Análisis de Clustering de Literatura Científica sobre COVID-19

Daniela Lopera, Carlos Andrés Jaramillo, Luis Miguel Caicedo
Maestría en Ciencia de Datos y Analítica
Universidad de EAFIT

1. Problema de investigación y objetivos

La COVID-19 es una enfermedad infecciosa causada por el virus SARS-CoV-2 que tuvo origen en la ciudad de Wuhan, China, en diciembre de 2019. Desde su aparición, la COVID-19 se propago rápidamente por todo el mundo, convirtiéndose en una pandemia global que ha tenido un profundo impacto en la salud pública, la economía y la vida cotidiana de las personas. En este sentido, en la búsqueda de generar una rápida respuesta a la expansión del virus se dio una aceleración significativa en la investigación y producción de literatura por parte de la comunidad científica en esta área, generando una gran cantidad de información en un corto período de tiempo.

La gran cantidad de literatura desarrollada generó un reto para el análisis y consulta por parte de la comunidad científica y de investigación médica dado que se dificulta la identificación y clasificación de tópicos de investigación realizada. En este proyecto de investigación se propone la implementación de técnicas de minería de datos basados en clustering para la agrupación de los datos de investigación abierta de COVID-19 (CORD-19); esto con el objetivo de identificar patrones, agrupar temas y descubrir tendencias emergentes en la investigación sobre COVID-19, SARS-CoV-2 y coronavirus relacionados en la literatura disponible Finalmente, se espera poder contribuir a la comprensión y la toma de decisiones informadas en relación con la investigación científica desarrollada.

1.1 Objetivo general:

El objetivo general de este proyecto de investigación es aplicar técnicas de minería de datos basados en clustering y análisis de tendencias en el conjunto de datos de investigación abierta de COVID-19 (CORD-19) con el fin de identificar patrones emergentes y tendencias significativas en la literatura científica relacionada con COVID-19, SARS-CoV-2 y coronavirus.

1.2 Objetivos específicos:

- Seleccionar y aplicar un algoritmo de clustering adecuado, considerando la magnitud de los datos y la naturaleza de la investigación científica, para agrupar los documentos académicos en clúster basados en similitudes temáticas.

- Analizar los resultados del clustering y determinar los temas y enfoques predominantes en cada clúster, identificando patrones emergentes y áreas de interés.
- Utilizar técnicas de visualización, como PCA o t-SNE, para representar gráficamente los clústeres en un espacio bidimensional, permitiendo una comprensión intuitiva de los patrones de agrupación.
- Identificar tendencias emergentes en la investigación sobre COVID-19 mediante el análisis de los temas más frecuentes y las áreas de enfoque que están ganando importancia en la literatura científica.
- Sintetizar los resultados en un informe detallado que presente las tendencias identificadas, los patrones emergentes y las implicaciones para la comunidad de investigación médica y las autoridades de salud.

1. Revisión de Literatura y Estado del Arte

El campo de la clasificación de documentos, las tecnologías y enfoques evolucionan constantemente debido al avance de la inteligencia artificial y el aprendizaje automático [1], [2]. Dentro de estos avances se destacan varios frentes de desarrollo conjunto entre el aprendizaje profundo y el procesamiento de texto como son: (i) embeddings de palabras y documentos [3] , (ii) aprendizaje profundo, (iii) lenguaje natural [1] y (iv) el desarrollo de los Transformers [4].

Las técnicas de embeddings o esquemas de incrustación de palabras, como Word2Vec y Doc2Vec, permiten representar documentos y palabras en un espacio vectorial, lo que facilita la clasificación y búsqueda de documentos similares. Los esquemas de incrustación de palabras son aplicaciones de aprendizaje profundo. Algunos esquemas como como word2vec y vectores globales (GloVe), se han empleado con éxito para tareas de procesamiento del lenguaje natural, como clasificación de texto, análisis de sentimientos e identificación de sarcasmo [5]. Con el uso de esquemas de incrustación de palabras, se pueden extraer y representar relaciones sintácticas y semánticas entre las palabras/frases. Estos esquemas permiten capturar características latentes de colecciones de texto mediante capas de la arquitectura de red neuronal profunda. Por lo tanto, se han eliminado las etapas involucradas en la extracción de características y el preprocesamiento de datos en los esquemas de minería de texto convencionales [3], [6].

Como se indicó, los esquemas basados en la incrustación de palabras se han empleado con éxito para tareas de procesamiento del lenguaje natural. Sin embargo, los esquemas de incrustación de palabras, como word2vec y GloVe, requieren un corpus muy grande para entrenar y construir la representación vectorial [7].Por esta razón, los modelos de incrustación de palabras previamente entrenados se pueden emplear en tareas de procesamiento del lenguaje natural basadas en el aprendizaje profundo. El rendimiento predictivo de tales esquemas puede no ser prometedor, ya que el modelo previamente entrenado no es específico para la tarea/datos correspondientes [6], [8]. Es posible que los cálculos de vectores no tengan en cuenta el contexto de los documentos y no se hayan modelado las relaciones entre palabras que no son literalmente concurrentes [7]. Para eliminar los problemas de los esquemas de incrustación de palabras, varias contribuciones de

investigación incorporaron diferentes esquemas de incrustación de palabras en una representación vectorial de conjunto.

En cuanto aprendizaje profundo (Deep Learning), las redes neuronales profundas, como las redes neuronales convolucionales (CNN) y las redes neuronales recurrentes (RNN), se han utilizado para abordar la clasificación de documentos de texto y otros tipos de documentos [3]. Las redes neuronales profundas han demostrado ser eficaces para aprender características complejas y realizar clasificaciones precisas. Esta técnica ha revolucionado la clasificación de documentos de texto, especialmente con el desarrollo procesamiento de texto basado en Transformers [4]. Modelos como BERT y GPT (Generative Pre-trained Transformer) se han destacado por su capacidad para comprender y clasificar texto de manera precisa[9]. Los Transformers se han convertido en un modelo destacado de aprendizaje profundo que ha sido ampliamente adoptado en diversos campos, como el procesamiento del lenguaje natural (NLP), la visión por computadora (CV) y el procesamiento del habla [2].

El primer Transformer se propuso originalmente como un modelo de secuencia a secuencia [10] para traducción automática. Trabajos posteriores muestran que los modelos pre-entrenados (PTM) basados en Transformer pueden lograr rendimientos de última generación en diversas tareas. Como consecuencia, Transformer se ha convertido en la arquitectura de referencia en PNL, especialmente para PTM.

Además de las aplicaciones relacionadas con el lenguaje, los Transformer también se han adoptado en CV, procesamiento de audio e incluso otras disciplinas, como la química y las ciencias de la vida [10]. Debido al éxito, se lanzaron una variedad de variantes de Transformer (a.k.a. formadores) se han propuesto en los últimos años. Estos X-formers mejoran el Transformer básico (conocido como vanilla Transformer) desde diferentes perspectivas: (i) la eficiencia del modelo, (ii) la generalización del modelo, (iii) la adaptación del modelo.

Estas tres mejoras en el Transformer básico han sido un avance importante en los últimos años, ahora el nuevo horizonte este trazado bajo la premisa de poder lograr una integración de datos multimodales. Esta integración será útil para mejorar el desempeño en la resolución de tareas complejas, y poder capturar las relaciones semánticas entre diferentes modalidades[4], [10].

Enfocándo la atención en BERT (Bidirectional Encoder Representations from Transformers), es un modelo de lenguaje pre-entrenado desarrollado por Devlin para mejorar la calidad y eficiencia de las soluciones de Procesamiento del Lenguaje Natural (NLP). BERT consta de 12 capas de codificadores de transformers, cada una con un tamaño oculto de 768, y el valor de h en la capa de atención auto-dirigida de multi-cabezas de 12. Esta arquitectura permite a BERT evaluar la importancia de una palabra en un documento basándose en su contexto [11].

El enfoque basado en características con BERT extrae características fijas del modelo BERT pre-entrenado. También conocido como "embedding" de palabras contextualizadas, en este enfoque

cada palabra se mapea a un espacio vectorial donde las palabras con significados similares están relativamente cercanas en ese espacio. Este enfoque tiene dos ventajas en comparación con el ajuste fino directo del modelo BERT: (i) permite añadir una arquitectura específica para un problema dado, ya que no todos los problemas de NLP se pueden resolver con una arquitectura de codificador de transformer, y aumenta la eficiencia computacional, ya que la costosa pre-computación de la representación se realiza solo una vez y se puede utilizar en varios experimentos. Además, al utilizar un modelo BERT pre-entrenado, es escalable para su uso en grandes conjuntos de datos[11], [12].

El proceso de obtención de representación textual mediante el enfoque basado en características de BERT implica alimentar un texto de entrada a BERT. Este texto se tokeniza utilizando el Modelo WordPiece antes de ser introducido en BERT. Para un documento que contiene n tokens, la representación textual obtenida es de n vectores numéricos con una dimensión de 768. El vector de salida de todas las palabras en el documento se puede organizar en una matriz de tamaño 768×768.

Por otro lado, según Beltagy (2022), se tiene el modelo Transformer que ha logrado resultados sobresalientes en una amplia gama de tareas de procesamiento de lenguaje natural (NLP), gracias a su componente de auto-atención que le permite capturar información contextual de secuencias completas. Sin embargo, su operación de auto-atención tiene una limitación importante: su requerimiento computacional y de memoria crece de manera cuadrática con la longitud de la secuencia, lo que lo hace ineficiente o muy costoso para procesar secuencias largas. Para superar esta limitación, se presenta el Longformer, una arquitectura Transformer modificada cuya operación de auto-atención escala linealmente con la longitud de la secuencia. Esto lo hace adecuado para procesar documentos largos. El mecanismo de atención del Longformer combina una atención local en ventanas con una atención global motivada por la tarea, y puede reemplazar la auto-atención estándar en los Transformers pre-entrenados, mejorando el rendimiento en una variedad de tareas de NLP a nivel de documento.

A nivel de aplicación en el sector científico, el trabajo desarrollado se demuestra la aplicabilidad y efectividad de modelos basados en BERT para la normalización de entidades biomédicas y clínicas. La normalización de entidades (EN) es crucial en el procesamiento del lenguaje natural (NLP) y se refiere a mapear menciones de entidades nombradas a términos en un vocabulario controlado. Los resultados mostraron la eficacia de los modelos basados en BERT para la normalización de entidades biomédicas y clínicas. El estudio concluye que estos modelos pueden ser ajustados finamente para normalizar cualquier tipo de entidades nombradas, alcanzando un buen rendimiento. En este mismo estudio [10], los autores describen cómo generaron un modelo de aprendizaje de representación clínica utilizando BERT y notas de registros de salud electrónicos (EHR). Usando la implementación de BERT de PyTorch desarrollada por Hugging Face, recopilaron aproximadamente 1.5 millones de notas EHR del UMass Memorial Medical

Center con la aprobación de las Juntas de Revisión Institucional de la Universidad de Massachusetts Medical School.

2. Metodología

La metodología implementada para el desarrollo del proyecto consta de las siguientes etapas: i) descripción del conjunto de datos, ii preprocesamiento de texto, iii construcción de la matriz de representación de documentos, iv clustering de documentos basado en K-means y LDA, v) métricas de evaluación, y, vi) visualización y comparación de resultados. En la Figura 1, se muestra el flujo de trabajo y la arquitectura utilizada para la clusterización de documentos

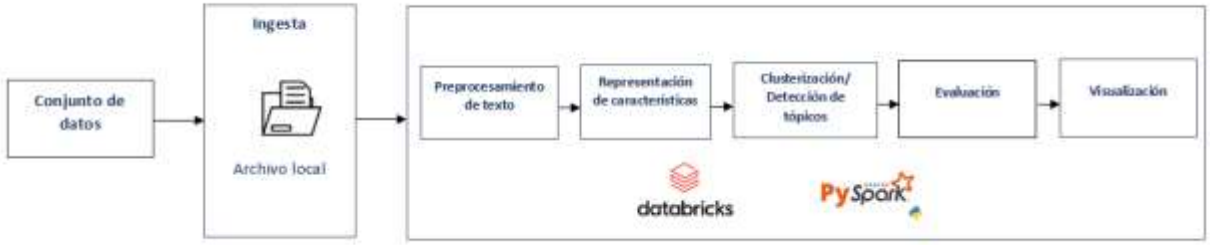

Figura 1. Arquitectura clustering de documentos

Tabla 1. Dataset utilizado métodos para representación de características clusterización de documentos y detección de tópicos; métricas de evaluación y técnicas usadas para visualización de resultados.

Conjunto de datos
Conjunto de datos ODIR19, 1.056.660 abstract Muestra de 50.000 abstract para análisis
Métodos
Preprocesamiento de texto **MC** Normalización de texto / eliminación de Stopwords **LDA** Normalización de texto / eliminación de Stopwords/ Lematización / Tokenización
Representación de características RC 1 Transformers SBERT RC 2 TF - IDF
Método de clusterización **MC** K-means
Modelo de Tópicos **LDA** Latent Dirichlet Allocation topic model
Evaluación **MC** Inercia / Silueta Score **LDA** Log Likelihood / Log Perplexity
Visualización **MC - LDA** T-SNE / gráfico de barras / nubes de palabras

4.1 Conjunto de datos

En respuesta a los retos generados por la pandemia de COVID-19 en materia de investigación científica, la Casa Blanca y una coalición de grupos de investigación lideres pusieron a disposición de la comunidad científica el Conjunto de datos de investigación abierta de COVID-19 (CORD-19) como un recurso gratuito disponible para que se apliquen avances recientes en el procesamiento del lenguaje natural y otras técnicas de inteligencia artificial. CORD-19[1] es un recurso de más de 1.000.000 de artículos académicos, incluidos más de 400.000 con texto completo, sobre COVID-19, SARS-CoV-2 y coronavirus relacionados.

El conjunto de datos CORD-19 tiene 19 variables donde cada registro corresponde a un artículo científico sobre COVID-19. Las variables disponibles para análisis están relacionadas con el título, año de publicación, autores, revista, código DOI, archivo pdf en formaro json abastract, entre otras. Para el desarrollo del ejercicio se seleccionó la variable que contenía los abstract de cada documento; asimismo, se tomó una muestra de 50.000 documentos teniendo en cuenta los recursos de cómputo disponibles.

4.2 Preprocesamiento de texto

El preprocesamiento de texto y análisis fue realizado en Databricks usando un cómputo de 14GB con 4 núcleos. El lenguaje de programación utilizado fue Python, utilizando la biblioteca PySpark el cual proporciona una interfaz con Apache Spark para aprovechar los beneficios del procesamiento distribuido y escalabilidad horizontal proporcionado por Spark.

En la etapa de preprocesamiento de texto inicialmente se identificó por medio de la librería *langdetect* la distribución de idiomas en el conjunto de datos, posteriormente se eliminaron los documentos escritos en un idioma diferente al inglés. De los 50.000 documentos tomados para el análisis un total de 49.127 estaban escritos en idioma inglés y 873 estaba en otra variedad de idiomas como el español, francés, entre otros. Posteriormente, se normalizó el texto mediante la eliminación de caracteres especiales y conversión de texto a minúsculas para el método de detección de tópicos se implementó un proceso de lematización adicional a lo anteriormente descrito. Finalmente, se eliminaron "stopwords[2]" mediante el uso de la librería *nltk* en los dos métodos.

4.3 Representación de características

La representación de características se basa en la transformación de documentos y/o texto en un formato numérico vectorial representativo que pueda ser utilizada como entrada por algoritmos de aprendizaje automático. Para la representación de los documentos del conjunto de datos se seleccionaron dos técnicas algoritmo Transformers SBERT para la representación de

[1] Tomados de: https://www.kaggle.com/code/acmiyaguchi/parquet-and-bigquery-dataset-for-cord-19/output
[2] son palabras que se filtran o eliminan durante el procesamiento de texto debido a que son comunes y no aportan un significado semántico importante al análisis.

características para K-means y TF-IDF para el LDA. A continuación, se realiza una breve descripción del algoritmo y su implementación en cada uno de los casos:

4.3.1 Transformers SBERT: los Tansformers son modelos preentrenados en grandes conjuntos de datos que se ajustan para tareas específicas o dominios utilizando conjuntos de datos más pequeños. Según [13]un Transformer, se basa en una secuencia de capas transformadoras que producen una representación contextualizada de una secuencia de entrada de tokens pertenecientes a un texto. Los Transformers generalmente se componen por una sucesión de autoatención multi-cabezal, un primer normalizador, una red neuronal feed-forward y un segundo normalizador.

SBERT es un tipo de Transformer categorizado como Sentences Transformer, el cual es un framework diseñado para generar representaciones de características de última generación para oraciones, texto e imágenes. La representación de oraciones generada por SBERT se utiliza comúnmente para comparar la similitud semántica entre oraciones. Esto es útil en tareas como la búsqueda semántica, la recuperación de información, la agrupación de documentos y la clasificación de similitud. [14].

Para la realización de la representación de características con SBERT se importa el modelo preentrenado de sentences transformes *"all-MiniLM-L6-v2"* de la biblioteca Hugging Face Transformers. Posteriormente, se utiliza un tokenizador disponible en el modelo preentrenado para convertir el texto de entrada en tokens y luego calcular los embeddings de cada documento. Es importante resaltar que dado que SBERT es una técnica basada en modelos de representación de texto que utiliza embeddings para capturar la semántica de oraciones completas fue necesario dividir cada abstract en oraciones antes de ingresar la entrada de texto al Transformers. Para ajustar el modelo preentrenado a el conjunto de datos se obtuvo la última capa oculta del modelo que contiene representaciones detalladas del token en el texto; posteriormente, se calculan los embeddings promedio ponderados por la máscara de atención para manejar las variaciones en la longitud de los textos. Finalmente, se normalizan los embeddings resultantes para asegurar que tengan una longitud constante, lo que facilita la compresión y análisis.

4.3.2 TF-IDF: es un método de ponderación que se emplea para transformar un documento a un formato estructurado. Se trata de un valor numérico que indica la relevancia de una palabra para un documento dentro de una colección o corpus. Comúnmente, se utiliza como factor de ponderación en procesos como la recuperación de información y la minería de textos [15].

En la implementación del LDA antes de realizar el proceso de representación de características se convirtió el conjunto de datos preprocesado a tokens por medio de la librería *Tokenizer* de PySpark A continuación, se utiliza *ContVectorizer* para convertir el texto en una representación de bolsa de palabras, y luego se aplica TF-IDF para ponderar estas características según la importancia en el conjunto de datos. Se realizo el proceso de optimización de hiperparametros para *ContVectorizer* y se obtuvo un valor óptimo para de tamaño máximo de vocabulario

(VocabSize o palabras únicas que se incluirán en la representación de las bolsas de palabras igual a 1000; y de umbral mínimo de frecuencia del documento minDF para incluir una palabra en el vocabulario de 5, es decir, la cantidad mínima de documentos en los que una palabra debe aparecer para ser considerada en el vocabulario es 5.

4.4 Métodos de agrupación

En este proyecto se realizó la comparación de dos métodos usados para la clusterización de documentos, K-means, el cual, es un método tradicional usado en procesos de agrupación y el Latent Dirichlet Allocation (LDA) que es un modelo probabilístico generativo que se utiliza para descubrir temas latentes dentro de un conjunto de documentos. A continuación, se da una breve descripción de cada uno de los métodos implementados:

4.4.1 K-means: en el contexto del clustering de documentos, K-means es un algoritmo de aprendizaje no supervisado que se utiliza para agrupar documentos similares en clústeres. La idea principal detrás de K-means es asignar cada documento a uno de los K clústeres predefinidos, de manera que los documentos dentro de un mismo clúster sean más similares entre sí que con los documentos en otros clústeres [16].

4.4.1 Latent Dirichlet Allocation(LDA): LDA es una aplicación extendida del modelo Bayesiano Jerárquico [17]. LDA es un modelo generativo que asume que cada documento en un conjunto de documentos se genera mediante una mezcla de tópicos. Cada tópico es una distribución de palabras y cada documento es una mezcla de estos tópicos. La idea principal del LDA es tratar un documento como una colección de palabras, cada documento como una combinación de múltiples temas y cada tema como compuesto por varias palabras [18]

4.5 Evaluación

Antes de realizar la implementación de los modelos se realizó un proceso de optimización para hallar el número de cluster y tópicos óptimos Para el K-means[3] se valúo la inercial o método del codo, la cual, mide cuánto varían los puntos de datos dentro de un mismo clúster y se define como la suma de las distancias cuadráticas de cada punto de datos al centroide de su clúster asignado. En bases a los resultados de la inercia se selecciona un K=10, en la
se muestran los resultados obtenidos

Por otra parte, en el caso del LDA se tomó como métrica de evaluación el Log Perplexity y el Log-Likelihood. La perplexity es una medida de qué tan bien un modelo de lenguaje puede predecir un conjunto de datos, por lo tanto, un valor más bajo de log perplexity indica que el modelo está haciendo mejores predicciones y tiene una mejor capacidad para explicar los datos observados. Por otra parte, el el log-likelihood es una medida de la probabilidad de que los datos observados

[3] No se evalúo con el método de la silueta por eficiencia computacional dado que no encontraba incluido en las funciones disponibles por PySpark.

sean generados por el modelo y un Un log-likelihood más alto sugiere que el modelo es más consistente con los datos observados. El resultado obtenido para el LDA fue K=20, sin embargo, este dato es extremo es un dato y muestra que el conjunto de datos tiene una alta complejidad y el modelo está detectando una gran variedad de patrones heterogéneos, por esto se tomó la decisión de igualar el número de tópicos a 10 para que los resultados fueran comparables con el K-means El valor obtenido de perplexity para K=10 fue de 6.25, lo que indica que el modelo es bastante bueno para predecir las palabras en el conjunto de datos; y, el valor de log-likelihood es de -60293276.18, este resultado sugiere que el modelo es bastante probable dada la evidencia observada.

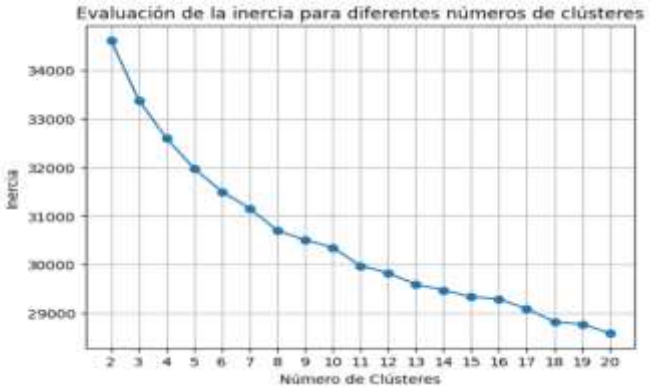

Figura 1. Evaluación de la inercia (método del codo)

4.6 Uso de herramientas Big data

Las técnicas y métodos usados para la implementación del proyecto permiten una escalabilidad futura a más volumen o velocidad, esto se explica por razones como que el código está desarrollado utilizando Apache Spark en Databricks por medio de PySpark, lo que proporciona un entorno distribuido escalable para el procesamiento de grandes conjuntos de datos que permite aprovechar la capacidad de cómputo distribuido para manejar grandes volúmenes de datos de manera eficiente. Por otra parte, en la búsqueda de optimizar el rendimiento de los recursos computacionales se ajusta el número de particiones del conjunto de datos en concordancia con el número de núcleos disponibles. Distribuir los datos de manera uniforme entre las particiones permite una mejor utilización de los recursos del clúster, evitando desequilibrios que podrían ralentizar ciertas operaciones.

Asimismo, en los casos donde se esperaba que un conjunto de datos fuera utilizado varias veces en operaciones subsiguientes se configuró el almacenamiento en caché lo que reduce la necesidad de recalcular los mismos resultados en cada iteración. Cuando ya es necesario tener los datos en caché se libera, lo que ayuda a gestionar el uso de memoria. Finalmente, se emplean funciones definidas por el usuario para aplicar operaciones a nivel de fila de manera distribuida en el clúster

Spark; esto es fundamental para mantener el paralelismo y la eficiencia en el procesamiento de grandes volúmenes de datos.

5. Resultados

En la *Figura* , se muestran los resultados obtenidos para cada uno de los modelos implementados. Para el K-means se observa que los cluster se encuentran superpuestos lo que sugiere que los límites entre los grupos no son claros y definidos; lo que indica que hay regiones en el espacio de características donde los puntos de clústeres comparten características similares. De igual manera, en el caso del LDA se observa superposición de tópicos, mostrando que los conceptos o temas representados por los tópicos comparten características.

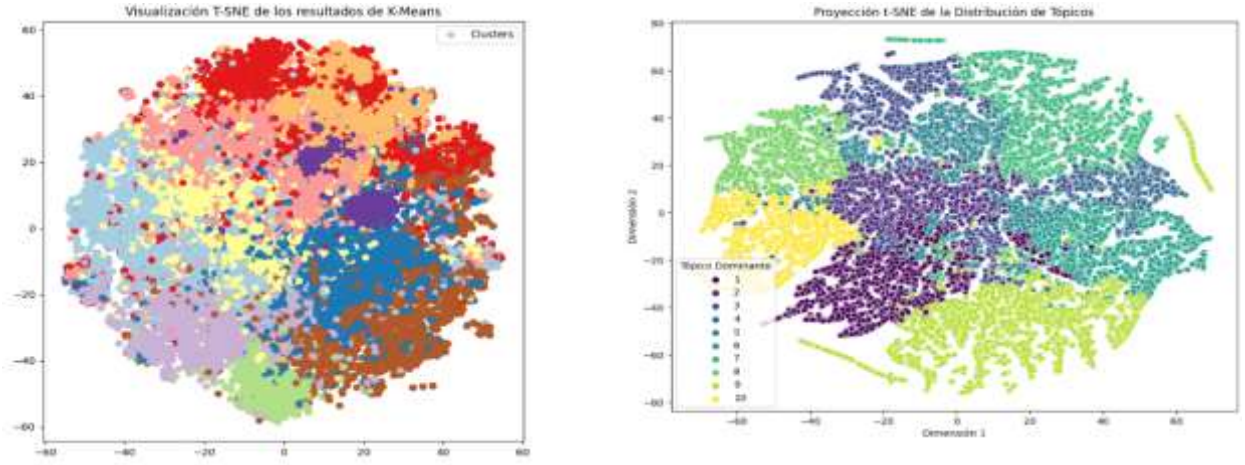

Figura 2. Visualización T-SNE

El número de documentos por clúster para el caso del K-means es homogéneo, lo que muestra cierta estabilidad en el resultado del agrupamiento. El número máximo de documentos asignados a un clúster es de 5600 (11.4%) y el mínimo es de 3469; por otra parte, en el caso del LDA el número máximo de documentos asignados a un clúster es de 7280 (14.82%) y el mínimo es de 2894. En la *Figura* se muestran los resultados obtenidos para la distribución de documentos.

Por otra parte, En la Figura 5 se encuentran las nubes de palabras generadas para los tópicos del LDA. A continuación, se realiza una breve descripción de los posibles temas relacionados en cada tópico:

- **Tópico 0**: tratamiento para enfermedades como la diabetes.;
- **Tópico 1:** cuidado de la salud en hospitales para pacientes y trabajadores de salud;
- **Tópico 2:** sistema, red y equipos de ventilación para el manejo del COVID-19r;
- **Tópico 3:** uso de mascarillas en entornos de la manipulación;
- **Tópico 4:** impacto del COVID-19 en niños con cáncer o en pacientes en cirugía;
- **Tópico 5:** manejo en un hospital con un paciente positivo para COVID-19;

- **Tópico 6:** leyes para manejo del COVID-19 y sus efectos en la economía global;
- **Tópico 7:** salud mental, estrés, ansiedad, psicología;
- **Tópico 8:** vacunación; comportamiento del virus en las células y genética.
- **Tópico 9:** Programas educativos y entrenamiento para aprender sobre el manejo del
- COVID-19.

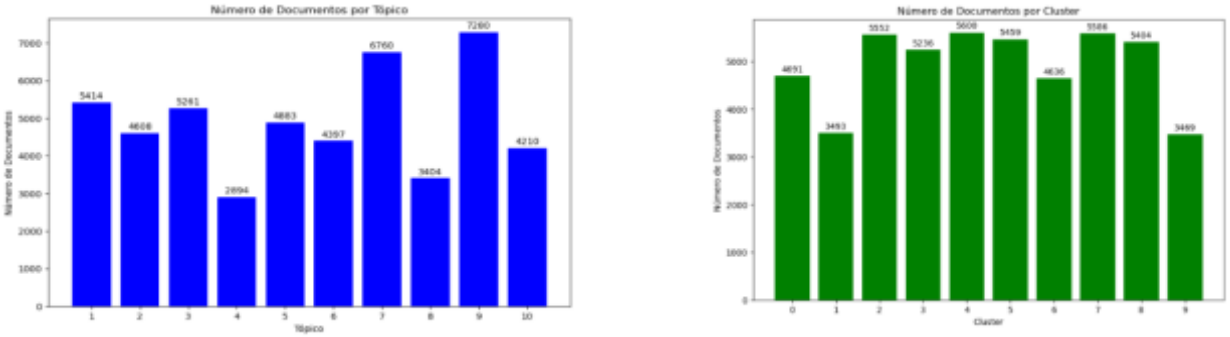

Figura 3. Distribución de documentos

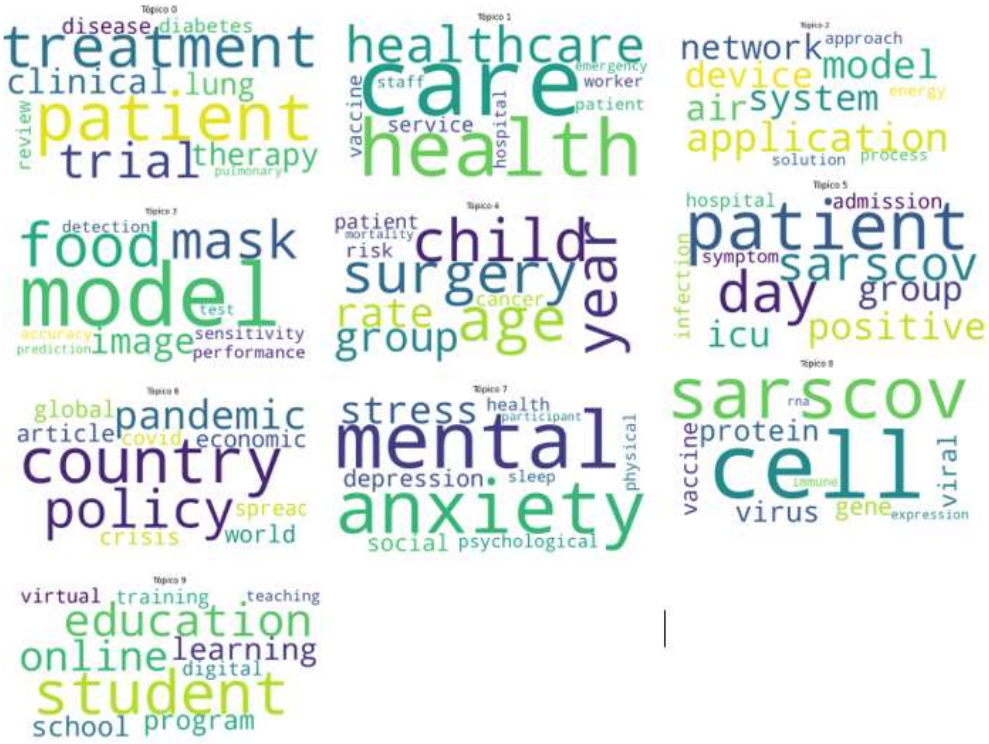

Figura 4. Nubes de palabras para tópicos LDA

Finalmente, en la Figura se muestran las diez palabras más frecuentes en cada uno de los cluster generados por el K-means, se observa que en términos generales las palabras generadas en todos los cluster son las mismas y no permite hacer una distinción clara de los temas que pueden estar implícitos en cada grupo de documento.

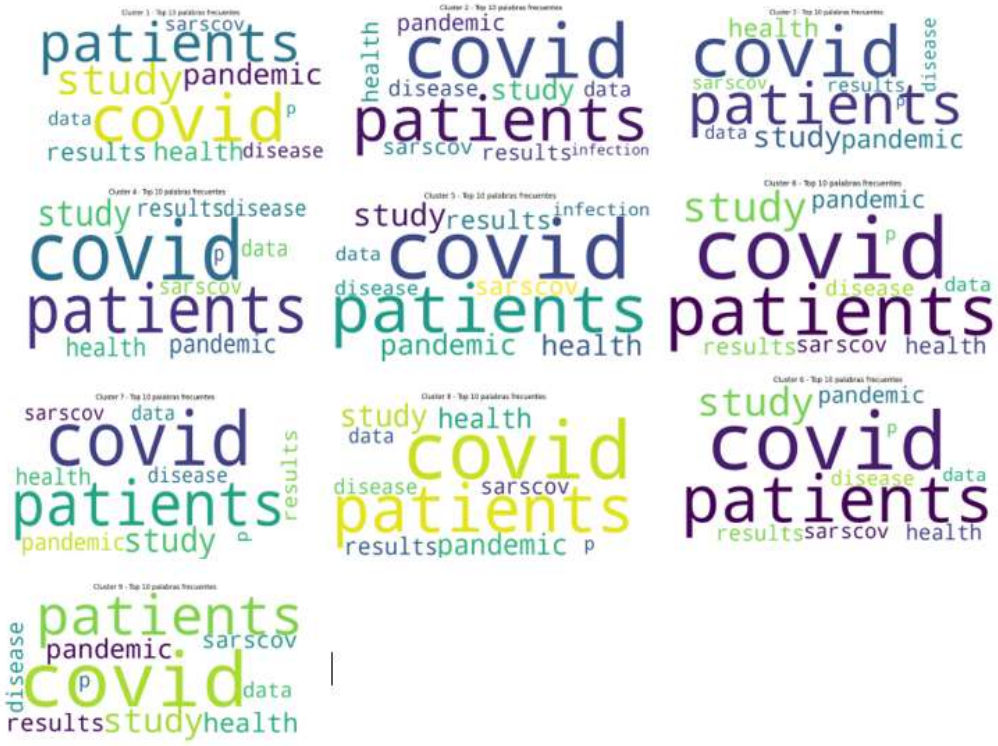

Figura 5. Nubes de palabras para K-means

6.Conclusiones y trabajo futuro

Se observó una diferencia importante en el número de tópicos óptimos encontrados por el método de clusterización en comparación con el LDA. Esto se puede deber a que el LDA puede verse afectado por la diversidad y complejidad de los documentos lo que explica el valor mayor de tópicos en este caso. Tanto los tópicos como los clusters se encuentran superpuestos y presentan poca separación entre clústers lo que puede indicar que los temas se encuentran altamente relacionados o que el algoritmo no logró realizar una buena diferenciación de los grupos en los

datos. El método de LDA es parece tener un mejor desempeño en cuanto identificación de temáticas internas en los documentos, el cluster realizado por medio de K-means es más difícil de interpretar dado que no permite identificar posibles temáticas en los grupos de documentos. Se corrobora con los resultados del K-means que los documentos se encuentran altamente correlacionados y por lo tanto para realizar una separación significativa de los mismo se requieren emplear otras estrategias que capturen adecuadamente la complejidad de los datos.

Los resultados obtenidos son una aproximación para identificar las temáticas relacionadas con el COVID19 en la literatura y permite agrupar documentos por temáticas de análisis para su recuperación y análisis. Para un trabajo futuro se plantea una revisión de otro grupo de algoritmos con el objetivo de comparar los resultados obtenidos por otros métodos de clusterización en conjunto con otros métodos para representación de características. Adicionalmente, se requiere la implementación de estrategias más avanzadas que permitan identificar las características y/o temas de los diferentes grupos generados por el K-means como el análisis de expertos.

7. Ejecución del plan

Para la ejecución de plan se ajustaron relativamente bien los tiempos para los procesos de comprensión del negocio hasta preparación de los datos. En la fase de modelado y resultados se tuvieron algunos inconvenientes, dado que el código se ejecutó con inicialmente con una muestra de los datos y no se tuvo en cuenta el tiempo real que podría tomar generar resultados para el conjunto de datos completo, lo que retraso el proceso. Se toma como lección aprendida en el trabajo con Big Data tomar el tiempo de ejecución como una variable relevante a tener en cuenta en el momento de plantear el cronograma de trabajo.

Actividades	Inicio	Final	Septiembre				Octubre				Noviembre
			del 01 al 08	del 9 al 17	del 18 al 25	del 24 al 30	del 01 al 10	del 11 al 18	del 19 al 24	del 24 al 27	del 01 al 07
1. Comprensión del negocio	1/09/2023	8/09/2023									
2. Comprensión de los datos	9/09/2023	17/09/2023									
3. Preparación de los datos	18/09/2023	25/09/2023									
4. Modelado	26/09/2023	10/10/2023									
5. Evaluación	11/10/2023	18/10/2023									
6. Resultados	19/10/2023	24/05/2023									
7. Desarrollo documento final	24/10/2023	27/10/2023									

Tabla 1. Cronograma de actividades inicial

Actividades	Inicio	Final	Septiembre				Octubre				Noviembre	
			del 01 al 08	del 9 al 17	del 18 al 25	del 24 al 30	del 01 al 10	del 11 al 18	del 19 al 24	del 24 al 27	del 01 al 07	del 07 al 14
1. Comprensión del negocio	1/09/2023	8/09/2023										
2. Comprensión de los datos	9/09/2023	17/09/2023										
3. Preparación de los datos	18/09/2023	25/09/2023										
4. Modelado	1/10/2023	30/10/2023										
5. Evaluación	1/10/2023	7/10/2023										
6. Resultados	1/11/2023	14/11/2023										
7. Desarrollo documento final	24/10/2023	15/10/2023										

Tabla 1. Cronograma de actividades final

5. Implicaciones éticas

El conjunto de datos de investigación abierta de COVID-19 (CORD-19) es de carácter público y no contiene información sensible sobre pacientes u otra información que pueda comprometer la confidencialidad o anonimidad de los datos. En paralelo, el avance del proyecto de investigación podría proporcionar a la comunidad médica y a las autoridades de salud una vía más accesible para explorar la literatura relacionada con COVID-19, permitiendo también la identificación de áreas cruciales de investigación. No obstante, es importante reconocer la posibilidad de que los resultados estén influenciados por sesgos involuntarios, lo que podría generar conclusiones incorrectas.

6. Aspectos legales y comerciales

El proyecto de investigación propuesto es de carácter académico y sin fines de lucro. Las conclusiones obtenidas pueden ser usadas por la comunidad médica para el entendimiento de cuáles son las temáticas y áreas de investigación sobre la que se ha desarrollado producción científica sobre COVID-19. En cuanto a los aspectos legales, los datos usados para la investigación cumplen con las leyes de privacidad y protección de datos personales. Del mismo modo, los artículos analizados están exentos de restricciones legales en su utilización. Los resultados alcanzados se consideran propiedad de los autores del proyecto, sin generar ningún tipo de conflicto legal con los investigadores que originalmente contribuyeron a los artículos académicos sometidos al análisis.

Referencias

[1] A. W. Olthof *et al.*, "Machine learning based natural language processing of radiology reports in orthopaedic trauma," *Comput Methods Programs Biomed*, vol. 208, p. 106304, 2021, doi: 10.1016/j.cmpb.2021.106304.

[2] S. Casola, I. Lauriola, and A. Lavelli, "Pre-trained transformers: an empirical comparison," *Machine Learning with Applications*, vol. 9, no. May, p. 100334, 2022, doi: 10.1016/j.mlwa.2022.100334.

[3] R. Bai, R. Huang, Y. Chen, and Y. Qin, "Deep multi-view document clustering with enhanced semantic embedding," *Inf Sci (N Y)*, vol. 564, pp. 273–287, Jul. 2021, doi: 10.1016/j.ins.2021.02.027.

[4] A. Veltman, D. W. J. Pulle, and R. W. De Doncker, "The Transformer," *Power Systems*, no. Nips, pp. 47–82, 2016, doi: 10.1007/978-3-319-29409-4_3.

[5] M. Allahyari *et al.*, "A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques," Jul. 2017, [Online]. Available: http://arxiv.org/abs/1707.02919

[6] A. Onan, "Two-Stage Topic Extraction Model for Bibliometric Data Analysis Based on Word Embeddings and Clustering," *IEEE Access*, vol. 7, pp. 145614–145633, 2019, doi: 10.1109/ACCESS.2019.2945911.

[7] J. Singh, M. Wazid, D. P. Singh, and S. Pundir, "An embedded LSTM based scheme for depression detection and analysis," *Procedia Comput Sci*, vol. 215, pp. 166–175, 2022, doi: 10.1016/j.procs.2022.12.019.

[8] C. C. Aggarwal and C. X. Zhai, "A survey of text clustering algorithms," in *Mining Text Data*, vol. 9781461432234, Springer US, 2012, pp. 77–128. doi: 10.1007/978-1-4614-3223-4_4.

[9] Q. Grail, J. Perez, and E. Gaussier, "Globalizing BERT-based Transformer Architectures for Long Document Summarization," 1810.

[10] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, vol. 3, no. October, pp. 111–132, 2022, doi: 10.1016/j.aiopen.2022.10.001.

[11] U. H. Govindarajan, D. K. Singh, and H. A. Gohel, "Forecasting cyber security threats landscape and associated technical trends in telehealth using Bidirectional Encoder Representations from Transformers (BERT)," *Comput Secur*, vol. 133, no. January, p. 103404, 2023, doi: 10.1016/j.cose.2023.103404.

[12] B. Ay, F. Ertam, G. Fidan, and G. Aydin, "Turkish abstractive text document summarization using text to text transfer transformer," *Alexandria Engineering Journal*, vol. 68, pp. 1–13, 2023, doi: 10.1016/j.aej.2023.01.008.

[13] Q. Grail, J. Perez, and E. Gaussier, "Globalizing BERT-based Transformer Architectures for Long Document Summarization," 1810.

[14] Nils Reimers, "SentenceTransformers Documentation," https://www.sbert.net/.

[15] Bafna Prafulla, Dhanya Pramod, and Vaidya Anagha, "Document Clustering: TF-IDF approach," 2016.

[16] V. K. Singh, N. Tiwari, and S. Garg, "Document clustering using K-means, heuristic K-means and fuzzy C-means," in *Proceedings - 2011 International Conference on Computational Intelligence and Communication Systems, CICN 2011*, 2011, pp. 297–301. doi: 10.1109/CICN.2011.62.

[17] I.-C. Chang *et al.*, "Applying Text Mining, Clustering Analysis, and Latent Dirichlet Allocation Techniques for Topic Classification of Environmental Education Journals," *Sustainability 2021, Vol. 13, Page 10856*, vol. 13, no. 19, p. 10856, Sep. 2021, doi: 10.3390/SU131910856.

[18] J. R. Millar, G. L. Peterson, and M. J. Mendenhall, "Document Clustering and Visualization with Latent Dirichlet Allocation and Self-Organizing Maps." [Online]. Available: www.aaai.org

[5] Janani, R., & Vijayarani, S. (2019). Text document clustering using Spectral Clustering algorithm with Particle Swarm Optimization. *Expert Systems with Applications, Volume 134.*, 192-200.

[6] Tong, Z., & Zhang, H. (2016, May). A text mining research based on LDA topic modelling. In International conference on computer science, engineering and information technology (pp. 201-210).