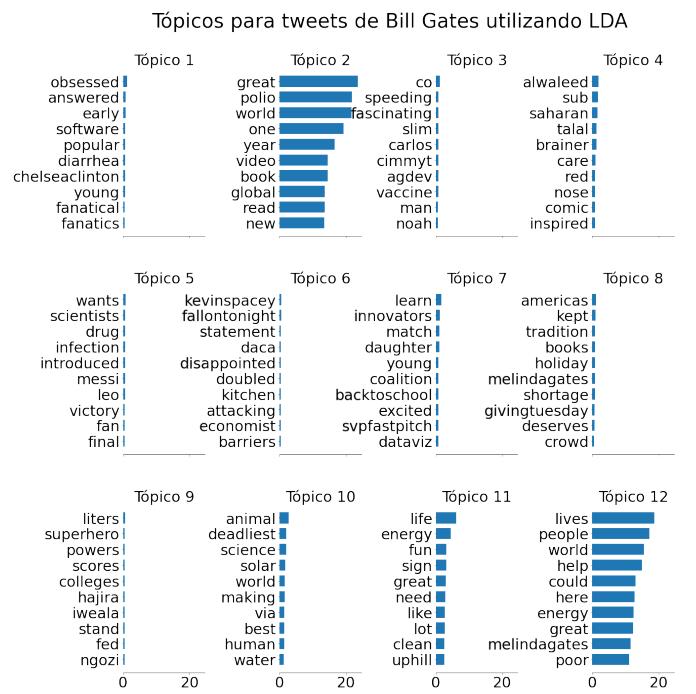


Figure A12. Extracción de tópicos de tweets de Bill Gates con las palabras más representativas para cada uno utilizado LDA (Latent Dirichlet Allocation) y usando Tf-idf

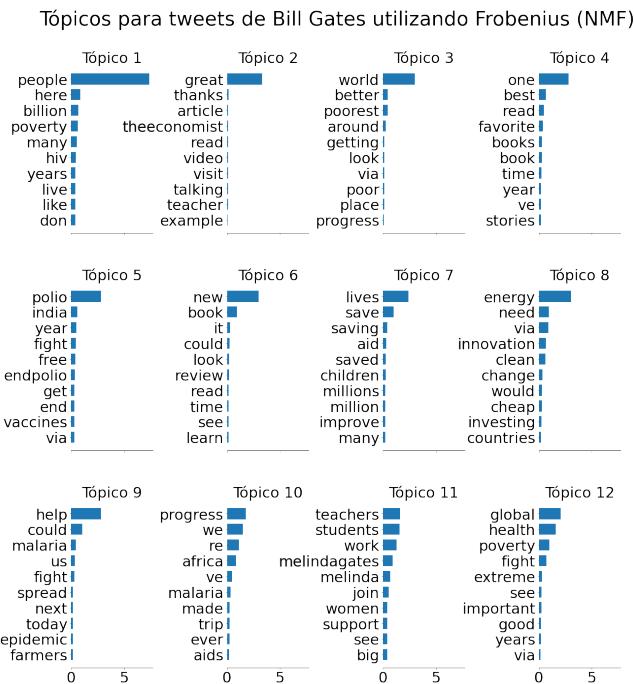


Figure A13. Extracción de tópicos de tweets de Bill Gates con las palabras más representativas para cada uno utilizando la distancia de Frobenius y usando “bag of words”

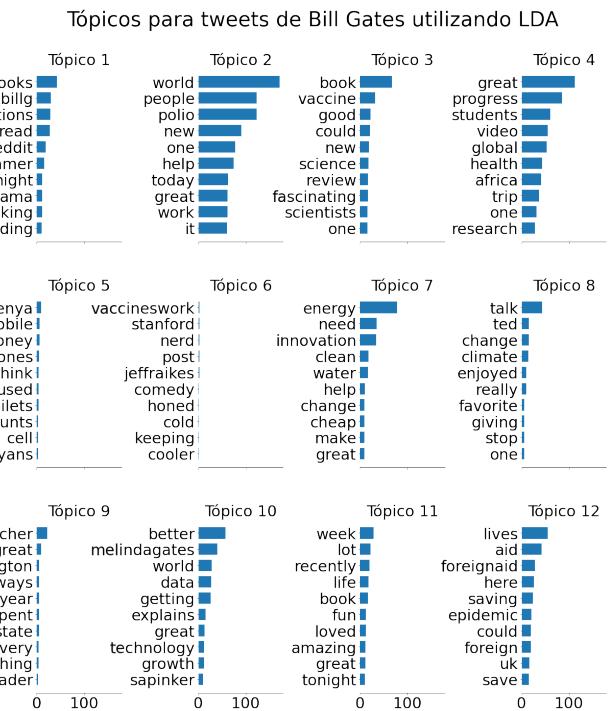


Figure A15. Extracción de tópicos de tweets de Bill Gates con las palabras más representativas para cada uno utilizado LDA (Latent Dirichlet Allocation) y usando “bag of words”

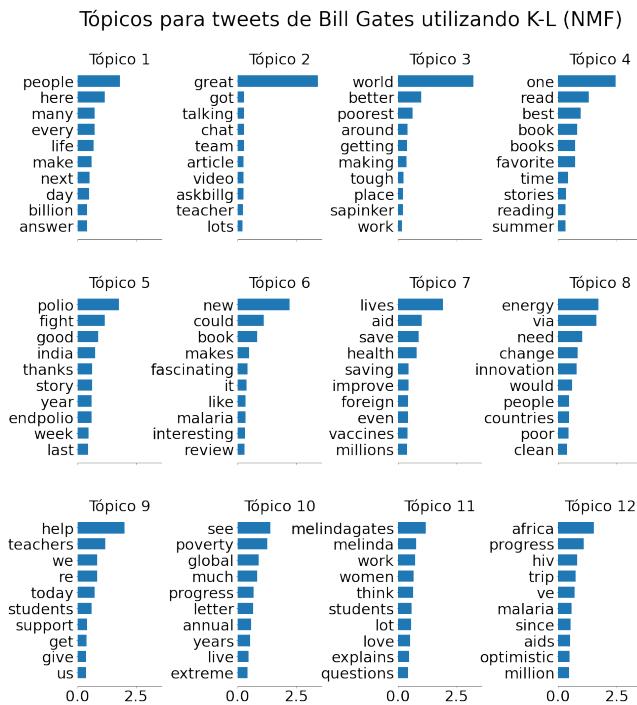
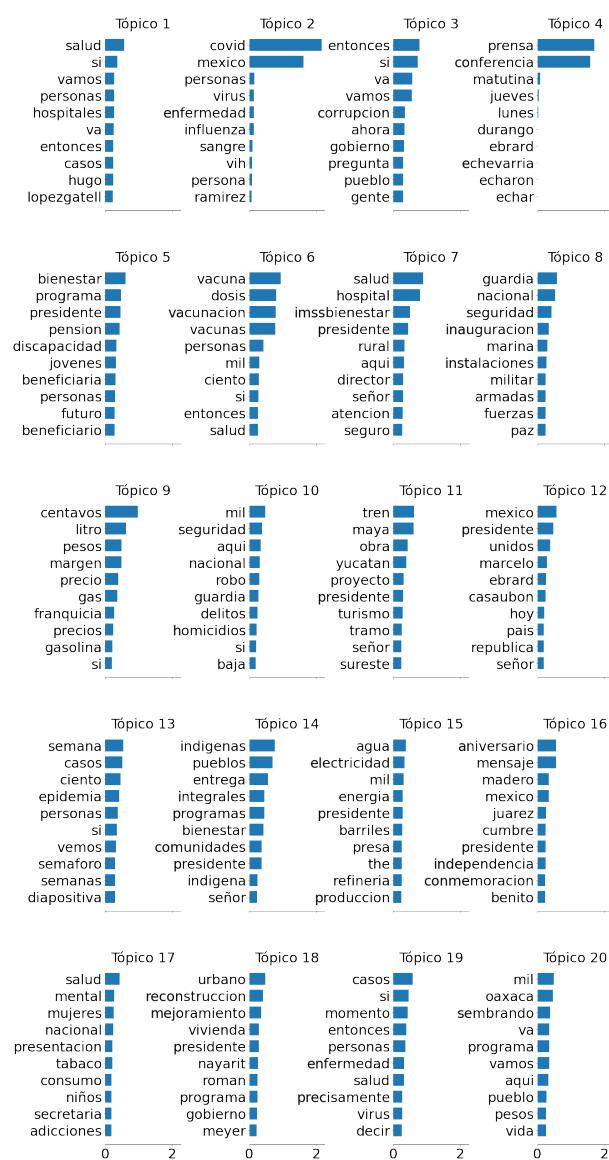


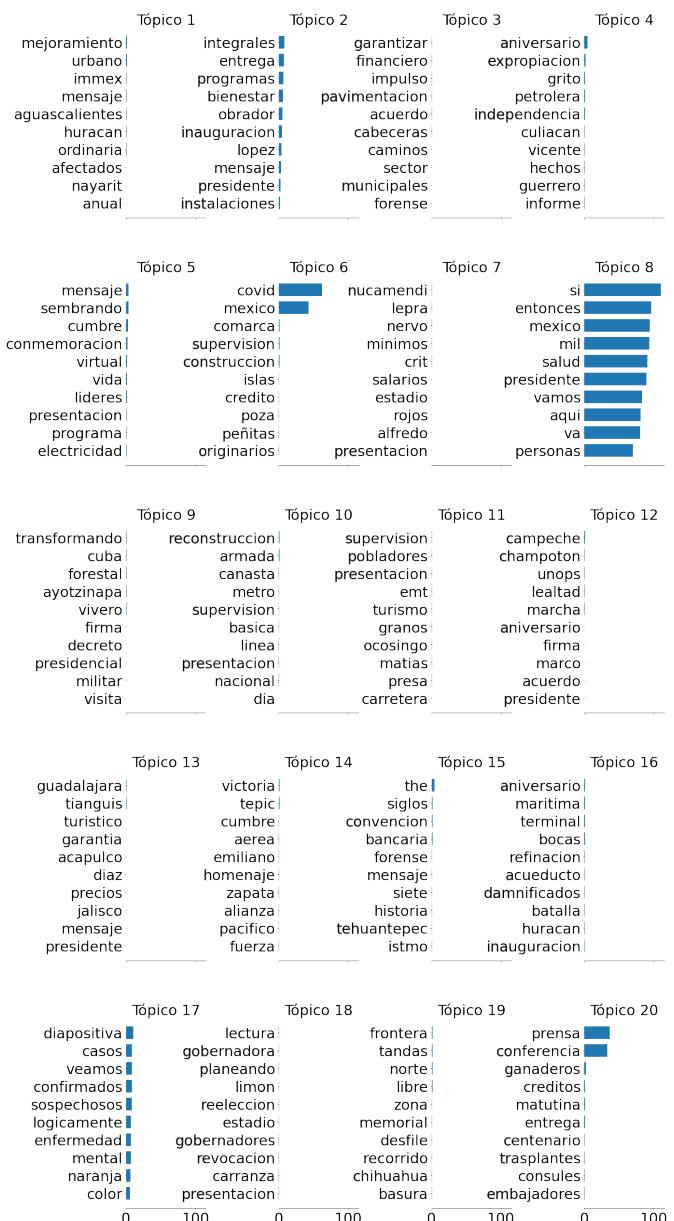
Figure A14. Extracción de tópicos de tweets de Bill Gates con las palabras más representativas para cada uno utilizando la divergencia del Kullback-Leibler y usando “bag of words”

A.1.3. Figuras para el ejemplo de AMLO

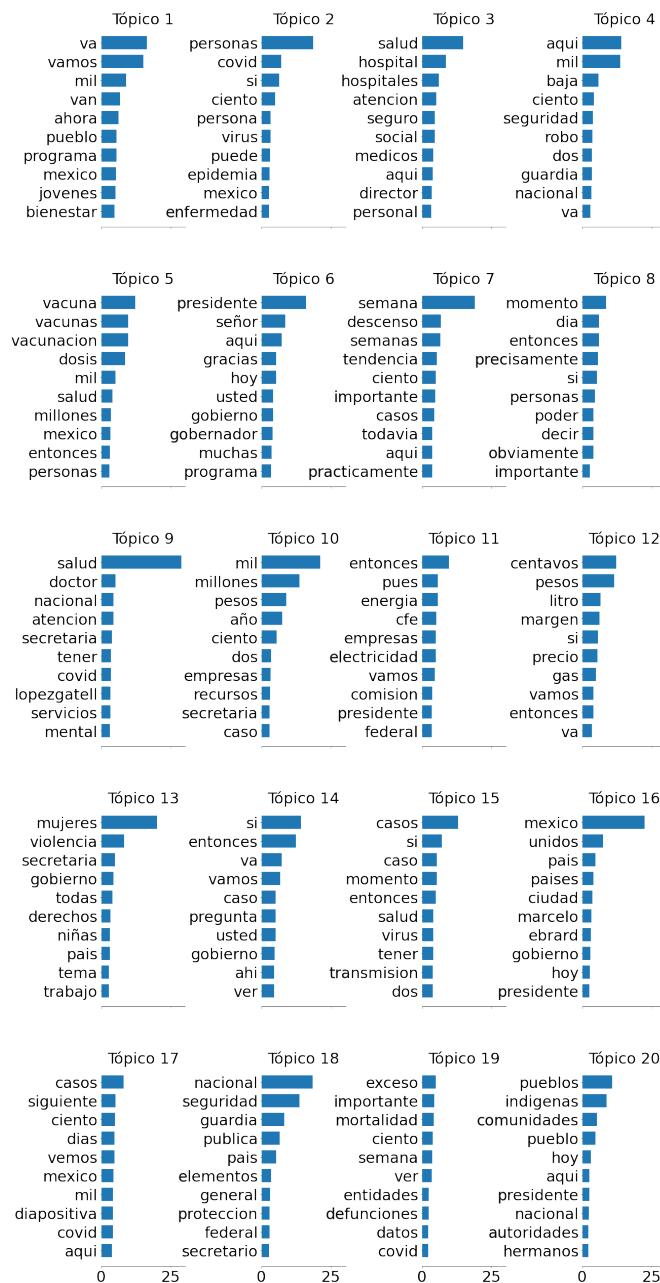
Tópicos para AMLO utilizando la divergencia de Kullback-Leiblers (NMF)



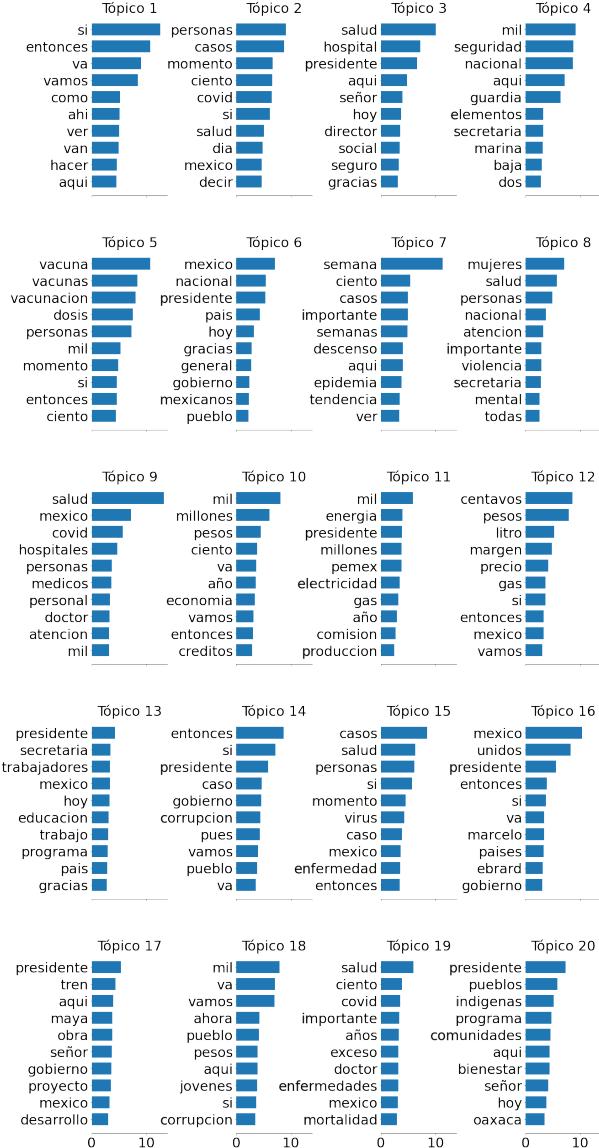
Tópicos para AMLO utilizando LDA



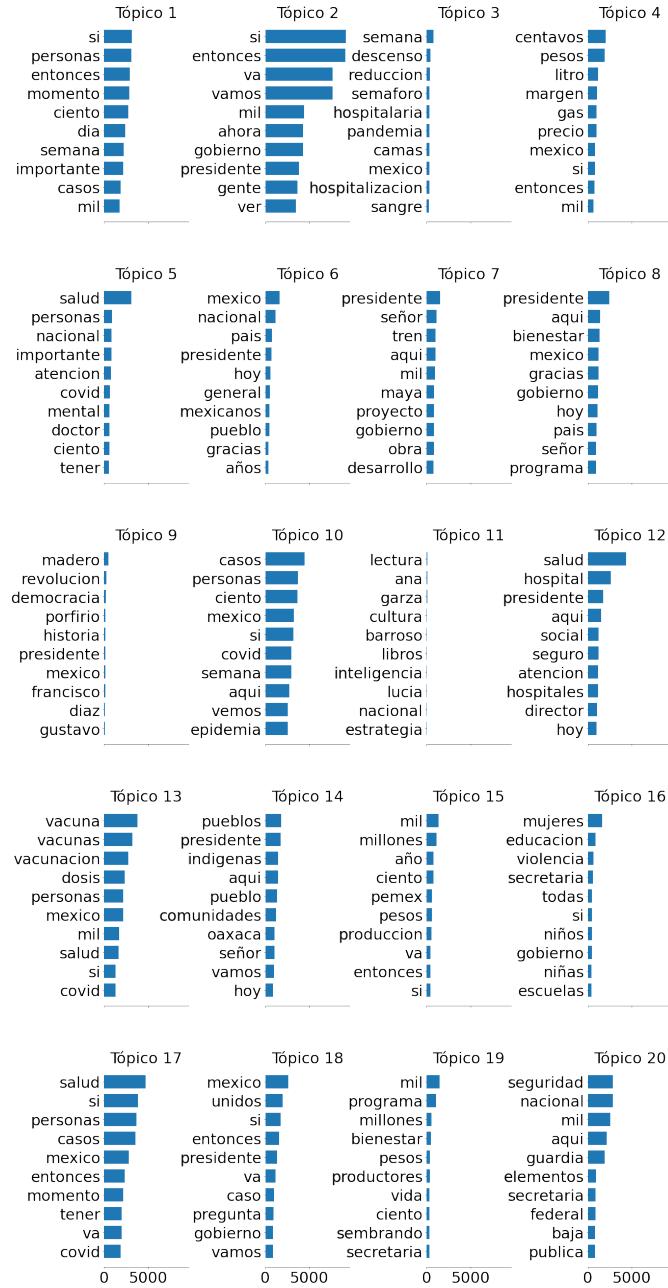
Tópicos para AMLO utilizando la norma de Frobenius (NMF)

**Figure A18.** Extracción de tópicos de las mañanera de AMLO con las palabras más representativas para cada uno utilizado la distancia de Frobenius y usando “bag of words”

Tópicos para AMLO utilizando la divergencia de Kullback-Leiblers (NMF)

**Figure A19.** Extracción de tópicos de las mañanera de AMLO con las palabras más representativas para cada uno utilizado la la divergencia del Kullback-Leibler y usando “bag of words”

Tópicos para AMLO utilizando LDA



A.1.4. Figuras para Clustering

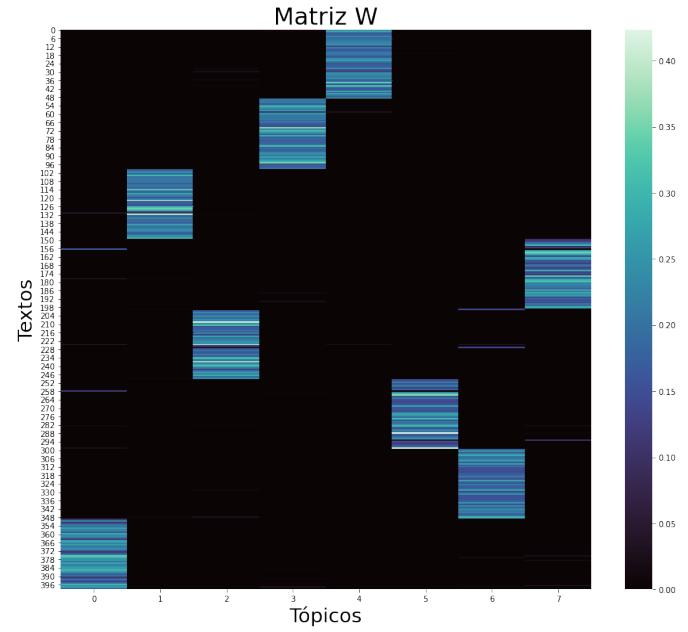
Figure A21. Heat Map de la matriz W

Figure A20. Extracción de tópicos de las mañanera de AMLO con las palabras más representativas para cada uno utilizado LDA (Latent Dirichlet Allocation) y usando “bag of words”