
Tarea 4

CENTRO DE INVESTIGACIÓN EN MATEMÁTICAS



Maestría en Cómputo Estadístico
Ciencia de Datos

Isaias Siliceo Guzmán

18 de mayo de 2024

1. Problema 1

Calcula lo siguiente:

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix} + \begin{pmatrix} 7 & 9 \end{pmatrix}$$

Usa broadcasting de tal forma que la operación esté bien definida. Antes, averigua y describe qué es broadcasting, en el contexto de numpy.

Broadcasting

El término "broadcasting" describe cómo NumPy trata con los arreglos que tienen diferentes formas durante operaciones aritméticas. El arreglo que tiene menor tamaño es transmitido (*broadcast* en inglés) a lo largo de toda la forma del arreglo más grande. Cuando se realiza una operación, NumPy compara las formas elemento por elemento, comienza con la dimensión *trailing* (la última a la derecha) y recorre hacia la izquierda. En general, dos dimensiones son compatibles cuando

1. Son iguales.
2. Una es 1.

Cuando esto ocurre, las dimensiones del arreglo que se transmite se copian para generar un arreglo de la misma forma que el más grande.¹

Ejercicio

Note que la primera operación a la izquierda sí está bien definida ya que los arreglos ambos son (2×2) , el resultado será un arreglo (2×2) ,

$$\underbrace{\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}}_{(2 \times 2)} \underbrace{\begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix}}_{(2 \times 2)} + \underbrace{\begin{pmatrix} 7 & 9 \end{pmatrix}}_{(1 \times 2)} = \underbrace{\begin{pmatrix} 4 & 7 \\ 8 & 15 \end{pmatrix}}_{(2 \times 2)} + \underbrace{\begin{pmatrix} 7 & 9 \end{pmatrix}}_{(1 \times 2)}$$

La segunda operación necesita un ajuste que NumPy hace automáticamente. La dimensión correspondiente a las columnas (*trailing dimension*) cumple con la regla de Numpy para hacer el *broadcasting*, ya que son iguales.

$$\underbrace{\begin{pmatrix} 4 & 7 \\ 8 & 15 \end{pmatrix}}_{(2 \times 2)} + \underbrace{\begin{pmatrix} 7 & 9 \end{pmatrix}}_{(1 \times 2)} = \underbrace{\begin{pmatrix} 4 & 7 \\ 8 & 15 \end{pmatrix}}_{(2 \times 2)} + \underbrace{\begin{pmatrix} 7 & 9 \end{pmatrix}}_{(2 \times 2)} = \underbrace{\begin{pmatrix} 11 & 16 \\ 15 & 24 \end{pmatrix}}_{(2 \times 2)}$$

En la Figura 1.1 se muestra el resultado de hacerlo directamente con NumPy. En muchas ocasiones el broadcasting resulta ser una herramienta eficiente para el ahorro de memoria.

¹Recuperado de Documentación NumPy el 11/05/2024

```

import numpy as np
A = np.array([
    [1,2],
    [3,4]
])
B = np.array([
    [0,1],
    [2,3]
])
c = np.array([7,9])

print(np.matmul(A,B)+c)

```

[1] ✓ 0.1s
... [[11 16]
[15 24]]

Figura 1.1: Operación de Numpy. El *broadcasting* se realiza de forma automática.

2. Problema 2

Considera un problema de clasificación multiclase y una red neuronal densa con una capa oculta, como se muestra en la Figura 2.1. Consideraremos también el uso de la función sigmoide como activación de las unidades ocultas, la función softmax para las estimaciones en la capa de salida y cross-entropy como función de costo.

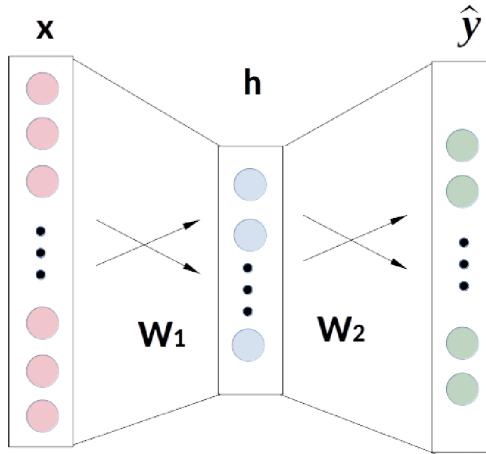


Figura 2.1: Red neuronal densamente conectada con una sola capa oculta.

- a) Muestra que softmax es invarianta a traslaciones (constantes) del vector de entrada, es decir, para cualquier vector \mathbf{x} y cualquier constante c :

$$\text{softmax}(\mathbf{x}) = \text{softmax}(\mathbf{x} + c),$$

donde la operación $\mathbf{x} + c$ se realiza con broadcasting. Recuerda que

$$\text{softmax}(\mathbf{x})_i = \frac{e^{\mathbf{x}_i}}{\sum_j e^{\mathbf{x}_j}}$$

Lo anterior es útil cuando se escoge $c = -\max(\mathbf{x})$, es decir, quitando el valor mayor en todos los elementos de \mathbf{x} , para estabilidad numérica.

Considerando la función softmax(\mathbf{x}) hacemos una operación de traslación en el argumento sobre la i -ésima componente.

$$\text{softmax}(\mathbf{x} + c)_i = \frac{e^{\mathbf{x}_i + c}}{\sum_j e^{\mathbf{x}_j + c}} = \frac{e^{\mathbf{x}_i} e^c}{\sum_j e^{\mathbf{x}_j} e^c} = \frac{e^{\mathbf{x}_i} e^c}{e^c \sum_j e^{\mathbf{x}_j}} = \frac{e^{\mathbf{x}_i}}{\sum_j e^{\mathbf{x}_j}} = \text{softmax}(\mathbf{x})_i$$

Esto se cumple para cada una de las i componentes de \mathbf{x} , de modo que se concluye

$$\text{softmax}(\mathbf{x}) = \text{softmax}(\mathbf{x} + c),$$

la función softmax(\mathbf{x}) es invariante ante traslaciones. \square

- b) Para un escalar x , muestra que el gradiente de la función sigmoide es

$$\sigma(x)(1 - \sigma(x))$$

La función sigmoide para un escalar x se define como

$$\sigma(x) = \frac{1}{1 + e^{-x}},$$

tomando el gradiente de la función sigmoide

$$\begin{aligned} \frac{d\sigma(x)}{dx} &= \frac{d}{dx} \left(\frac{1}{1 + e^{-x}} \right) \\ &= -\frac{1}{(1 + e^{-x})^2} \cdot \frac{d}{dx}(1 + e^{-x}) \\ &= \frac{e^{-x}}{(1 + e^{-x})^2} \\ &= \left(\frac{1}{1 + e^{-x}} \right) \left(\frac{e^{-x}}{1 + e^{-x}} + \underbrace{1 - 1}_{=0} \right) \\ &= \left(\frac{1}{1 + e^{-x}} \right) \left(1 + \frac{e^{-x}}{1 + e^{-x}} - \frac{1 + e^{-x}}{1 + e^{-x}} \right) \\ &= \left(\frac{1}{1 + e^{-x}} \right) \left(1 - \frac{1}{1 + e^{-x}} \right) \end{aligned}$$

Finalmente, identificamos que las funciones escritas en este último renglón no son más que la función sigmoide

$$\boxed{\frac{d\sigma(x)}{dx} = \sigma(x)(1 - \sigma(x))} \quad (2.1)$$

- c) Muestra que el gradiente en la capa de salida es

$$\frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{z}} = \hat{\mathbf{y}} - \mathbf{y},$$

donde $\hat{\mathbf{y}} = \text{softmax}(\mathbf{z})$, para algún vector \mathbf{z} que proviene de la capa de salida.

La función de costo, como mencionamos al inicio, es la cross-entropy:

$$L(\mathbf{y}, \hat{\mathbf{y}}) = - \sum_i y_i \log (\hat{y}_i)$$

donde \mathbf{y} es un vector *one-hot* de las clases y $\hat{\mathbf{y}}$ es el vector de probabilidades estimadas.

Para la k -ésima neurona de salida

$$\begin{aligned} \frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial z_k} &= - \sum_i y_i \frac{\partial}{\partial z_k} \log \left(\frac{e^{z_i}}{\sum_j e^{z_j}} \right) \\ &= - \sum_i y_i \frac{\partial}{\partial z_k} \left[\log (e^{z_i}) - \log \left(\sum_j e^{z_j} \right) \right] && (\log(a/b) = \log(a) - \log(b)) \\ &= - \sum_i y_i \frac{\partial}{\partial z_k} \left[z_i - \log \left(\sum_j e^{z_j} \right) \right] && (\log(e^a) = a) \\ &= - \sum_i y_i \left[\frac{\partial z_i}{\partial z_k} - \frac{\partial}{\partial z_k} \log \left(\sum_j e^{z_j} \right) \right] && \left(\frac{d}{dx}(f(x) - g(x)) = \frac{d}{dx}g(x) - \frac{d}{dx}f(x) \right) \\ &= - \sum_i y_i \left[\delta_{ik} - \frac{1}{\sum_j e^{z_j}} \frac{\partial}{\partial z_k} \left(\sum_j e^{z_j} \right) \right] && \left(\frac{d}{dx} \log(u) = \frac{1}{u} \frac{du}{dx} \right) \\ &= - \sum_i y_i \left[\delta_{ik} - \frac{1}{\sum_j e^{z_j}} \left(\sum_j e^{z_j} \frac{\partial}{\partial z_k} e^{z_j} \right) \right] && \left(\frac{d}{dx} \sum_i f_i(x) = \sum_i \frac{d}{dx} f_i(x) \right) \\ &= - \sum_i y_i \left[\delta_{ik} - \frac{1}{\sum_j e^{z_j}} \left(\sum_j e^{z_j} \frac{\partial z_j}{\partial z_k} \right) \right] && \left(\frac{d}{dx} e^u = e^u \frac{du}{dx} \right) \\ &= - \sum_i y_i \delta_{ik} + \sum_i y_i \frac{1}{\sum_j e^{z_j}} \left(\sum_j e^{z_j} \delta_{jk} \right) && \left(\frac{\partial x_i}{\partial x_j} = \delta_{ij} \right) \end{aligned}$$

donde δ_{ik} y δ_{jk} son deltas de Kronecker, cuyo valor es 1 cuando $i = k$ y $k = j$ respectivamente. En el caso contrario, cuando $i \neq k$ y $k \neq j$ ambos son 0. De tal forma que la derivada se reescribe como

$$\frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial z_k} = -y_k + \sum_i y_i \frac{e^{z_k}}{\sum_j e^{z_j}} = \sum_i y_i \hat{y}_k - y_k$$

Ya que y_i es un vector *one hot*, este tiene un valor de 1 en la k -ésima categoría y ceros en todas las demás. De modo que cuando $i = k$, se tiene que la derivada de la *Cross entropy* para cada una de las k neuronas de salida está dada por

$$\frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial z_k} = \hat{y}_k - y_k$$

(2.2)

De modo que para algún vector \mathbf{z} en la capa de salida, el gradiente de la *Cross entropy* está dado por

$$\boxed{\frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{z}} = \hat{\mathbf{y}} - \mathbf{y}} \quad (2.3)$$

- d) Considerando los incisos anteriores, obtén los gradientes respecto a los parámetros del modelo calculando

$$\frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{x}},$$

para obtener de ésta forma, las ecuaciones de backpropagation de la red. Recuerda que el paso forward calcula las activaciones $\mathbf{h} = \sigma(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1)$ y $\hat{\mathbf{y}} = \text{softmax}(\mathbf{W}_2 \mathbf{h} + \mathbf{b}_2)$. Recuerda también que la función de activación en un vector, se aplica la entrada por entrada.

Con el objetivo de aprovechar las definiciones previas, considere que $\mathbf{z} = \mathbf{W}'_2 \mathbf{h} + \mathbf{b}_2$ y $\mathbf{u} = \mathbf{W}'_1 \mathbf{x} + \mathbf{b}_1$. Entonces, procedemos a determinar los siguientes resultados para los gradientes de \mathbf{z} y \mathbf{u}

$$\frac{\partial \mathbf{z}}{\partial \mathbf{W}_2} = \mathbf{h} = \sigma(\mathbf{W}'_1 \mathbf{x} + \mathbf{b}_1) \quad (2.4)$$

$$\frac{\partial \mathbf{z}}{\partial \mathbf{h}} = \mathbf{W}_2 \quad (2.5)$$

$$\frac{\partial \mathbf{z}}{\partial \mathbf{b}_2} = \mathbf{1} \quad (2.6)$$

$$\frac{\partial \mathbf{u}}{\partial \mathbf{W}_1} = \mathbf{x} \quad (2.7)$$

$$\frac{\partial \mathbf{u}}{\partial \mathbf{b}_1} = \mathbf{1} \quad (2.8)$$

Considerando que los parámetros del modelo son \mathbf{W}_1 , \mathbf{W}_2 , \mathbf{b}_1 y \mathbf{b}_2 . Además, obviamos las dependencias de la función de costo L . Aplicando la regla de la cadena, procedemos como sigue,

$$\frac{\partial L}{\partial \mathbf{W}_2} = \underbrace{\frac{\partial L}{\partial \mathbf{z}}}_{(2.3)} \underbrace{\frac{\partial \mathbf{z}}{\partial \mathbf{W}_2}}_{(2.4)} = (\hat{\mathbf{y}} - \mathbf{y}) \mathbf{h} = \boxed{(\hat{\mathbf{y}} - \mathbf{y}) \sigma(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1)}$$

$$\frac{\partial L}{\partial \mathbf{W}_1} = \underbrace{\frac{\partial L}{\partial \mathbf{z}}}_{(2.3)} \underbrace{\frac{\partial \mathbf{z}}{\partial \mathbf{h}}}_{(2.5)} \underbrace{\frac{\partial \mathbf{h}}{\partial \mathbf{u}}}_{(2.1)} \underbrace{\frac{\partial \mathbf{u}}{\partial \mathbf{W}_1}}_{(2.7)} = \boxed{(\hat{\mathbf{y}} - \mathbf{y}) \mathbf{W}_2 \sigma(\mathbf{u}) [\mathbf{1} - \sigma(\mathbf{u})] \mathbf{x}}$$

$$\frac{\partial L}{\partial \mathbf{b}_2} = \underbrace{\frac{\partial L}{\partial \mathbf{z}}}_{(2.3)} \underbrace{\frac{\partial \mathbf{z}}{\partial \mathbf{b}_2}}_{(2.6)} = \boxed{(\hat{\mathbf{y}} - \mathbf{y}) \mathbf{1}}$$

$$\frac{\partial L}{\partial \mathbf{b}_1} = \underbrace{\frac{\partial L}{\partial \mathbf{z}}}_{(2.3)} \underbrace{\frac{\partial \mathbf{z}}{\partial \mathbf{h}}}_{(2.5)} \underbrace{\frac{\partial \mathbf{h}}{\partial \mathbf{u}}}_{(2.1)} \underbrace{\frac{\partial \mathbf{u}}{\partial \mathbf{b}_1}}_{(2.8)} = \boxed{(\hat{\mathbf{y}} - \mathbf{y}) \mathbf{W}_2 \sigma(\mathbf{u}) [\mathbf{1} - \sigma(\mathbf{u})] \mathbf{1}}$$

3. Problema 3

Considera de nuevo los textos de transcripciones de las conferencias matutinas de la presidencia de México que usaste en la tarea 3. En éste ejercicio implementarás un método de análisis de tópicos mediante un algoritmo de clústering.

- a) Usando vocabulario obtenido mediante los textos por semana, obtén las representaciones de las palabras usando word2vec^a pre-entrenado en español. En éste *espacio semántico* obtén un modelo de tópicos usando Fuzzy k -means, eligiendo el tamaño adecuado de k . Representa cada tópico mediante un *word cloud* usando la probabilidad máxima como criterio para elegir las palabras más representativas de cada tópico.

 - ¿Puedes asignar un "nombre" representativo de cada tópico?
 - ¿Qué diferencias notas respecto a lo que obtuviste con la representación TF-IDF?

“Define un mecanismo para manejar las palabras fuera del vocabulario y menciónalo en tu reporte.”

En este ejercicio se hará una revisión de las transcripciones diarias de las Conferencias Matutinas del Presidente Andrés Manuel López Obrador², con el objetivo de elaborar un análisis de tópicos que muestre los temas más relevantes durante estos eventos durante el periodo correspondiente a los años 2019-2023.

La lectura de los datos se realizó para obtener una serie temporal con las transcripciones agrupadas por semana. Dando como resultado 259 semanas con texto. A continuación, en la figura 3.1 se muestra la nube de palabras de todas las conferencias durante el periodo de estudio habiendo eliminado *stop words*. Además, se eligió que las palabras conservaran sus acentos con el objetivo de no perder sentido al compararse con otras. En la figura se observa que existen todavía palabras que no son clasificables en algún tópico. De modo que se elaboraron funciones que permitieran retirar adjetivos, adverbios y otras palabras no clasificables a criterio del autor de este ejercicio, el vocabulario removido se encuentra en el *notebook* anexo a este reporte.

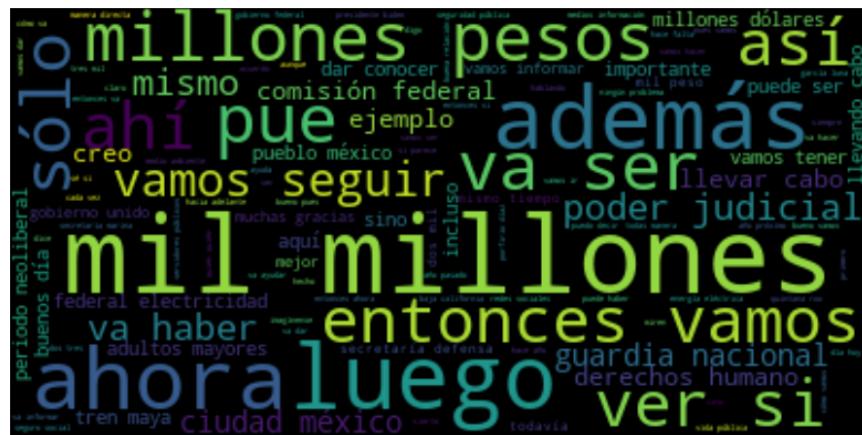


Figura 3.1: Nube de palabras con el texto previo al procesamiento. Se puede observar que la mayoría de palabras no son clasificables. En el sentido de no poder catalogarse en algún tópico en particular. Es necesario realizar un procesamiento para removerlas.

²Recuperado del repositorio conferencias_matutinas_amlo

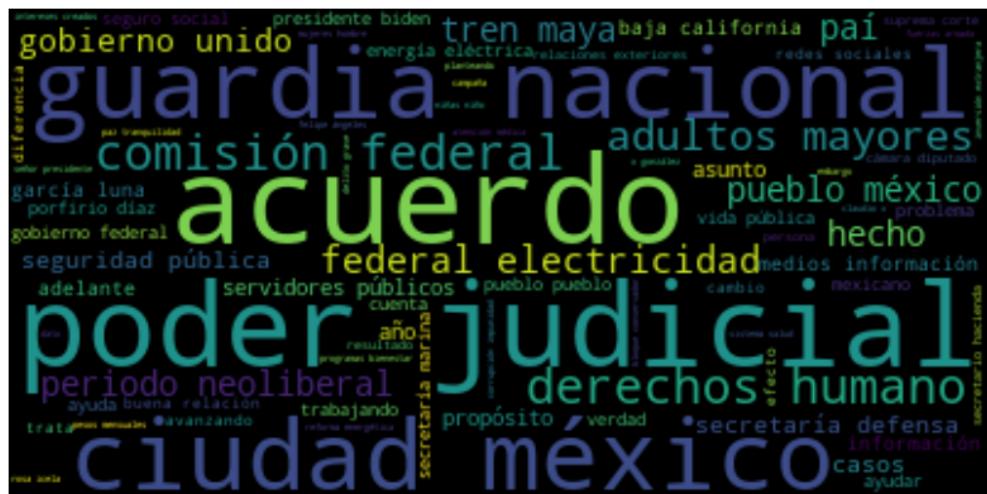


Figura 3.2: Nube de palabras con las palabras más destacables durante las conferencias matutinas de AMLO en el periodo 2019-2023. La mayor parte de las palabras que se encuentran aquí son clasificables en algún tópico.

Luego del procesamiento de los datos, la nube de palabras de las transcripciones se puede observar en la figura 3.2. Se puede observar en esta figura que la mayoría de palabras que se muestran son palabras clasificables. En el sentido de que podemos asignar un tópico a ellas. Para hacer un análisis de tópicos, se eligió un vocabulario con las 500 palabras más repetidas, algunas de estas pueden observarse en grande en la figura 3.2.

3.1. Embeddings Word2Vec

A continuación en la figura 3.3 se presentan los resultados para las nubes de palabras determinadas utilizando el método de clusterización "Fuzzy K- means" utilizando los embeddings de un algoritmo Word2Vec preentrenado³ con el corpus *Spanish Billion Word Corpus*, el cual contiene 1 millón de palabras. Cada embedding es un vector de dimensión 300 que fue clasificado en alguno de los clústers mostrados en la figura 3.3. Además, en la descripción de cada clúster se añade un nombre que engloba de manera general cada nube de acuerdo a las palabras más destacables de la misma. Al buscar el vocabulario de tamaño 500 en estos embeddings se observó que no se cuentan con las palabras: '*pemex*', '*juárez*', '*biden*', '*yucatán*', '*covid*', '*zedillo*'. De modo que el vocabulario se redujo a 494 palabras.

Para este ejercicio se eligieron 7 clústers y un parámetro de *fuzzyness* de 3. En comparación al proyecto realizado con la matriz TF-IDF, esta vez se aumentó el número de palabras en el vocabulario y esto propició a que más palabras de diferentes tipos se agruparan en un mismo clúster, de modo que se aumentó el número de clústers de 5 a 7. En comparación al uso de la matriz TF-IDF, utilizando estos embeddings se observó que los clústers contienen más palabras.

³Embeddings recuperados del repositorio spanish-word-embeddings



Figura 3.3: Nubes de palabras clasificadas en un clúster de acuerdo a un algoritmo de clusterización *Fuzzy K-means*. En la descripción de cada nube se añade un nombre que engloba de manera general cada grupo de palabras.

- b) Como en la tarea anterior, considera cada una de las conferencias del presidente durante los años del estudio como tus "documentos". Obtén la representación vectorial respectiva calculando *el promedio* de los embeddings de las palabras que componen cada uno de ellos. Posteriormente, obtén la asignación de cada documento en su tópico correspondiente usando el modelo ajustado en el inciso anterior. Usa visualizaciones de baja dimensión basadas en PCA, Kernel PCA y t-SNE de la asignación de tópicos que obtuviste y reporta los patrones y hallazgos que identifiques.

Ya que las palabras tienen una representación vectorial, es posible determinar cuál fue la palabra promedio que más se utilizó durante cada semana. De este modo, se obtuvieron 261 vectores promedio de dimensión 300. Cada uno de estos vectores nuevamente puede ser clasificado en alguno de los tópicos utilizando la distancia más cercana a uno de los centroides encontrados utilizando *Fuzzy K-means*. Estos vectores se representan en visualizaciones de baja dimensión (PCA, Kernel-PCA, t-SNE) con el objetivo de visualizar patrones o proporciones. Los resultados se muestran en las figuras 3.4, 3.5 y 3.6. Es evidente que en ninguna de estas figuras se observa algún patrón destacable. Pero la palabra promedio destaca algunos tópicos más que otros. Los tópicos

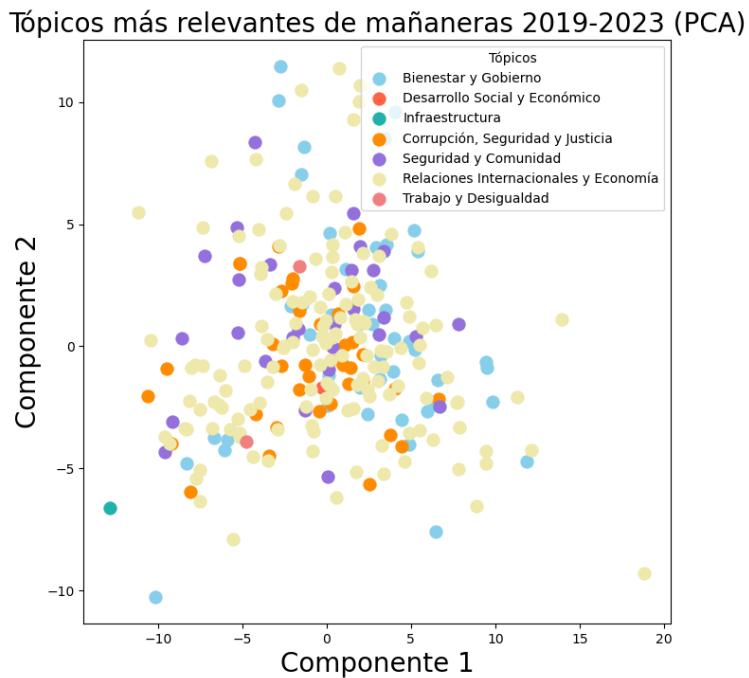


Figura 3.4: Representación en baja dimensión utilizando PCA. Se puede observar que hay un mayor número de palabras promedio asociadas a tópicos como lo son *Relaciones Internacionales y Economía*, en cambio, hay apenas una palabra promedio en el tópico de *Infraestructura*.

Tópicos más relevantes de mañaneras 2019-2023 (Kernel PCA)

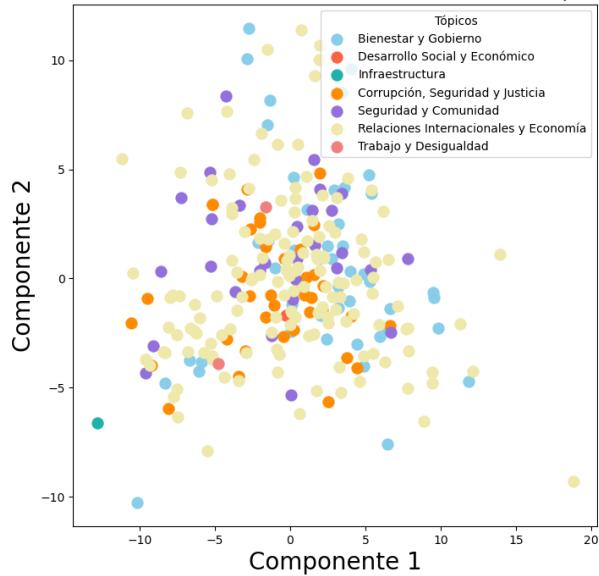


Figura 3.5: Representación en baja dimensión utilizando Kernel-PCA. Se puede observar que, al igual que utilizando PCA, la mayoría de palabras promedio nuevamente pertenecen al tópico de *Relaciones Internacionales y Economía*.

Tópicos más relevantes de mañaneras 2019-2023 (t-SNE)

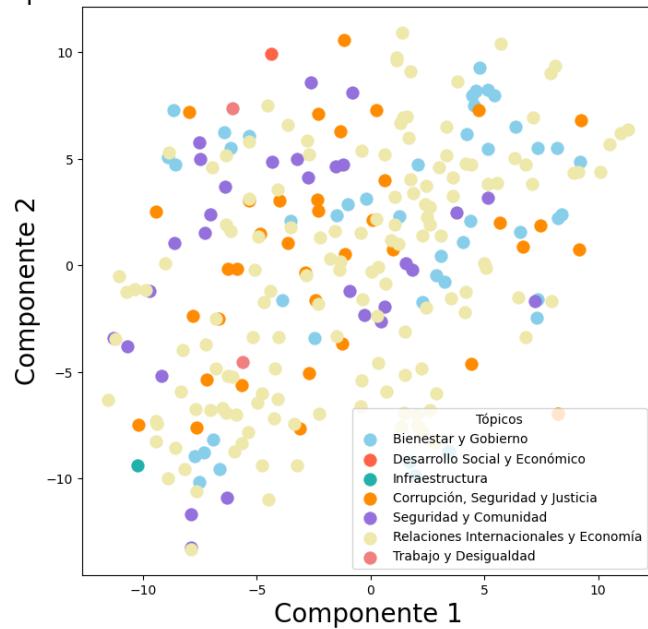


Figura 3.6: Representación en baja dimensión utilizando t-SNE. Incluso en esta proyección, no se observa un patrón interesante de los datos. Probablemente relacionado a la reducción de dimensión tan drástica.

- c) Construye un indicador semanal para cada uno de los k tópicos durante el periodo de estudio y repite el inciso 4e de la tarea anterior.

Para construir el indicador semanal, se recuperó un diccionario de "palabra:frecuencia" para cada unas de las 261 semanas en el periodo. Estos diccionarios sólo contienen palabras que pertenecen al vocabulario y que se encuentran en una semana particular. De modo que podemos dividir este diccionario en sub-diccionarios asociados a cada tópico según la clasificación obtenida con *Fuzzy K-means*. El resultado es una tabla de frecuencias las palabras correspondientes a cada tópico por semana la cual escaló para obtener un indicador de la probabilidad de ocurrencia de un tópico en particular. En la figura 3.7 se muestra la serie temporal de cada uno de los tópicos. Estas series temporales son casi estacionarias, no se observan comportamientos relevantes como aquellos que se observaron en el proyecto con TF-IDF, donde se observaba un cambio drástico en los temas cuando ocurrió la pandemia de 2020 (El tópico de Salud pública incrementó drásticamente).

Relevancia de tópicos en las Mañaneras del Presidente AMLO (2019-2023)(w2v)

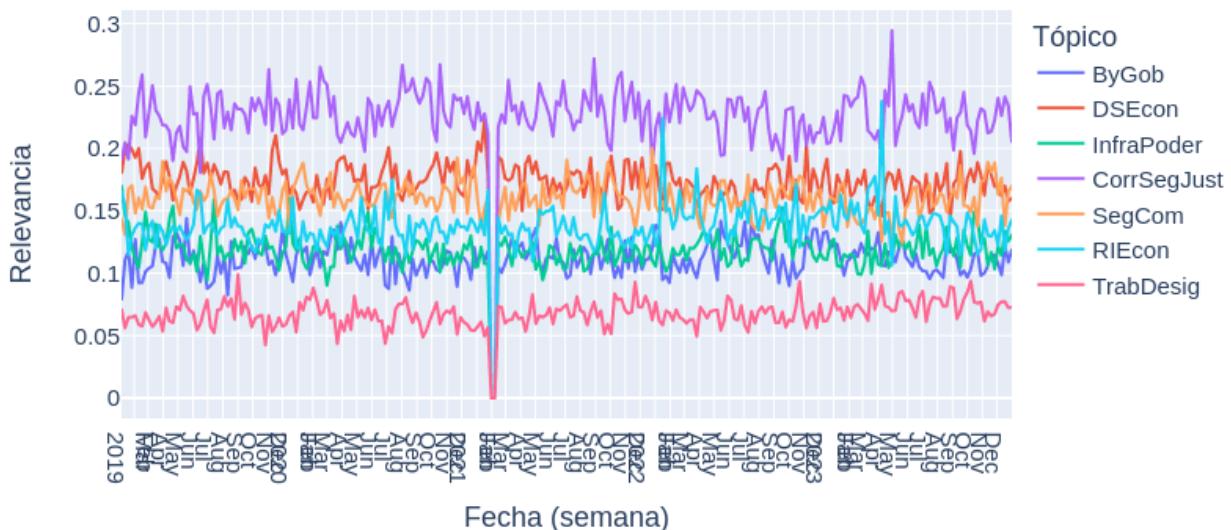


Figura 3.7: Serie de tiempo correspondiente al periodo 2019-2023. En esta figura se recupera el comportamiento general de los tópicos a lo largo del tiempo. Además, se incluyen las series temporales por año en el Apéndice de este reporte utilizando los embeddings de Word2Vec.

- d) Repite los incisos anteriores usando fastText.

3.2. Embeddings FastText

Como punto de partida, se tiene el mismo vocabulario de 500 palabras que en los incisos previos. En esta ocasión se recuperaron los embeddings de un modelo preentrenado con el algoritmo de FastText, utilizando esta vez el corpus *Spanish Unannotated Corpora*⁴. Este contiene 1.3 millones de embeddings de tamaño 300. Se realizó una búsqueda del vocabulario y sólo no se encontró una

⁴Recuperado del repositorio spanish-word-embeddings

palabra: '*covid*', de modo que para hacer la clusterización sólo se utilizaron 499 palabras. De este modo, utilizando el algoritmo de clusterización *Fuzzy K-means* se seleccionaron esta vez 4 clústers. Ya que utilizando 7 había muchas palabras que no entraban en algún tópico. Lo cual desequilibró mucho el vocabulario en cada nube de palabras. Sin embargo, ajustando los parámetros de *fuzziness* y el número de clústers, se llegó a que un análisis con 4 tópicos era adecuado. En la figura 3.8 se muestran las nubes de palabras obtenidas de la clasificación en cada clúster.



Figura 3.8: ubes de palabras clasificadas en un clúster de acuerdo a un algoritmo de clusterización *Fuzzy K-means*. En la descripción de cada nube se añade un nombre que engloba de manera general cada grupo de palabras.

3.3. Representaciones en baja dimensión

Además, como en el inciso b) de este ejercicio, se realizaron visualizaciones en baja dimensión de los embeddings promedio por semana. Esta vez utilizando aquellos obtenidos con FastText. En las figuras se muestran los resultados. En esta ocasión, es posible observar una buena separación entre dos de los tópicos *Gobierno Nacional* y *Reformas y Programas Sociales*, sin embargo, los dos tópicos restantes no son tan comunes.

Tópicos más relevantes de mañaneras 2019-2023 (PCA)

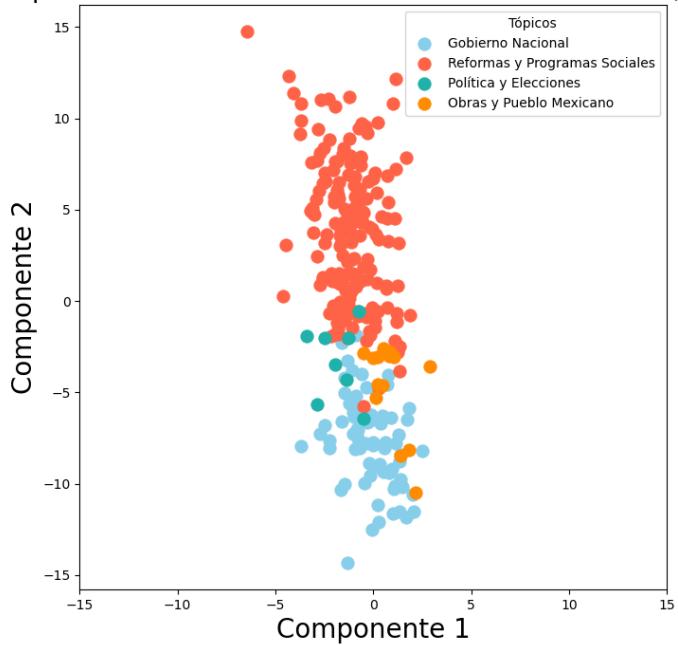


Figura 3.9: Representación en baja dimensión utilizando PCA. Se puede observar una separación entre dos de los tópicos, *Gobierno Nacional* y *Reformas y Programas Sociales*, lo cual refleja que los embeddings promedio se encuentran en desbalance con respecto a los otros tópicos.

Tópicos más relevantes de mañaneras 2019-2023 (Kernel PCA)

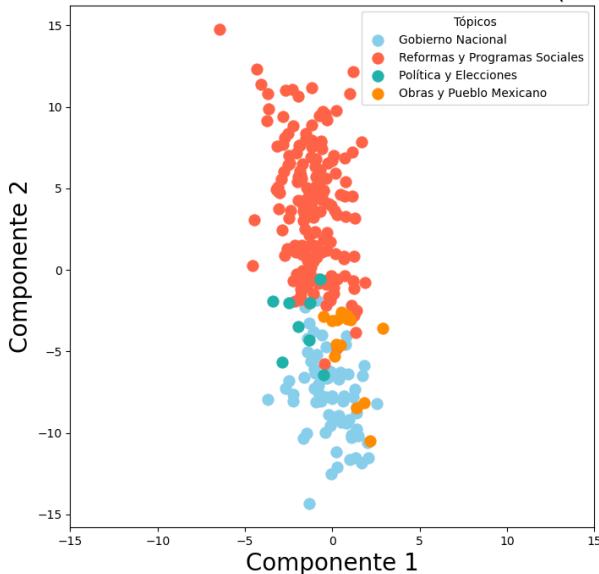


Figura 3.10: Representación en baja dimensión utilizando Kernel-PCA. Al igual que con PCA, se puede observar una separación entre dos de los tópicos, *Gobierno Nacional* y *Reformas y Programas Sociales*, lo cual refleja que los embeddings promedio se encuentran en desbalance con respecto a los otros tópicos. En ambas representaciones se observa que los embeddings promedio correspondientes a otros tópicos no son tan comunes.

Tópicos más relevantes de mañaneras 2019-2023 (t-SNE)

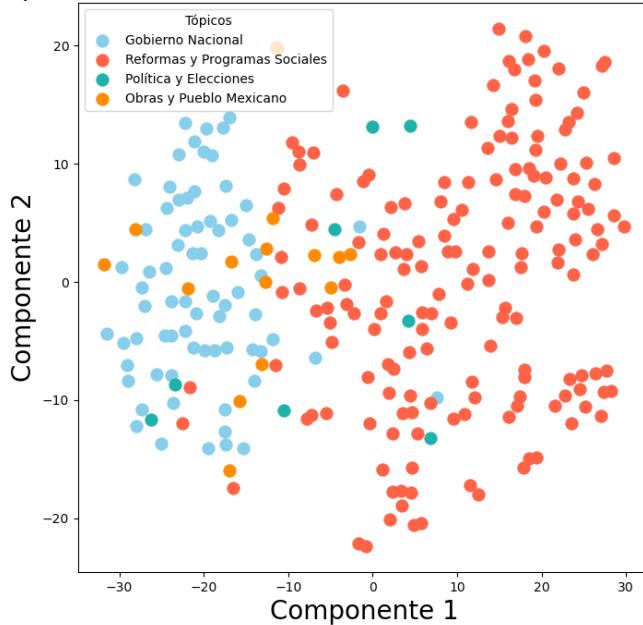


Figura 3.11: Representación en baja dimensión utilizando t-SNE. Al igual que con las representaciones anteriores, se puede observar una separación entre dos de los tópicos, *Gobierno Nacional* y *Reformas y Programas Sociales*. Sin embargo, en esta representación los datos muestran mayor dispersión.

3.4. Indicador semanal

De forma análoga al inciso anterior, se realizó el mismo procedimiento para obtener las series temporales con las probabilidades de ocurrencia de cada uno de los tópicos. En la figura 3.12 se muestra la serie temporal correspondiente al periodo de 2019 a 2023 utilizando los embeddings de FastText. En esta se observa un comportamiento similar, no hay un cambio significativo que permita ver sucesos importantes reflejados en los tópicos que se analizaron. Como referencia, observe que estos cambios sí ocurren utilizando la representación TF-IDF en la figura 3.13. Lo cual refleja que es muy importante la selección y la longitud del vocabulario en este estudio.

Relevancia de tópicos en las Mañaneras del Presidente AMLO (2019-2023)(FT)

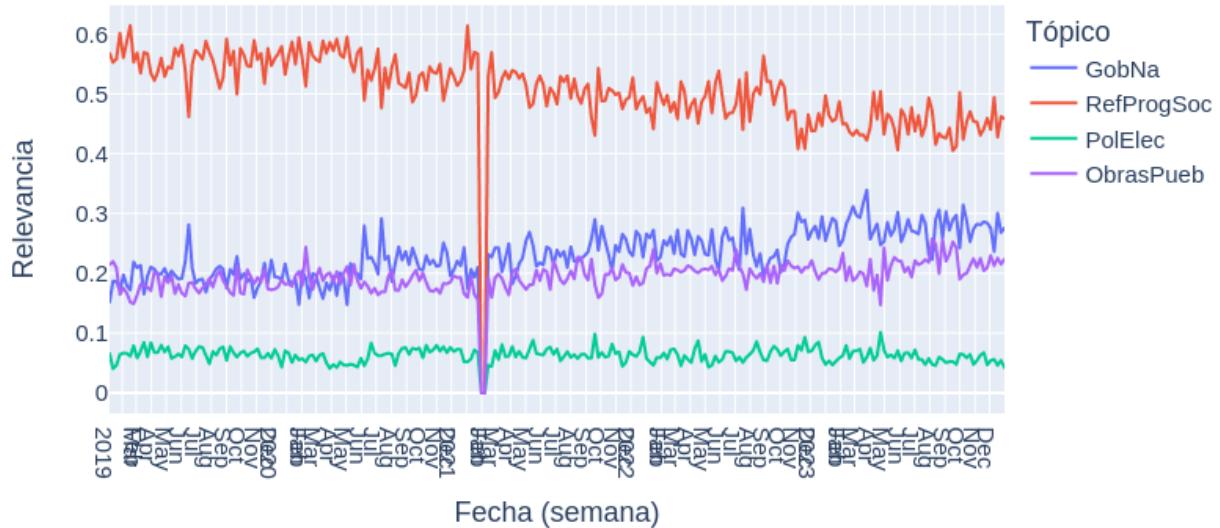


Figura 3.12: Serie de tiempo correspondiente al periodo 2019-2023. En esta figura se recupera el comportamiento general de los tópicos a lo largo del tiempo. Además, se incluyen las series temporales por año en el Apéndice de este reporte utilizando los embeddings de FastText.

Relevancia de tópicos en las Mañaneras del Presidente AMLO (2019-2023)

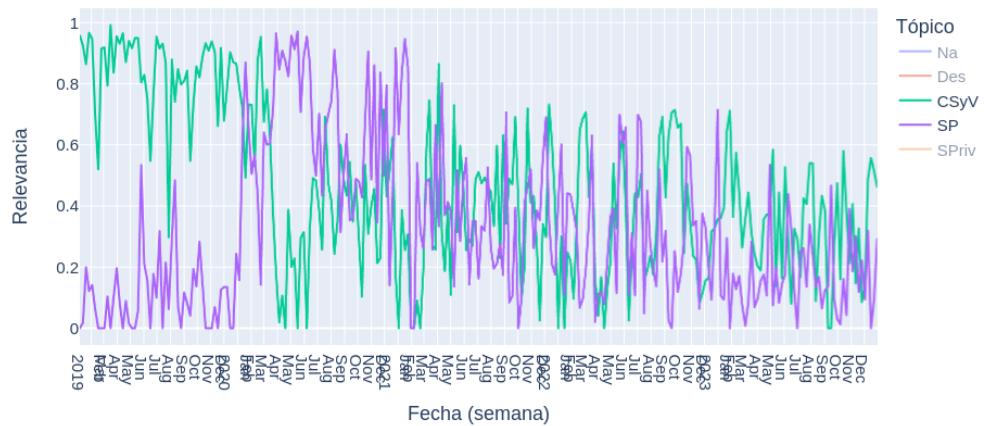


Figura 3.13: En esta serie de tiempo se encuentran los Tópicos "Corrupción, Seguridad y Violencia" (CSyV) y "Salud Pública" (SP), ambos tópicos han perdido relevancia a lo largo de los 5 años de estudio. Al comienzo de 2019, los tópicos de CSyV son muy representativos, mientras que al inicio del año 2020 se ve un incremento abrupto en los temas de salud. Posiblemente, debido en gran medida al inicio de la pandemia de *SARS-Covid 19* en el año 2020.

- e) Realiza un reporte ejecutivo que resuma tus análisis, hallazgos y conclusiones, resaltando las ventajas, desventajas y comparación entre los diferentes métodos que usaste para analizar éste conjunto de datos. Incluye sugerencias para mejorar el análisis.

3.5. Conclusiones

El uso de embeddings depende mucho del vocabulario utilizado. Los resultados presentados en este proyecto pueden variar mucho si se utiliza un vocabulario diferente.

3.6. Ventajas, desventajas y comparación con TF-IDF

Ventajas:

1. Las palabras tienden a agruparse más entre ellas debido a la ventana contextual que tienen.
2. Las palabras tienen una representación vectorial y se pueden realizar operaciones con ellas.
3. Se pueden visualizar las palabras y su cercanía con otras gracias a las representaciones en baja dimensión.

Desventajas:

1. Los parámetros para realizar la clasificación con *Fuzzy K-means*. Elegir estos parámetros puede cambiar mucho el cómo se distribuyen las palabras.

En comparación al utilizar la representación TF-IDF, las series temporales tenían bastante sentido con la realidad del país, e.g. el aumento en el tópico de Salud pública durante la pandemia de 2020. En este proyecto únicamente se observan series de tiempo que no varían mucho.

3.7. Sugerencias para mejorar el análisis

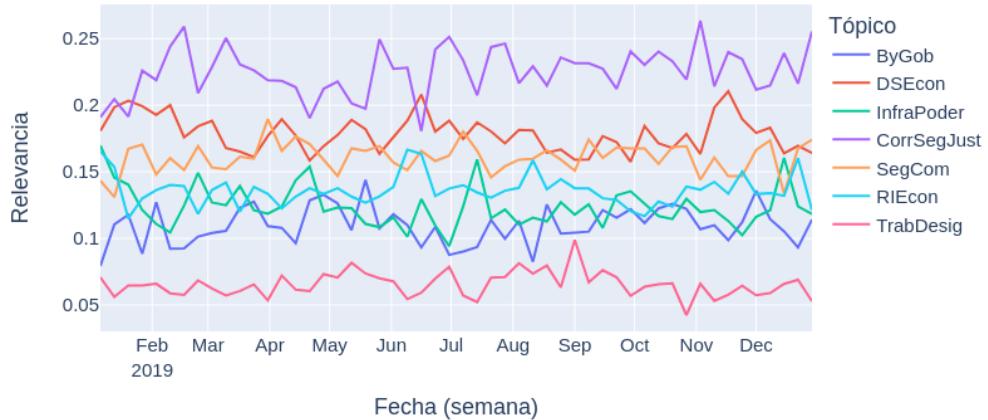
1. Hacer un preproceso adecuado de los documentos.
 - a) Lematización.
 - b) Eliminar acentos.
 - c) Remover palabras menores a 4 letras.
2. Tomar los embeddings promedio por día para equilibrar las proporciones de cada tópico en las representaciones de baja dimensión.

Apéndice A

A continuación, se incluyen las series temporales correspondientes al periodo 2019-2023. Organizadas por año para una visualización más detallada.

Series temporales Word2Vec

Relevancia de tópicos en las Mañaneras del Presidente AMLO 2019-(word2vec)



Relevancia de tópicos en las Mañaneras del Presidente AMLO 2020-(word2vec)

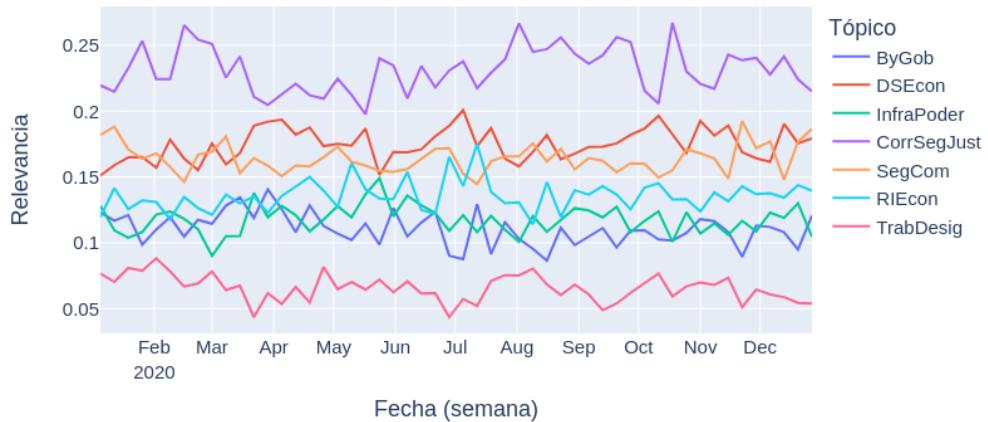
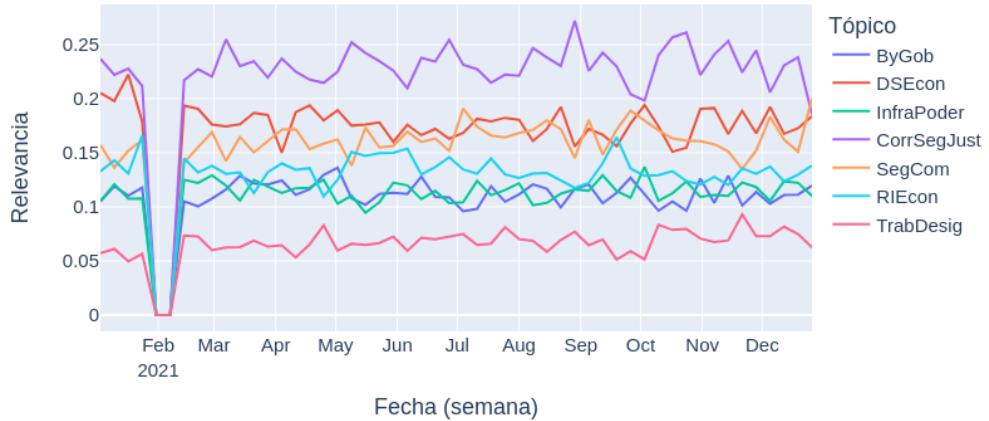
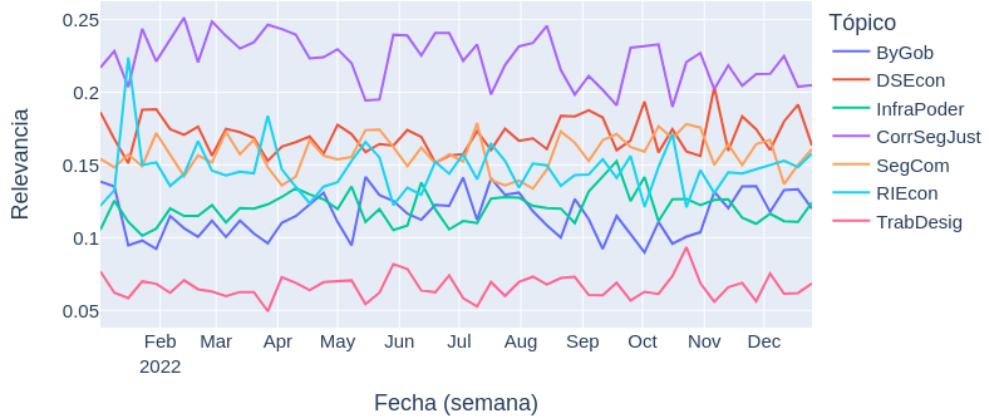


Figura 3.1: Series temporales correspondientes a los años 2019 y 2020 utilizando los embeddings de Word2Vec

Relevancia de tópicos en las Mañaneras del Presidente AMLO 2021-(word2vec)



Relevancia de tópicos en las Mañaneras del Presidente AMLO 2022-(word2vec)



Relevancia de tópicos en las Mañaneras del Presidente AMLO 2023-(word2vec)

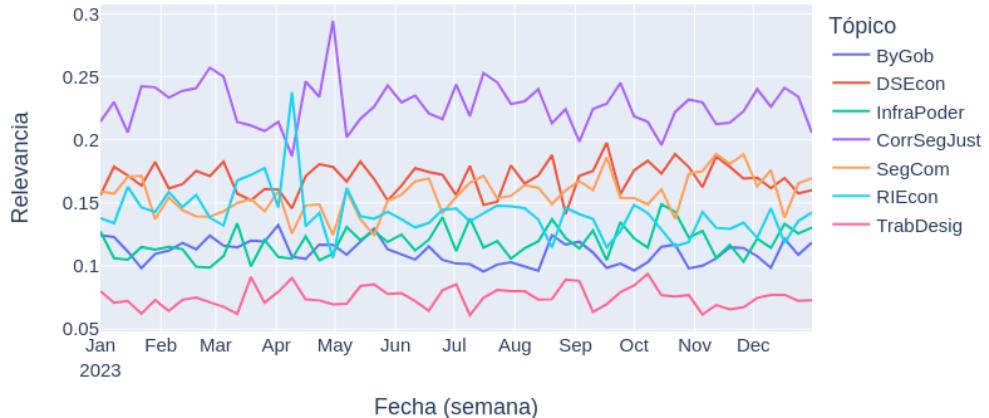
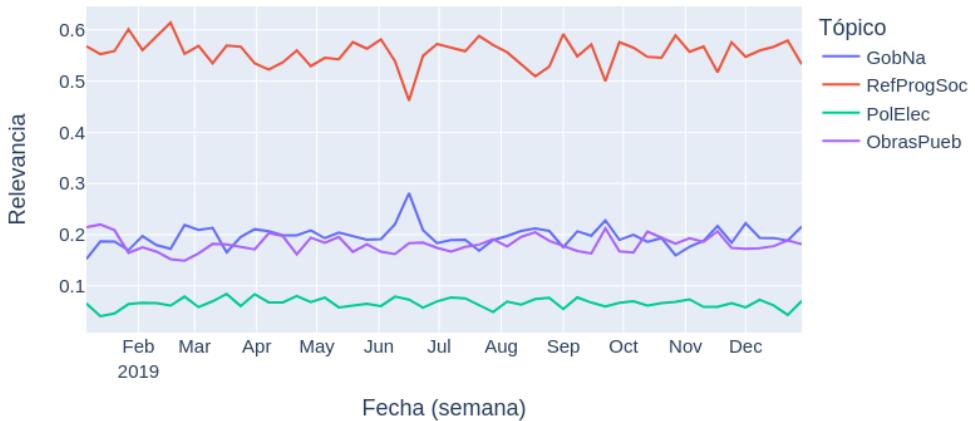


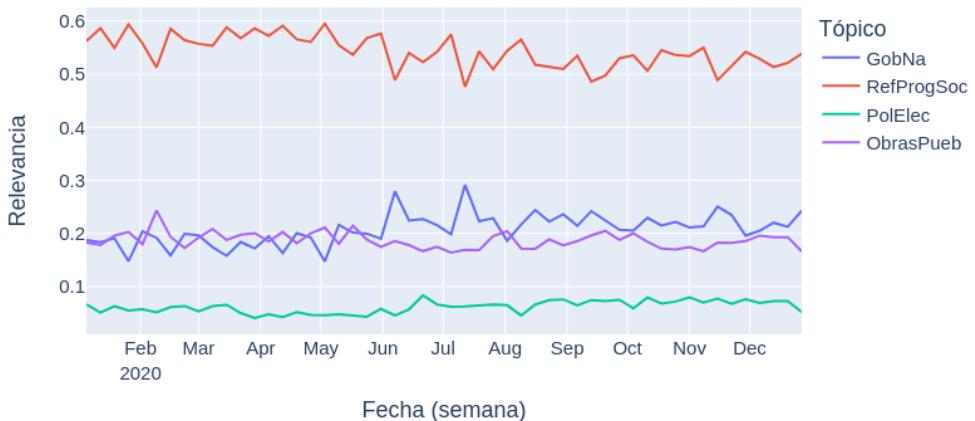
Figura 3.2: Series temporales correspondientes a los años 2021 a 2023 utilizando los embeddings de Word2Vec

Series temporales FastText

Relevancia de tópicos en las Mañaneras del Presidente AMLO 2019(FastText)



Relevancia de tópicos en las Mañaneras del Presidente AMLO 2020(FastText)



Relevancia de tópicos en las Mañaneras del Presidente AMLO 2021(FastText)

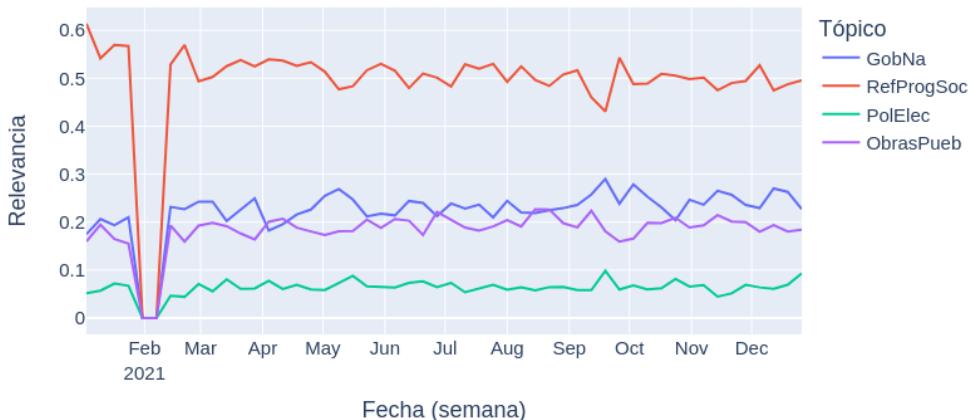
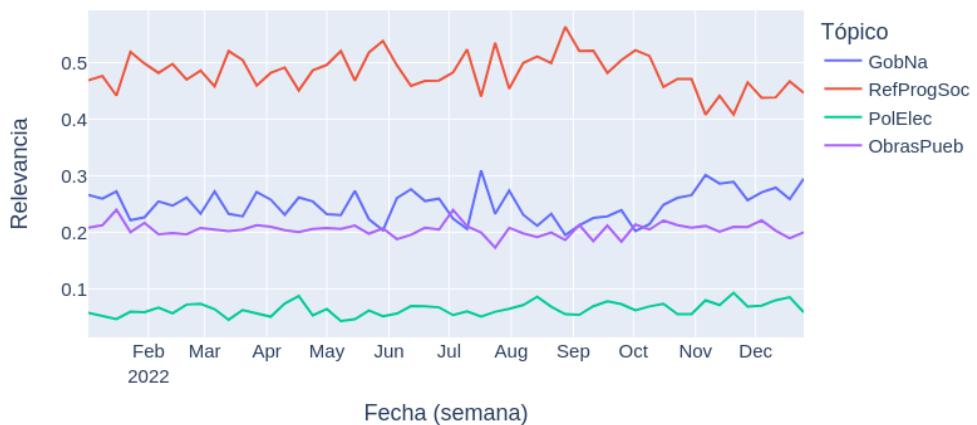


Figura 3.3: Series temporales correspondientes a los años 2019 a 2021 utilizando los embeddings de FastText

Relevancia de tópicos en las Mañaneras del Presidente AMLO 2022(FastText)



Relevancia de tópicos en las Mañaneras del Presidente AMLO 2023(FastText)

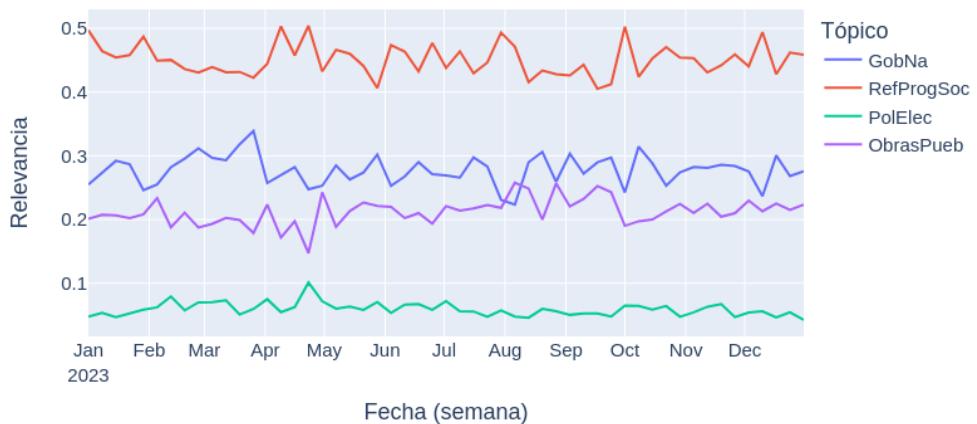


Figura 3.4: Series temporales correspondientes a los años 2022 y 2023 utilizando los embeddings de FastText