

2. PROBLEMA

Este ejercicio es sobre análisis de tópicos. Un tópico es una variable latente que representa o resume conceptos importantes de un texto, como el significado o las ideas principales del mismo. Un tópico, se conforma por varias palabras relacionadas semánticamente entre sí de acuerdo a cierto contexto. En el área de procesamiento de lenguaje natural (NLP), forma parte de una tarea general llamada *recuperación de información* (IR). Para nosotros, desde la perspectiva de machine learning, la consideraremos como una tarea de aprendizaje no-supervisado a partir de una representación vectorial particular de los textos.

Considera una representación documento-término como las que vimos en clase. Una forma sencilla de extraer estructuras latentes entre documentos y términos es usando análisis semántico latente (LSA), el cual se basa en factorizaciones apropiadas de esa matriz. Sea $A_{m \times n}$ la matriz TF-IDF de rango r , con m renglones (documentos) y n columnas (términos). Una aproximación de rango k de esta matriz, está dada por la factorización SVD $A \approx A^{(k)} = U^{(k)} \Sigma^{(k)} V^{(k) \prime}$, donde $\Sigma^{(k)}$ es diagonal¹ con los k eigenvalores más grandes de A y $U^{(k)}$, $V^{(k)}$ contienen los correspondientes eigenvectores izquierdos y derechos que definen una base ortogonal para los espacios columna y renglón, respectivamente. Al aplicar esta factorización en matrices documento-término, podemos extraer las relaciones semánticas y conceptuales entre documentos y términos expresadas en un conjunto de componentes (o tópicos) k , mediante representaciones densas y de baja dimensión, donde $V_{n \times k}^{(k)}$ y $U_{n \times k}^{(k)}$ nos proporcionan una representación de los términos y documentos, respectivamente en términos de los k tópicos, y $\Sigma^{(k)}$ nos proporcionan a la importancia de cada tópico. En Python, puedes usar la implementación de `sklearn.decomposition.TruncatedSVD`.

En este ejercicio, realizarás un análisis de tópicos en las transcripciones de las conferencias matutinas de la presidencia de México, los cuales puedes acceder en este repositorio². Para construir tu modelo de tópicos, considera los textos de las conferencias por semana durante los años 2019 a 2023, usando las transcripciones que corresponden al presidente, contenido en los archivos “PRESIDENTE ANDRES MANUEL LOPEZ OBRADOR.csv”³.

- a) Obtén una representación TF-IDF de los textos. Define el tamaño del vocabulario y realiza el preprocess que consideres necesario en los textos, considerando que, para un análisis de tópicos, no es recomendable que el vocabulario sea tan grande, y es mejor conservar palabras cuyo uso dentro del texto, pueda asociarse con tópicos. Documenta y justifica tus parametrizaciones.

¹ Estamos considerando la representación reducida de SVD, donde se han removido todas las entradas cero de Σ , y las correspondientes columnas de U y V , y aún más, se han remplazado por ceros los $k - r$ valores propios más pequeños.

² Recomiendo hacer un git clone a todo el repositorio, para mantener la misma estructura de archivos.

³ Ten cuidado con las diferentes variaciones del nombre de los archivos, e.g: LOPEZ y LOPEZ...

- b) Obtén k tópicos mediante la descomposición SVD. Elige un k adecuado y justifícalo. Representa cada tópico mediante un **Word Cloud**⁴ de los términos que forman cada tópico según la importancia expresada en las magnitudes de los renglones de $V^{(k)}$. ¿Puedes asignar un “nombre” representativo de cada tópico?
- c) Usando el modelo de tópicos ajustado en el paso previo, obtén la representación correspondiente de cada una de las conferencias del presidente durante los años del estudio, calculando la matriz documento-tópico mediante el producto $XV^{(k)}$ (o con el método **transform** de **TruncatedSVD**). Asigna cada conferencia a su tópico correspondiente usando como criterio el valor máximo de cada renglón de la matriz. Usa visualizaciones de baja dimensión basadas en PCA, Kernel PCA y t-SNE de la asignación de tópicos que obtuviste. ¿Observas patrones interesantes? Describe brevemente tus hallazgos.
- d) Un problema que surge al usar SVD es la falta de interpretabilidad, ya que no es claro cómo pueden considerarse los valores negativos en las matrices U y V . Una forma de resolver este problema es usar una factorización no-negativa de matrices (NMF), que es adecuada para matrices con entradas no negativas, como las TF-IDF. Para una matriz A de rango r con entradas no-negativas, NMF calcula una aproximación de rango $k < r$ mediante la factorización $A \approx A(k) = W^{(k)}H^{(k)}$, donde $W^{(k)}, H^{(k)} \geq 0$. En **scikit-learn** puedes usar la clase NMF del módulo `sklearn.decomposition.NMF`. Repite los incisos anteriores usando esta descomposición. ¿Cuál te parece mejor y por qué?
- e) Usando los resultados del método que te parezca más conveniente, (SVD, NMF) construye un indicador semanal para cada uno de los k tópicos durante el periodo de estudio, basado en su frecuencia de aparición. Normalízalos de manera adecuada para que sean comparables y grafícalos como una serie de tiempo. Lo anterior, puede darte un panorama general de la dinámica de los temas que se han tratado en las conferencias matutinas. Realiza un reporte ejecutivo de tus análisis y hallazgos, resaltando las ventajas y desventajas de las metodologías exploradas y da tus conclusiones, incluyendo sugerencias para mejorar el análisis⁵.

⁴ Puedes usar el módulo `wordcloud` de Python, el cual tiene bastantes ejemplos, incluyendo la opción `generate_from_frequencies`, que puede ser de utilidad.

⁵ En el Moodle del curso, hay un par de artículos de referencia sobre LSA, que quizás puedan servir para ampliar algunos detalles del ejercicio, en caso de ser necesario. En este ejercicio no está permitido usar módulos especializados en LSA, solo aquellos que se mencionan en los incisos.

2.1. SOLUCIÓN.

R. Para proceder, necesitamos cargar los archivos de repositorio. Para ello hemos creado la clase “AMLOMañaneras” en el archivo RepositorioAMLO.py. Con este archivo y clase cargamos todas las conferencias de prensa en una matriz de textos donde cada fila contiene una conferencia del presidente a través de la función matriz_mañaneras del mismo archivo.

La clase permite recuperar las fechas de las mañanera, el texto, el número de semana total de gobierno entre otras cosas, en particular preprocesa los archivos inicialmente mediante el uso del archivo preprocesar.py que contiene la función de preprocesamiento de texto que vimos en la respectiva ayudantía. La configuración que se hace inicialmente es:

- Quitar Acentos
- Remover Stopwords
- Lematizar
- Remover Puntuación.

Lo anterior con el objetivo de limpiar y facilitar la interpretabilidad. Esto inicialmente permite la reducción del vocabulario y facilitara el análisis.

Posteriormente hacemos un segundo preproceso implícito en el uso de TfidfVectorizer, pero antes creamos un lista mayor de stopwords recolectando las stopwords de las librerías: spaCy, NLTK, stopwordsiso y una lista sugerida. Con la unión de ellas generamos una lista final de stopwordsiso que pasaremos al vectorizador de TfidfVectorizer.

La configuración del TfidfVectorizer es la siguiente

- max_df=0.95: Para eliminar palabras demasiado frecuentes que carezcan de significado. Dada la gran cantidad de textos y vocabulario la establecemos en un valor alto.
- min_df=min_df: Este parámetro se establece en función de una cantidad mínima de vocabulario (1000), sin embargo, para el análisis total es de 0.01.
- max_features=tamaño_propuesto: Para nuestro caso usamos un vocabulario de 1200 palabras con el fin de capturar las más importantes. Además de haber probado empíricamente ser un buen número.
- use_idf=True: Para dar más peso a las palabras no tan comunes pero que pueden ser informativas.
- strip_accents='unicode': Quitar acentos que pudieran quedar.
- stop_words=final_stopwords_list.
- ngram_range=(1, 2): Hacemos máximo bigramas que pueden ser informativos. Más de 3 es poco común.

Esta configuración la hacemos dentro de la clase TF_IDF en el archivo AnalisisTopicos.py que mediante su uso obtenemos la representación TF-IDF buscada.

2.2. SOLUCIÓN.

R. La decisión de elegir un k adecuado se tomó ya con las implementaciones hechas. Primero corrimos un test de estabilidad con el criterio Spearman analizando de 2 hasta 35 tópicos logrando la mayor estabilidad con el número 15. Tomando este criterio como base se analizaron posibles k alrededor de ese valor.

Consultamos con un experto en política y se analizaron las dimensiones resultantes, se probó $k = 8, 10, 12, 14, 15, 16$ y se acordó que los mejores modelos eran para $k = 12$ y $k = 15$ bajo el criterio de ser diversos y enfocar temas relevantes.

SVD (12)	NMF (12)	SVD (15)	NMF (15)
Corrupción	Corrupción	Corrupción	Corrupción
Salud	Salud	Salud	Salud
CFE/Energía	CFE/Energía	CFE/Energía	CFE/Energía
Tren Maya	Tren Maya	Tren Maya	Tren Maya
COVID/Vacuna	COVID/Vacuna	COVID/Vacuna	COVID/Vacuna
Seguridad	Seguridad	Seguridad	Seguridad
EU/RE/Migración	EU/RE/Migración	EU/RE/Migración	EU/RE/Migración
Otros	Electoral y Política	Desastre Acapulco	Política
Economía	Economía	Programas Económicos	Programas Económicos
Electoral y Política	PEMEX/Petróleo	Opositores Políticos	PEMEX/Petróleo
Educación	Educación	Educación	Educación
Nuevo Aeropuerto	Nuevo Aeropuerto	Nuevo Aeropuerto	Nuevo Aeropuerto
	Otros	Electoral	
	Economía	Economía	
	Electoral	Desastre Acapulco	

Tabla 2.1: Propuesta de Tópicos para modelos $k = 12, 15$ SVD y NMF

Elegidos 15 pues es más diverso además de ser la propuesta que da el criterio de estabilidad.

Para generar los Word Cloud correspondientes usamos la librería recomendada, además observamos inicialmente que los Word Cloud iniciales no eran muy representativos, por lo mismo aplicamos una rotación para obtener cargas más “extremas”.

Los Word Clouds son:





Figura 2.1. Word Cluods con SVD rotado.

Los nombres asignado a estos Word Cloud se reflejan en la tabla 2.1.

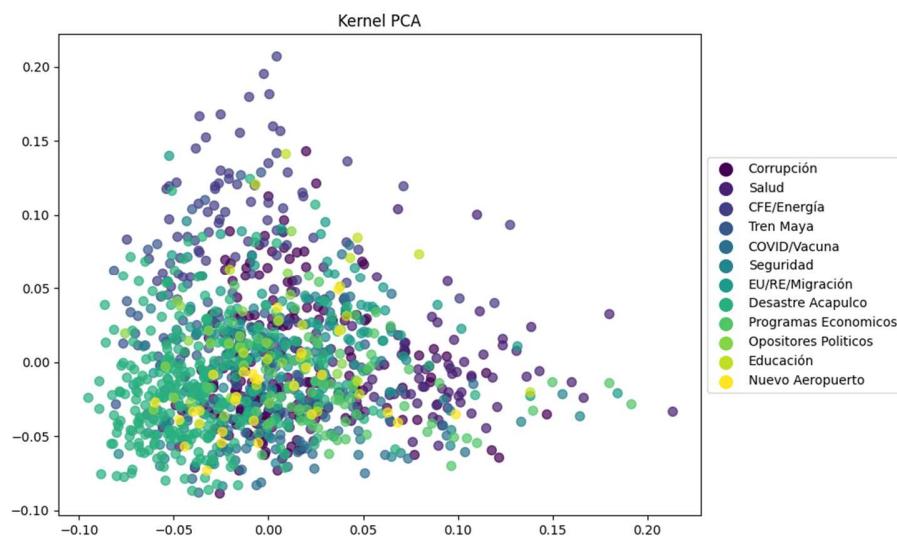


Figura 2.2. Kernel PCA de las mañaneras

2.3. SOLUCIÓN.

R. Usamos el método propuesto y englobamos todos los pasos en una sola clase llamada Top_SVD. Como podemos ver en las furas 2.2, 2.3 y 2.4 la representación no parece arrojar patrones claros. Lo que si podemos observar es la gran cantidad de conferencias asignadas a Corrupción y opositores políticos (se observa mejor ocultando los puntos en el Ploty generado en el ipynb del ejercicio).

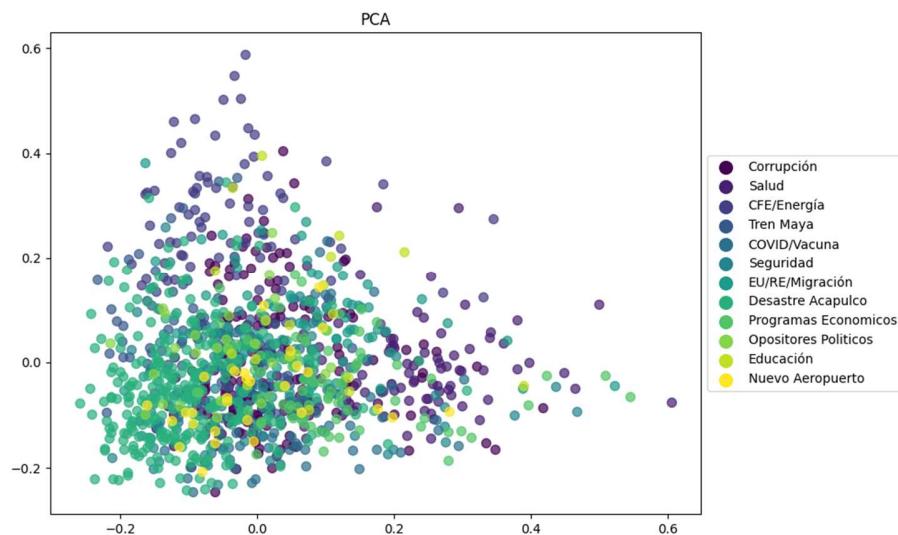


Figura 2.3. PCA de las mañaneras

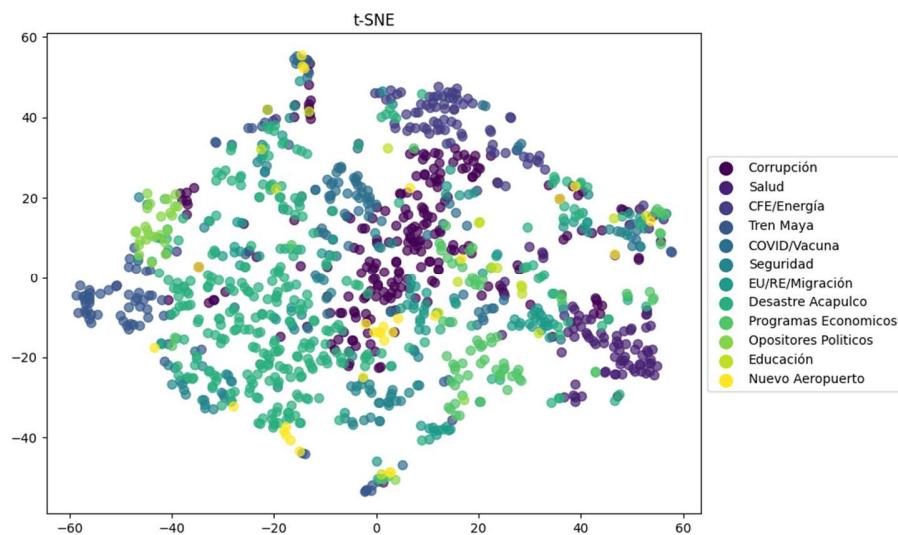


Figura 2.4. t-SNE de las mañaneras

2.4. SOLUCIÓN.

R. Realizamos una clase similar a Top_SVD, llamada Top_NMF donde desarrollamos todos los pasos realizados con SVD. Los Word Clouds son los siguientes:





Figura 2.5. Word Clouds con NMF.

Igualmente, los nombres se pueden consultar en la tabla 2.1.

Este modelo parece separar mejor los componentes, el t-SNE muestra la mejorar en la separación (figura 2.6), por otro lado, Kernel PCA y PCA presentan una mejoría (figuras 2.7 y 2.8).

Nota: Las observaciones son más fácilmente observables en los Ploty interactivo desarrollados en el ipynb del ejercicio.

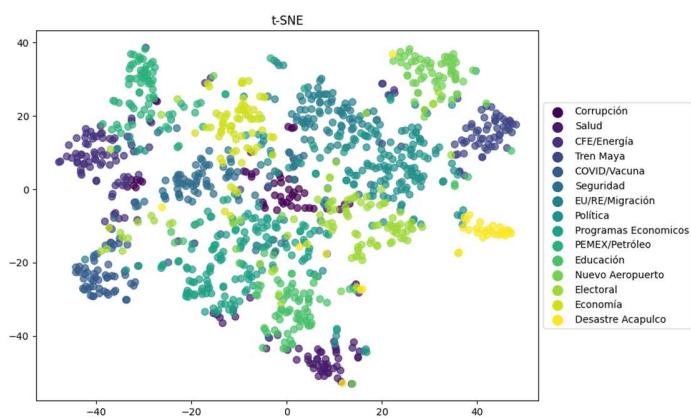


Figura 2.6. t-SNE de las mañaneras NMF

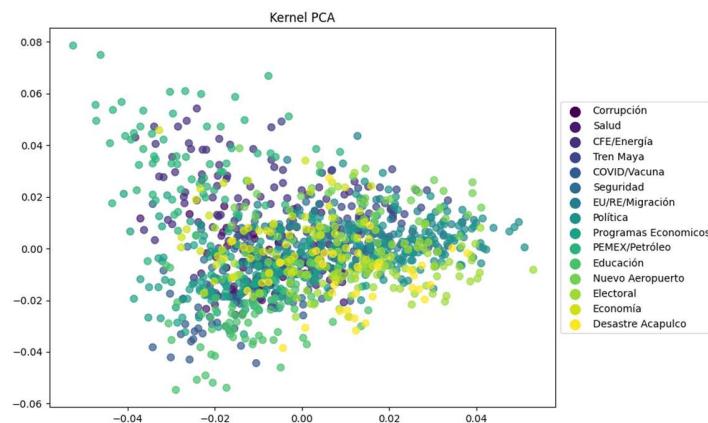


Figura 2.7. Kernel PCA de las mañaneras NMF

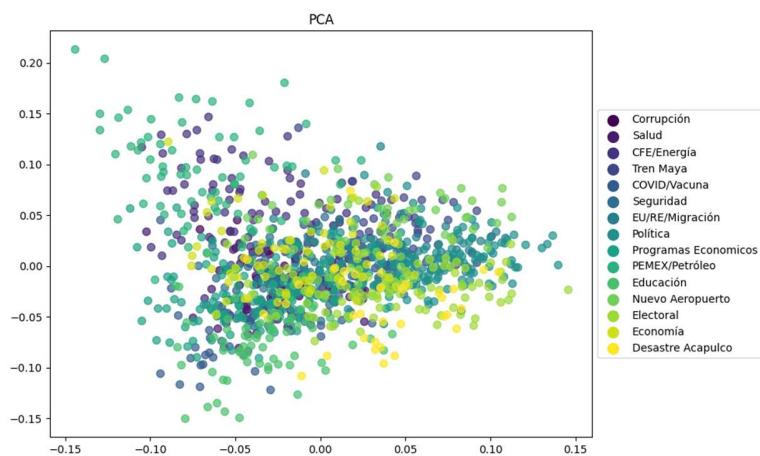


Figura 2.8. PCA de las mañaneras

¿Cuál te parece mejor y porqué?

R. En general ambos métodos me parecen correctos, sin embargo, NMF parece ser mejor a la hora de clasificar los datos. Al realizar la prueba con diferentes k por lo general era más fácil asignarle un nombre al tópico. Además, el hecho tener clara la estructura a diferencia de SVD, donde no se sabe que son los valores negativos me parece fundamental para obtener confiabilidad y certeza.

2.5. SOLUCIÓN.

R. El índice propuesto es:

$$I_{i,j} = \frac{f_{i,j}}{\max_j(f_{i,j})} \times \frac{s_i}{\max_i(s_i)} \times e^{-H_j}$$

Donde:

- $f_{i,j}$: Representa la frecuencia de aparición del tópico i en la semana j .
- $\max_j(f_{i,j})$: Es la máxima frecuencia de cualquier tópico durante la semana j , lo que normaliza $f_{i,j}$ en una escala de 0 a 1, donde 1 significa que el tópico fue el más discutido esa semana.
- s_i : Es el puntaje acumulado del tópico i a lo largo de un periodo (mensual), que refleja la importancia general del tópico.
- $\max_i(s_i)$: Es el máximo puntaje acumulado de cualquier tópico en el mismo periodo, utilizado para normalizar s_i .
- e^{-H_j} : Es un factor exponencial basado en el Índice de Shannon H_j , que mide la diversidad de tópicos en la semana j . Este término reduce el índice en proporción a la diversidad, significando que un mayor valor de H_j (mayor diversidad) reduce la relevancia percibida del tópico.

Adonde:

$$H_j = - \sum_{i=1}^n p_i \log(p_i)$$

- p_i : Representa la proporción del total de discusiones que el tópico i representa en la semana j .
- n : Es el número total de tópicos diferentes discutidos durante la semana j .
- $\log(p_i)$: Se utiliza el logaritmo natural de p_i , reflejando el aporte de cada tópico a la diversidad total.

Este índice busca atrapar la relevancia de los tópicos en las mañaneras, considerando tanto la importancia general del tópico en un marco temporal (en este caso, mensual) pero no global (pues puede ser poco significativo) y la diversidad (Incluyendo el índice de Shannon) de los tópicos discutidos cada semana. Pueden checar el reporta para más detalle,

La normalización elegida es min-max, reduciendo todo a un intervalo 0-1. Sin embargo, los resultados son muy similares a los obtenidos estandarizando.

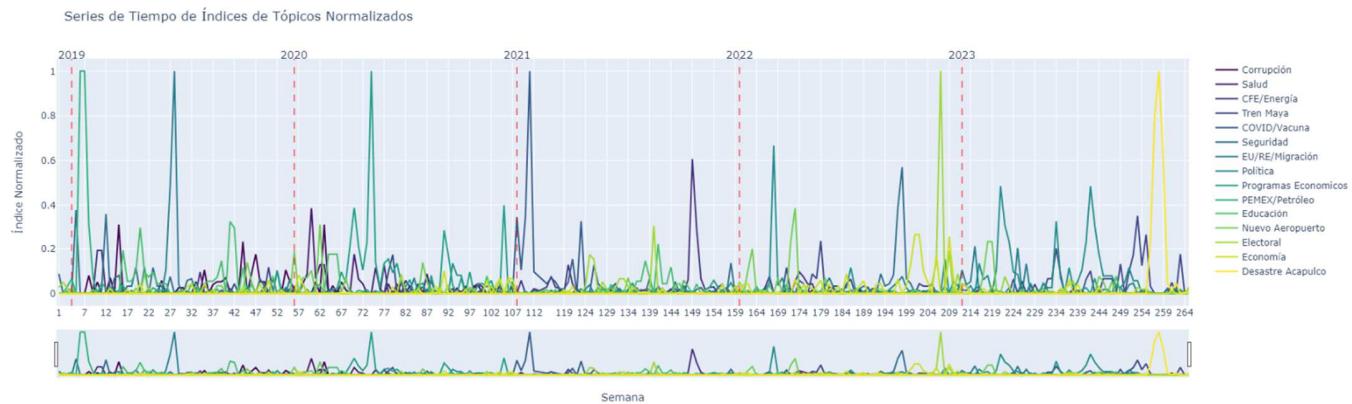


Figura 2.9. Lineal del tiempo de Tópicos Normalizados 2019-2023

Se anexa el reporte de este ejercicio.