Universidad Autónoma de Entre Ríos

Facultad de Ciencia y Tecnología

MODELADO DE TÓPICOS EN TEXTOS CORTOS E INFORMALES: ADAPTACIÓN Y EVALUACIÓN PARA EL ESPAÑOL ARGENTINO

Tesistas

Federico Ruhl y Gonzalo Ramos Muzio

Director

Dr. Carlos Germán Sedano

Docentes

Lic. Monica Vera, Lic. Pamela Bonadeo y Lic. Damian Cian

2026

Dedicatoria

A nuestras familias y parejas, por su amor incondicional y apoyo constante.

A nuestros amigos, por su comprensión y paciencia en los momentos de mayor dedicación.

A todos aquellos que creyeron en nosotros cuando nosotros mismos dudábamos.

Agradecimientos

En primer lugar, queremos expresar nuestro más sincero agradecimiento a nuestro director de tesis, Dr. Carlos Germán Sedano, por su guía, paciencia y apoyo genuino e inagotable durante todo este proceso. Su gran conocimiento y sus valiosas sugerencias han sido fundamentales para el desarrollo de este trabajo.

Agradecemos también a los miembros del tribunal de tesis por dedicar su tiempo a la evaluación de este trabajo y por sus constructivos comentarios.

A nuestros profesores y mentores a lo largo de nuestra formación académica, quienes nos proporcionaron las herramientas necesarias para enfrentar este desafío.

Finalmente, y no por ello menos importante, queremos agradecer a nuestras familias y amigos.

A todos ustedes, nuestro más profundo agradecimiento.

Índice de Contenidos

Índice de Tablas

Índice de Figuras

Resumen

El modelado de tópicos es una técnica de procesamiento de lenguaje natural que permite descubrir temas relevantes en grandes volúmenes de texto. Sin embargo, aplicarlo a textos cortos e informales en español argentino, como los que circulan en redes sociales, presenta desafíos particulares por las características propias del idioma y el alto nivel de ruido textual. El presente trabajo aborda estos desafíos comparando modelos de modelado de tópicos específicamente adaptados para este contexto. Primeramente, se realizó la validación en el idioma inglés usando el entorno OCTIS conjuntamente con BERTopic, planteada por el autor. Luego, se llevó a cabo una comparativa en español argentino, analizando publicaciones de redes sociales de la República Argentina. Finalmente, se evaluaron tres modelos: Latent Dirichlet Allocation (LDA), Non-negative Matrix Factorization (NMF) y BERTopic usando métricas de coherencia y diversidad, destacando sus ventajas y limitaciones para este tipo de datos.

A partir de la evaluación realizada, se propone una metodología para el preprocesamiento y modelado de tópicos adecuada para las particularidades del español argentino en textos cortos e informales.

Palabras claves: modelado de tópicos, español argentino, BERTopic, LDA, NMF.

Abstract

Topic modeling is a natural language processing technique that enables the discovery of relevant themes in large volumes of text. However, applying it to short and informal texts in Argentinian Spanish — such as those found on social media — presents particular challenges due to the linguistic characteristics of the language and the high level of textual noise. This work addresses these challenges by comparing topic modeling approaches specifically adapted for this context. First, the proposed approach is validated in English using the OCTIS framework along with the incorporation of BERTopic, as proposed by the author. Then, a comparative analysis is conducted in Argentinian Spanish, using social media posts from Argentina. Finally, three models — Latent Dirichlet Allocation (LDA), Non-negative Matrix Factorization (NMF), and BERTopic — are evaluated using coherence and diversity metrics, highlighting their strengths and limitations for this type of data.

Based on the evaluation conducted, a methodology is proposed for preprocessing and topic modeling that is well-suited to the particularities of Argentinian Spanish in short and informal texts.

Introducción

En los últimos años, se ha observado un notable incremento en el uso de las tecnologías de la información y la comunicación (TIC), lo cual ha dado lugar a la generación y manejo de grandes volúmenes de datos en diversos campos de investigación (Milne y Watling, 2019).

Parte de estos datos se producen o generan a través de comentarios y mensajes en redes sociales, las cuales juegan un rol fundamental, ya que abordan una amplia diversidad de tópicos de interés general. Pero esta ventaja en la generación de grandes volúmenes de datos, trae aparejado como desventaja que, este tipo de fuentes, presenta una predominancia de textos cortos e informales, que requieren un minucioso trabajo de procesamiento previo a su utilización.

Esta característica textual, ha motivado que numerosos autores investiguen y desarrollen diversas técnicas de preprocesamiento, especialmente en el idioma inglés, tanto de manera general (Chai, 2023; Larriva-Novo y cols., 2021; Tabassum y Patil, 2020; Vijayarani y cols., 2015) como específica para el modelado de tópicos (Zimmermann y cols., 2024; Churchill y Singh, 2021; Wachirapong, 2023). En cuanto al español latinoamericano, la literatura disponible se enfoca en contextos generales (Talamé y cols., 2019; Tessore y cols., 2019; Orellana y cols., 2018), sin encontrarse estudios específicos sobre el modelado de tópicos.

Debido a la naturaleza de los textos provenientes de bancos de datos extraídos de redes sociales, su contenido puede variar considerablemente según la red social involucrada, el período temporal en que se obtuvieron los datos y los regionalismos lingüísticos asociados al origen geográfico y sociocultural del usuario.

Por ello, uno de los desafíos del presente trabajo es determinar las técnicas y herramientas idóneas para preparar los textos provenientes de redes sociales, en el idioma

español argentino, para emprender la próxima etapa que consta de la validación y comparación de los modelos.

Modelos de tópicos

Para la selección de modelos se tienen en cuenta tanto aquellas alternativas convencionales como también emergentes, de manera similar al análisis comparativo de Egger y Yu (2022).

La comparación de estos modelos elegidos, permitirá evaluar su aplicabilidad o uso en contextos hispanohablantes, especialmente en textos informales y de alta variabilidad lingüística, como los que se extraen de redes sociales.

Para evaluar la calidad de los modelos, se emplearán métricas ampliamente utilizadas en el ámbito del modelado de tópicos, como son la coherencia y la diversidad. En particular, dos variantes de la métrica de coherencia temática, tales como:

- *Normalized Pointwise Mutual Information* (NPMI): métrica que mide la coherencia semántica entre los términos más representativos de cada tópico en base a su co-ocurrencia dentro de un corpus de referencia.

- *Coherence Value* (CV): métrica híbrida que combina medidas indirectas de similitud semántica y estadísticas de co-ocurrencia para proporcionar una evaluación robusta y estable (Röder y cols., 2015).

- *Diversity*: métrica que permite cuantificar la proporción de términos únicos en los tópicos extraídos, penalizando la redundancia léxica y promoviendo la generación de tópicos más informativos y diferenciados (Dieng y cols., 2019).

Las métricas descriptas precedentemente, permiten establecer comparaciones objetivas entre los modelos seleccionados, facilitando así la identificación del enfoque más adecuado para representar el contenido de textos breves e informales en idioma español argentino.

En este contexto, se analizarán los siguientes modelos:

- Latent Dirichlet Allocation (LDA): modelo ampliamente utilizado en diferentes contextos y propuesto originalmente por Blei y cols. (2003).
- Non-negative Matrix Factorization (NMF): modelo que destaca por su capacidad de descomponer matrices de datos en componentes interpretables, facilitando la identificación de patrones ocultos en textos (D. D. Lee y Seung, 1999).
- BERTopic: modelo que permite combinar técnicas de machine learning con modelos de lenguaje a fin de generar tópicos con mayor nivel de coherencia y relevancia, adaptándose dinámicamente a la estructura de los datos (Grootendorst, 2022).

En cuanto a la redacción del informe, se tomó como referencia el proceso de investigación descrito por Hernández-Sampieri y cols. (2014), así como las directrices APA de citación y presentación recogidas en el *Publication Manual* de la American Psychological Association (American Psychological Association, 2020).

Objetivos

Objetivo General

Evaluar modelos de modelado de tópicos en textos cortos e informales escritos en idioma español argentino e identificar aquel con mejor desempeño en términos de coherencia y diversidad, utilizando OCTIS como marco de referencia.

Objetivos Específicos

- Seleccionar y validar los modelos de modelado de tópicos en inglés, en base a las métricas de coherencia y diversidad.
- Investigar, seleccionar técnicas y herramientas para el preprocesamiento de textos cortos e informales.
- Adaptar las técnicas y herramientas seleccionadas para el idioma español argentino y preprocesar los datos.
- Comparar y seleccionar el modelo con mejor desempeño para los conjuntos de datos preprocesados interpretando los resultados obtenidos.

El propósito de la computación es la comprensión, no los números.

— Richard Hamming

Capítulo 1: Análisis Temático en Textos Informales

En el presente capítulo se analizarán los fundamentos del procesamiento del lenguaje natural, los tipos de modelado de tópicos, sus implementaciones más reconocidas y el preprocesamiento de textos informales.

1.1 Procesamiento del lenguaje natural

El procesamiento del lenguaje natural (NLP, por sus siglas en inglés) es un campo que combina los aportes de la lingüística computacional, las ciencias de la computación y la inteligencia artificial (IA), con el objetivo de permitir que las máquinas comprendan, analicen y extraigan el significado del lenguaje humano (Albalawi y cols., 2020).

A continuación, se presenta una definición de las tres áreas de conocimiento, nombradas anteriormente, con la finalidad de comprender la íntima relación necesaria para el abordaje del presente trabajo:

- Según Grishman (1986), se puede definir la lingüística computacional como “el estudio de los sistemas de computación utilizados para la comprensión y la generación de las lenguas naturales”.

- Las ciencias de computación, por su parte, tratan el estudio sistemático de los procesos algorítmicos que describen y transforman la información: su teoría, análisis, diseño, eficiencia, implementación y aplicación (Denning, 1985)

- La inteligencia artificial es acuñada por McCarthy (1956) como la ciencia e ingeniería de hacer máquinas inteligentes, especialmente programas de computación inteligentes.

De acuerdo a Sandoval (1998) "La Inteligencia Artificial (IA) se encarga de codificar en un programa facultades cognitivas como la inferencia, la toma de decisiones y la

adquisición de conocimiento experto. En este sentido, la LC[1] es una parte integrante de la IA, de la misma forma que para muchos lingüistas la Lingüística es parte de la Psicología por tratar de una de las capacidades cognitivas por excelencia, el lenguaje. De hecho, las aproximaciones al NLP desde la IA adoptan una perspectiva más global: el tratamiento de la estructura lingüística (sintaxis y semántica, básicamente) se integra como un módulo de entrada/salida dentro de un sistema compuesto por más conocimientos."

El NLP tiene como objetivo crear modelos computacionales del lenguaje lo suficientemente detallados que permitan escribir programas informáticos que realicen las diferentes tareas donde interviene el lenguaje natural (Allen, 1995).

Según Jurafsky y Martin (2025), el desarrollo alcanzado en este campo permite realizar múltiples tareas, tales como:

- Traducción (por ejemplo, con Google Translate o DeepL).
- Respuesta a preguntas y consulta de información (por ejemplo, con Google Search).
- Chatbots y sistemas de dialogo (por ejemplo, con ChatGPT de OpenAI).
- Conversión de voz a texto, y viceversa (por ejemplo, con Google Speech-to-Text).

Entre las múltiples tareas que permite el procesamiento del lenguaje natural, se destaca la posibilidad de analizar grandes volúmenes de datos, con el fin de identificar patrones temáticos subyacentes. Este análisis resulta especialmente útil en contextos donde el contenido es abundante, heterogéneo y no estructurado, como ocurre con los datos provenientes de redes sociales.

En este marco, surge el modelado de tópicos, una técnica estadística, no supervisada, que permite identificar los temas ocultos, presentes en un conjunto de documentos, a partir de las palabras que los componen (Blei, 2012).

[1] Lingüística Computacional.

1.2 Modelado de tópicos

El modelado de tópicos comprende un conjunto de técnicas computacionales diseñadas para descubrir la estructura temática oculta en grandes colecciones de textos. Mediante este enfoque, se identifican grupos de palabras que tienden a aparecer de manera conjunta; dichos conjuntos se interpretan como tópicos, los cuales se definen formalmente como distribuciones probabilísticas sobre un vocabulario fijo (Blei, 2012).

A continuación se definen los principales conceptos del modelado de tópicos, según Blei y cols. (2003):

- Palabra: unidad básica de los datos discretos, definida como un elemento de un vocabulario indexado por $\{1, \ldots, V\}$. Las palabras pueden representarse mediante vectores unitarios de dimensión V, donde el componente correspondiente a la palabra es igual a uno y el resto es cero.
- Documento: secuencia de N palabras, denotada por $w = (w_1, w_2, \ldots, w_N)$, donde w_n es la n-ésima palabra de la secuencia.
- Corpus: colección de M documentos, denotada por $\mathcal{D} = \{w_1, w_2, \ldots, w_M\}$.

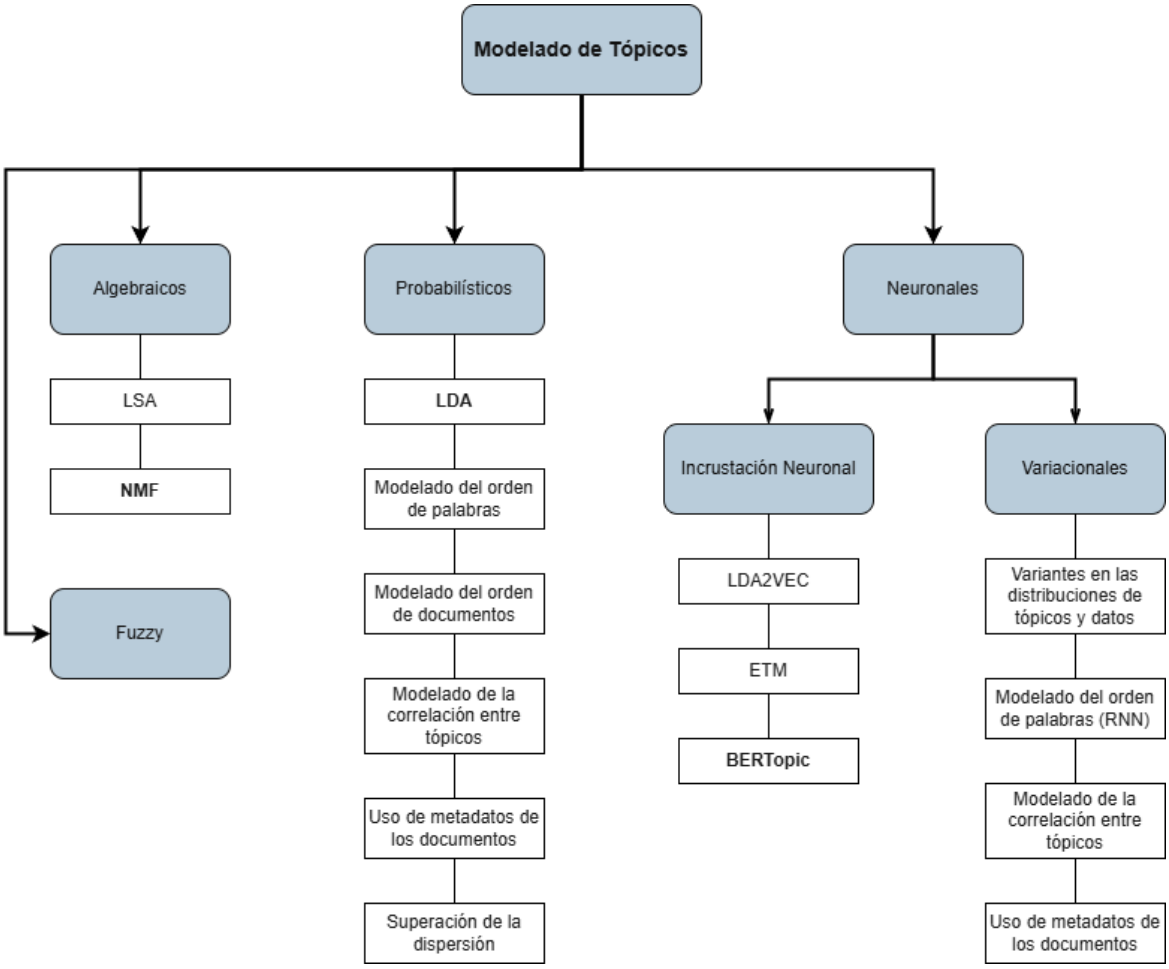
Con la terminología establecida, las distintas aproximaciones al modelado de tópicos que la literatura ha propuesto a lo largo de los años se agrupan en cuatro categorías bien definidas —algebraico, probabilístico, fuzzy y neuronal—, cada una con ventajas, limitaciones y ámbitos de aplicación propios.

1.2.1 Clasificación de modelos de modelado de tópicos

La siguiente figura resume la taxonomía de los modelos de modelado de tópicos adoptada en la literatura.

Figura 1

Modelado de tópicos.



Nota. Adaptado de “Topic modeling algorithms and applications: A survey”, por A. Abdelrazek, Y. Eid, E. Gawish, W. Medhat y A. Hassan Yousef, 2022, *Information Systems*, *112*. https://doi.org/10.1016/j.is.2022.102131. Derechos de autor por Elsevier.

Modelos Algebraicos. Este tipo de modelo se distingue por su simplicidad conceptual, facilidad de implementación y por no requerir la configuración de hiperparámetros complejos. Contempla modelos descriptivos que permiten identificar patrones latentes en los datos, pero sin modelar explícitamente el proceso generativo que los origina: no intentan explicar cómo se generan los documentos a partir de una estructura latente de tópicos.

El enfoque algebraico, en lugar de asumir una distribución probabilística subyacente, se basa en descomposiciones algebraicas —como la factorización matricial— que capturan

correlaciones estáticas entre términos y documentos. Esta característica puede limitar su capacidad de generalización en contextos con ruido —entendido como cualquier información que no aporta valor semántico relevante para el análisis y que debe ser eliminada o normalizada (Symeonidis y cols., 2018)—, escasez de datos o alta variabilidad, donde los modelos probabilísticos suelen ofrecer una mayor robustez y flexibilidad.

El primer modelo representativo de este enfoque fue Latent Semantic Indexing (LSI) (Deerwester y cols., 1990), el cual emplea una representación de bolsa de palabras (BoW, por sus siglas en inglés), modelando el corpus mediante una matriz documento-término (DTM), sobre la cual se aplica una Descomposición en Valores Singulares (SVD, por sus siglas en inglés). A través de esta técnica, se extraen los tópicos como componentes latentes asociados a los valores singulares más altos, que representan los patrones dominantes en el corpus.

No obstante, aunque LSI permite descubrir asociaciones semánticas implícitas entre los términos, presenta limitaciones en cuanto a su interpretabilidad y su capacidad para manejar polisemia o ambigüedad semántica.

Además, otro modelo representativo de este enfoque es Non-Negative Matrix Factorization (NMF), el cual descompone matrices de alta dimensión en representaciones de menor dimensión preservando la restricción de no negatividad.

Modelos Probabilísticos. En segundo lugar, los modelos Bayesiano probabilísticos emplean un enfoque estadístico que utiliza el teorema de Bayes para modelar la incertidumbre y actualizar las creencias a medida que se obtienen nuevos datos. Entendiéndose como creencia a la distribución de probabilidad que representa incertidumbre sobre un parámetro desconocido (Gelman y cols., 2013). Estas distribuciones se clasifican en:

- *Prior*: distribución inicial que representa el conocimiento sobre un parámetro θ antes de observar datos. Para un dado justo:

$$P(\theta) = \text{Dirichlet}(\mathbf{1}_6) \quad \text{(Distribución para las seis caras)} \tag{1.1}$$

- *Posterior*: distribución actualizada al incorporar n observaciones $\mathbf{X} = (x_1, \ldots, x_n)$. Si observamos $k = 30$ veces el '6' en $n = 100$ tiradas:

$$P(\theta \mid \mathbf{X}) = \text{Dirichlet}(\mathbf{1}_6 + \mathbf{k}) \quad \text{con } \mathbf{k} = (0, 0, 0, 0, 0, 30) \tag{1.2}$$

Ejemplo numérico. El valor esperado de la probabilidad de la cara '6' bajo la distribución posterior se calcula como:

$$P(\theta_6 \mid \mathbf{X}) = \frac{1 + 30}{6 + 100} \approx 0{,}292 \quad (\text{vs. } 0{,}166 \text{ en el Prior}) \tag{1.3}$$

Sugiriendo un posible sesgo (carga) hacia dicha cara.

Los modelos temáticos de este tipo trabajan con grafos, donde los nodos representan variables aleatorias, y la presencia de aristas indica la dependencia probabilística entre dos variables.

Latent Dirichlet Allocation (LDA) es uno de los modelos generativos probabilísticos más utilizados debido a su relativa simplicidad y eficacia para modelar temas en grandes colecciones de texto, el mismo es un modelo jerárquico de tres niveles, en el que cada elemento perteneciente a una colección se modela como una combinación finita sobre un conjunto subyacente de temas. A su vez, cada tema es una distribución de probabilidad sobre un vocabulario finito de palabras (Blei y cols., 2003).

Además de LDA, existen otros modelos bayesianos para el análisis de tópicos, como Hierarchical Dirichlet Process (HDP), que permite un número potencialmente infinito de temas, o Correlated Topic Model (CTM), que permite capturar correlaciones entre temas al utilizar una distribución logística normal en lugar de una *Dirichlet*, como en el caso de LDA. (Lafferty y Blei, 2005; Teh y cols., 2006).

Modelos Fuzzy. Los modelos difusos (o fuzzy), se basan en la teoría de conjuntos difusos propuesta por Zadeh (1965), en la cual los documentos pueden pertenecer simultáneamente a múltiples tópicos con distintos grados de pertenencia (Yang, 1993). A diferencia de los enfoques rígidos (*hard clustering*), que asignan cada instancia a un único clúster, y también de los modelos probabilísticos como LDA, donde la pertenencia a temas se modela como una distribución de probabilidad que debe sumar uno, los modelos difusos utilizan valores de membresía independientes que no requieren dicha normalización. Esto les permite representar relaciones más flexibles entre tópicos, lo cual resulta particularmente útil en textos breves o en dominios con alta superposición semántica.

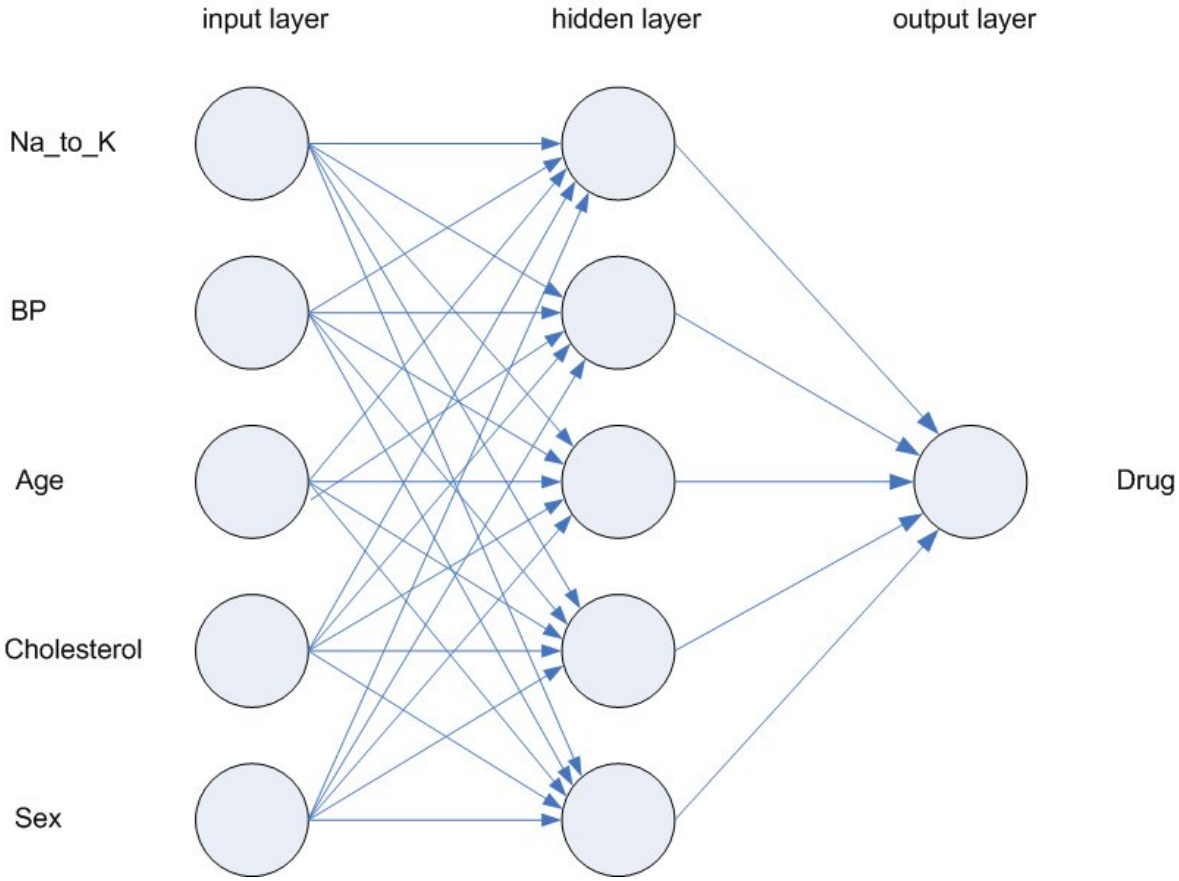
Entre sus principales ventajas, se destaca su capacidad de adaptación ante la escasez de datos (*sparsity*), ya que incorporan ponderaciones semánticas entre términos para representar grados de pertenencia continua a distintos temas, lo que permite capturar matices semánticos que otros enfoques tienden a forzar o descartar.

Asimismo, los modelos difusos tienden a mejorar la coherencia temática al incorporar el uso de similitudes léxicas. Por ejemplo, Akhtar y cols. (2019) demuestran que una representación difusa de documentos sobre una estructura LDA —donde las palabras clave se asignan con valores de pertenencia derivados de su similitud semántica— logra una calidad temática superior respecto al LDA clásico y a variantes con simple ponderación de términos.

Modelos Neuronales. Las redes neuronales son modelos simples del funcionamiento del sistema nervioso, que emulan el modo en que el cerebro humano procesa la información, analizando un número elevado de unidades de procesamiento interconectadas (IBM, s.f.).

Figura 2

Estructura de una red neuronal.



Nota. Adaptado de “Modelos de redes neuronales”, por IBM. Recuperado de https://www.ibm.com/docs/es/spss-modeler/saas?topic=networks-neural-model. Derechos de autor por IBM.

Una red neuronal típica se compone de tres tipos de capas: una capa de entrada, que recibe los datos; una o más capas ocultas, donde se lleva a cabo el procesamiento; y una capa de salida, que entrega el resultado del modelo, como se puede observar en la Figura 2.

La red “aprende” a partir de registros individuales. Al inicio, los pesos sinápticos se asignan de forma aleatoria y las predicciones son incorrectas. En el entrenamiento supervisado, cada ejemplo incluye un resultado conocido: la predicción de la red se compara con el valor esperado y el error se propaga hacia atrás mediante retropropagación, ajustando

gradualmente los pesos. Este ciclo se repite hasta alcanzar uno o varios criterios de parada; con cada iteración mejora la capacidad de predicción. Entrenada la red, se aplica a casos nuevos cuya etiqueta es desconocida.

Entre los modelos neuronales aplicados al modelado de tópicos, se encuentran aquellos que integran representaciones semánticas distribuidas, como LDA2Vec (Moody, 2016), el cual combina el enfoque temático de LDA con los vectores de palabras aprendidos por Word2Vec (Mikolov y cols., 2013), enriqueciendo así la representación de los documentos.

Word2Vec, en una de sus variantes denominada skip-gram, que predice las palabras del contexto a partir de una palabra central, define una ventana deslizante sobre el texto y, para cada palabra pivote en dicha ventana, busca maximizar la probabilidad de ocurrencia de las palabras del contexto. Esto genera representaciones vectoriales densas que capturan relaciones semánticas entre palabras basadas en su distribución contextual.

LDA2Vec extiende este esquema al introducir un vector adicional que representa al documento, el cual se suma al vector de la palabra pivote antes de predecir las palabras del contexto. Este vector de documento, sin embargo, resulta denso y poco interpretable. Para abordar esto, se lo descompone en una combinación lineal de vectores latentes que conforman una matriz de tópicos, ponderados por coeficientes que indican la proporción de cada tópico en el documento. Estas ponderaciones se regularizan mediante distribuciones previas de Dirichlet, lo que permite conservar la interpretabilidad temática del modelo. En conjunto, esta arquitectura logra capturar tanto el contexto local de las palabras como la estructura global de los temas en los documentos, superando en muchos casos a los enfoques puramente estadísticos.

Por otra parte, recientemente se han desarrollado modelos tales como BERTopic (Grootendorst, 2022), los cuales aprovechan representaciones semánticas generadas por modelos de lenguaje preentrenados basados en Transformers, como BERT o RoBERTa.

A partir de estos *embeddings* contextuales, aplica técnicas de reducción de dimensionalidad para proyectar los vectores en un espacio de menor dimensión y, sobre ese espacio, agrupamiento jerárquico no supervisado para identificar documentos similares y descubrir temas latentes. Este enfoque neuronal permite identificar tópicos con alta coherencia semántica, incluso en textos cortos o desestructurados.

Con las cuatro categorías anteriores —algebraicos, probabilísticos, fuzzy y neuronales— queda completada la descripción de la clasificación mostrada en la Figura 1, sirviendo de base para la descripción en profundidad de los modelos del presente trabajo.

1.2.2 Latent Dirichlet Allocation (LDA)

LDA es un modelo que, como su nombre lo indica, asigna (*Allocation*) palabras a tópicos ocultos (*Latent*) utilizando distribuciones de Dirichlet como prior, las cuales son distribuciones multivariadas sobre vectores de proporciones (valores positivos que suman uno) que modelan la mezcla de temas en cada documento.

LDA fue introducido por Blei y cols. (2003) como un modelo generativo, probabilístico, no supervisado y de naturaleza estadística, siguiendo un enfoque bayesiano, como se presentó previamente en la Figura 1, dentro de los modelos probabilísticos. El modelo se enmarca dentro de un esquema jerárquico de tres niveles donde cada documento se representa como una distribución de probabilidad sobre un conjunto finito de tópicos, y cada tópico, a su vez, se define mediante una distribución de probabilidad sobre el vocabulario[2]. Una de sus principales ventajas es que no requiere supervisión y permite inferir los tópicos directamente de una colección de documentos, sin necesidad de contar con conocimientos previos sobre su contenido.

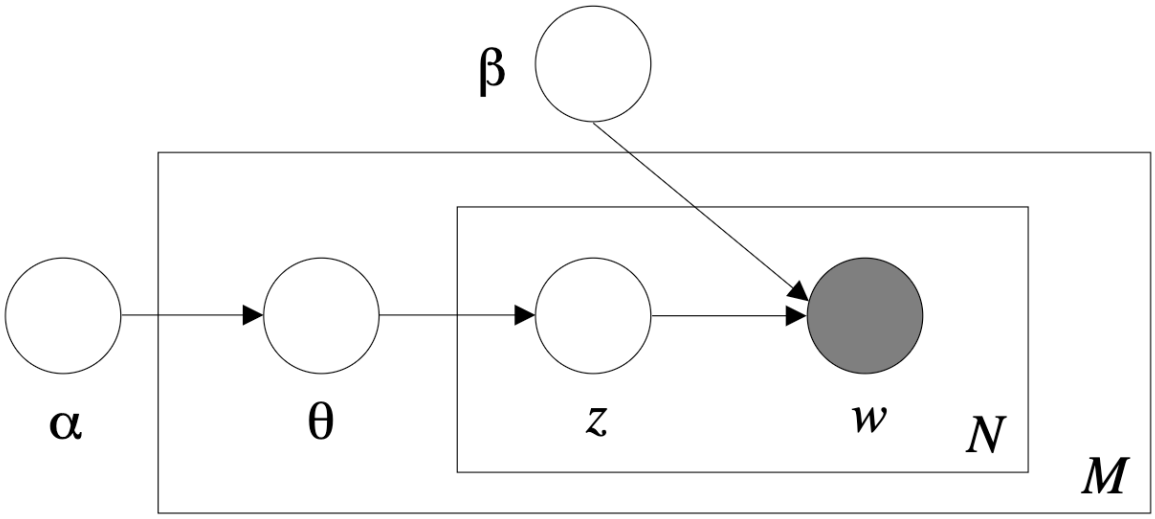
En el modelado de tópicos, LDA es el modelo más utilizado (Shen y cols., 2022; Weisser y cols., 2022; Vukanti y Jose, 2021; Mohammed y cols., 2020), disponible en herramientas ampliamente reconocidas como Machine Learning for Language Toolkit (MALLET),

[2] En este contexto, el vocabulario se refiere al conjunto de todas las palabras distintas que aparecen en el corpus analizado.

Gensim y también Stanford Topic Modeling Toolbox (TMT) (Albalawi y cols., 2020). Asimismo, se encuentra integrado en el sistema Optimización y Comparación de Modelos de Tópicos es Simple (OCTIS, por sus siglas en inglés) (Terragni y cols., 2021), una plataforma destacada por su capacidad de facilitar la optimización y comparación de distintos modelos de modelado de tópicos, a través de diversas métricas, bajo condiciones controladas y reproducibles.

Figura 3

Estructura de LDA.



Nota. Tomado de “Latent dirichlet allocation”, por D. M. Blei, A. Y. Ng y M. I. Jordan, 2003, *Journal of Machine Learning Research*, *3*.
https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf.

Como se presentó en la Figura 3, la estructura del modelo LDA se organiza en tres niveles jerárquicos. En el diagrama se representan el hiperparámetro α, las distribuciones θ, las asignaciones z, las palabras observadas w, la matriz β, y los índices M y N_d (número de documentos y de palabras por documento); el hiperparámetro η y las distribuciones φ_k forman parte del modelo pero no figuran en la representación gráfica estándar.

Significado de las Variables del Modelo. A modo de simplificación se presentan las principales variables del modelo LDA:

Tabla 1: Significado de las variables del modelo LDA

Símbolo	Descripción
α	Hiperparámetro de la distribución Dirichlet previa sobre temas por documento
θ_d	Distribución de temas para el documento d
η	Hiperparámetro de la distribución Dirichlet previa sobre palabras por tema
φ_k	Distribución de palabras para el tema k
β	Matriz de parámetros cuyas filas corresponden a las distribuciones φ_k
z_{dn}	Tema asignado a la palabra n del documento d
w_{dn}	Palabra observada en la posición n del documento d
M	Número total de documentos en el corpus
N_d	Número de palabras en el documento d
K	Número de temas

Nota. En el modelo original de LDA (Blei y cols., 2003), β representa la matriz de parámetros: el conjunto de todas las distribuciones de palabras para cada tema (φ_k). En este enfoque, β se estima directamente a partir de los datos y no se le asigna una distribución previa. Sin embargo, en el mismo trabajo se presenta una variante denominada Smoothed LDA Model, en la que sí se introduce un prior Dirichlet sobre cada fila de β, parametrizado por η. Este enfoque bayesiano completo ha sido adoptado y extendido en implementaciones modernas, como LDA Online (Hoffman y Blei, 2010). Por lo tanto, en la literatura actual, es importante distinguir entre β (la matriz de parámetros) y η (el hiperparámetro del prior Dirichlet sobre palabras por tema).

- Nivel de corpus: según Blei y cols. (2003), α y η son hiperparámetros a nivel de corpus, asumidos como conocidos durante el proceso generativo. De ellos, α es un vector de valores positivos que parametriza la distribución Dirichlet previa sobre las proporciones de temas (θ). Por otra parte, η es el hiperparámetro de la distribución Dirichlet previa sobre las distribuciones de palabras por tema. Ambos hiperparámetros determinan las propiedades de las distribuciones resultantes: α controla cuán concentrada o dispersa será la mezcla de temas en cada documento, mientras que η define qué tan concentradas o diversas serán las palabras dentro de cada tema.

- Nivel de documento: como parte del proceso generativo, para cada documento d se genera una distribución de temas θ_d. Esta distribución se obtiene aleatoriamente a partir de una distribución Dirichlet con parámetro α. En este contexto, θ_d representa las proporciones con las que se combinarán los distintos temas en el documento.

- Nivel de palabras: una vez determinada la distribución de temas para un documento (θ_d), se genera cada una de las N_d palabras del documento mediante el siguiente proceso:

    1. Se selecciona un tema z_{dn} para la palabra en la posición n del documento d, eligiéndolo aleatoriamente según la distribución de temas θ_d.
    2. Luego, se elige una palabra w_{dn} a partir de la distribución de palabras asociada al tema z_{dn}. Esta distribución está parametrizada por la matriz de parámetros β, que define la probabilidad de cada palabra dentro de cada tema.

  Cabe destacar que el índice d, que representa a cada documento, no aparece explícitamente como nodo en el gráfico. Esto se debe a que la caja exterior implica que el proceso descrito se repite M veces, una por cada documento del corpus (Ver Figura 3).

Este procedimiento refleja la suposición generativa del modelo: que las palabras de un documento se generan eligiendo, para cada palabra, un tema z_{dn} de acuerdo con una distribución multinomial parametrizada por θ_d. Posteriormente, se selecciona una palabra w_{dn} a partir de la distribución de palabras correspondiente al tema asignado.

En este modelo, θ_d y z_{dn} son variables latentes: θ_d es una distribución de temas específica del documento d, que se asume generada a partir de una distribución Dirichlet con parámetro α; y z_{dn} representa la asignación de tema para la n-ésima palabra del documento d. Por su parte, β es la matriz de parámetros a nivel de corpus: una matriz de tamaño $K \times V$ que define, para cada uno de los K temas, una distribución de probabilidad sobre el vocabulario de tamaño V.

En la práctica, los documentos ya están escritos, y una vez determinados los hiperparámetros α y η, el objetivo principal del modelo es inferir las variables latentes θ_d y z_{dn} que revelan la estructura temática subyacente. La determinación de los hiperparámetros puede realizarse como un paso previo, ya sea fijándolos a priori en base al conocimiento del dominio o estimándolos mediante técnicas de maximización. En este contexto, los autores del modelo sugieren un enfoque bayesiano empírico para determinar estos hiperparámetros (Blei y cols., 2003).

Inferencia y Actualización de Parámetros. Como se estableció anteriormente en el apartado 1.2.2, el objetivo principal del modelo es inferir las variables latentes θ_d y z_{dn} que mejor explican las observaciones. Las principales variables latentes en LDA incluyen las distribuciones de temas por documento (θ_d) y las asignaciones de temas por palabra (z_{dn}), ya que determinan la estructura temática de los documentos.

Además, los hiperparámetros globales α y η, que definen los *priors* del modelo, pueden fijarse a priori, estimarse mediante maximización de la verosimilitud marginal, o

ajustarse empíricamente utilizando métricas externas de calidad, tales como la coherencia temática o la diversidad de los tópicos generados.

Descubrir la estructura temática oculta en los documentos requiere estimar las distribuciones de temas más probables para cada documento y el tema más probable para cada palabra, dado el texto observado y los hiperparámetros del modelo. Dicha estimación, fundamental en el modelo, se sustenta en lo que se conoce como distribución posterior (Gelman y cols., 2013; Blei y cols., 2003), que se deriva del teorema de Bayes, expresado en su forma general como:

$$\textit{Distribución posterior} = \frac{\textit{Verosimilitud} \times \textit{Distribución previa}}{\textit{Evidencia}}$$

Aplicado al modelo LDA, se expresa matemáticamente como:

$$p(\theta, \mathbf{z} \mid \mathbf{w}, \alpha, \eta) = \frac{p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \eta)}{p(\mathbf{w} \mid \alpha, \eta)} \tag{1.4}$$

donde,

θ: denota las distribuciones de temas por documento

$\mathbf{z}$: representa las asignaciones de temas a cada palabra

$\mathbf{w}$: es el conjunto de palabras observadas en los documentos

α y η: son los hiperparámetros del modelo.

Cabe destacar que el numerador de esta expresión, $p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \eta)$, corresponde a la distribución conjunta del modelo, la cual puede descomponerse en tres términos: la verosimilitud de las palabras dadas las asignaciones de temas, $p(\mathbf{w} \mid \mathbf{z}, \eta)$; la probabilidad de las asignaciones de temas dadas las proporciones de temas, $p(\mathbf{z} \mid \theta)$; y el prior sobre las proporciones de temas, $p(\theta \mid \alpha)$. De esta forma, la distribución conjunta refleja tanto la verosimilitud como los priors definidos por la estructura jerárquica del modelo:

$$p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \eta) = p(\mathbf{w} \mid \mathbf{z}, \eta) \times p(\mathbf{z} \mid \theta) \times p(\theta \mid \alpha)$$

Realizar el cálculo por medio del modelo descripto precedentemente resulta, en general, costoso computacionalmente (Blei y cols., 2003), debido a la necesidad de computar el denominador de la distribución posterior, $p(\mathbf{w} \mid \alpha, \eta)$, que implica integrar sobre todas las posibles distribuciones de temas por documento y asignaciones de temas por palabra. Es por ello que la inferencia en LDA se basa en métodos aproximados que permiten estimar dicha distribución de manera eficiente, entre los cuales los más utilizados son la aproximación de Laplace, la inferencia variacional y los métodos de Monte Carlo por cadenas de Markov (MCMC), como se describe en Jordan y cols. (1999). Dichos métodos permiten actualizar iterativamente las variables latentes y los parámetros del modelo en función de los datos observados, implementando así un enfoque bayesiano: partiendo de distribuciones previas (Dirichlet) sobre los parámetros latentes y, al incorporar los datos, se obtiene una distribución posterior que refleja una creencia actualizada sobre la estructura temática del corpus.

Ejemplo Ilustrativo del Modelo LDA. A continuación se presenta un ejemplo simplificado, a modo de ilustración del funcionamiento del modelo descrito, que permite entender cómo se identifican temas a partir de un conjunto reducido de documentos.

Considérese que el usuario especifica tres temas ($K = 3$) para todo el corpus. El modelo no conoce los temas de antemano, sino que los descubre automáticamente durante el entrenamiento, a partir de qué palabras tienden a aparecer juntas en los documentos. Una vez entrenado, el modelo proporciona las palabras más relevantes de cada tema, permitiéndole al usuario interpretar y etiquetar cada tópico.

Asúmase que, tras el entrenamiento, el modelo estima la siguiente distribución de temas para el primer documento:

$$\theta_1 = [0{,}2, 0{,}5, 0{,}3]$$

Esto indica que dicho documento aborda aproximadamente un 20 % del tema 1, un 50 % del tema 2 y un 30 % del tema 3. Es importante destacar que estos valores no son elegidos arbitrariamente, sino que resultan del proceso de inferencia del modelo, que ajusta θ_1 para reflejar la mezcla de temas que mejor explica el contenido del documento.

La distribución θ_1 se genera a partir de una distribución Dirichlet con parámetro α, como se describió en el nivel de documento (véase apartado 1.2.2); ello permite modelar la incertidumbre y la diversidad temática de los documentos. En particular, la Dirichlet permite que diferentes documentos tengan diferentes niveles de concentración temática: algunos documentos pueden estar muy enfocados en un solo tema (valores cercanos a [1, 0, 0]), mientras que otros pueden tratar múltiples temas de manera más equilibrada (valores como [0.3, 0.4, 0.3]). Esta flexibilidad es crucial para capturar la realidad de que los documentos raramente tratan de un único tema.

El siguiente proceso describe cómo el modelo asume que se generan las palabras; en la práctica, el algoritmo infiere θ y φ a partir de los documentos observados. Para cada palabra del documento, el proceso generativo sigue dos pasos:

1. Primero se selecciona aleatoriamente un tema conforme a la distribución θ_1 estimada para el documento. Por ejemplo, si para la primera palabra del documento se obtiene $z_{11} = 2$, significa que la palabra se asignó al tema 2, que tenía un 50 % de probabilidad según θ_1.

2. Una vez asignado el tema, la palabra observada w_{11} se genera a partir de la distribución de palabras φ_2 correspondiente al tema 2. Por ejemplo, si el tema seleccionado tuviese como palabras más relevantes:

   - “inversión” con probabilidad 0,08
   - “mercado” con probabilidad 0,06
   - “capital” con probabilidad 0,05

- (y otras palabras con probabilidades menores)

Entonces, es más probable que se genere la palabra “inversión”. Con estas palabras, el tema 2 puede interpretarse o etiquetarse como “Economía”.

Este proceso se repite para cada palabra del documento, utilizando siempre la misma distribución θ_1 para muestrear el tema. La distribución de palabras φ_k es global para cada tema y también se aprende durante el entrenamiento del modelo, ajustándose para reflejar las palabras más representativas de cada tópico.

En la práctica, el modelo infiere tanto las distribuciones de temas por documento como las distribuciones de palabras por tema a partir de los datos observados, utilizando técnicas de inferencia bayesiana o aproximada. Esto significa que, dado un corpus de documentos, el algoritmo analiza qué palabras tienden a aparecer juntas en los documentos y estima qué combinación de temas y distribuciones de palabras es más probable que haya generado los documentos observados. Por ejemplo, si el algoritmo encuentra que las palabras “inversión”, “mercado” y “capital” aparecen frecuentemente juntas en varios documentos, inferirá que probablemente pertenecen al mismo tema y ajustará las distribuciones φ_k y θ_d en consecuencia.

Formulación Matemática. Para formalizar el modelo LDA, se considera un documento como una secuencia de palabras $\mathbf{w} = (w_1, \ldots, w_N)$. El modelo asume que cada documento posee una distribución de temas θ que determina qué proporción del documento trata sobre cada tema. Para cada palabra w_n, se asigna un tema específico z_n según la distribución θ. El parámetro α controla la distribución de temas por documento, mientras que η define las distribuciones de palabras para cada tema.

La probabilidad de observar un documento con sus palabras específicas, dados los parámetros del modelo, se obtiene integrando sobre todas las posibles distribuciones de

temas y sumando sobre todas las asignaciones posibles de temas a palabras (Blei y cols., 2003):

$$p(\mathbf{w} \mid \alpha, \eta) = \int p(\theta \mid \alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n \mid \theta) p(w_n \mid z_n, \eta) \right) d\theta.$$

Esta expresión calcula la probabilidad de que el documento se genere con sus palabras actuales, dados los parámetros α y η. Cabe destacar que esta probabilidad, $p(\mathbf{w} \mid \alpha, \eta)$, corresponde al denominador de la distribución posterior presentada anteriormente en la Ecuación 1.4, y representa la evidencia del modelo para el documento: la probabilidad de generar las palabras $\mathbf{w}$ bajo el modelo, teniendo en cuenta todas las posibles combinaciones de temas por documento y asignaciones de tema a palabras.

Dicha expresión considera todas las posibles formas en que los temas podrían distribuirse en el documento y todas las posibles formas en que las palabras podrían asignarse a esos temas.

- $p(\theta \mid \alpha)$: es la distribución de Dirichlet que determina cómo se mezclan los tópicos en el documento.
    - Ejemplo: si $\alpha = [0{,}1, 0{,}1, 0{,}1]$ para tres tópicos, la distribución de Dirichlet resultante tiende a concentrar la mezcla en pocos tópicos dominantes. Una muestra de esta distribución podría ser $\theta = [0{,}7, 0{,}2, 0{,}1]$, lo que indica que el documento está compuesto, por ejemplo, en un 70 % por “deportes”, 20 % por “política”, y 10 % por “tecnología”.
    - Interpretación: se enfoca en cómo se distribuyen los tópicos en el documento, determinando qué temas son más importantes.
- $p(z_n \mid \theta)$: es la probabilidad de que una palabra específica w_n pertenezca al tema z_n, basándose en la mezcla de tópicos θ del documento.

  - Ejemplo: para cada palabra del documento, como “jugador” o “gol”, hay un 70 % de probabilidad de que pertenezca al tema “deportes”, un 20 % de que pertenezca a “política”, y un 10 % de que pertenezca a “tecnología”.
  - Interpretación: se enfoca en la asignación de temas a cada palabra individual del documento.

- $p(w_n \mid z_n, \eta)$: es la probabilidad de que la palabra w_n sea generada por el tema z_n, determinada por la distribución de palabras en ese tema, que está parametrizada por η.
  - Ejemplo: si el tema “deportes” tiene una alta probabilidad asignada a palabras como “fútbol” y “gol”, esto indica que estas palabras son características del tema y, por lo tanto, es más probable que aparezcan en documentos donde “deportes” es un tema predominante.
  - Interpretación: se enfoca en cómo se distribuyen las palabras dentro de un tema.

Extensión al Corpus. El objetivo del modelo es extender el análisis anterior a un corpus completo compuesto por múltiples documentos. Para ello, es necesario considerar cómo se relacionan las probabilidades de documentos individuales. La probabilidad del corpus completo (conjunto de M documentos) se obtiene como el producto de las probabilidades marginales[3] de cada documento (Blei y cols., 2003):

$$p(D \mid \alpha, \eta) = \prod_{d=1}^{M} \int p(\theta_d \mid \alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn} \mid \theta_d) p(w_{dn} \mid z_{dn}, \eta) \right) d\theta_d.$$

Esta segunda ecuación extiende la anterior a todo el corpus, asumiendo que los documentos son independientes entre sí. Esta independencia es una suposición fundamental del modelo LDA, donde cada documento posee su propia distribución de temas θ_d que se genera de manera independiente según los parámetros α y η compartidos. Los parámetros

[3] Probabilidad de ocurrencia de un solo evento, sin tener en cuenta otros eventos relacionados.

α y η son hiperparámetros a nivel de corpus que definen cómo se distribuirán los temas. Para cada documento, se genera una distribución de temas θ_d específica. Finalmente, para cada palabra en el documento, se selecciona un tema z_{dn} y se genera la palabra w_{dn} correspondiente. Esta estructura jerárquica permite al modelo capturar tanto la diversidad temática entre documentos como la coherencia semántica dentro de cada tema.

Es importante diferenciar LDA de un modelo básico de clustering; en un modelo tradicional de clustering, la estructura se compone de dos niveles: en el primer nivel, se establece una única distribución Dirichlet para todo el corpus, y en el segundo nivel, cada documento recibe una única asignación de tema mediante una variable multinomial. Esta asignación determina la generación de todas las palabras del documento, restringiendo su contenido a un único tema. En contraste, LDA implementa una estructura de tres niveles: primero, una distribución Dirichlet genera las proporciones de temas para cada documento; segundo, para cada palabra del documento, se selecciona un tema específico según estas proporciones; y tercero, se genera la palabra basándose en la distribución de palabras del tema seleccionado. Esta estructura permite que cada documento contenga palabras de diferentes temas, reflejando la naturaleza multifacética de los textos reales (Blei y cols., 2003).

Por su parte, la elección de la distribución de Dirichlet como prior en LDA se fundamenta por las siguientes razones matemáticas y computacionales (Blei y cols., 2003). En primer lugar, la Dirichlet es una distribución definida sobre el simplex: genera vectores de números reales no negativos que suman exactamente uno, lo que la hace ideal para modelar proporciones, como las mezclas de tópicos en documentos o la distribución de palabras en tópicos. Además, pertenece a la familia exponencial, lo que facilita la aplicación de métodos eficientes para la inferencia estadística. Una ventaja computacional importante es que, dados los hiperparámetros, la actualización de la distribución posterior se realiza mediante

conteos simples de asignaciones (como las apariciones de tópicos o palabras), independientemente del tamaño total del corpus. Finalmente, la propiedad de conjugación con la distribución multinomial permite actualizar los parámetros de manera analítica y eficiente al observar nuevos datos.

LDA como Modelo Bayesiano Jerárquico. LDA se interpreta, en el modelado estadístico bayesiano, como un modelo jerárquico (Gelman y cols., 2003), o más precisamente como un modelo jerárquico condicionalmente independiente (Kass y Steffey, 1989). En la literatura, este tipo de estructuras también se asocia con los modelos bayesianos empíricos paramétricos, término que alude no solo a la arquitectura del modelo, sino también a los métodos utilizados para estimar sus parámetros (C. N. Morris, 1983).

En este tipo de modelos, las variables se organizan en distintos niveles jerárquicos, lo cual permite capturar dependencias complejas en los datos, existiendo diferentes enfoques para estimar los hiperparámetros α y η en LDA.

Un primer enfoque consiste en *maximizar la verosimilitud marginal* de los datos encontrando los valores de α y η que hacen más probable observar el corpus bajo el modelo. Este método se basa en principios estadísticos formales del enfoque bayesiano empírico: los hiperparámetros se consideran fijos pero desconocidos, y se estiman maximizando la probabilidad de los datos, integrando sobre las variables latentes, sin tener en cuenta la calidad semántica de los tópicos, sino únicamente el ajuste estadístico del modelo a los datos observados.

En contraste, existe un segundo enfoque que prioriza la *interpretabilidad semántica* de los tópicos sobre el ajuste estadístico estricto. En este método, se ajustan α y η de manera que las distribuciones generadas por el modelo se alineen con patrones observables en los datos, revisándolos manualmente.

Para el parámetro α, si se observa empíricamente que la mayoría de los documentos contienen predominantemente uno o dos temas (en lugar de una mezcla uniforme), se selecciona un valor pequeño de α (típicamente $\alpha < 1$), ya que valores pequeños de α generan distribuciones de Dirichlet que favorecen la concentración de probabilidad en pocos componentes. Por el contrario, valores grandes de α (típicamente $\alpha > 1$) generan distribuciones más uniformes.

Para el parámetro η, se ajusta considerando la estructura general del vocabulario en el corpus. Valores pequeños de η favorecen distribuciones de palabras más concentradas en cada tópico, generando tópicos más específicos y diferenciados. Valores grandes de η generan distribuciones más uniformes de palabras por tópico, resultando en tópicos más generales y menos específicos. La elección de η se basa en el balance deseado entre especificidad y generalidad de los tópicos.

Este ajuste puede realizarse mediante validación cruzada (Stone, 1974) utilizando métricas como la coherencia de tópicos, y tiene como objetivo principal mejorar la calidad interpretativa del modelo más que optimizar su ajuste probabilístico formal.

En síntesis, el primer enfoque (maximización de la verosimilitud marginal) busca un ajuste estadístico óptimo, mientras que el segundo (orientado a la interpretabilidad semántica) prioriza la correspondencia cualitativa entre el modelo y la estructura observada en los datos.

Por otra parte, también existe la posibilidad de optar por un análisis utilizando un enfoque *bayesiano completo*, en el cual se especifican distribuciones previas para α y η, y se infieren sus distribuciones posteriores junto con las demás variables latentes del modelo, como las asignaciones de temas y las proporciones de tópicos por documento. Este enfoque implica una inferencia más compleja, pero ofrece una representación más completa de la incertidumbre sobre los hiperparámetros.

Limitaciones y Consideraciones Prácticas. Es importante señalar que LDA presenta limitaciones que deben considerarse al implementarlo. Los resultados no son completamente estables, ya que el orden de los documentos ingresados durante el entrenamiento afecta al modelo y, en consecuencia, altera los temas generados.

Además, LDA requiere establecer explícitamente el número de temas a priori, lo cual resulta difícil de determinar sin conocimiento previo del corpus. Métodos como la coherencia de tópicos o la perplejidad ayudan a identificar el número óptimo de temas.

Tal como se expone en *LDA como Modelo Bayesiano Jerárquico*, la elección de α y η incide en el comportamiento del modelo.

El espacio de búsqueda de los parámetros posibles es considerable, ya que incluye tanto los hiperparámetros α y η como el número de temas K. Para α, típicamente se consideran valores entre 0.01 y 1.0; para η, valores entre 0.01 y 0.1; y para K, valores que pueden variar desde 2 hasta cientos dependiendo del tamaño y complejidad del corpus. Por lo tanto, pueden utilizarse técnicas de optimización como la validación cruzada para encontrar configuraciones óptimas que mejoren el rendimiento del modelo para el corpus específico sobre el que se quiera trabajar.

1.2.3 Non-negative Matrix Factorization (NMF)

El modelo Non-negative Matrix Factorization (NMF) fue descrito por D. D. Lee y Seung (1999) y publicado formalmente en los dos años posteriores (Seung y Lee, 2001), presentándose algoritmos prácticos de factorización basados en restricciones de no negatividad.

Actualmente, existen múltiples herramientas prácticas para su implementación, como Gensim y OCTIS, mencionadas previamente en la Subsección 1.2.2, y otras como scikit (Pedregosa y cols., 2011), la cual fue diseñada para ser simple, eficiente y accesible; contando con una documentación extensa, rendimiento optimizado, facilidad de validación cruzada y ajuste de hiperparámetros.

Por su parte NMF, como se mostró en la Figura 1, es una técnica algebraica utilizada para descomponer una matriz no negativa en el producto de dos matrices también no negativas. En el contexto del modelado de tópicos, ha demostrado ser más propenso a producir temas de mayor calidad que LDA en textos cortos como tweets (Chen y cols., 2019).

Vectorización del Corpus: TF-IDF. Previo a utilizar NMF para el modelado de tópicos, es necesario representar el corpus de documentos de una forma matricial. Una técnica ampliamente utilizada para ello es el esquema Term Frequency–Inverse Document Frequency (TF-IDF), que transforma el texto en una matriz numérica que refleja la importancia relativa de cada término en cada documento, ponderando cada entrada de la matriz según dos factores:

- TF (frecuencia de término): permite cuantificar la frecuencia de un término en un documento, identificando su relevancia local.
- IDF (frecuencia inversa de documento): penaliza los términos que aparecen en múltiples documentos, debido a que aportan poca discriminación temática.

El producto de ambos factores permite obtener una matriz informativa, para simplemente contar palabras (como en el modelo Bag of Words), ya que produce una matriz dispersa y no negativa, en la que cada fila representa un término y cada columna un documento. Esta matriz, típicamente denotada como $\mathbf{V} \in \mathbb{R}_{+}^{m \times n}$, es la entrada sobre la que opera el algoritmo de NMF.

Representación Matricial. Dado un corpus de documentos representado por una matriz de términos $\mathbf{V} \in \mathbb{R}_{+}^{m \times n}$, donde m es el número de términos únicos y n es el número de documentos, NMF busca dos matrices no negativas:

$$\mathbf{V} \approx \mathbf{WH} \tag{1.5}$$

Donde:

- $\mathbf{W} \in \mathbb{R}_{+}^{m \times k}$ representa la matriz de tópicos, donde cada columna define un tópico como una distribución sobre palabras.
- $\mathbf{H} \in \mathbb{R}_{+}^{k \times n}$ representa la matriz de pesos de tópicos en los documentos, donde cada columna define un documento como una combinación de tópicos.
- k es el número de tópicos predefinido (hiperparámetro).

Funcionamiento del Algoritmo. El modelo NMF emplea una técnica iterativa basada comúnmente en reglas de actualización multiplicativas para ajustar las matrices $\mathbf{W}$ y $\mathbf{H}$, donde el objetivo es minimizar la diferencia (o error de reconstrucción) entre V y WH bajo la restricción de no-negatividad de W y H. Esta factorización produce una representación "de partes" de los datos, ya que las columnas de W pueden interpretarse como componentes básicas (o "templates") y las columnas de H como ponderaciones de dichos componentes en los datos originales.

Algoritmo Iterativo con Reglas Multiplicativas. Como se indicó previamente, el método para obtener la NMF, propuesto por Seung y Lee (2001), utiliza una estrategia de iteración y se basa en la actualización alternada de W y H denominada descenso en bloque (*coordinate descent*).

En cada iteración se efectúan los siguientes pasos:

1. Inicialización. Se asignan valores positivos iniciales a todas las entradas de W y H (por ejemplo, aleatorios estrictamente positivos) para asegurar que no ocurran divisiones por cero.
2. Actualización de H. Se fija W y se actualizan las entradas de H mediante una regla de actualización multiplicativa que mantiene su no negatividad.

3. Actualización de W. Se fija H y se actualiza W con una regla análoga, que también garantiza que los valores de W permanezcan no negativos.

4. Iteración. Se repiten los pasos 2 y 3 hasta alcanzar la convergencia.

Las reglas (multiplicativas) fueron propuestas por Seung y Lee (2001) e implican que la función de coste no aumenta con cada actualización.

Formulación Matemática de la Optimización y Función de Coste. El problema de NMF se formula como un problema de optimización no lineal con restricciones:

$$\min_{W,H} \ f(W,H) \quad \text{sujeto a } W \geq 0,\ H \geq 0,$$

donde una elección común para la función de coste es la distancia Euclidiana (o norma de Frobenius) entre V y WH. Esta función de coste mide el error de reconstrucción y se minimiza al ajustar W y H. Alternativamente, algoritmos de NMF utilizan la divergencia de Kullback-Leibler (KL), que también es no negativa y se anula si y solo si $V = WH$ (Hien y Gillis, 2021; D. Lee y Seung, 2000).

Evaluación de Convergencia. El algoritmo NMF no garantiza hallar el óptimo global, pero sí converge a un mínimo local. El criterio de parada habitual es verificar el error de reconstrucción. Por ejemplo, se puede calcular el error actual $E = \|V - WH\|_F$ después de cada iteración y detener el proceso cuando la reducción relativa de E sea menor que un umbral prefijado, o cuando se alcance un número máximo de iteraciones.

De igual manera, el modelo permite monitorear la variación de los factores W, H entre iteraciones. Dado que el algoritmo asegura una disminución (o invariancia) de la función de coste en cada paso, la convergencia se caracteriza típicamente por estabilización del error de reconstrucción.

En resumen, el procedimiento completo de NMF itera las siguientes operaciones: iniciar $W, H \geq 0$; actualizar alternadamente H y W usando las fórmulas multiplicativas

mostradas; y repetir hasta que el error $\|V - WH\|_F$ sea suficientemente pequeño o se agote el número de iteraciones. Estas reglas garantizan mantener la no-negatividad y asegurar una mejora (no aumento) del ajuste en cada paso.

Ventajas de NMF en Modelado de Tópicos. Entre las principales ventajas del modelo se destacan las siguientes:

- Las representaciones obtenidas son fácilmente interpretables, ya que tanto los tópicos como las combinaciones de documentos están expresados en términos positivos (Casalino y cols., 2016).
- A diferencia de otros modelos como LDA o BERTopic, NMF es computacionalmente eficiente y escalable (Egger y Yu, 2022).
- Un documento puede pertenecer a varios tópicos (Egger y Yu, 2022).
- Sencillo de implementar (Egger y Yu, 2022).

1.2.4 BERTopic

BERTopic es un enfoque moderno para el modelado de tópicos basado en representaciones semánticas (*embeddings*) obtenidas con modelos del tipo BERT (*Bidirectional Encoder Representations from Transformers*), propuesto por (Grootendorst, 2022). A diferencia de enfoques tradicionales como LDA o NMF, no trabaja con representaciones basadas en frecuencia de palabras; en cambio, captura el significado de los documentos de modo que aquellos semánticamente similares se agrupen juntos, independientemente de su frecuencia de aparición en el corpus. Para ello, emplea modelos de lenguaje preentrenados que, combinados con técnicas de reducción de dimensionalidad y de agrupamiento ya introducidas en la clasificación de modelos, permiten identificar temas de forma más flexible y precisa.

Modelos de Representación Semántica. La primera etapa del proceso de BERTopic consiste en transformar los documentos de texto en representaciones vectoriales que capturen su significado semántico. Estas representaciones, conocidas como *embeddings*, son vectores de números reales que capturan el significado semántico de los documentos en función de su uso en el lenguaje, incorporando tanto aspectos semánticos como sintácticos (B. Wang y cols., 2019). Para realizar esta tarea, existen múltiples alternativas disponibles como SpaCy, SBERT y Transformers, entre otras, que son librerías Python que sirven como intermediario para facilitar el uso de modelos de lenguaje preentrenados.

BERT: Bidirectional Encoder Representations from Transformers. Entre los modelos preentrenados se destaca BERT (Devlin y cols., 2019), un modelo revolucionario desarrollado por Google que marcó un punto de inflexión en el procesamiento de lenguaje natural. BERT fue entrenado con grandes volúmenes de datos como BooksCorpus (800 millones de palabras) y English Wikipedia (2.500 millones de palabras).

La principal innovación de BERT radica en su capacidad de generar representaciones contextuales bidireccionales. A diferencia de embeddings estáticos como Word2Vec, donde cada palabra tiene una única representación independiente del contexto, BERT genera vectores diferentes para la misma palabra según su uso en la oración. Por ejemplo, la palabra “banco” tendrá representaciones distintas en “me senté en el banco del parque”, y en “saqué dinero del banco”.

BERT utiliza la arquitectura Transformer (Vaswani y cols., 2023), basada en mecanismos de atención que permiten al modelo ponderar la relevancia de cada palabra en relación con las demás en una oración. Esto posibilita capturar dependencias de largo alcance y relaciones semánticas complejas entre términos.

De BERT a Sentence-BERT: Embeddings de Oraciones. Si bien BERT genera representaciones de alta calidad a nivel de *token*, no fue diseñado originalmente para producir embeddings de oraciones completas de manera directa y eficiente. Sentence-BERT (SBERT) (Reimers y Gurevych, 2019) surge como una adaptación de BERT específicamente optimizada para tareas de similitud semántica entre oraciones y documentos.

BERTopic utiliza por defecto la librería SBERT y el modelo all-MiniLM-L6-v2. Este modelo, basado en la arquitectura MiniLM (W. Wang y cols., 2020), produce representaciones vectoriales de 384 dimensiones que preservan las relaciones semánticas entre documentos mediante un proceso de destilación de conocimiento desde modelos más grandes.

Además del modelo por defecto, BERTopic permite la integración de otros modelos de representación semántica adaptados a contextos específicos.

Entre las alternativas disponibles se encuentra BETO (Cañete y cols., 2020), una versión de BERT entrenada específicamente para el español, y también HIIAMSID (hiiamsid, 2021), un modelo BERT preentrenado para el análisis de contenido en redes sociales en español.

Debe tenerse en consideración que la selección del modelo depende directamente de las características del corpus seleccionado, por ende siempre es aconsejable comenzar con el modelo por defecto y probar distintas alternativas comparando resultados.

Reducción de Dimensionalidad. Los embeddings generados por los modelos preentrenados suelen tener alta dimensionalidad. Esta característica implica que cada documento es representado en un espacio matemático de muchas dimensiones, lo que puede aumentar considerablemente el costo de procesamiento computacional y dificultar la visualización o interpretación directa de los datos. Por este motivo, es habitual aplicar técnicas de reducción de dimensionalidad, que permiten proyectar estos vectores en un espacio de

menor dimensión, conservando la mayor cantidad posible de información relevante para el análisis posterior.

BERTopic aplica una técnica de reducción de dimensionalidad llamada *Uniform Manifold Approximation and Projection* (UMAP)[4] (McInnes y cols., 2020) la cual permite proyectar los vectores originales en un espacio de menor dimensión, preservando la estructura local de los datos y manteniendo las relaciones de proximidad entre documentos que comparten significados similares.

Este paso resulta clave ya que conserva la organización semántica aprendida por el modelo de lenguaje, facilitando el posterior agrupamiento de documentos. En el supuesto de que se detecte que el modelo no está generando una reducción de dimensionalidad adecuada, se puede optar por utilizar una técnica de reducción de dimensionalidad diferente gracias a la arquitectura modular de BERTopic.

Agrupamiento. Para identificar documentos similares que puedan interpretarse como temas, BERTopic utiliza un algoritmo de clustering jerárquico basado en densidad llamado *HDBSCAN* (McInnes y cols., 2017), el cual no requiere especificar previamente el número de grupos. De hecho, el autor destaca esto como una ventaja, ya que permite que el propio modelo defina las agrupaciones basándose en la estructura inherente del corpus en lugar de imponerlas a priori. Si bien el autor prioriza esta adaptación automática, de ser necesario puede optarse por una reducción de tópicos una vez obtenidos. El algoritmo permite también aislar aquellos documentos que no pertenecen a ningún grupo.

HDBSCAN es particularmente adecuado para BERTopic debido a su capacidad de identificar grupos de forma automática sin necesidad de definir parámetros como el número de *clusters* o la densidad mínima de puntos, entendida como la cantidad de documentos que deben estar cercanos entre sí para formar un clúster válido. El algoritmo determina de forma automática qué documentos pertenecen al mismo grupo basándose en su proximidad

[4] En español: aproximación y proyección de variedad uniforme.

en el espacio reducido, donde los documentos que comparten características semánticas similares tienden a agruparse naturalmente.

Además, HDBSCAN se encarga de identificar aquellos documentos que no pertenecen estrictamente a ningún grupo, ya que no aportan información relevante para el análisis. Si se considerase que dichos documentos en realidad sí deberían haber sido incluidos en algún grupo, puede optarse por utilizar una técnica de agrupamiento diferente gracias a la arquitectura modular de BERTopic, mencionada previamente.

Extracción y Representación de Temas. Una vez agrupados los documentos, según su similitud semántica, BERTopic procede a identificar y representar los temas presentes en cada grupo. Para ello, primero se realiza la *tokenización* de los documentos, segmentando el texto en unidades léxicas significativas (tokens).

Posteriormente, el modelo aplica una variante del esquema Frecuencia de Término - Frecuencia Inversa de Documento (TF-IDF, por sus siglas en inglés) denominada class-based TF-IDF (c-TF-IDF) (Grootendorst, 2022).

Mientras que TF-IDF tradicional evalúa la importancia de una palabra en un documento individual respecto al corpus completo, c-TF-IDF trata cada clúster de documentos como una única clase, calculando la importancia de las palabras a nivel de tópico.

De esta manera, se identifican las palabras que son informativas para cada conjunto temático, resaltando términos que son frecuentes dentro del tópico pero raros en los demás. Las palabras con mayor puntuación c-TF-IDF dentro de cada clúster son seleccionadas como las más representativas del tema (típicamente 10-15 palabras), conformando su etiqueta semántica. Esta representación permite interpretar el contenido de cada grupo sin tener la necesidad de inspeccionar todos los documentos que lo componen, facilitando de esta forma el análisis cualitativo posterior.

1.2.5 Métricas de evaluación

A fin de determinar y evaluar temáticas subyacentes capturadas en un conjunto de datos, existen diferentes métricas y criterios que permiten evaluar la calidad y efectividad de los modelos de modelado de tópicos. Estas métricas y criterios se clasifican en:

- Evaluación intrínseca: se enfoca en analizar las propiedades internas del modelo y los tópicos generados. Aquí se consideran aspectos tales como la coherencia de los tópicos y la estabilidad, entre otros, que permiten indicar la calidad y fiabilidad del modelo en sí mismo, independientemente de cualquier tarea posterior.
- Evaluación extrínseca: se evalúa el rendimiento del modelo en tareas específicas o en aplicaciones reales, como modelado de tópicos, donde los tópicos se usan como herramientas para mejorar otros procesos, por lo que la efectividad en estas tareas o procesos indican qué tan útil es el modelo en contextos prácticos.

Perplejidad. La perplejidad (*perplexity*) es una métrica comúnmente utilizada para evaluar modelos de lenguaje probabilísticos, incluyendo modelos de tópicos como Latent Dirichlet Allocation (LDA) (Blei y cols., 2003).

Matemáticamente, la perplejidad mide la capacidad del modelo de predecir una muestra nueva basándose en la información “aprendida", y se define como:

$$\text{Perplejidad}(D) = \exp\left\{-\frac{1}{N}\sum_{d=1}^{|D|}\log p(w_d)\right\} \tag{1.6}$$

donde D representa el corpus de documentos de prueba, w_d son las palabras del documento d, $p(w_d)$ es la probabilidad de observar las palabras de dicho documento según el modelo, y N es el número total de palabras en el conjunto de prueba (Blei y cols., 2003).

Una perplejidad baja, indica que el modelo asigna mayores probabilidades a las palabras observadas, lo cual se debe interpretar como una mejor capacidad de generalización del modelo (Blei y cols., 2003). Pero una perplejidad baja, no necesariamente implica que

los tópicos generados sean interpretables para humanos, ya que esta métrica evalúa la calidad estadística del modelo, no su coherencia semántica (Chang y cols., 2009).

Es importante destacar que el resultado de la perplejidad con que se evalúa un corpus varía dependiendo del vocabulario utilizado, ya que asigna probabilidades a un mayor número de palabras poco frecuentes.

Para mitigar este problema, se implementa la perplejidad normalizada (Unigram-Normalized Perplexity o PPLu) mediante un modelo de unigramas (Roh y cols., 2020). Entendiéndose a éste como un modelo probabilístico de lenguaje que asume independencia entre palabras y estima la probabilidad de cada palabra individualmente, sin considerar el contexto de palabras adyacentes.

$$\text{PPL}_{\text{norm}} = \frac{\text{Perplejidad}_{\text{modelo}}}{\text{Perplejidad}_{\text{unigrama}}} \tag{1.7}$$

donde $\text{Perplejidad}_{\text{modelo}}$ es la perplejidad del modelo evaluado y $\text{Perplejidad}_{\text{unigrama}}$ corresponde a la perplejidad de un modelo de unigramas entrenado sobre el mismo corpus. Esta normalización proporciona una medida más robusta y comparable del rendimiento del modelo (Roh y cols., 2020).

Coherencia. En el contexto del modelado de tópicos, la coherencia es una métrica que evalúa la calidad semántica de los tópicos generados, midiendo el grado de asociación entre los términos más representativos de cada tópico. Una alta coherencia indica que los términos del tópico tienden a aparecer juntos en contextos similares, lo que sugiere una mayor interpretabilidad desde una perspectiva humana.

Normalized Pointwise Mutual Information (NPMI). Bouma (2009) describe una de las métricas utilizadas para cuantificar la coherencia de un corpus, denominada Normalized Pointwise Mutual Information.

En la misma, dada una lista de M términos $\{w_1, w_2, ..., w_M\}$ que representan un tópico, NPMI se calcula para cada par de términos (w_i, w_j) con $i < j$ como:

$$\text{NPMI}(w_i, w_j) = \frac{\log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}}{-\log P(w_i, w_j)} \tag{1.8}$$

donde $P(w_i)$ y $P(w_j)$ son las probabilidades individuales de los términos, y $P(w_i, w_j)$ es la probabilidad de co-ocurrencia conjunta de ambos términos, dentro de una ventana de contexto en un corpus de referencia. Esta ventana define el límite máximo de proximidad (ya sea una cantidad de palabras o la extensión de un documento) necesario para considerar que dos términos aparecen juntos.

Esta normalización, permite acotar el resultado en un rango definido, facilitando su interpretación ya que NPMI toma valores entre -1 y 1, donde:

- 1: co-ocurrencia perfecta (los términos aparecen siempre juntos).
- 0: indica independencia, la co-ocurrencia observada es la esperada por azar.
- -1: co-ocurrencia inexistente (los términos no aparecen juntos).

Cabe destacar que NPMI surge como una solución a las limitaciones de su antecedente, la métrica PMI (Pointwise Mutual Information), la cual fue introducida originalmente en el campo de la lingüística computacional por Church y Hanks (1990), y se basa en la teoría de la información de Fano (1961). PMI mide el grado de asociación entre dos eventos como:

$$\text{PMI}(x, y) = \log \frac{P(x, y)}{P(x)P(y)} \tag{1.9}$$

A pesar de su utilidad, PMI presenta limitaciones importantes. Por ejemplo, tiende a asignar valores altos a pares de términos poco frecuentes, debido a sus bajas probabilidades marginales. Además, no está acotada superiormente, lo que dificulta la interpretación

comparativa de sus valores y puede favorecer asociaciones artificialmente elevadas (Bouma, 2009).

Para abordar estos problemas, Bouma (2009) propuso la métrica NPMI, que normaliza el valor de PMI al dividirlo por $-\log P(x, y)$, acotando el rango y facilitando su interpretación. Esto permite penalizar asociaciones artificialmente elevadas debido a bajas frecuencias, haciendo la medida más robusta y menos sensible a términos de baja frecuencia.

En el contexto de evaluación de tópicos, diversos estudios han mostrado que NPMI resulta más confiable que otras métricas, incluida PMI clásica, al capturar la coherencia semántica de tópicos desde una perspectiva humana (Aletras y Stevenson, 2013; Lau y cols., 2014).

Además, esta métrica resulta más alineada con la percepción humana que otras como la perplejidad la cual, como se indicó previamente, se basa exclusivamente en criterios probabilísticos sin considerar si las palabras tienen sentido juntas.

Cabe destacar que existe un compromiso entre ambas métricas: la optimización de la perplejidad suele conducir a una disminución en la coherencia, y viceversa. Además, hiperparámetros del modelo —como la cantidad de tópicos— también influyen directamente en estos valores (Chang y cols., 2009).

Coherencia C_V. Otra métrica ampliamente utilizada es la coherencia C_V, propuesta por Röder y cols. (2015), que combina múltiples elementos para evaluar la coherencia semántica de un conjunto de términos.

La misma tiene una alta correlación con evaluaciones humanas, ha demostrado un desempeño consistente con diversos conjuntos de datos y se compone de cuatro componentes descritos a continuación (Röder y cols., 2015):

1. Segmentación: se generan pares de términos relacionados a partir del conjunto de palabras del tópico mediante la estrategia de segmentación S:

$$S = \{(w_i, W_i) \mid w_i \in W,\, W_i \subset W,\, w_i \notin W_i\}$$

   donde W es la lista de términos del tópico. Típicamente, se utiliza una estrategia de ventana deslizante (*sliding window*), en la cual se recorre el texto con una ventana de tamaño fijo (por ejemplo, 10 palabras) y se registran las coocurrencias de términos dentro de dicha ventana. Esto permite capturar asociaciones locales entre términos sin depender de la estructura de oraciones o párrafos.

2. Estimación de Probabilidades: se calculan las frecuencias de coocurrencia de los términos en un corpus de referencia utilizando una ventana deslizante.

3. Medida de confirmación: para cada par de términos se cuantifica la relación semántica mediante NPMI.

4. Agregación: se construyen vectores de características para cada término basados en las NPMI calculadas, y se calcula la similitud del coseno entre estos vectores para obtener una puntuación de coherencia global del tópico.

Este enfoque modular y jerárquico permite a C_V capturar tanto relaciones estadísticas como estructurales entre los términos, lo que contribuye a una mayor correlación con juicios humanos en tareas de interpretación de tópicos (Röder y cols., 2015).

Otras Métricas: C_{UMass} y C_{UCI}. Además de NPMI y C_V, otras métricas comunes de coherencia son C_{UMass} y C_{UCI}, ambas basadas en la coocurrencia de términos, pero con diferencias clave en sus enfoques:

- C_{UMass}: también conocida como coherencia intrínseca, fue introducida por Mimno y cols. (2011). Se basa en la coocurrencia de pares de palabras en los documentos del corpus de entrenamiento. La coherencia de un conjunto de palabras $W = \{w_1, w_2, ..., w_N\}$ se calcula como:

$$C_{UMass}(W) = \frac{2}{N(N-1)} \sum_{i=2}^{N} \sum_{j=1}^{i-1} \log \frac{D(w_i, w_j) + 1}{D(w_j)}$$

  donde $D(w_i, w_j)$ representa la cantidad de documentos en los que ambas palabras w_i y w_j aparecen juntas, y $D(w_j)$ es la cantidad de documentos que contienen w_j.

- C_{UCI}: esta métrica, propuesta por Newman y cols. (2010), emplea la metrica de PMI sin normalizar, utilizando una colección externa como corpus de referencia. Para un conjunto de palabras $W = \{w_1, w_2, ..., w_N\}$, la coherencia se calcula como:

$$C_{UCI}(W) = \frac{2}{N(N-1)} \sum_{i=2}^{N} \sum_{j=1}^{i-1} \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}$$

  donde $P(w_i, w_j)$ es la probabilidad de coocurrencia de w_i y w_j en una ventana deslizante dentro del corpus de referencia, y $P(w_i)$, $P(w_j)$ son las probabilidades marginales de aparición de cada palabra.

Diversidad. Otra métrica utilizada en la evaluación de modelos de tópicos es la diversidad temática, la cual mide el grado de diferenciación semántica entre los distintos tópicos generados.

Idealmente, un modelo debería producir tópicos que sean informativamente distintos entre sí, lo que se refleja en un puntaje alto en esta métrica. Por el contrario, un valor bajo indica redundancia temática, ya que varios tópicos comparten términos principales o abordan contenidos similares (Dieng y cols., 2020).

Al igual que ocurre en otros modelos con hiperparámetros (como LDA, véase Subsección 1.2.2), el valor de este indicador puede verse afectado significativamente por su configuración. En particular, la cantidad de tópicos especificada juega un papel crucial: un número excesivo de tópicos tiende a generar solapamientos entre ellos, mientras que una cantidad demasiado reducida puede dar lugar a tópicos demasiado amplios o poco específicos, dificultando su interpretación (Röder y cols., 2015).

Estabilidad. La estabilidad es una métrica que evalúa el grado de variación de los tópicos generados a través de múltiples ejecuciones del modelo bajo condiciones similares (Belford y cols., 2018).

Para su medición, se identifican los principales tópicos obtenidos en cada ejecución y se comparan sus términos más representativos. Esta comparación puede realizarse mediante evaluación humana o utilizando métricas cuantitativas como el PMI, el índice de Jaccard o el RBO (Rank-Biased Overlap), entre otras.

Un alto grado de estabilidad indica que el modelo produce resultados consistentes y reproducibles, mientras que una baja estabilidad refleja sensibilidad a factores como la inicialización aleatoria o a variaciones menores en los datos de entrada.

Eficiencia. La eficiencia computacional es un aspecto relevante cuando se trabaja con grandes volúmenes de texto (Abdelrazek y cols., 2022). Un aspecto central es el tiempo de ejecución requerido para entrenar el modelo: los tiempos varían considerablemente entre enfoques. Cuando dos modelos alcanzan resultados de calidad similares, la eficiencia puede inclinar la elección hacia el más rápido.

Frente a grandes volúmenes de datos, pueden emplearse herramientas de procesamiento paralelo y en lotes, como Dask (Dask Development Team, 2024), que permiten repartir la carga de trabajo y optimizar el consumo de memoria del sistema mediante tareas diferidas y ejecución distribuida.

Flexibilidad. La flexibilidad representa un aspecto relevante al evaluar modelos de tópicos, sin ser una métrica cuantificable en sentido estricto. Se refiere a la capacidad del modelo para adaptarse a diferentes características del corpus, como variaciones en la distribución de palabras, longitud de los documentos o requerimientos de preprocesamiento.

Consideraciones Finales. Las métricas desarrolladas en esta sección son de carácter intrínseco. No obstante, tanto la eficiencia como la flexibilidad, previamente desarrolladas en esta misma sección, se clasifican como extrínsecas según el contexto de evaluación. En este marco, las métricas presentadas abordan distintos aspectos de la evaluación de los modelos; no obstante, al tratarse de una evaluación estadística de resultados, no hay un procedimiento único de evaluación aplicable a todos los casos.

Esto se debe a que cada métrica ofrece una perspectiva distinta y su relevancia varía de forma significativa según el objetivo del análisis y la interpretación profesional del mismo. Por ejemplo, en aplicaciones donde la interpretación de los tópicos es crucial, la coherencia adquiere mayor peso que la perplejidad. En otros escenarios, como tareas automatizadas a gran escala, la eficiencia o la estabilidad resultan más determinantes.

Además, concentrarse exclusivamente en la optimización de una métrica específica conduce a un ajuste excesivo de los parámetros del modelo, generando resultados que puntúan alto en dicha métrica, pero sin garantizar una mejora del desempeño en términos generales o en el contexto práctico deseado. Por ello, la selección de métricas debe responder al caso de uso concreto y los resultados deben ser evaluados por profesionales entrenados en la materia.

1.3 Preprocesamiento de textos informales

Una vez establecidos los fundamentos teóricos del procesamiento del lenguaje natural (Sección 1.1) y del modelado de tópicos (Sección 1.2), resulta fundamental abordar

los aspectos prácticos relacionados con la preparación de los textos de entrada. En el modelado de tópicos, la calidad y adecuación del preprocesamiento constituyen factores determinantes que influyen significativamente en la confiabilidad y validez de los resultados obtenidos.

El desarrollo de esta sección se organiza en dos ejes complementarios: las técnicas esenciales de preprocesamiento (Subsección 1.3.1) y la selección de herramientas para su implementación (Subsección 1.3.2).

El preprocesamiento de textos permite normalizar, depurar y transformar los documentos en representaciones adecuadas para su procesamiento automático. Esta etapa adquiere especial relevancia en textos provenientes de redes sociales, donde coexisten elementos paratextuales (enlaces, imágenes, emoticones) con formas de escritura informal (abreviaciones, errores ortográficos, jerga y variación léxica).

Para mitigar esta complejidad, autores como Symeonidis y cols. (2018) enfatizan en la necesidad de eliminar o normalizar el ruido presente en los textos antes de proceder a su análisis. Como señalan Guo y cols. (2021), los modelos de aprendizaje automático (ML, por sus siglas en inglés), requieren estructuras de datos consistentes como entrada. Al tratarse de algoritmos diseñados para aprender patrones sin ser programados explícitamente (Wiederhold y McCarthy, 1992), un preprocesamiento adecuado no solo mejora la calidad de los datos, sino que también evita un deterioro significativo en su rendimiento.

1.3.1 Técnicas esenciales para textos informales

En el análisis de textos provenientes de redes sociales, un preprocesamiento eficaz requiere la aplicación de ciertas técnicas fundamentales (Sailunaz y Alhajj, 2019; Kumar y Garg, 2019; Al Amrani y cols., 2018), entre las que se destacan:

1. Eliminación de elementos no informativos: incluye la remoción de *stopwords* (palabras vacías o stopwords son términos sin significado propio, como artículos, pronombres o preposiciones), caracteres especiales (signos de puntuación, hashtags, menciones,

emojis y emoticonos), números y palabras con errores ortográficos o irrelevantes para el análisis.

2. Normalización léxica: aplicación de técnicas como *stemming* o lematización, las cuales reducen las palabras a sus raíces bajo el supuesto de que los términos que comparten raíz tienen significados similares, simplificando así el vocabulario sin perder contenido semántico (Kumar y Garg, 2019). Adicionalmente, incluye la normalización del lenguaje inclusivo y las expresiones de risa, contribuyendo a reducir la variabilidad léxica para agrupar las distintas formas de escribir una misma idea, evitando que el algoritmo las interprete por error como conceptos diferentes.

Estas técnicas resultan especialmente críticas en contextos de lenguaje informal, caracterizados por la ausencia de convenciones gramaticales estrictas, el uso frecuente de jerga y expresiones coloquiales. Entre ellas, la lematización consiste en reducir las palabras a su forma canónica (lema), para unificar variantes morfológicas con significado equivalente. Symeonidis y cols. (2018) realizaron un estudio empírico evaluando dieciséis técnicas de preprocesamiento mediante algoritmos tanto clásicos (Logistic Regression, Naïve Bayes) como avanzados (Convolutional Neural Networks), concluyendo que la lematización, el reemplazo de repeticiones de signos de puntuación y la eliminación de números son particularmente efectivas para preservar la información semántica relevante.

Dado que la implementación manual de estas técnicas puede resultar compleja y propensa a errores, la selección de herramientas especializadas de procesamiento de lenguaje natural se vuelve fundamental para garantizar tanto la eficiencia como la reproducibilidad del preprocesamiento.

1.3.2 Herramientas disponibles

A continuación, se presentan las principales librerías consideradas para el preprocesamiento de textos en español:

- spaCy: ofrece modelos preentrenados para español (es_core_news_md), con soporte para tokenización[5], etiquetado morfosintáctico[6] (POS, por sus siglas en inglés) y reconocimiento de entidades nombradas (NER, por sus siglas en inglés). Su rapidez y precisión la hacen ideal para procesar grandes volúmenes de datos (Honnibal y Montani, 2023).

- Stanza: desarrollada por la Universidad de Stanford, permite dividir textos en oraciones y palabras, aplicar lematización, NER, etiquetado POS y realizar análisis sintácticos. Incluye modelos preentrenados para más de 70 idiomas, siendo especialmente útil para análisis estructurales y semánticos en español (Stanford NLP Group, 2020).

- StanfordNLP: versión previa a Stanza que incluye herramientas para tokenización, segmentación de oraciones, lematización y etiquetado POS en múltiples idiomas (Qi y cols., 2018).

- NLTK: proporciona funciones para tokenización, lematización, etiquetado POS, listas de *stopwords* por idioma, análisis sintáctico y razonamiento semántico. Incluye interfaces a más de 50 corpus y recursos léxicos (NLTK Project, 2023).

- spanish_nlp: especializada en español, incluye lematización adaptada a variantes regionales y recursos específicos para el español rioplatense (Ortiz-Fuentes, 2022).

- unidecode: convierte caracteres Unicode en sus equivalentes ASCII, buscando una representación cercana a la elección que haría un humano usando un teclado estándar. Permite normalizar textos con tildes, diéresis o caracteres especiales, facilitando la homogeneización de corpus multilingües (Solc, 2009).

[5] Proceso de segmentar un texto en unidades mínimas, llamadas tokens, para facilitar el procesamiento.

[6] El etiquetado morfosintáctico o *Part-of-Speech tagging* (POS) consiste en asignar a cada token su categoría gramatical (sustantivo, verbo, adjetivo), facilitando la comprensión estructural del texto.

- LanguageTool: corrector gramatical de código abierto que ofrece reglas específicas para español y soporte para más de 20 idiomas adicionales, útil para detectar errores ortográficos y gramaticales en textos informales (et al., 2025).

- pyspellchecker: corrector ortográfico que identifica errores mediante la comparación con un diccionario de palabras frecuentes. Cuando encuentra una palabra incorrecta, genera sugerencias basadas en similitudes ortográficas y prioriza las correcciones más comunes según su frecuencia de uso. Incluye soporte para múltiples idiomas (Barrus, 2025).

La selección y configuración de las herramientas presentadas constituye un aspecto fundamental para garantizar la calidad y consistencia del preprocesamiento; su combinación permite abordar las particularidades específicas de los textos informales y las variaciones lingüísticas propias de las redes sociales. A partir de ello, el siguiente paso consiste en implementar la metodología de preprocesamiento y proceder con el entrenamiento de los modelos.

Capítulo 2: Metodología

En el presente capítulo se describe la metodología empleada en esta investigación, que comprende el preprocesamiento de los textos —en línea con lo expuesto en la Sección 1.3 del Capítulo 1— y el entrenamiento comparativo de los modelos de modelado de tópicos LDA, NMF y BERTopic, expuestos en las Subsecciones 1.2.2, 1.2.3 y 1.2.4, respectivamente. Dicha metodología se aplica sobre los conjuntos de datos seleccionados para el trabajo: *datasets* internacionales empleados en la validación de las implementaciones y el corpus argentino destinado al análisis comparativo de los modelos.

Una vez establecidos los fundamentos teóricos del modelado de tópicos y del preprocesamiento resulta fundamental abordar los aspectos prácticos relacionados con la selección, preparación y análisis de los datos que servirán como base empírica para la validación de los modelos propuestos. En el modelado de tópicos, la calidad y adecuación de los datos de entrada constituyen factores determinantes que influyen significativamente en la confiabilidad y validez de los resultados obtenidos.

La metodología desarrollada se centra en dos aspectos críticos: la optimización del preprocesamiento para preservar información semánticamente relevante mientras se elimina el ruido característico de los medios sociales, y la implementación y mejora de los modelos en el contexto específico del español argentino.

La metodología se articula en tres etapas: (1) selección de *datasets*; (2) preprocesamiento del corpus argentino; (3) evaluación comparativa e interpretación de los resultados. Cada etapa se presenta en las Secciones 2.1 a 2.3.

2.1 Etapa 1: Selección de datasets

Con el objetivo de garantizar la robustez metodológica del trabajo, en esta etapa se incorporaron tres *datasets* de procedencia internacional ampliamente utilizados en la literatura. Estos conjuntos de datos permiten reproducir resultados previamente reportados (Grootendorst, 2022) y establecer un marco de referencia consistente previo a la aplicación de los modelos en contextos locales. Los *datasets* incluidos son:

- 20NewsGroups: contiene 16.309 artículos de noticias categorizados en 20 temáticas, empleado como referencia para evaluar la capacidad de clasificación y agrupamiento de tópicos (Lang, 1995).
- BBC News: formado por 2.225 artículos periodísticos del sitio web de noticias de la BBC en el periodo del 2004 al 2005, útil para validar modelos en dominios de lenguaje formal (Greene y Cunningham, 2006).
- Trump: conjunto de 44.253 tuits originales publicados por el presidente Donald Trump entre el 2009 y el 2021, excluyendo los retuits (Archive, 2023).
- Elecciones presidenciales argentinas: conjunto de 1.143.424 publicaciones de la red social Twitter[1], vinculadas con las elecciones presidenciales argentinas, correspondientes al período comprendido entre el 5 y el 22 de noviembre del 2015 (Robins y cols., 2016).

Cabe destacar que los tres primeros conjuntos de datos presentados permitirán validar los modelos de modelado de tópicos LDA, NMF y BERTopic, mencionados al inicio del presente capítulo. En el marco de esta investigación, dicha validación consiste en replicar los experimentos documentados en la literatura para contrastar el rendimiento de las implementaciones desarrolladas contra los valores de referencia reportados por el autor

[1] Desde julio de 2023, la plataforma se denomina X; en este trabajo se conserva el nombre Twitter por corresponder al período del corpus y a la nomenclatura de la fuente original.

original. Específicamente, el objetivo es reproducir los resultados obtenidos en métricas de evaluación previamente detalladas, tales como la coherencia y la diversidad de tópicos. Esta etapa resulta fundamental no solo para certificar empíricamente la correcta calibración de los algoritmos y la solidez del entorno de pruebas, sino también para sentar una base metodológica transparente que garantice la reproducibilidad de la investigación. De este modo, se asegura que el proceso pueda replicarse con exactitud antes de su aplicación sobre el dominio de estudio principal, correspondiente al corpus de elecciones presidenciales argentinas en Twitter.

En síntesis, los conjuntos de datos seleccionados ofrecen tanto variedad temática como diversidad de estilos lingüísticos, lo que posibilita evaluar la robustez de los modelos en diferentes contextos. Sin embargo, mientras que los *datasets* 20NewsGroups y BBC News ya cuentan con un preprocesamiento estándar en sus versiones públicas, el de Trump y, especialmente, el de las elecciones presidenciales argentinas, requirieron un tratamiento adicional para normalizar, depurar y estructurar los textos antes de ser utilizados por los modelos.

2.2 Etapa 2: Preprocesamiento del corpus argentino

A continuación se enumeran, en el orden requerido, los pasos de preprocesamiento del corpus. Dado que el orden de las operaciones incide en el resultado final, la secuencia es obligatoria. La implementación se detalla en el Anexo A.

1. Separación de hashtags: Considerando que los hashtags cumplen una función estructural en Twitter, permitiendo agrupar contenido y facilitar su descubrimiento, se contemplan en el análisis y se separan en palabras, de modo que el modelo los interprete correctamente. Esta funcionalidad se implementa de forma personalizada, sin recurrir a bibliotecas externas.

   Los hashtags que contienen números no se separan, dado que posteriormente se eliminan todos los caracteres numéricos; un procesamiento adicional genera un costo

computacional innecesario. Asimismo, los hashtags escritos completamente en mayúsculas o en minúsculas, que contienen múltiples palabras concatenadas, no se separan por eficiencia computacional.

2. Conversión a minúsculas: Una vez completada la separación de los hashtags, la cual requiere preservar las mayúsculas y minúsculas para identificar correctamente los límites de palabras, se procede a la conversión de cada tuit a minúsculas. Esta normalización facilita el procesamiento posterior y evita la duplicación de tokens que representan la misma palabra.

3. Decodificación de entidades HTML: Se aplica una etapa de decodificación para transformar las entidades HTML codificadas a sus caracteres correspondientes, evitando que sean interpretadas como palabras erróneamente.

4. Descarte de tuits en otros idiomas: Con la finalidad de dirigir el análisis a documentos escritos en español, se descartó cualquier tuit escrito en portugués haciendo uso del software de Danilak (2021). Por otro lado, dado que existe la posibilidad de que un tuit argentino esté compuesto también por expresiones en inglés, no se implementó la misma lógica para dicho idioma, sino que se identificaron manualmente las 33 palabras con más apariciones en documentos completamente escritos en inglés y se descartaron aquellos tuits que las contengan (véase la lista del Anexo A).

5. Remoción de enlaces a imágenes: Se descartó el análisis de imágenes con el objetivo de reducir la complejidad en este primer acercamiento al modelado de tópicos sobre el *dataset* argentino. Esta funcionalidad fue implementada de forma personalizada, sin recurrir a la instalación de paquetes adicionales.

6. Normalización de lenguaje inclusivo: Se aplicó un mapeo que convierte palabras inclusivas como “todes”, “nosotres”, “amiges” hacia sus equivalentes estándar “todos”, “nosotros”, “amigos”.

7. Eliminación de retuits y menciones: Dado que el *dataset* argentino está compuesto por un importante número de retuits y menciones, se eliminaron los fragmentos de texto con dichas referencias manteniendo únicamente el contenido original del mensaje.

8. Eliminación de palabras incompletas: Se identificó que el *dataset* argentino presenta un límite de 140 caracteres por tuit. Cuando un mensaje excede esta longitud, se trunca en el carácter 139 y se agrega el símbolo "…" para indicar el corte. Este símbolo, propio del *dataset*, difiere de los puntos suspensivos convencionales ("..."), por lo que se procedió a eliminar estos fragmentos incompletos para evitar ruido en el análisis.

9. Normalización de risas: Como es esperable en redes sociales, las onomatopeyas de la risa suelen expresarse de diversas formas. Por ello, se desarrolló una funcionalidad que permite normalizar estas variantes bajo una forma única: "jaja".

10. Spanish NLP (presentada en la Subsección 1.3.2): Librería que permitió aplicar una serie de transformaciones adicionales, configuradas mediante los parámetros de una clase personalizada (SpanishPreprocess), que permite activar o desactivar cada paso de forma modular.

    Entre los parámetros se encuentran:

    a) Conversión a minúsculas: por defecto está activado. Fue desactivado debido a su aplicación previa (véase el paso 2).

    b) Eliminación de URLs: por defecto está activado y se lo mantuvo de esa forma, considerando los enlaces como información no relevante para este análisis.

    c) Eliminación de hashtags: por defecto está desactivado, valor que se decidió sostener ya que se consideró que la separación de los mismos aportaba más valor al análisis que la remoción total.

d) Separación de hashtags: por defecto está activado, pero se decidió desactivarlo ya que se detectó el problema de que si el hashtag estaba al inicio de una oración, no funcionaba correctamente. Por ende, se codificó este proceso manualmente como se comentó en el paso 1.

e) Normalización de saltos de línea: por defecto activado. Se decidió mantenerlo así, permitiendo reemplazar múltiples saltos de línea por uno solo.

f) Eliminación de emoticones: se mantuvo su valor activado por defecto, con la finalidad de darle prioridad al texto.

g) Eliminación de emojis: se mantuvo su valor activado por defecto, con la finalidad de darle prioridad al texto. Los emoticones refieren a combinaciones de caracteres del teclado que representan expresiones faciales; por ejemplo: :) :(;-) XD :P. En cambio, los emojis son caracteres gráficos Unicode.

h) Conversión de emoticones: se mantuvo su valor desactivado por defecto, ya que se optó por la eliminación.

i) Conversión de emojis: se mantuvo su valor desactivado por defecto, ya que se optó por la eliminación.

j) Normalización de lenguaje inclusivo: por defecto desactivado. Se optó por mantenerlo desactivado, pero crear una conversión personalizada a fin de agregar más palabras de las que trae por defecto y alterar el orden de ejecución.

k) Reducción de spam: se mantuvo su valor activado por defecto, a fin de reducir el número de palabras repetidas.

l) Eliminación de vocales con acentos: por defecto activado. Se optó por desactivarlo debido a que las correcciones en etapas posteriores reintroducen los acentos, incurriendo en un procesamiento ineficiente.

m) Eliminación de espacios múltiples: se mantuvo su valor activado por defecto, para normalizar el texto.

n) Eliminación de signos de puntuación: se mantuvo su valor activado por defecto. Esta etapa es particularmente relevante porque:

- conserva letras, números, guiones bajos (_), espacios y caracteres en español (incluyendo acentos y "ñ").
- reemplaza todos los caracteres no deseados (que no coinciden con los anteriores) por un espacio.

ñ) Eliminación de caracteres no imprimibles: se mantuvo su valor activado por defecto, descartándose caracteres como "©".

o) Eliminación de números: se mantuvo su valor activado por defecto, descartando todos los caracteres numéricos.

p) Eliminación de stopwords: por defecto desactivado. Se activó para enfocar el análisis en las palabras con mayor significado semántico.

q) Listado de stopwords: este parámetro en particular solo es requerido en caso de que "Eliminación de stopwords" esté activada. Para este trabajo se utilizó la lista de stopwords de NLTK (véase Subsección 1.3.2).

r) Lematización: se mantuvo su valor desactivado por defecto. Esta tarea se realizó posteriormente con la librería Stanza (véase Subsección 1.3.2).

s) Stem: se mantuvo desactivado por defecto, ya que se optó por utilizar lematización en su lugar (véase Subsección 1.3.1). Aunque la lematización es más costosa computacionalmente, ofrece mayor precisión lingüística y capacidad para preservar el significado semántico de las palabras en contexto. A diferencia del *stemming*, que reduce las palabras a su raíz mediante reglas de truncamiento y

puede generar formas no válidas, la lematización retorna la forma canónica o lema de cada palabra, manteniendo su integridad morfológica y semántica (Khyani y cols., 2021).

t) Eliminación de etiquetas HTML: se mantuvo su valor activado por defecto. Cabe aclarar que tras revisar el corpus se verificó que no contenía etiquetas HTML, por lo que si bien se considera una etapa útil, no tuvo relevancia en este trabajo.

11. Normalización de palabras: Se eliminaron caracteres no alfabéticos y aquellas palabras con longitud menor a tres caracteres.

12. Corrección de palabras: En la presente etapa se emplearon distintas fuentes de datos:

    - Diccionario de nombres en español (migalpha, 2018): por medio de la plataforma Kaggle se descargó un diccionario de nombres masculinos y otro de nombres femeninos, posteriormente unificados para su trabajo en conjunto.
    - Diccionario de palabras en español de la RAE (Lérin, 2024): diccionario en español con todas las palabras de la Real Academia Española.
    - Diccionario de palabras en español recopiladas de diversas fuentes (Fenollosa, 2019): diccionario en español con información adicional además de las palabras, como: tipo, género, número, raíz, afijo, tónica, sílabas, locale, origen y sinónimos.
    - Diccionario de palabras en inglés (Tatman, 2017): formado por las mil palabras en inglés con mayor frecuencia y que además cuenten con longitud mayor a dos caracteres.

    En este paso se procede a verificar si las palabras pertenecientes al tuit que se está procesando se encuentran en el diccionario de nombres, los diccionarios de palabras en español o el diccionario de palabras en inglés. En caso de que no se encuentre una palabra, se aplica un proceso de detección de secuencia de caracteres no numéricos

repetidos al final de la palabra, y se reemplaza con una sola instancia de dicho carácter, como por ejemplo “holaaa” se convierte en “hola” (esta lógica se aplica recién en esta instancia para evitar corromper palabras tales como “cree”, que se convertiría erróneamente a “cre”).

Posteriormente, se procede a buscar la palabra resultante en los diccionarios mencionados previamente. Si la palabra sigue sin estar presente, se aplica una corrección con el software LanguageTool (J. Morris, 2024), el cual es una interfaz escrita en Python para utilizar la herramienta de corrección gramatical y ortográfica de código abierto LanguageTool, comúnmente conocida como el corrector de OpenOffice. Esta elección se enmarca en la evaluación de herramientas de NLP presentada en la Subsección 1.3.2. Es importante resaltar que se evaluaron otras alternativas a LanguageTool, como pyspellchecker (Barrus, 2018), que también demostró capacidad para detectar errores comunes, aunque el primero reconocía un mayor número de términos propios del español argentino, sugiriendo un vocabulario más adaptado a la variedad local. Debido a ello, en el código del preprocesamiento (véase Anexo A) se dejaron ambas implementaciones disponibles, de manera tal que sea el usuario quién decida qué librería utilizar.

Por otro lado, en el supuesto de que la palabra se haya podido corregir, se la agrega a un conjunto de palabras corregidas, y en caso de que no, a un conjunto de palabras no encontradas, ambos creados para facilitar el análisis del funcionamiento del algoritmo y realizar mejoras posteriores. Es importante destacar que las correcciones posteriores se aplicaron únicamente sobre aquellas palabras mal corregidas con una frecuencia mayor o igual a 200 (umbral fijado para manejar la alta dimensionalidad de los documentos).

Esta revisión fue realizada manualmente, y las palabras se fueron organizando en diferentes conjuntos según su naturaleza:

- Nombres propios: añadidos a un conjunto específico, luego integrado al diccionario de nombres.
- Palabras mal lematizadas, siglas o abreviaturas: agregadas a un conjunto de lemas, con el fin de remplazarlas por su forma completa ya lematizada.
- Palabras regionales: en los casos donde la corrección automática resultaba errónea debido a regionalismos, se almaceno en un conjunto destinado a complementar los diccionarios en español.

13. Lematización de palabras: Tal como se definió en la Subsección 1.3.1, la lematización reduce las palabras a su forma canónica (lema) para unificar variantes morfológicas (por ejemplo, “corriendo”, “corrió” y “correrá” se transforman en “correr”). Para aplicar esta técnica se implementó la solución propuesta por Stanza (Qi y cols., 2020), como alternativa a la implementación directa de Spacy (Honnibal y cols., 2020), ya integrado dentro de Spanish NLP. La razón detrás de esta decisión fue que Stanza permite agregar lemas personalizados, logrando capturar también términos locales como “abg” mapeado a “abogado”, “kretina” a “cristina”, “cba” a “córdoba”, entre muchos otros; esta opción también nos permitió alterar el orden de ejecución para ponerlo exactamente al final del proceso; además de obtener resultados más acertados en las pruebas exploratorias realizadas.

2.3 Etapa 3: Procesamiento

En esta etapa se describe el proceso metodológico llevado a cabo para la implementación, la evaluación comparativa y la interpretación de los resultados obtenidos con los modelos seleccionados previamente.

2.3.1 Selección del marco de evaluación

Para LDA (véase Subsección 1.2.2), considerado el modelo más utilizado dentro del paradigma del modelado de tópicos según Shen y cols. (2022); Weisser y cols. (2022);

Vukanti y Jose (2021); Mohammed y cols. (2020), se evaluaron inicialmente múltiples alternativas de implementación. En primera instancia, se intentó implementar la versión original publicada por su autor (blei-lab, 2016). Sin embargo, al no encontrarse disponible el conjunto de datos propuesto en la publicación original, no fue posible validar el modelo bajo las condiciones especificadas, por lo que se descartó esta alternativa.

Posteriormente, se evaluó la posibilidad de utilizar una implementación desarrollada en el lenguaje de programación R (Grün y Hornik, 2024), que funciona como interfaz de la implementación original en C. No obstante, esta alternativa presentaba la limitación de no incluir la coherencia como métrica de evaluación, aspecto considerado fundamental para el presente trabajo, razón por la cual también fue descartada.

Entre las variantes analizadas se incluyó LDA Online (Hoffman y Blei, 2010). Tras este análisis exhaustivo de alternativas, se adoptó el marco de evaluación comparativa propuesto por el autor de BERTopic (véase Subsección 1.2.4) y documentado en un repositorio público (M. Grootendorst, 2025). Dicho marco consiste en entrenar cada modelo de modelado de tópicos variando sistemáticamente la cantidad de tópicos —con incrementos de 10 unidades en la propuesta original— y registrar, para cada configuración, métricas de coherencia y diversidad en archivos de salida en formato JSON. De este modo se obtiene un protocolo unificado aplicable a LDA, NMF y BERTopic, que garantiza resultados metodológicamente consistentes y comparables entre los tres algoritmos.

2.3.2 Configuración del entorno de desarrollo

Una vez definido el marco de evaluación en la subsección precedente, a continuación se expone su implementación técnica y las adaptaciones introducidas.

Como primera medida, se optó por utilizar Docker[2] para garantizar la reproducibilidad y portabilidad del entorno de ejecución, asegurando que el despliegue fuera independiente del sistema operativo subyacente (Windows, Linux o macOS). Una vez configurado

[2] Docker es una plataforma que permite empaquetar aplicaciones con todas sus dependencias en contenedores, garantizando ejecución consistente en cualquier entorno.

el entorno, se desarrolló el código necesario para la validación sistemática de los modelos LDA, NMF y BERTopic, tomando como referencia el marco de evaluación adoptado (véase Subsección 2.3.1).

2.3.3 Mejoras implementadas

Se incorporaron diversas mejoras al marco de evaluación original con el objetivo de enriquecer la robustez del análisis comparativo, enumeradas a continuación:

1. **Ampliación del conjunto de métricas de evaluación.** Considerando que la implementación de referencia empleaba exclusivamente las métricas de coherencia NPMI y diversidad (véase Subsección 1.2.5) para la evaluación comparativa, se decidió incorporar la variante CV de la métrica de coherencia. Esta adición permitió obtener una evaluación más robusta del rendimiento de los modelos, al considerar diferentes perspectivas de medición de la calidad temática.

2. **Refinamiento de la granularidad paramétrica.** El marco de evaluación adoptado (véase Subsección 2.3.1) contempla incrementos de 10 unidades en el número de tópicos. No obstante, con el propósito de obtener una caracterización más detallada del comportamiento de los modelos, se modificó esta configuración para utilizar incrementos de 5 unidades, duplicando efectivamente la resolución en el espacio de parámetros evaluado y permitiendo una identificación más precisa del número óptimo de tópicos.

3. **Implementación de optimización automática de hiperparámetros.** Se incorporó un algoritmo de optimización automática de hiperparámetros desarrollado en la plataforma OCTIS (véase Subsección 1.2.2), basado en métricas configurables por el usuario. Esta implementación sirve como marco de referencia para escenarios donde se requiera optimizar el rendimiento más allá de los valores predeterminados, mediante la configuración del espacio de búsqueda y la métrica objetivo correspondiente.

2.3.4 Herramientas de análisis y visualización

Se desarrolló un conjunto de herramientas para facilitar el análisis sistemático y la interpretación de los resultados obtenidos:

- Optimización del almacenamiento de resultados: se modificó el formato de salida de los archivos JSON para priorizar tanto los valores numéricos de las métricas como las palabras más representativas de cada tópico, facilitando así la revisión manual y la interpretación cualitativa de los resultados.

- Visualización gráfica automatizada: se desarrolló un algoritmo que genera automáticamente gráficos de barras para visualizar la variación de las métricas de coherencia (CV y NPMI) y diversidad en función del número de tópicos, proporcionando una representación visual inmediata del rendimiento de los modelos.

- Visualización interactiva avanzada: se implementó un sistema de visualización interactiva en dos dimensiones para los tópicos generados por BERTopic, que permite comprender mejor las relaciones entre tópicos e identificar las palabras más relevantes de cada uno.

- Sistema de agregación de resultados: se desarrolló un script especializado para la consolidación y visualización sistemática de resultados, organizando las salidas individuales por modelo para facilitar el análisis comparativo.

- Implementación de embedding alternativo: se implementó el modelo de embedding BETO (véase apartado 1.2.4.3) mediante la librería Transformers (Wolf y cols., 2020), que permite el acceso directo a modelos preentrenados disponibles en Hugging Face, la principal plataforma *open source* para modelos de inteligencia artificial. Además, se implementó el modelo de embedding HIIAMSID (véase apartado 1.2.4.3) mediante la librería SentenceTransformer, mantenida también por la empresa Hugging Face.

Debido a que no se obtuvieron mejoras sustanciales con este último modelo, se decidió no incluirlo en el análisis comparativo, ya que correspondería más bien a una comparativa propia de modelos de embedding en BERTopic.

2.3.5 Limitaciones en el procesamiento de grandes volúmenes de datos

Con el objetivo de abordar los desafíos computacionales asociados al análisis de grandes volúmenes de datos, se exploró la utilización de la librería Dask (véase apartado 1.2.5.8) para el procesamiento distribuido en lotes.

Sin embargo, no fue posible completar satisfactoriamente el procesamiento del corpus completo con el modelo BERTopic, ya que el proceso finalizaba de manera forzada por agotamiento de recursos del sistema. Los equipos en los cuales fue probado el modelo contaban con 16 GB de memoria RAM.

Ante esta situación, se decidió entrenar y evaluar todos los modelos sobre un subconjunto de 200.000 tuits, lo que permitió continuar con la investigación en condiciones controladas y mantener la coherencia metodológica del estudio.

2.3.6 Documentación y garantía de reproducibilidad

Se documentó tanto los resultados de las validaciones como la especificación detallada de los comandos y procedimientos necesarios para la replicación completa del proceso experimental. Adicionalmente, se rediseñó la estructura organizacional de directorios siguiendo criterios de modularidad y mantenibilidad, optimizando tanto la organización del proyecto como su gestión colaborativa por parte del equipo de investigación. El acceso al código desarrollado se detalla en el Anexo A.

2.3.7 Obtención de resultados

Conforme al marco de evaluación adoptado (véase Subsección 2.3.1), la ejecución de cada modelo genera como resultado tres archivos en formato JSON, correspondientes a las métricas de evaluación configuradas. Estos documentos almacenan los valores calculados para distintas cantidades de tópicos. Finalmente, el rendimiento global de cada algoritmo se determina mediante el cálculo de un promedio general de dichos resultados.

Capítulo 3: Resultados

En el presente capítulo se exponen los resultados obtenidos durante la investigación, organizados según las tres etapas de la metodología (Capítulo 2): (1) selección de *datasets* (Sección 3.1); (2) preprocesamiento del corpus argentino (Sección 3.2); (3) evaluación comparativa e interpretación de los resultados de LDA, NMF y BERTopic (Sección 3.3).

3.1 Etapa 1: Selección de datasets

En coherencia con la Sección 2.1 del Capítulo 2, la Etapa 1 incorporó los conjuntos de datos 20NewsGroups, BBC News, Trump y elecciones presidenciales argentinas. Sobre los tres *datasets* internacionales se validaron las implementaciones de LDA, NMF y BERTopic mediante la replicación de experimentos documentados en la literatura, con el fin de contrastar coherencia y diversidad temática frente a los valores de referencia. Esta validación confirmó la calibración de los algoritmos y del entorno de pruebas antes de aplicar los modelos al corpus argentino. Dado que el aporte empírico de esta etapa consistió en certificar la reproducibilidad del *pipeline*, el detalle de los conjuntos seleccionados, del procedimiento y de los resultados de la validación se documenta en la metodología citada y en el Anexo A.

3.2 Etapa 2: Preprocesamiento del corpus argentino

A continuación se presentan los resultados de la Etapa 2 (Sección 2.2), aplicados al *dataset* argentino. Las transformaciones documentadas en esta sección reproducen, en el mismo orden secuencial, los pasos allí definidos. En la ejecución sobre el *dataset* completo, dado que el orden de las operaciones condiciona el resultado final en el procesamiento de texto, cada paso toma como entrada la salida del inmediatamente anterior. El título de cada ítem enlaza con la definición correspondiente en dicha sección; la implementación completa se detalla en el Anexo A.

Los recuadros *antes/después* (o *descartado*, cuando corresponde) tienen un fin exclusivamente ilustrativo. En cada ejemplo se aplica únicamente la operación del ítem que se está documentando: no se ejecutan sobre el mismo texto las etapas previas del preprocesamiento. Se utilizan extracciones reales del corpus de tuits, elegidas para evidenciar de forma aislada el efecto de esa transformación; por ello, el contenido del recuadro *antes* no coincide con la salida acumulada que tendría un documento tras recorrer todo el flujo sobre el *dataset*.

3.2.1 Pasos del preprocesamiento

1. Separación de hashtags:

   ANTES

   RT @melimendez1987: A Quien Votaste? #Decision2015 #Elecciones2015 #HoyVotoMACRI #ArgentinaDecide #BuenDomingo #votoporcronica

   DESPUÉS

   RT @melimendez1987: A Quien Votaste? Decision2015 Elecciones2015 Hoy Voto MACRI Argentina Decide Buen Domingo votoporcronica

2. Conversión a minúsculas:

   ANTES

   VAMOS ARGENTINA TODAVIA !!!

   DESPUÉS

   vamos argentina todavia !!!

3. Decodificación de entidades HTML:

   ANTES

   @Donramongo y si ganara Scioli ¿a un Nac&Pop? Ah no, cierto que cerró. Mala mía. Bueno, a un Burger.

DESPUÉS

@Donramongo y si ganara Scioli ¿a un Nac&Pop? Ah no, cierto que cerró. Mala mía. Bueno, a un Burger.

4. Descarte de tuits en otros idiomas:

DESCARTADO

RT @Earth_Pics: Sky on fire. Mount Fitz Roy, Los Glaciers National Park, El Chalten, Argentina - Paul Weeks https://t.co/GevROAokvS

5. Remoción de enlaces a imágenes:

ANTES

Del altar a la urna: fueron a votar después de casarse #balotaje https://t.co/yvM4N1VXa9 pic.twitter.com/Vy5IO... https://t.co/EkVXzWxlfq

DESPUÉS (los enlaces restantes no correspondían a imágenes)

Del altar a la urna: fueron a votar después de casarse #balotaje https://t.co/yvM4N1VXa9 ... https://t.co/EkVXzWxlfq

6. Normalización de lenguaje inclusivo:

ANTES

HOY es un día relevante para todes les habitantes de la Argentina.

DESPUÉS

HOY es un día relevante para todos los habitantes de la Argentina.

7. Eliminación de retuits y menciones:

ANTES

RT @alfanograce: Canchera para el día d las elecciones . @BravabySilvina

DESPUÉS

Canchera para el día d las elecciones .

8. Eliminación de palabras incompletas:

ANTES

RT @Lanataenel13: Hoy Argentina dice BASTA a los atropellos institucionales, al Estado que persigue, a la corrupción y a la mentira. ¡Viv. . .

DESPUÉS

RT @Lanataenel13: Hoy Argentina dice BASTA a los atropellos institucionales, al Estado que persigue, a la corrupción y a la mentira.

9. Normalización de risas:

ANTES

@gabisol65 dice CAMBIEMOS y pone votoascioli JUAJUAJUAJAJAJAJAAAAA @DANYD8

DESPUÉS

@gabisol65 dice CAMBIEMOS y pone votoascioli jaja @DANYD8

10. Spanish NLP:

b) Eliminación de URLs:

ANTES

RT @nicklaud84: Bruce, veo gente muerta. https://t.co/h75BYE79ey

DESPUÉS

RT @nicklaud84: Bruce, veo gente muerta.

e) Normalización de saltos de línea:

ANTES

RT @kpe1: LLEGO EL GRAN DÍA A LAS 20 TOD@S AL OBELISCO EN CÓRDOBA TOD@S AL PATIO OLMOS CON ALEGRÍA HOY CAMBIEMOS #YaVoteMACRI #YoVoteAM...

DESPUÉS

RT @kpe1: LLEGO EL GRAN DÍA A LAS 20 TOD@S AL OBELISCO EN CÓRDOBA TOD@S AL PATIO OLMOS CON ALEGRÍA HOY CAMBIEMOS #YaVoteMACRI #YoVoteAM...

m) Eliminación de espacios múltiples:

ANTES

RT @EmyEnRed: SEÑORAS Y SEÑORES #YoVoteAScioli en un domingo de victoria!!!!!!!

DESPUÉS

RT @EmyEnRed: SEÑORAS Y SEÑORES#YoVoteAScioli en un domingo de victoria!!!!!!!

OBSERVACIÓN

Se detectó un error en este paso: como se observa en el ejemplo "... SEÑORES#YoVoteAScioli..." se perdió el espaciado que debía mantenerse.

n) Eliminación de signos de puntuación: El error expuesto en el punto anterior se solventa en este paso mediante el agregado de espaciados durante el remplazo de caracteres.

ANTES

RT @Bracesco: El FPV quería alquilar el Luna Park para mañana.

DESPUÉS

RT Bracesco El FPV quería alquilar el Luna Park para mañana

ñ) Eliminación de caracteres no imprimibles:

ANTES

©Walter Velázquez

DESPUÉS

Walter Velázquez

o) Eliminación de números:

ANTES

RT @juanpedro1950: Que lindo es verla mendigando 20 minutos de cámara..!

DESPUÉS

RT @juanpedro: Que lindo es verla mendigando minutos de cámara..!

p) Eliminación de stopwords:

ANTES

@diegobranca Vota a Macri, pensa en el pais

DESPUÉS

@diegobranca Vota Macri, pensa pais

11. Normalización de palabras:

ANTES

RT @NotiArgentina: Elecciones 2015: Calu Rivero, una presidente de mesa feliz - LA NACION (Argentina) https://t.co/eInvFuYi6C #Argentina

DESPUÉS

NotiArgentina Elecciones Calu Rivero una presidente mesa feliz NACION Argentina Argentina

12. Corrección de palabras:

ANTES

@endlessgravano Votar a scioli es votar a cristina si quieren un cambio voten a macri y dejen de rompero las volas sepan elejir

DESPUÉS

votar scioli votar cristina quieren cambio voten macri dejen romper bolas sepan elegir

13. Lematización de palabras:

ANTES

@endlessgravano Votar a scioli es votar a cristina si quieren un cambio voten a macri y dejen de rompero las volas sepan elejir

DESPUÉS

votar scioli votar cristina querer cambio votar macri dejar romper bola saber elegir

3.3 Etapa 3: Procesamiento

Implementando la Etapa 3 descrita en la metodología (Sección 2.3), se obtuvieron los siguientes resultados para BERTopic, LDA y NMF sobre el *dataset* de tuits argentinos preprocesados.

En la presente sección, cada una de las tablas de métricas muestran los valores de mayor relevancia obtenidos al variar el número de tópicos para cada modelo. A su vez, las gráficas se generaron por medio del archivo JSON que contenia dichos valores.

3.3.1 BERTopic: embedding all-mpnet-base-v2

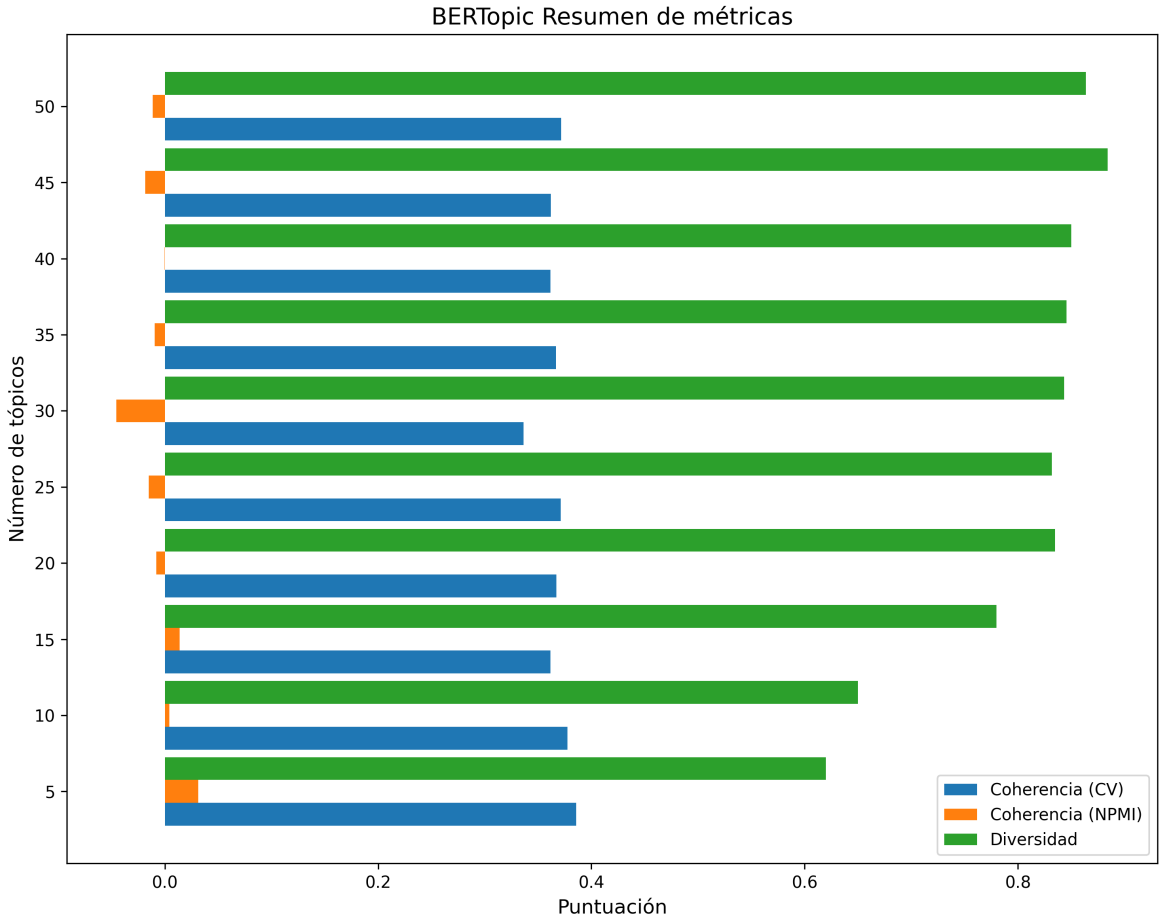
A continuación se presentan los resultados obtenidos con BERTopic (Subsección 1.2.4) utilizando el *embedding* `all-mpnet-base-v2`, conforme a lo descrito en la metodología (Capítulo 2).

Tabla 2: Resultados de las métricas de evaluación para BERTopic (all-mpnet-base-v2)

Métrica	**Valor**
cv	0.38586337480854027
npmi	0.03135620038798654
diversity	0.62

Figura 4

Resultados de las métricas en función del número de tópicos para BERTopic (all-mpnet-base-v2).



Nota. El gráfico compara los valores de coherencia (C_V), NPMI y diversidad temática al variar el número de tópicos. Los resultados corresponden a BERTopic aplicado al corpus de tuits argentinos preprocesados, con el embedding `all-mpnet-base-v2`.

A continuación se presentan las palabras representativas de cada tópico identificado por el modelo:

Tabla 3: Tópicos identificados por BERTopic (embedding all-mpnet-base-v2)

Tópico	**Palabras características**
1	scioli, votar, ganar, mejor, macri, argentino, elección, daniel, vamos, encuesta

Tabla 3: Tópicos identificados por BERTopic (embedding all-mpnet-base-v2) (continuación)

Tópico	Palabras características
2	macri, presidente, cambiar, scioli, mauricio, hoy, cambio, argentino, votar, historia
3	elección, violar, veda, electoral, denunciar, fraude, kirchnerismo, scioli, argentino, decir
4	voto, hoy, histórico, macri, día, blanco, scioli, argentino, votar, cambiar
5	democracia, argentino, elección, voto, hoy, minuto, macri, decisión, pueblo, votar

Interpretación Semántica de Cada Tópico. Los tópicos identificados se interpretan de la siguiente manera:

- Tópico 1: se observa una narrativa centrada en la competencia electoral entre Scioli y Macri, con énfasis en la acción de votar, la intención de ganar y la búsqueda de lo mejor para el país.
- Tópico 2: este tópico destaca la figura de Macri como presidente.
- Tópico 3: aborda cuestiones relacionadas con la veda electoral, posibles violaciones y denuncias de fraude, con el foco en el kirchnerismo y el candidato Scioli.
- Tópico 4: el foco está en el voto, especialmente en el contexto del día de la elección, resaltando su carácter histórico. Se mencionan tanto el voto en blanco como las referencias a los principales candidatos.
- Tópico 5: resalta valores democráticos y la participación del pueblo argentino en la elección. Palabras como “decisión”, “minuto” y “democracia” reflejan la relevancia del acto de votar como expresión de la voluntad popular y la importancia del momento electoral para el país.

3.3.2 BERTopic: embedding BETO

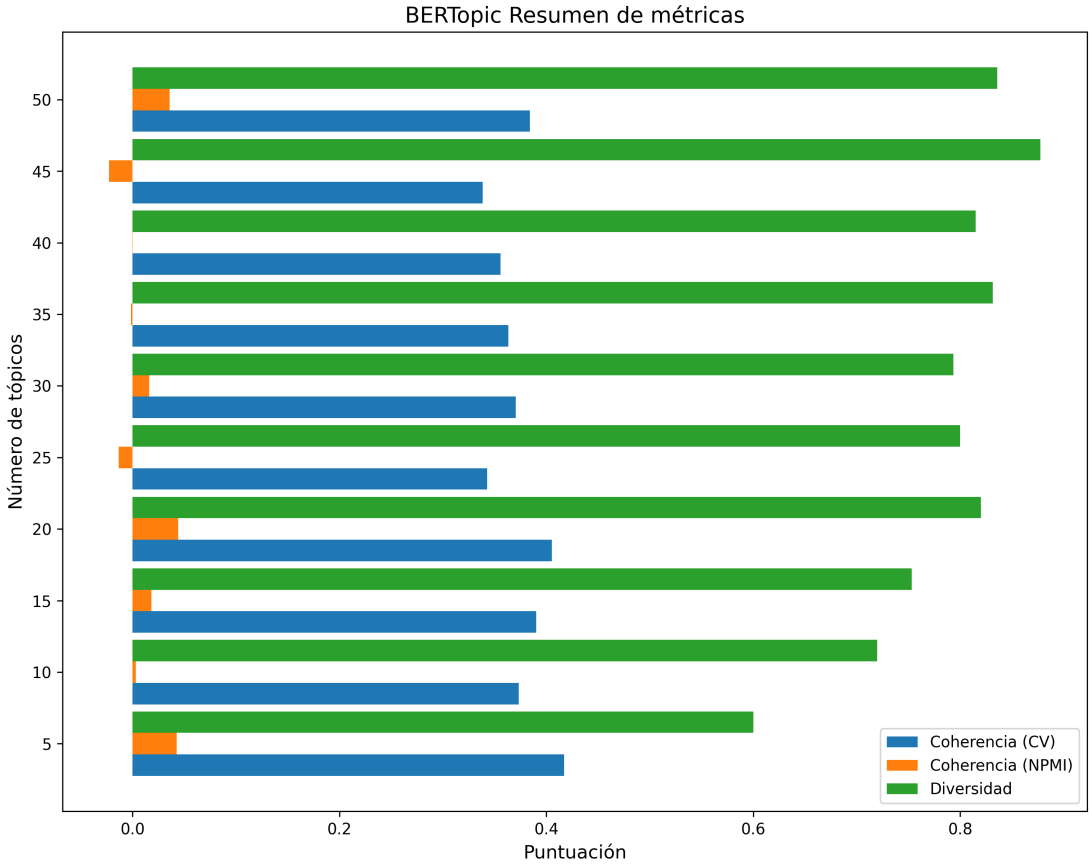
A continuación se presentan los resultados obtenidos con BERTopic (Subsección 1.2.4) utilizando el *embedding* BETO, conforme a lo descrito en la metodología (Capítulo 2).

Tabla 4: Resultados de las métricas de evaluación para BERTopic (BETO)

Métrica	Valor
cv	0.41734500589172185
npmi	0.04254980645483734
diversity	0.6

Figura 5

Resultados de las métricas en función del número de tópicos para BERTopic (BETO).



Nota. El gráfico compara los valores de coherencia (C_V), NPMI y diversidad temática al variar el número de tópicos. Los resultados corresponden a BERTopic aplicado al corpus de tuits argentinos preprocesados, con el embedding BETO.

Resultados Representativos por Tópicos. A continuación se presentan las palabras representativas de cada tópico identificado por el modelo:

Tabla 5: Tópicos identificados por BERTopic (embedding BETO)

Tópico	Palabras características
1	scioli, votar, mejor, macri, ganar, pensar, hablar, optar, argentino, elección
2	voto, hoy, macri, histórico, blanco, día, scioli, votar, argentino, elección
3	presidente, macri, scioli, daniel, cambio, votar, domingo, argentino, hoy, buen
4	vamos, argentino, hoy, elección, scioli, presidente, cambiar, macri, votar, voto
5	elección, veda, violar, denunciar, electoral, scioli, romper, cristino, cristina, hacer

Interpretación Semántica de Cada Tópico. Los tópicos identificados se interpretan de la siguiente manera:

- Tópico 1: en el presente tópico se puede indetificar un debate comparativo, donde se nombran los candidatos que llegaron a la segunda vuelta de la elección presidencial, y se expresa la intención de dar espacio al pensamiento crítico y reflexivo al momento de optar por uno de ellos, con el fin, lógicamente, de elegir al mejor. Compuesto probablemente de sugerencias y opiniones.

- Tópico 2: en el presente tópico nuevamente se hace mención a los candidatos, pero esta vez contextualizandolo en el día de la votación, dando a entender que se trata de un suceso histórico. Además, al estar la palabra “blanco” uno podría interpretar la intención de que se evite el voto en blanco, y se vote con convicción.

- Tópico 3: en el presente tópico se puede identificar un solapamiento con los tópicos mencionados anteriormente, además de una palabra que no debe pasar desapercibida: “cambio”. Este término puede referirse bien al suceso en sí del cambio de figura

presidencial, o más bien al nombre de la coalición política liderada por el candidato Mauricio Macri.

- Tópico 4: en el presente tópico se vuelve a mencionar a los candidatos, como es esperado debido a la tendencia monotemática del período analizado, y se insta a votar, a movilizarse y cumplir con los deberes cívicos.

- Tópico 5: en el presente tópico aparece un nuevo actor político dentro del análisis: Cristina Kirchner.

3.3.3 LDA

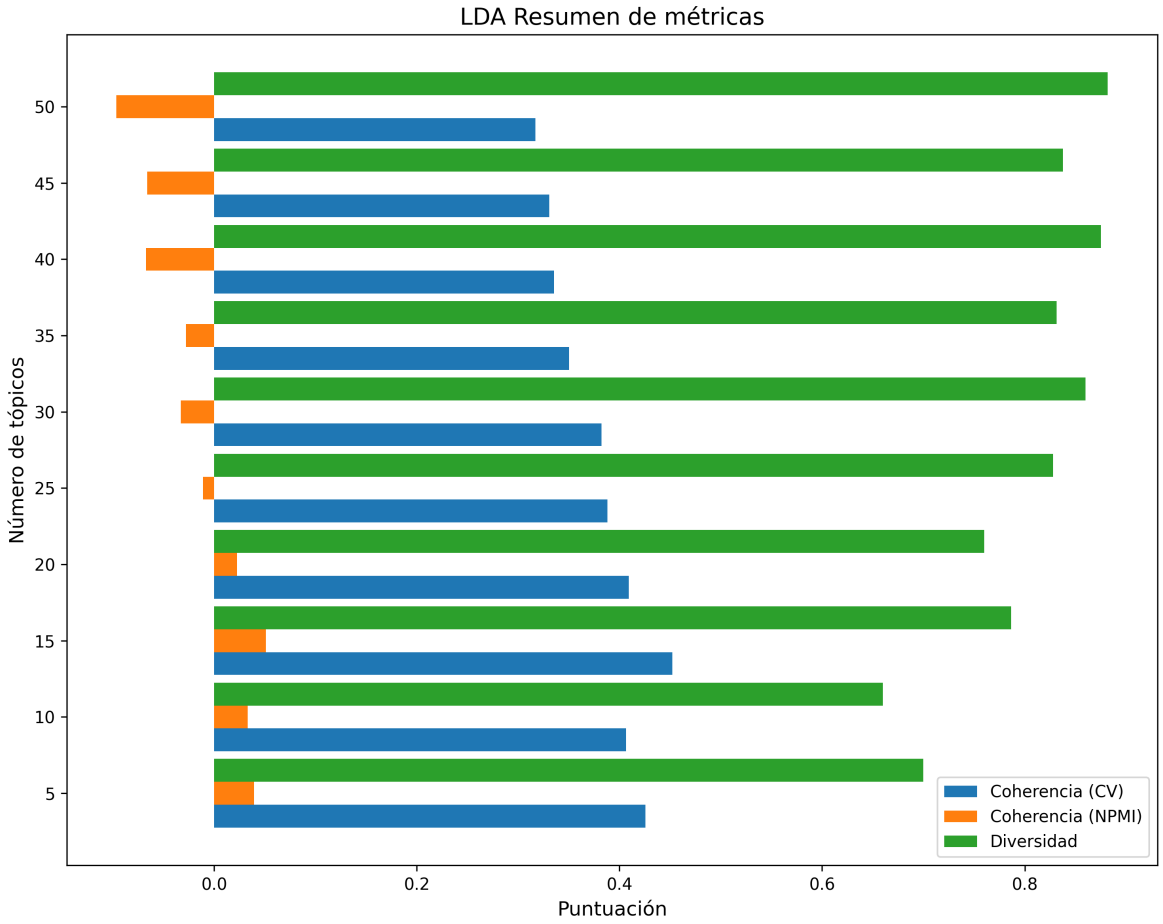
A continuación se presentan los resultados obtenidos con LDA (Subsección 1.2.2), conforme a lo descrito en la metodología (Capítulo 2).

Tabla 6: Resultados de las métricas de evaluación para LDA

Métrica	**Valor**
cv	0.4522636029267376
npmi	0.05114334271030066
diversity	0.7866666666666666

Figura 6

Resultados de las métricas en función del número de tópicos para LDA.



Nota. El gráfico compara los valores de coherencia (C_V), NPMI y diversidad temática al variar el número de tópicos. Los resultados corresponden a LDA aplicado al corpus de tuits argentinos preprocesados.

Resultados Representativos por Tópicos. A continuación se presentan las palabras representativas de cada tópico identificado por el modelo:

Tabla 7: Tópicos identificados por LDA

Tópico	Palabras características
1	scioli, votar, presidente, mejor, voto, daniel, victoria, vamos, falta, respetar
2	cambiar, fiscal, fpv, voto, vez, fraude, gran, único, pro, cada

Tabla 7: Tópicos identificados por LDA (continuación)

Tópico	Palabras características
3	balotaje, elección, esperar, venir, decisión, venezuela, mesa, resultado, acá, final
4	elección, votar, argentino, hablar, quién, deber, bastar, derecho, vuelta, pueblo
5	poder, ser, jaja, gente, seguir, él, tener, pasar, dejar, decir
6	ganar, macri, scioli, blanco, pensar, pueblo, ir, diferencia, decir, hablar
7	día, hoy, buen, histórico, balotaje, domingo, argentino, votar, encuesta, así
8	votar, foto, dar, feliz, creer, boleta, poder, corrupto, suerte, hoy
9	votar, ir, querer, ver, más, poner, nunca, volver, salir, pedir
10	argentino, vamos, hoy, nuevo, elegir, votar, decidir, democracia, país, cambio
11	llegar, candidato, vivir, ahora, minuto, favor, dos, momento, grande, votar
12	veda, violar, electoral, saber, tú, gracia, cristina, denunciar, elección, vía
13	macri, presidente, hoy, voto, cambio, cambiar, mauricio, próximo, fiscalizar, esperanza
14	urna, boleta, boca, todo, escuela, nacional, primero, tarde, vergüenza, listo
15	hacer, decir, scioli, año, campaña, cristino, kirchnerismo, él, después, votar

Interpretación Semántica de Cada Tópico. Los tópicos identificados se interpretan de la siguiente manera:

- Tópico 1: se observa una narrativa vinculada directamente con Scioli, apoyando su candidatura así como la necesidad de respetar el proceso de votación electoral.
- Tópico 2: refleja preocupaciones sobre transparencia electoral, mencionando explícitamente “fiscal”, “fraude” y “FPV”. Parece centrarse en la competencia entre el FPV y el PRO, destacando la posibilidad de prácticas irregulares.
- Tópico 3: este tópico refiere específicamente al balotaje y al proceso electoral en sí, con un énfasis en la espera de resultados y en la expectativa generada. La mención de “Venezuela” podría indicar preocupaciones geopolíticas.

- Tópico 4: se centra en el rol ciudadano y democrático, donde se apela al derecho y deber de votar. Sugiriendo una reflexión sobre la responsabilidad ciudadana en la segunda vuelta electoral.
- Tópico 5: contiene expresiones más genéricas y coloquiales (“jaja”, “seguir”, “pasar”). Probablemente reúne comentarios informales o menos sustanciales sobre la situación política.
- Tópico 6: refleja la polarización entre Macri y Scioli, incorporando también menciones al voto en blanco y al llamado a un ejercicio reflexivo por parte del pueblo al momento de tomar su elección.
- Tópico 7: hace alusión directa al día del balotaje, resaltando su carácter histórico. A su vez, transmite entusiasmo y relevancia del acontecimiento.
- Tópico 8: este tópico sugiere contenido visual compartido. Por otro lado, las menciones a corrupción y suerte reflejan tanto manifestaciones de apoyo como críticas hacia los candidatos.
- Tópico 9: expresa motivaciones personales para ir a votar.
- Tópico 10: se identifica un discurso de entusiasmo colectivo hacia un “nuevo” comienzo democrático. Aparece la idea de país, democracia y cambio, con un enfoque esperanza hacia el futuro.
- Tópico 11: se enfoca en la inmediatez de la jornada electoral, reflejando la tensión del cierre de campaña.
- Tópico 12: hace referencia a la veda electoral y a posibles violaciones de la misma, vinculando a Cristina Kirchner en el debate público.
- Tópico 13: en contraste con el primer tópico, este se centra en Mauricio Macri como proxima figura presidencial.

- Tópico 14: se centra en las críticas e irregularidades respecto a la organización del proceso electoral.
- Tópico 15: aborda cuestiones más amplias de campaña, donde se mencionan a Scioli, Cristina y el kirchnerismo. Se trata de un discurso retrospectivo, que revisa lo hecho en años anteriores y plantea críticas o reflexiones hacia el futuro, con menciones a promesas y declaraciones.

3.3.4 NMF

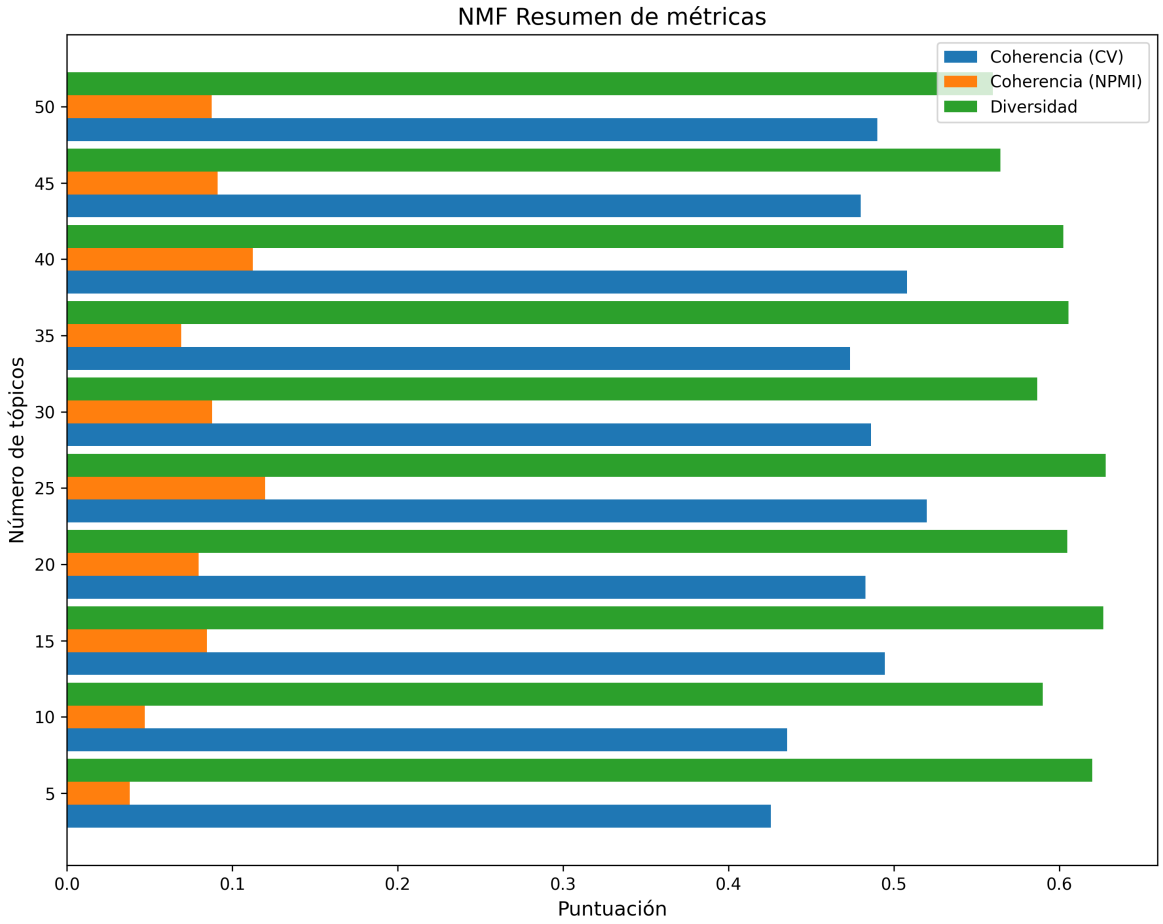
A continuación se presentan los resultados obtenidos con NMF (Subsección 1.2.3), conforme a lo descrito en la metodología (Capítulo 2).

Tabla 8: Resultados de las métricas de evaluación para NMF

Métrica	**Valor**
cv	0.5199723977300792
npmi	0.11981636579835561
diversity	0.628

Figura 7

Resultados de las métricas en función del número de tópicos para NMF.



Nota. El gráfico compara los valores de coherencia (C_V), NPMI y diversidad temática al variar el número de tópicos. Los resultados corresponden a NMF aplicado al corpus de tuits argentinos preprocesados.

Resultados Representativos por Tópicos. A continuación se presentan las palabras representativas de cada tópico identificado por el modelo:

Tabla 9: Tópicos identificados por NMF

Tópico	**Palabras características**
1	argentino, decidir, nuevo, elegir, venezuela, comenzar, etapa, vuelta, futuro, segundo
2	votar, hablar, conciencia, ir, diferencia, gente, pedir, memoria, padrón, sólo

Tabla 9: Tópicos identificados por NMF (continuación)

Tópico	Palabras características
3	elección, hablar, presidencial, domingo, quién, mesa, decisión, democracia, decidir, seguir
4	cristina, odio, derecho, llevar, querés, solo, odiar, ódiala, él, kirchner
5	hoy, pueblo, gente, mucho, confianza, duda, él, nunca, mentira, bastar
6	urna, boca, seguir, encuesta, primero, barómetro, votaste, fpv, sólo, dar
7	día, histórico, buen, feliz, vida, elección, llegar, llego, gran, música
8	voto, macri, hoy, blanco, cuidar, cada, emitir, más, exterior, querer
9	ser, seguir, hablar, jaja, año, país, querer, poder, pelotudo, después
10	mejor, esperar, país, optar, domingo, ir, buen, conciencia, papa, pedir
11	veda, violar, electoral, denunciar, cristino, cristina, campaña, ley, cfk, romper
12	presidente, cambio, nuevo, elegir, próximo, compatriota, sar, candidato, futuro, grande
13	macri, cambio, ir, país, compatriota, buen, aire, ganar, venezuela, scioli
14	saber, así, favor, gente, encuesta, ir, votar, ver, gracia, hora
15	poder, fiscal, dar, creer, boleta, corrupto, cagar, asco, enterar, él
16	ganar, scioli, gente, pueblo, ir, país, confianza, duda, mucho, llegar
17	hacer, él, campaña, jaja, querer, ir, año, pensar, reconocecer, obligar
18	cambiar, fiscal, matanza, fraude, culatazo, pro, evitar, fuego, arma, herido
19	boleta, escuela, mucho, robo, problema, lugar, masivo, conurbano, reportar, laferrere
20	vamos, cambio, elección, gracia, corazón, mundo, junto, carajo, ver, poder
21	decir, ir, país, jaja, fpv, kirchnerismo, aquel, bastar, momento, fiscal
22	macri, mauricio, hablar, diferencia, querer, ver, tú, vida, pasar, etapa
23	scioli, voto, daniel, mejor, jaja, presidente, optar, vamos, votar, victoria
24	ahora, búnker, argentino, fpv, deber, cambiar, llegar, momento, pasar, grande
25	balotaje, quién, primero, encuesta, histórico, presidencial, feliz, paz, minuto, elijar

Interpretación Semántica de Cada Tópico. Los tópicos identificados se interpretan de la siguiente manera:

- Tópico 1: las menciones a “nuevo”, “futuro” y “etapa” sugieren un énfasis en la idea de un comienzo, mientras que la aparición de “Venezuela” probablemente se usa en un tono comparativo o de advertencia.
- Tópico 2: este tópico refleja un llamado a la conciencia cívica y responsable, instando a utilizar la memoria histórica para marcar una diferencia con el voto.
- Tópico 3: aborda el proceso electoral con un enfoque en la deliberación ciudadana (“hablar”, “decidir”, “seguir”).
- Tópico 4: este tópico gira en torno a Cristina Kirchner con expresiones de fuerte rechazo y odio hacia su figura política.
- Tópico 5: se expresa desconfianza y escepticismo, transmitiendo la idea de que se ha perdido la fe en sus promesas.
- Tópico 6: se enfoca en las encuestas y sondeos electorales, mostrando interés por medir y predecir el resultado final del proceso.
- Tópico 7: se observan sentimientos de celebración y optimismo, percibiendo el día electoral como un acontecimiento positivo y trascendente.
- Tópico 8: hace un llamado al cuidado del voto y la participación.
- Tópico 9: este tópico tiene un tono más coloquial y hasta burlón, con expresiones como “jaja” y “pelotudo”. Se mezclan reflexiones sobre el país con lenguaje despectivo, lo que refleja la dimensión más espontánea e informal del discurso en redes.
- Tópico 10: nuevamente se observa un llamado a la reflexión con palabras como “conciencia”, “esperar” y “optar”, similar al tópico 2.

- Tópico 11: se centra en denuncias específicas sobre violación a la veda electoral, apuntando directamente a Cristina Kirchner.
- Tópico 12: expresa esperanza en un futuro mejor a través del cambio.
- Tópico 13: concentra el debate entre Macri y Scioli. Además se menciona nuevamente (como en el tópico 1) a “Venezuela”.
- Tópico 14: este tópico aborda la expectativa y seguimiento del proceso electoral, más que en juicios políticos o posturas partidarias.
- Tópico 15: en este caso se observan discursos de denuncia contra la corrupción y críticas conectadas con preocupaciones por la transparencia electoral.
- Tópico 16: pone el acento en Scioli, la confianza y el apoyo popular. Sin embargo, también aparecen dudas, lo que refleja tanto respaldo como desconfianza hacia el candidato.
- Tópico 17: presenta críticas a las estrategias políticas acompañadas de un tono irónico y despectivo.
- Tópico 18: tópico importante ya que introduce un eje muy sensible: el fraude y la violencia electoral. Palabras como “fraude”, “arma”, “herido” y “culatazo” sugieren denuncias graves y posibles enfrentamientos en el proceso electoral. “Fiscal” y “pro” indican referencias a partidos políticos involucrados en las denuncias.
- Tópico 19: reitera denuncias de irregularidades electorales en el conurbano bonaerense. Se habla específicamente de la localidad de Laferrere, reportando robos de boletas logrando que el votante tenga dificultades al votar. Esto se debe a que el sistema de votación de ese entonces consistía en guardar la boleta deseada en un sobre y llevarlo a la urna; no en un sistema de boleta única.

- Tópico 20: se enmarca en un tono emocional y militante. Refleja entusiasmo, unidad y movilización política, probablemente ligadas a consignas de campaña. Se hace mención también al cambio protagonizado por la alianza política liderada por Macri.
- Tópico 21: refleja críticas al kirchnerismo y al FPV, con un tono de hartazgo. La presencia de “fiscal” sugiere también preocupaciones por el control electoral.
- Tópico 22: gira en torno al candidato Mauricio Macri como figura central, mostrándolo desde un lado más humano y reflexivo.
- Tópico 23: este tópico está enfocado en el candidato Daniel Scioli, mostrando apoyo con referencias a su nombre e incluso posiblemente a su partido político “Frente para la Victoria”.
- Tópico 24: resalta la inminencia de los resultados y la concentración de los políticos en los búnkers. Además, se refleja la necesidad de un cambio, indicando que es el momento de intentar algo distinto en la política.
- Tópico 25: hace foco en el balotaje como evento central, reflejando la incertidumbre en torno al desenlace y la urgencia del momento. Expresiones como “feliz” y “paz” sugieren expectativas positivas en torno al desenlace. Cabe aclarar que se detectó un error en la palabra “elijar” la cual quedó mal lematizada.

3.3.5 Resultados cuantitativos y cualitativos

En este punto, es importante destacar que no existen umbrales universales aceptados para las métricas de evaluación empleadas (véase Subsección 2.3.3), dado que los valores dependen fuertemente del corpus, por ende se adoptó un enfoque de optimización comparando múltiples configuraciones y seleccionando la que obtenga mejores valores en las tres métricas allí indicadas.

Es importante contextualizar estos resultados, ya que, aunque los valores obtenidos puedan interpretarse como bajos en términos absolutos, deben compararse con los resultados documentados por Grootendorst (2022), en especial con aquel referido al corpus de tuits de Donald Trump debido a su naturaleza similar, donde se obtuvo una coherencia NPMI de 0.066 y una diversidad temática de 0.663; por ende se mantiene consistencia con los valores de referencia reportados en textos similares.

En lo que respecta a los resultados del presente trabajo, las métricas obtenidas permiten determinar que NMF demostró una capacidad superior para capturar la granularidad temática del corpus, manteniendo una coherencia (CV y NPMI) superior a los modelos LDA y BERTopic. A su vez, fue capaz de identificar 25 tópicos distintos que abarcan desde aspectos procedimentales del proceso electoral (violación de veda electoral, irregularidades en mesas de votación, fiscalización) hasta dimensiones emocionales y políticas del debate público (apoyo a candidatos, críticas a mandatarios políticos, expresiones de esperanza o desconfianza, denuncia de fraude y violencia electoral). Por otra parte, en lo que respecta a diversidad temática, NMF presentó un rendimiento inferior a LDA, presentando valores similares a BERTopic.

Una de las ventajas más notables de NMF fue su capacidad para distinguir entre tópicos aparentemente similares, como aquellos centrados en Macri (tópicos 8, 13, 22) y Scioli (tópicos 1, 16, 23), capturando diferencias en el tono, las preocupaciones asociadas y el contexto de las menciones, demostrando su capacidad para este tipo de análisis.

Este hallazgo es especialmente relevante para el análisis de redes sociales en períodos de alta actividad alrededor de eventos específicos, donde el contenido tiende a concentrarse en un conjunto limitado de narrativas, pero con matices importantes que deben ser diferenciadas.

Conclusiones

En el presente trabajo se analizó el desempeño de tres modelos de modelado de tópicos (BERTopic, LDA y NMF) aplicados a un corpus monotemático de tuits en español argentino, relacionados con el balotaje presidencial de 2015. Los resultados obtenidos demostraron que Non-negative Matrix Factorization (NMF) superó a los demás en las métricas de evaluación de coherencia seleccionadas, posicionando a NMF como el modelo más adecuado para identificar tópicos coherentes y semánticamente consistentes. No obstante, en diversidad temática fue inferior a LDA y mantuvo valores similares a BERTopic.

La metodología de preprocesamiento propuesta para el español argentino incorpora decisiones específicas para fenómenos frecuentes en medios sociales, como hashtags, lenguaje inclusivo, risas, abreviaturas y errores ortográficos. Este aporte metodológico constituye un insumo transferible para futuras investigaciones en contextos hispanohablantes, donde los recursos y estudios aplicados al modelado de tópicos aún son escasos en comparación al inglés.

Dentro de las limitaciones identificadas, se reconoce el carácter monotemático del corpus principal, el costo computacional asociado al preprocesamiento integral y la dificultad de explorar exhaustivamente todas las configuraciones de hiperparámetros en cada modelo. Aun con dichas limitaciones, la evidencia obtenida confirma la viabilidad del modelado de tópicos para analizar discursos políticos en español argentino y establece una base sólida para la ampliación del estudio a nuevos corpus.

Futuras Líneas de Investigación

Para futuras líneas de investigación, se sugiere la búsqueda o creación de nuevos conjuntos de datos en español argentino pertinentes para el modelado de tópicos, con una

mayor variedad de temas y contextos, lo cual por sí mismo ya confiere una complejidad significativa.

Respecto al preprocesamiento de datos, aunque en el presente trabajo se implementaron técnicas de normalización y limpieza tales como la separación de hashtags, la conversión a minúsculas, la decodificación de entidades HTML, el descarte de tuits en otros idiomas, la remoción de enlaces a imágenes, la normalización de lenguaje inclusivo, la eliminación de retuits y menciones, la eliminación de palabras incompletas, la normalización de risas, la aplicación de la librería Spanish NLP, la normalización de palabras, la corrección ortográfica y la lematización, se propone diferenciar la conveniencia de cada técnica según el tipo de corpus. Un caso podría ser el uso de *stemming* como alternativa al lematizado, que podría reducir significativamente los requerimientos computacionales. Esta exploración es especialmente relevante dado que las limitaciones de memoria encontradas durante el procesamiento del dataset completo podrían mitigarse mediante esta estrategia.

Además, si bien se probaron diferentes configuraciones de parámetros para cada modelo, se propone profundizar este análisis mediante una búsqueda sistemática de la configuración óptima, incorporando métricas de validación complementarias. En el caso de BERTopic, debido a su estructura modular, se recomienda un análisis exhaustivo de alternativas en cada etapa del *pipeline*: modelos de embedding, métodos de reducción de dimensionalidad, algoritmos de clustering, y técnicas de extracción de palabras representativas.

En este sentido, se propone desarrollar una segmentación más robusta de hashtags no delimitados (por ejemplo, escritos completamente en mayúsculas o minúsculas), generando candidatos de subcadenas y seleccionando las particiones más plausibles mediante diccionarios. Asimismo, se sugiere ampliar el alcance a tuits en otros idiomas (por ejemplo, portugués), incorporando detección de idioma y estrategias de procesamiento/representación acordes a un escenario multilingüe. También resulta relevante integrar el análisis de

imágenes asociadas a los tuits, evitando su descarte en el preprocesamiento y explorando representaciones multimodales, así como incorporar el análisis de emojis y emoticones preservando y modelando su contribución semántica y pragmática. Por último, se propone continuar evaluando embeddings alternativos a los considerados, particularmente variantes comparables a BETO y otros modelos en español mencionados en el marco teórico, y realizar experimentos con volúmenes de datos significativamente mayores, utilizando herramientas de cómputo distribuido como Dask (u otras alternativas) para mejorar escalabilidad y robustez.

Finalmente, se propone enriquecer el análisis de resultados mediante visualizaciones interactivas que faciliten la interpretación de los tópicos descubiertos y las relaciones entre sí, permitiendo así una mejor comprensión de cómo cada modelo captura las particularidades del español argentino.

Referencias

Abdelrazek, A., Eid, Y., Gawish, E., Medhat, W., y Hassan Yousef, A. (2022, 10). Topic modeling algorithms and applications: A survey. *Information Systems*, *112*, 102131. doi: 10.1016/j.is.2022.102131

Akhtar, N., Sufyan Beg, M. M., y Javed, H. (2019). Topic modelling with fuzzy document representation. En M. Singh, P. Gupta, V. Tyagi, J. Flusser, T. Ören, y R. Kashyap (Eds.), *Advances in computing and data sciences* (pp. 577–587). Singapore: Springer Singapore.

Al Amrani, Y., Lazaar, M., y El Kadiri, K. E. (2018). Random forest and support vector machine based hybrid approach to sentiment analysis. *Procedia Computer Science*, *127*, 511-520. Descargado de https://www.sciencedirect.com/science/article/pii/S1877050918301625 (PROCEEDINGS OF THE FIRST INTERNATIONAL CONFERENCE ON INTELLIGENT COMPUTING IN DATA SCIENCES, ICDS2017) doi: https://doi.org/10.1016/j.procs.2018.01.150

Albalawi, R., Yeap, T. H., y Benyoucef, M. (2020). Using topic modeling methods for short-text data: A comparative analysis. *Frontiers in artificial intelligence*, *3*, 42.

Aletras, N., y Stevenson, M. (2013, marzo). Evaluating topic coherence using distributional semantics. En A. Koller y K. Erk (Eds.), *Proceedings of the 10th international conference on computational semantics (IWCS 2013) – long papers* (pp. 13–22). Potsdam, Germany: Association for Computational Linguistics. Descargado de https://aclanthology.org/W13-0102/

Allen. (1995). *Natural language understanding*. Pearson Education. Descargado de https://books.google.com.ar/books?id=l0DOH4NHneIC

American Psychological Association. (2020). *Publication manual of the american psychological association* (7th ed.). Washington, DC: American Psychological Association.

Archive, T. T. (2023). *Frequently asked questions.* Descargado de https://www.thetrumparchive.com/faq (Accessed: 2023-11-08)

Barrus, T. (2018). *pyspellchecker: Pure python spell checking library.* https://github.com/barrust/pyspellchecker. (MIT License. Accessed: 2025-08-25)

Barrus, T. (2025). *pyspellchecker: Pure python spell checking based on peter norvig's algorithm.* https://pypi.org/project/pyspellchecker/. (Versión 0.8.3)

Belford, M., Mac Namee, B., y Greene, D. (2018). Stability of topic modeling via matrix factorization. *Expert Systems with Applications*, *91*, 159-169. Descargado de https://www.sciencedirect.com/science/article/pii/S0957417417305948 doi: https://doi.org/10.1016/j.eswa.2017.08.047

Blei, D. M. (2012, abril). Probabilistic topic models. *Commun. ACM*, *55*(4), 77–84. Descargado de https://doi.org/10.1145/2133806.2133826 doi: 10.1145/2133806.2133826

Blei, D. M., Ng, A. Y., y Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993–1022. Descargado de https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf

blei-lab. (2016). *lda-c: C implementation of variational EM for Latent Dirichlet Allocation.* https://github.com/blei-lab/lda-c. (GitHub repository)

Bouma, G. (2009). Normalized (Pointwise) Mutual Information in Collocation Extraction. En *Proceedings of the biennial conference of the gscl* (pp. 31–40).

Casalino, G., Mencar, C., y Nicoletta, D. B. (2016, 01). Non negative matrix factorizations for intelligent data analysis. En (p. 49-74). doi: 10.1007/978-3-662-48331-2_2

Cañete, J., Chaperon, G., Fuentes, R., Ho, J.-H., Kang, H., y Pérez, J. (2020). Spanish pre-trained bert model and evaluation data. En *Pml4dc at iclr 2020.*

Chai, C. P. (2023). Comparison of text preprocessing methods. *Natural Language Engineering*, *29*(3), 509–553. doi: 10.1017/S1351324922000213

Chang, J., Boyd-Graber, J., Gerrish, S., Wang, C., y Blei, D. M. (2009). Reading tea leaves: how humans interpret topic models. En *Proceedings of the 23rd international conference on neural information processing systems* (p. 288–296). Red Hook, NY, USA: Curran Associates Inc.

Chen, Y., Zhang, H., Liu, R., Ye, Z., y Lin, J. (2019). Experimental explorations on short text topic mining between lda and nmf based schemes. *Knowledge-Based Systems*, *163*, 1-13. Descargado de `https://www.sciencedirect.com/science/article/pii/S0950705118304076` doi: https://doi.org/10.1016/j.knosys.2018.08.011

Church, K. W., y Hanks, P. (1990). Word Association Norms, Mutual Information, and Lexicography. *Computational Linguistics*, *16*(1), 22–29.

Churchill, R., y Singh, L. (2021). textprep: A text preprocessing toolkit for topic modeling on social media data. En *Proceedings of the 10th international conference on data science, technology and applications - volume 1: Data,* (p. 60-70). SciTePress. doi: 10.5220/0010559000600070

Danilak, M. M. (2021). *langdetect.* `https://pypi.org/project/langdetect/`. (Python package, Apache Software License)

Dask Development Team. (2024). *Dask: Library for dynamic task scheduling.* `https://www.dask.org/`. (Accedido el 19 de mayo de 2026)

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., y Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information*

Science, *41*(6), 391-407. Descargado de https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-4571%28199009%2941%3A6%3C391%3A%3AAID-ASI1%3E3.0.CO%3B2-9 doi: https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9

Denning, P. J. (1985). The science of computing: What is computer science? *American Scientist*, *73*(1), 16–19. Descargado 2025-04-01, de http://www.jstor.org/stable/27853057

Devlin, J., Chang, M.-W., Lee, K., y Toutanova, K. (2019, junio). BERT: Pre-training of deep bidirectional transformers for language understanding. En J. Burstein, C. Doran, y T. Solorio (Eds.), *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics. Descargado de https://aclanthology.org/N19-1423/ doi: 10.18653/v1/N19-1423

Dieng, A. B., Dieng, A. B., Ruiz, F. J. R., Ruiz, F., Ruiz, F. J. R., Ruiz, F. J. R., ... Blei, D. M. (2020). Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*. doi: 10.1162/tacl_a_00325

Dieng, A. B., Ruiz, F. J. R., y Blei, D. M. (2019). Topic modeling in embedding spaces. *CoRR*, *abs/1907.04907*. Descargado de http://arxiv.org/abs/1907.04907

Egger, R., y Yu, J. (2022). A topic modeling comparison between lda, nmf, top2vec, and bertopic to demystify twitter posts. *Frontiers in sociology*, *7*, 886498.

et al., D. N. (2025). *Languagetool: Corrector gramatical de estilo y ortografía.* https://github.com/languagetool-org/languagetool. (Accedido el 9 de septiembre de 2025)

Fano, R. M. (1961). *Transmission of information: A statistical theory of communications*. MIT Press.

Fenollosa, C. (2019). *Spanish free dictionary.* Dataset. Descargado de https://cfenollosa.com/blog/tag_nlp.html (Licencia GNU FDL. Recuperado el 21 de agosto de 2025)

Gelman, A., Carlin, J., Stern, H., y Rubin, D. (2003). *Bayesian data analysis, second edition*. Taylor & Francis. Descargado de https://books.google.com.ar/books?id=TNYhnkXQSjAC

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., y Rubin, D. B. (2013). *Bayesian data analysis* (Third ed.). Boca Raton, Florida: CRC. Descargado de https://stat.columbia.edu/~gelman/book/

Greene, D., y Cunningham, P. (2006). Practical solutions to the problem of diagonal dominance in kernel document clustering. En *Proceedings of the 23rd international conference on machine learning* (pp. 377–384).

Grishman, R. (1986). *Computational linguistics: An introduction*. Cambridge University Press.

Grootendorst, M. (2022). Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Grün, B., y Hornik, K. (2024). topicmodels: Topic models [Manual de software informático]. Descargado de https://CRAN.R-project.org/package=topicmodels (R package version 0.2-17) doi: 10.32614/CRAN.package.topicmodels

Guo, K., Yang, Z., Yu, C.-H., y Buehler, M. J. (2021). Artificial intelligence and machine learning in design of mechanical materials. *Mater. Horiz.*, *8*, 1153-1172. Descargado de http://dx.doi.org/10.1039/D0MH01451F doi: 10.1039/D0MH01451F

Hernández-Sampieri, R., Fernández-Collado, C., y Baptista Lucio, P. (2014). *Metodología de la investigación* (6a ed.). México: McGraw-Hill.

Hien, L. T. K., y Gillis, N. (2021, mayo). Algorithms for nonnegative matrix factorization with the kullback–leibler divergence. *Journal of Scientific Computing*, *87*(3). Descargado de http://dx.doi.org/10.1007/s10915-021-01504-0 doi: 10.1007/s10915-021-01504-0

hiiamsid. (2021). *sentence_similarity_spanish_es.* https://huggingface.co/hiiamsid/sentence_similarity_spanish_es. HuggingFace. (Accedido: [fecha de acceso])

Hoffman, M., y Blei, D. (2010, 11). Online learning for latent dirichlet allocation. En (Vol. 23, p. 856-864).

Honnibal, M., y Montani, I. (2023). *spacy: Industrial-strength natural language processing in python.* https://spacy.io. (Accedido el 03/04/2025)

Honnibal, M., Montani, I., Van Landeghem, S., y Boyd, A. (2020). spaCy: Industrial-strength Natural Language Processing in Python.
doi: 10.5281/zenodo.1212303

IBM. (s.f.). *Modelos de redes neuronales.* Descargado de https://www.ibm.com/docs/es/spss-modeler/saas?topic=networks-neural-model (Recuperado el 13 de mayo de 2025)

Jordan, M., Ghahramani, Z., Jaakkola, T., y Saul, L. (1999, 01). An introduction to variational methods for graphical models. *Machine Learning*, *37*, 183-233. doi: 10.1023/A:1007665907178

Jurafsky, D., y Martin, J. H. (2025). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition with language models* (3rd ed.). Descargado de https://web.stanford.edu/~jurafsky/slp3/ (Online manuscript released January 12, 2025)

Kass, R. E., y Steffey, D. (1989). Approximate bayesian inference in conditionally independent hierarchical models (parametric empirical bayes models). *Journal of the American Statistical Association*, *84*(407), 717–726. Descargado 2025-06-24, de http://www.jstor.org/stable/2289653

Khyani, D., B S, S., Niveditha, N., M., D., y Y M, D. (2021, 01). An interpretation of lemmatization and stemming in natural language processing. *Shanghai Ligong Daxue Xuebao/-Journal of University of Shanghai for Science and Technology*, *22*, 350-357.

Kumar, A., y Garg, G. (2019, March). Sarc-m: sarcasm detection in typo-graphic memes. En *International conference on advances in engineering science management & technology (icaesmt)-2019, uttaranchal university, dehradun, india.*

Lafferty, J., y Blei, D. (2005). Correlated topic models. En Y. Weiss, B. Schölkopf, y J. Platt (Eds.), *Advances in neural information processing systems* (Vol. 18). MIT Press. Descargado de https://proceedings.neurips.cc/paper_files/paper/2005/file/9e82757e9a1c12cb710ad680db11f6f1-Paper.pdf

Lang, K. (1995). Newsweeder: Learning to filter netnews. En *Machine learning proceedings 1995* (pp. 331–339). Elsevier.

Larriva-Novo, X., Villagrá, V. A., Vega-Barbas, M., Rivera, D., y Sanz Rodrigo, M. (2021). An iot-focused intrusion detection system approach based on preprocessing characterization for cybersecurity datasets. *Sensors*, *21*(2). Descargado de https://www.mdpi.com/1424-8220/21/2/656 doi: 10.3390/s21020656

Lau, J. H., Newman, D., y Baldwin, T. (2014, abril). Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. En S. Wintner, S. Goldwater, y S. Riezler (Eds.), *Proceedings of the 14th conference of the European chapter of the association for computational linguistics* (pp. 530–539). Gothenburg, Sweden: Association

for Computational Linguistics. Descargado de https://aclanthology.org/E14-1056/ doi: 10.3115/v1/E14-1056

Lee, D., y Seung, H. S. (2000). Algorithms for non-negative matrix factorization. En T. Leen, T. Dietterich, y V. Tresp (Eds.), *Advances in neural information processing systems* (Vol. 13). MIT Press. Descargado de https://proceedings.neurips.cc/paper_files/paper/2000/file/f9d1152547c0bde01830b7e8bd60024c-Paper.pdf

Lee, D. D., y Seung, H. S. (1999, 10). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*(6755), 788-791. Descargado de https://doi.org/10.1038/44565 doi: 10.1038/44565

Lérin, J. D. (2024). *Diccionario con las palabras del español en formato txt.* Dataset. Descargado de https://github.com/JorgeDuenasLerin/diccionario-espanol-txt (Actualizado el 22 de mayo de 2024)

M. Grootendorst. (2025). *BERTopic_evaluation: Code and experiments for BERTopic: Neural topic modeling with a class-based TF-IDF procedure.* https://github.com/MaartenGr/BERTopic_evaluation. (GitHub repository, accessed 2025-09-02)

McCarthy, J. (1956). *What is artificial intelligence?* (Stanford University Computer Science Department)

McInnes, L., Healy, J., y Astels, S. (2017). hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, *2*(11), 205. Descargado de https://doi.org/10.21105/joss.00205 doi: 10.21105/joss.00205

McInnes, L., Healy, J., y Melville, J. (2020). *Umap: Uniform manifold approximation and projection for dimension reduction.* Descargado de https://arxiv.org/abs/1802.03426

migalpha. (2018). *Spanish names.* Kaggle dataset. Descargado de https://www.kaggle.com/datasets/migalpha/spanish-names/data (Licencia GPL 2. Recuperado el 21 de agosto de 2025)

Mikolov, T., Chen, K., Corrado, G., y Dean, J. (2013). *Efficient estimation of word representations in vector space.* Descargado de https://arxiv.org/abs/1301.3781

Milne, D., y Watling, D. (2019). Big data and understanding change in the context of planning transport systems. *Journal of Transport Geography*, *76*(C), 235-244. Descargado de https://ideas.repec.org/a/eee/jotrge/v76y2019icp235-244.html doi: 10.1016/j.jtrangeo.2017.11.004

Mimno, D., Wallach, H., Talley, E., Leenders, M., y McCallum, A. (2011, julio). Optimizing semantic coherence in topic models. En R. Barzilay y M. Johnson (Eds.), *Proceedings of the 2011 conference on empirical methods in natural language processing* (pp. 262–272). Edinburgh, Scotland, UK.: Association for Computational Linguistics. Descargado de https://aclanthology.org/D11-1024/

Mohammed, S. A., Mohammed, S. H., y Al-augby, S. (2020). Lsa & lda topic modeling classification: comparison study on e-books. *Indonesian Journal of Electrical Engineering and Computer Science*. doi: 10.11591/ijeecs.v19.i1.pp%p

Moody, C. E. (2016). *Mixing dirichlet topic models and word embeddings to make lda2vec.* Descargado de https://arxiv.org/abs/1605.02019

Morris, C. N. (1983). Parametric empirical bayes inference: Theory and applications. *Journal of the American Statistical Association*, *78*(381), 47–55. Descargado 2025-06-24, de http://www.jstor.org/stable/2287098

Morris, J. (2024). *language_tool_python: A free python grammar checker.* GitHub repository. Descargado de https://github.com/jxmorris12/language_tool_python (Licencia GPL-3.0. Última actualización: 3 de junio de 2025. Consultado el 21 de agosto de 2025)

Newman, D., Lau, J. H., Grieser, K., y Baldwin, T. (2010, junio). Automatic evaluation of topic coherence. En R. Kaplan, J. Burstein, M. Harper, y G. Penn (Eds.), *Human language technologies: The 2010 annual conference of the north American chapter of the association for computational linguistics* (pp. 100–108). Los Angeles, California: Association for Computational Linguistics. Descargado de https://aclanthology.org/N10-1012/

NLTK Project. (2023). *Natural language toolkit (nltk).* https://www.nltk.org/. (Accedido el 03/04/2025)

Orellana, G., Arias, B., Orellana, M., Saquicela, V., Baculima, F., y Piedra, N. (2018). A study on the impact of pre-processing techniques in spanish and english text classification over short and large text documents. En *2018 international conference on information systems and computer science (inciscos)* (p. 277-283). doi: 10.1109/INCISCOS.2018.00047

Ortiz-Fuentes, J. (2022). *Spanish nlp: A python library for natural language processing in spanish.* https://github.com/jorgeortizfuentes/spanish_nlp. GitHub.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, É. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Qi, P., Dozat, T., Zhang, Y., y Manning, C. D. (2018, October). Universal dependency parsing from scratch. En *Proceedings of the CoNLL 2018 shared task: Multilingual parsing*

from raw text to universal dependencies (pp. 160–170). Brussels, Belgium: Association for Computational Linguistics. Descargado de https://nlp.stanford.edu/pubs/qi2018universal.pdf

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., y Manning, C. D. (2020). Stanza: A Python natural language processing toolkit for many human languages. En *Proceedings of the 58th annual meeting of the association for computational linguistics: System demonstrations.* Descargado de https://nlp.stanford.edu/pubs/qi2020stanza.pdf

Reimers, N., y Gurevych, I. (2019, 11). Sentence-bert: Sentence embeddings using siamese bert-networks. En *Proceedings of the 2019 conference on empirical methods in natural language processing.* Association for Computational Linguistics. Descargado de https://arxiv.org/abs/1908.10084

Robins, D., Frati, F. E., Alvarez, J., y Texier, J. (2016). *Balotage in argentina 2015, a sentiment analysis of tweets.* Descargado de https://arxiv.org/abs/1611.02337

Röder, M., Both, A., y Hinneburg, A. (2015). Exploring the space of topic coherence measures. En *Proceedings of the eighth acm international conference on web search and data mining* (p. 399–408). New York, NY, USA: Association for Computing Machinery. Descargado de https://doi.org/10.1145/2684822.2685324 doi: 10.1145/2684822.2685324

Roh, J., Oh, S.-H., y Lee, S.-Y. (2020). Unigram-normalized perplexity as a language model performance measure with different vocabulary sizes. *arXiv preprint arXiv:2011.13220*.

Sailunaz, K., y Alhajj, R. (2019). Emotion and sentiment analysis from twitter text. *Journal of Computational Science*, *36*, 101003.

Sandoval, A. M. (1998). *Lingüística computacional: introducción a los modelos simbólicos, estadísticos y biológicos*. Editorial Síntesis. Descargado de https://books.google.com.ar/books?id=o01nQgAACAAJ

Seung, D., y Lee, L. (2001). Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, *13*(556-562), 3.

Shen, J., Huang, W., Wen, H., y Hu, Q. (2022). Picf-lda: a topic enhanced lda with probability incremental correction factor for web api service clustering. *Journal of Cloud Computing*. doi: 10.1186/s13677-022-00291-9

Solc, T. (2009). *Unidecode: Ascii transliterations of unicode text.* `https://pypi.org/project/Unidecode/`. (Versión 1.4.0)

Stanford NLP Group. (2020). *Stanza: A python nlp library for many human languages.* `https://stanfordnlp.github.io/stanza/`. (Accedido el 03/04/2025)

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, *36*(2), 111–147. Descargado 2025-06-26, de `http://www.jstor.org/stable/2984809`

Symeonidis, S., Effrosynidis, D., y Arampatzis, A. (2018). A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis. *Expert Systems with Applications*, *110*, 298-310. Descargado de `https://www.sciencedirect.com/science/article/pii/S0957417418303683` doi: https://doi.org/10.1016/j.eswa.2018.06.022

Tabassum, A., y Patil, D. R. R. (2020). A survey on text pre-processing & feature extraction techniques in natural language processing.. Descargado de `https://api.semanticscholar.org/CorpusID:235211496`

Talamé, L., Cardoso, A., y Amor, M. (2019, septiembre). Comparación de herramientas de procesamiento de textos en español extraídos de una red social para python. En *Xx simposio argentino de inteligencia artificial (asai 2019) - jaiio 48* (p. 53-67). Salta,

Argentina: Sociedad Argentina de Informática e Investigación Operativa. (Document type: Objeto de conferencia)

Tatman, R. (2017). *English word frequency: 1/3 million most frequent english words on the web.* Kaggle dataset. Descargado de https://www.kaggle.com/datasets/rtatman/english-word-frequency (Licencia MIT. Recuperado el 21 de agosto de 2025)

Teh, Y. W., Jordan, M. I., Beal, M. J., y and, D. M. B. (2006). Hierarchical dirichlet processes. *Journal of the American Statistical Association*, *101*(476), 1566–1581. doi: 10.1198/016214506000000302

Terragni, S., Fersini, E., Galuzzi, B. G., Tropeano, P., y Candelieri, A. (2021, abril). OCTIS: Comparing and optimizing topic models is simple! En D. Gkatzia y D. Seddah (Eds.), *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: System demonstrations* (pp. 263–270). Online: Association for Computational Linguistics. Descargado de https://aclanthology.org/2021.eacl-demos.31/ doi: 10.18653/v1/2021.eacl-demos.31

Tessore, J. P., Esnaola, L. M., Russo, C. C., y Baldassarri, S. (2019). Comparative analysis of preprocessing tasks over social media texts in spanish. En *Proceedings of the xx international conference on human computer interaction.* New York, NY, USA: Association for Computing Machinery. Descargado de https://doi.org/10.1145/3335595.3335632 doi: 10.1145/3335595.3335632

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2023). *Attention is all you need.* Descargado de https://arxiv.org/abs/1706.03762

Vijayarani, S., Ilamathi, J., y Nithya. (2015). Preprocessing techniques for text mining - an overview. *International Journal of Computer Science & Communication Networks*, *5*(1), 7–16. (Available at ResearchGate or the publisher's site)

Vukanti, V., y Jose, A. (2021). Business analytics: A case-study approach using lda topic modelling. *International Conference Computing Methodologies and Communication*. doi: 10.1109/iccmc51019.2021.9418344

Wachirapong, F. (2023). *Exploring the effect of preprocessing techniques on the topic modeling in social science data* (Tesis de Master no publicada). University of Helsinki, P. O. Box 68, 00014 University of Helsinki.

Wang, B., Wang, A., Chen, F., Wang, Y., y Kuo, C.-C. J. (2019). Evaluating word embedding models: methods and experimental results. *APSIPA Transactions on Signal and Information Processing*, *8*(1). Descargado de http://dx.doi.org/10.1017/ATSIP.2019.12 doi: 10.1017/atsip.2019.12

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., y Zhou, M. (2020). *Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers.* Descargado de https://arxiv.org/abs/2002.10957

Weisser, C., Gerloff, C., Thielmann, A., Python, A., Reuter, A., Kneib, T., y Säfken, B. (2022). Pseudo-document simulation for comparing lda, gsdmm and gpm topic models on short and sparse text using twitter data. *Computational Statistics*. doi: 10.1007/s00180-022-01246-z

Wiederhold, G., y McCarthy, J. (1992, 06). Arthur samuel: Pioneer in machine learning. *IBM Journal of Research and Development*, *36*, 329 - 331. doi: 10.1147/rd.363.0329

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... Rush, A. M. (2020, octubre). Transformers: State-of-the-art natural language processing. En *Proceedings of the 2020 conference on empirical methods in natural language processing: System demonstrations* (pp. 38–45). Online: Association for Computational Linguistics. Descargado de https://www.aclweb.org/anthology/2020.emnlp-demos.6

Yang, M.-S. (1993). A survey of fuzzy clustering. *Mathematical and Computer Modelling*, *18*(11), 1-16. Descargado de https://www.sciencedirect.com/science/article/pii/089571779390202A doi: https://doi.org/10.1016/0895-7177(93)90202-A

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, *8*(3), 338-353. Descargado de https://www.sciencedirect.com/science/article/pii/S001999586590241X doi: https://doi.org/10.1016/S0019-9958(65)90241-X

Zimmermann, J., Champagne, L. E., Dickens, J. M., y Hazen, B. T. (2024). Approaches to improve preprocessing for latent dirichlet allocation topic modeling. *Decision Support Systems*, *185*, 114310. Descargado de https://www.sciencedirect.com/science/article/pii/S016792362400143X doi: https://doi.org/10.1016/j.dss.2024.114310

Anexo A

El código y los scripts desarrollados en el marco de esta investigación se encuentran disponibles en los siguientes repositorios públicos de GitHub:

- Preprocesamiento: `https://github.com/FedeRuhl/preprocess`
- Modelado de tópicos (LDA, NMF y BERTopic): `https://github.com/gonzalo363rm/BERTopic`

En el repositorio de preprocesamiento se documentan las etapas descritas en la Sección 2.2.

En el repositorio de modelado de tópicos se incluyen las adaptaciones al marco de evaluación adoptado (véase Subsección 2.3.1), las herramientas de análisis y visualización, y los procedimientos necesarios para replicar las ejecuciones experimentales.

A.1 Material complementario del preprocesamiento

En el paso de descarte de tuits en otros idiomas se emplea la siguiente lista de palabras en inglés, identificadas manualmente a partir de los documentos completamente escritos en ese idioma:

- amazing
- bcbbf
- bjorloth
- chalten
- cup
- dead
- dibidik
- doodle
- drone
- eyes
- fitz
- fmln
- fond
- gelandang
- glaciers
- liverpool
- loves
- mount

- pleasure
- polls
- seconds
- sky
- spell
- strong
- taking
- trending
- voting
- wait
- weeks
- win
- winehouse
- yesterday
- 734b701c7b7b41f

Las palabras fueron seleccionadas por frecuencia de aparición en tuits íntegramente escritos en inglés; entre ellas figuran tokens atípicos (identificadores, nombres propios o cadenas alfanuméricas) que concentraban dichas apariciones.