UCUENCA

Facultad de Ingeniería

Maestría en Gestión Estratégica de Tecnologías de la Información II Cohorte

Clasificación de artículos académicos sobre la pandemia de la COVID-19, a través de técnicas de minería de texto

Trabajo de titulación previo a la obtención del título de Magíster en Gestión Estratégica de Tecnologías de la Información

Autor:

Bayron Fernando, Vásquez Vanegas

CI: 0104742424

Correo electrónico:

bayron.vasquezv@hotmail.com

Director:

Marcos Patricio, Orellana Cordero

CI: 0102668209

Cuenca, Ecuador

6 de enero de 2023

Resumen:

Debido a la aparición del virus SARS-CoV-2, y a la enfermedad del COVID-19 que provoca este virus, la comunidad científica así como los distintos actores y organizaciones, han visto la necesidad de obtener información que pueda aportar conocimiento sobre cómo evoluciona esta enfermedad y enfrentar los distintos problemas que la misma ha traído a la población mundial.

El estudio propone realizar la clasificación de artículos científicos mediante la aplicación de técnicas de *Machine Learning*, a través de mecanismos de representación semántica de palabras como es *Word Embeddings* y tecnologías basadas en redes neuronales, analizando los *abstracts* de artículos científicos disponibles en las fuentes de información como lo es LitCovid. El desarrollo del presente estudio está basado en la aplicación de la metodología CRISP-DM (*CRoss-Industry Standard Process for Data Mining*) (Wirth, 2000), la cual describe un modelo de procesos jerárquico que consta de seis fases que describen de manera natural el ciclo de vida de un proyecto de minería de datos, y debido a que tanto la minería de datos como la de texto buscan obtener conocimiento sea de grandes volúmenes de datos y de grandes volúmenes de documentos de texto respectivamente, se adopta como base para el desarrollo del presente estudio esta metodología.

Para lograr los objetivos propuestos se emplea la metodología adoptada y se evalúan los resultados de desempeño de aplicar dicha metodología y modelos propuestos.

Los resultados obtenidos demuestran que al aplicar la metodología propuesta se obtuvieron resultados aceptables para la clasificación, dando como resultado, que, al emplear FastText como modelo de representación semántica, se consiguieron métricas de

exactitud del 74%, en comparación con los modelos Word2Vec y Glove que alcanzaron el 72% y 65% respectivamente, siendo esta técnica una de las mejores opciones al momento de emplear modelos de representación semántica del texto.

Palabras clave: Word embedding. Machine learning. Procesamiento de lenguaje natural. Deep learning. Clasificación de texto. Redes neuronales. Abstracts. Vectores de palabras. Minería de texto. Minería de datos.

Abstract:

Due to the appearance of the SARS-CoV-2 virus, and the COVID-19 disease caused by this virus, the scientific community, as well as the different actors and organizations, have seen the need to obtain information that can provide knowledge about its evolution and how to deal with different problems that it has brought to the world population.

The present study proposes to carry out the classification of articles through the application of Machine Learning techniques, by mechanisms of semantic representation of words such as Word Embeddings and neural networks technologies, analyzing papers abstracts available in sources such as LitCovid. The development of this study is based on the application of the CRISP-DM methodology (CRoss-Industry Standard Process for Data Mining) (Wirth, 2000), which describes a hierarchical process model consisting of six phases that describe naturally the life cycle of a data mining project, since both data and text mining seek to obtain knowledge from large volumes of data and large volumes of text documents, respectively, is adopted as the basis for the development of this study this methodology.

To achieve the proposed objectives, the adopted methodology is used and the performance results of applying said methodology and proposed models are evaluated.

The results obtained have shown that when applying the proposed methodology, acceptable results were obtained for the classification, resulting in the fact that, when using FastText as a semantic representation model, accuracy metrics of 74% were achieved, compared to the Word2Vec and Glove models that reached 72% and 65% respectively, this technique being one of the best options when using the semantic representation of the text.

Keywords: Word embedding. Machine learning. Natural language processing. Deep learning. Text classification. Neuronal networks. Abstracts. Word vectors. Text mining. Data mining.

Índice

UCUENCA

Cláusula de Propiedad Intelectual

Bayron Fernando Vásquez Vanegas, autor del trabajo de titulación "Clasificación de artículos académicos sobre la pandemia de la COVID-19, a través de técnicas de minería de texto.", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor/a.

Cuenca, 6 de enero de 2023

Bayron Fernando Vásquez Vanegas

C.I: 0104742424

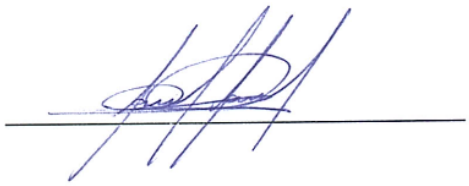
UCUENCA

Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Bayron Fernando Vásquez Vanegas en calidad de autor/a y titular de los derechos morales y patrimoniales del trabajo de titulación "Clasificación de artículos académicos sobre la pandemia de la COVID-19, a través de técnicas de minería de texto", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 6 de enero de 2023



Bayron Fernando Vásquez Vanegas

C.I: 0104742424

1. Introducción.

Con el surgimiento de la enfermedad del COVID-19, provocado por el nuevo coronavirus SARS-CoV-2 que inició en la ciudad de Wuhan en China a finales del 2019(Maguiña Vargas et al., 2020), ha llevado al mundo a una de las mayores crisis de la historia, tanto en aspectos de salud, económicos, sociales, afectando así múltiples aspectos de la vida cotidiana. La comunidad científica ha puesto su mayor esfuerzo para hacer frente a esta pandemia y así estudiar y entender el comportamiento de este nuevo virus y sus efectos en la salud, para desarrollar tratamientos, elaborar medidas de prevención, desarrollar vacunas, implementar políticas públicas para la gestión y control de la pandemia.

Como resultado de este esfuerzo, la producción de conocimiento científico acerca del COVID-19 y el nuevo coronavirus, ha crecido a un ritmo sin precedentes, según estudios realizados (Wang & Lo, 2021), hasta septiembre del 2020 se habían generado entre 55,000 a 100,000 artículos científicos acerca de esta enfermedad. Esto ha provocado una gran presión sobre médicos, investigadores y otras entidades que necesitan estar al día con información relacionada con esta nueva literatura. Debido a la gran cantidad de artículos publicados, extraer información de interés particular puede llevar mucho esfuerzo, ya que todos estos artículos de investigación pertenecen a distintos dominios, ya sean en el área de la salud, medicina, prevención, tratamientos, entre otros.

El interés de los investigadores es extraer información de los artículos basándose en un criterio o un objetivo determinado según el área de interés, por lo que herramientas que permitan realizar una clasificación automática de esta información son cada vez más importantes y requeridas por parte de la comunidad científica. Por tal motivo, la aplicación

de enfoques de Procesamiento de Lenguaje Natural[1] (PLN), permitirían obtener información más relevante o específica, basándose en un análisis del cuerpo de texto sobre las investigaciones científicas en las distintas bases científicas, y así obtener información clasificada que pueda ser útil en la investigación o consulta de información.

Es fundamental hoy más que nunca tener una visión completa del estado del arte de la literatura relacionada con el COVID-19 por las siguientes razones:

- Organizar y categorizar la literatura.
- Explorar temas de investigación.
- Identificar prioridades y necesidades para generar oportunidades de investigación.
- Entender la evolución de la pandemia.
- Reconocer a los líderes de la investigación en esta área (Investigadores, institutos y países líderes).
- Explorar conexiones entre temas y áreas de investigación.

Los buscadores proporcionan información no relacionada, por lo que las búsquedas clasificadas pueden ser ineficientes y requerir mucho tiempo. De esta forma, el presente proyecto propone realizar una clasificación de artículos científicos acerca del COVID-19, mediante la aplicación de técnicas de PLN, específicamente *Word Embedding*. Se utiliza la metodología CRISP-DM (CRoss-Industry Standard Process for Data Mining) para la minería de datos, esta metodología está compuesta de seis fases para la aplicación de proyectos de minería de datos para la extracción de conocimiento, en donde cada fase contiene tareas

[1] PLN o en inglés Natural Language Processing (NLP), es una rama de la Inteligencia Artificial cuyo objetivo es crear modelos computacionales que faciliten la comprensión y comunicación hombre máquina por medio del lenguaje humano.

que están relacionadas según los objetivos del proyecto de minería de datos. Esta metodología se revisará más adelante más a profundidad.

1.1. Objetivos

Objetivo General

Clasificar artículos científicos con temáticas relacionadas con la COVID-19, mediante la aplicación del enfoque de *Word Embeddings*.

Objetivos Específicos

- Revisar la literatura de la temática de estudio.
- Desarrollar una metodología de clasificación del texto de los resúmenes de los artículos académicos.
- Aplicar técnicas de minería de texto para la clasificación del texto de los resúmenes de los artículos académicos utilizando el enfoque de *Word Embedding*.
- Analizar los resultados obtenidos al aplicar la metodología.
- Desarrollar un artículo científico.

1.2. Problemática

Desde el origen de los distintos tipos o clases de coronavirus, como el SARS (*Severe Acute Respiratory Syndrome*) en el 2003 (Thompson, 2003), y el MERS (*Middle East Respiratory Syndrome Coronavirus (MERS-CoV)*, 2022) en el 2012, se han publicado varios artículos de investigación sobre las enfermedades que estos virus han causado, sin embargo, no es hasta la aparición del SARS-CoV-2 en el 2019 causante de la pandemia del COVID-19, que el número de publicaciones o artículos científicos creció a un ritmo acelerado, provocando una avalancha de información sobre esta nueva literatura. Estos artículos pertenecen a distintos dominios en el área de la salud, medicina, prevención, tratamientos, etc. El interés de los investigadores es extraer los artículos basándose en un criterio o una clase de objetivo según el área de interés, debido a que es un tema nuevo en muchos aspectos y a la falta de conocimiento sobre cómo el SARS-CoV-2 y la pandemia del COVID-19 actúa y sus efectos en la salud así como en muchos aspectos de la vida cotidiana, es aquí, donde la investigación tiene un rol importante para hacer frente a esta pandemia. Por ello, herramientas que permitan realizar una clasificación automática de esta información son cada vez más importantes y requeridas por parte de la comunidad científica (Chandrasekaran & Fernandes, 2020).

La pandemia del COVID-19, ha provocado que la comunidad científica realice investigaciones día tras día a un ritmo sin precedentes. Se puede tomar como ejemplo los resultados obtenidos en ***Dimensions (https://www.dimensions.ai/)***, que es una base de datos multidisciplinar, la misma que integra publicaciones, datos, ensayos clínicos, etc. Indexa contenido de revistas científicas, bases de datos como PubMed y preprints procedentes de distintos repositorios en acceso abierto. En la cual, cómo se puede observar en la siguiente Figura 1, que las publicaciones realizadas acerca del COVID-19 en el año de aparición de dicha enfermedad (2019), llegaron alrededor de 569, para el 2020 este

número creció exponencialmente, llegando a contabilizarse 272,606 publicaciones y en el transcurso del 2021 hasta noviembre ya se han realizado 386,269 publicaciones que contengan el término de búsqueda "***COVID-19***". Hay que considerar que para el ejemplo únicamente se ha tomado como referencia el término "COVID-19", sin embargo, los resultados pueden variar significativamente si incluimos dentro de los criterios de búsqueda términos como: "***SARS CoV 2, coronavirus, 2019n-CoV, corona virus, COVID19***".

La Figura 1 a continuación, muestra el crecimiento de publicaciones de artículos científicos acerca del COVID-19 desde el 2019, en la cual se muestra que desde el 2019 al 2020 se han publicado alrededor de 270,000 artículos y hasta 2021 ya alcanzaron alrededor de las 380,000 publicaciones, tal como se indicó anteriormente.

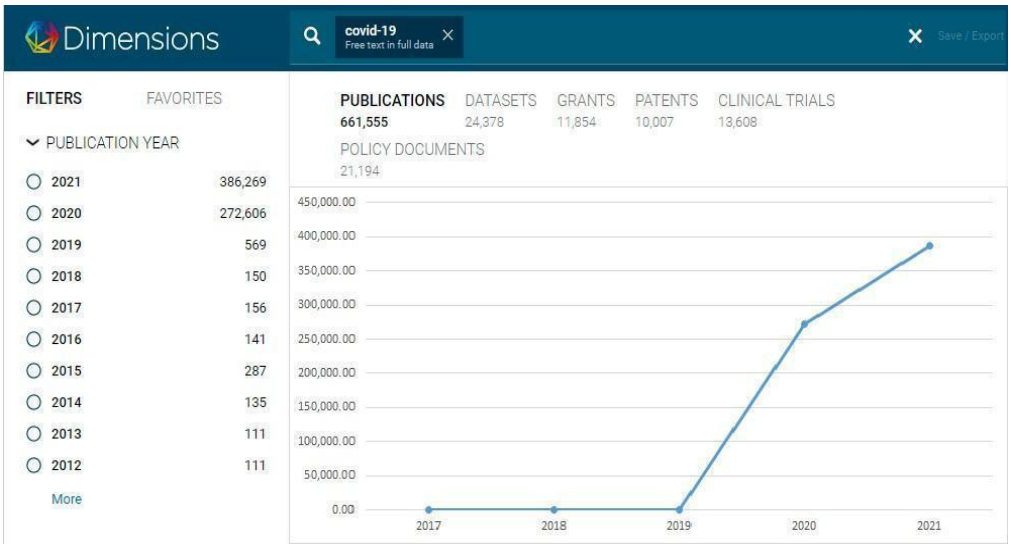


Figura 1: Curva de crecimiento de número de artículos publicados acerca del COVID-19, 2021 (https://www.dimensions.ai).

Extraer o buscar información manual de los artículos de investigación es una tarea que lleva un largo tiempo frente a la gran cantidad de artículos. Un investigador busca una solución guiada por un objetivo en particular, es decir, busca un conjunto de artículos que concuerden con su área de investigación de interés.

Es por ello, que una clasificación automática de documentación, mediante la aplicación de técnicas de Procesamiento de Lenguaje Natural (PLN), presenta un gran impacto al momento de organizar y clasificar artículos de interés por campos y temas, y de esta manera se facilita la tarea de búsqueda de información y brindar soporte a las tareas de investigación para esta nueva temática.

1.3. Justificación.

Según el artículo publicado (Torres-Salinas, 2020), la tasa de crecimiento bibliométrico según el análisis realizado en la base de datos *Dimensions* se calcula en $R^2 = 0.92$, el mismo que determina que la cantidad de publicaciones realizadas es de alrededor de 500 artículos diarios. Sin duda, toda esta cantidad de información es el reflejo de los esfuerzos de la comunidad científica para hacer frente a esta crisis sanitaria que ha afectado múltiples aspectos de la vida cotidiana alrededor del mundo.

Toda esta cantidad de publicaciones y artículos publicados son de naturaleza multidisciplinar, siendo así que cualquier entidad o persona interesada en realizar investigación sobre el COVID-19 con base en un criterio de interés particular debe realizar la búsqueda e ir clasificando los resultados obtenidos de manera manual. Esto supone un alto costo en términos de tiempo, siendo ahora más que nunca el recurso tiempo un factor primordial para hacer frente a esta pandemia.

Si bien varios estudios e investigaciones realizadas como: (Jimenez Gutierrez et al., 2020), (Jelodar et al., 2020) y (Dynomant et al., 2019), han abordado el tema de la problemática de clasificar artículos o documentos de texto acerca del COVID-19 y problemas de salud en general, es importante conocer si la técnica de PLN conocida como *Word Embedding* puede brindar una clasificación de artículos que permita extraer conocimiento relevante, pudiendo ser esta una técnica que pueda brindar soporte a la investigación científica.

2. Marco Teórico

Este capítulo aborda brevemente una introducción a la temática de la enfermedad del COVID-19, junto con los principales conceptos y fundamentos para entender como el texto puede ser clasificado empleando los distintos mecanismos y algoritmos de *Machine Learning* y *Deep Learning*. A su vez, es necesario también revisar los principales métodos de clasificación de texto que emplean los mecanismos y técnicas dentro del Procesamiento de Lenguaje Natural (PLN), y la manera que estos métodos utilizan para representar el texto computacionalmente. Si bien existen diversos algoritmos y técnicas de clasificación, únicamente se realiza una revisión de los principales que han sido desarrollados, ya que el presente estudio no se enfoca en un análisis de los algoritmos y técnicas de clasificación desarrollados hasta la actualidad.

2.1. COVID-19

La enfermedad de la COVID-19, fue declarada como pandemia por la OMS en marzo del 2020 (Adhanom Ghebreyesus, 2020), los primeros casos fueron reportados a finales del 2019 y estaban relacionados a un mercado de animales vivos y mariscos en Wuhan en la provincia de Hubei en China y ha provocado una emergencia sanitaria a nivel mundial por su alta capacidad infecciosa. Hasta el 31 de julio del 2020, la enfermedad se había extendido en más de 217 países, con alrededor de 17.1 millones de casos confirmados y 668.073 muertes. América reportaba 9.5 millones de casos, Europa 3.31 millones, Asia sudoriental 2 millones, Mediterráneo oriental 1.53 millones, África 0.75 millones y el Pacífico occidental 0.31 millones de casos (Aristovnik et al., 2020).

Los gobiernos del mundo adoptaron medidas para frenar su propagación como son: cuarentena obligatoria, aislamiento social, toques de queda y restricción social, para así,

evitar el colapso de los sistemas de salud., Esto provocó serios impactos en las dinámicas de la gente tanto económicas, sociales, laborales culturales, etc.

En cuanto a repercusiones de salud, el nuevo coronavirus, ha logrado afectar a grupos de todas las edades, siendo los grupos con tasas de mortalidad más alta, los de edad avanzada y pacientes con comorbilidades[2].

2.2. Machine Learning

Machine Learning (ML) o aprendizaje automático, es una rama que forma parte de la Inteligencia Artificial, la cual tiene como objetivo desarrollar mecanismos y algoritmos que partiendo de un conjunto de datos puedan realizar tareas específicas, sin que hayan sido programados específicamente para ello, esto quiere decir que los algoritmos de ML tienen como objetivo obtener una función *f,* la misma que producirá una salida de información $y \in Y$, a partir de los datos de entrenamiento $x \in X$, entonces esta función f debe ser capaz de generalizar el problema es decir $f: X \rightarrow Y$, y no únicamente especializarse en el subconjunto de datos de entrenamiento $(f: x \rightarrow y)$, es decir realizar predicciones sobre el dominio de datos completo.

2.3. Procesamiento de Lenguaje Natural

Hoy en día es de gran interés realizar tareas que procesan el lenguaje natural[3] mediante el empleo de técnicas o métodos de aprendizaje automático o *Deep Learning* (DL). El objetivo de PLN, es estudiar, analizar y emplear algoritmos y metodologías para desarrollar modelos computacionales que puedan ser capaces de procesar idiomas en

[2] La comorbilidad es un término que se refiere a la presencia de uno o más trastornos además de la enfermedad o trastorno primario.

[3] El lenguaje natural es la lengua o idioma hablado o escrito por humanos para propósitos generales de comunicación.

lenguaje natural, es decir, que permitan o faciliten la comunicación entre humanos y máquinas o realicen el procesamiento del habla o texto (Christopher D.Manning, 2021).

Los enfoques de PLN actualmente incorporan algoritmos de *Machine Learning* o aprendizaje automático, este enfoque desarrolla técnicas y algoritmos que aprenden a realizar ciertas tareas en particular desarrollando un modelo generalizado con un grupo de datos y haciendo predicciones sobre datos nuevos (Daud et al., 2017). Estas técnicas de *Machine Learning* podrían ser clasificadas en las siguientes categorías:

- Aprendizaje supervisado.
- Aprendizaje semi – supervisado.
- Aprendizaje no supervisado.

Para resolver tareas de PLN, la técnica predominante es de tipo supervisado, que consiste en inducir automáticamente reglas a partir de los datos de entrenamiento, este tipo de aprendizaje puede ser: (i) secuencial y (ii) no secuencial. Para el caso de (i) se puede mencionar entre los más conocidos HMM (Hidden Markov Model), CRF (Conditional Random Fields), MaxEnt (Maximum Entropy) y DL (*Deep Learning*). En el caso de (ii) están SVM (Support Vector Machines), DT (Decision Trees) y Naïve Bayes.

Para el caso del Aprendizaje Semi-Supervisado implica que se debe dar cierto grado de supervisión al modelo, aquí se puede mencionar la técnica de *bootstrapping*. Por el contrario para el Aprendizaje No Supervisado el modelo no se lo entrena sino que determinan intra e inter similitudes entre objetos, entre los cuales cabe mencionar técnicas como *clustering*. La siguiente figura muestra la clasificación de los métodos y algoritmos de ML descritos hasta ahora.

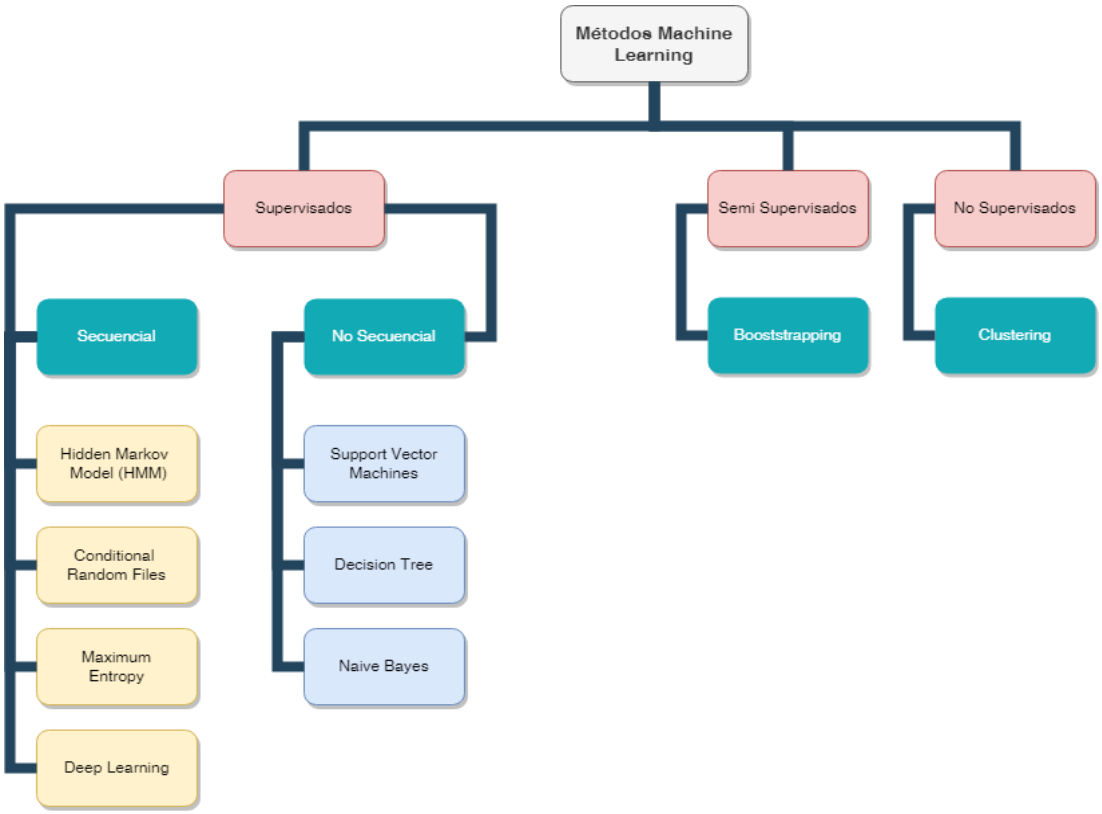


Figura 2: Métodos de Machine Learning

2.4. Representación del Texto

Una de las principales tareas de la clasificación de texto dentro de PLN, es la representación del mismo, el objetivo es representar de manera numérica los documentos de texto, para luego ser procesados computacionalmente, para ello es necesario representar los elementos textuales de los documentos como son palabras, caracteres, *n-grams* de palabras o incluso información morfológica como categorías gramaticales etc. Usualmente, existen dos tipos de representación que son *One-Hot* y representación distribuida o *Embeddings*.

2.4.1. One-Hot.

Consiste en la representación mediante un vector en el cual cada componente xi del vector está asociado al elemento i concreto del documento, esto se puede entender mediante la siguiente manera:

$$x = \{x1, x2, x3, \dots \dots \dots, xn\}$$

En donde, para cada elemento i del documento, está asociado al componente xi del vector x, para lo cual $xi = 1, xj = 0$ para todo $i \neq j$.

Una de las principales ventajas es la facilidad de implementación y la eliminación de errores por parte de algoritmos de aprendizaje. Sin embargo, las desventajas de este método es la incapacidad de establecer relación entre las palabras y el costo en cuanto al espacio, debido a que el tamaño del vector de representación de una palabra dentro del documento coincide con el tamaño del vocabulario del mismo.

En la figura a continuación se muestra un ejemplo de representación vectorial por One-Hot, para representar un diccionario de países.

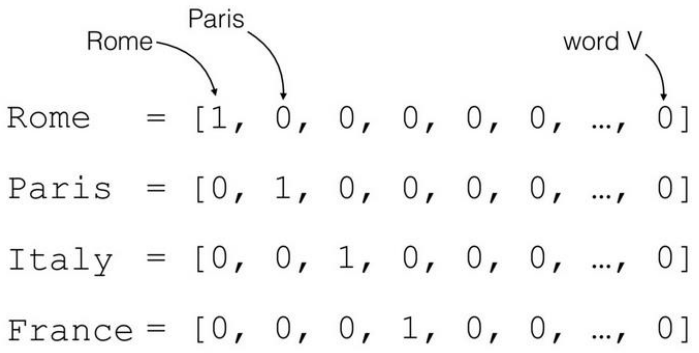


Figura 3: Representación One-Hot de palabras, 2020 (https://medium.com/intelligentmachines)

Como se puede observar en la figura, para cada palabra en una lista se tiene un vector de representación en el cual el elemento *i,* corresponde al elemento en cuestión y los demás elementos son 0, dando así que el tamaño de cada vector corresponde al tamaño del número de palabras del diccionario

2.4.2. Embeddings.

Este tipo de representación surge de la necesidad de resolver los problemas que tiene la representación por *One-Hot*, de manera que las palabras conserven sus propiedades lingüísticas como es el contexto o la similitud semántica entre palabras.

Es importante destacar que estas arquitecturas pueden extraer las propiedades lingüísticas complejas como por ejemplo el género o conceptos como la hiponimia[4]. Se observa, que en el espacio de representación de *Embeddings,* las relaciones entre estas representaciones están dadas por la aritmética de *Embeddings* (González Barba, 2017). La figura a continuación muestra un ejemplo de este tipo de representación en donde cada dimensión del vector representa la relación que existe entre la palabra y las distintas palabras en el diccionario.

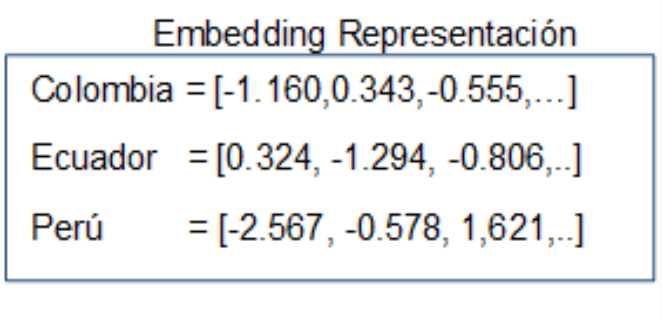


Figura 4: Representaciones vectorial de palabras mediante Embeddings

[4] La hiponimia es una relación que se establece entre una palabra de carácter más específico y otra de carácter más general.

2.4.2.1. Word Embeddings

Casi toda la información generada hoy en día es digital, el contenido textual en bases científicas es de gran volumen y contiene un amplio contenido en distintos campos, es por ello, que las áreas de investigación informáticas como PLN, tienen el propósito de desarrollar metodologías y modelos que permitan el procesamiento de estas grandes fuentes de información.

Las palabras dentro de distintos contextos pueden tener un significado diferente, a esto se lo conoce como los sentidos de las palabras, un ejemplo de esto es:

(a) El dinero fue retirado del ***banco.***

(b) El niño no se levantó del ***banco*** en el cual permanecía sentado.

Nótese en las oraciones (a) y (b) la palabra ***banco,*** tiene distintos sentidos o significados por el contexto en el cual es usada dicha palabra en cada una de las oraciones. La semántica distribucional (Harris, 2015), estudia los mecanismos y métodos que permiten categorizar la similitud semántica de las palabras en grandes cuerpos de texto con base en sus propiedades distribucionales.

Es decir, para el caso del ejemplo anterior, para obtener el significado de la palabra ***banco,*** se debe analizar un conjunto amplio de oraciones en donde aparezca dicha palabra y se podría determinar que si esta palabra se encuentra cerca de otras como, “dinero”, “financiamiento”, “interés”, “deuda”, etc., se puede tratar de dar el significado a ***banco*** en función de todas estas palabras. Esto puede ser interpretado mediante la definición de Hipótesis Distribucional, la cual manifiesta que elementos lingüísticos con distribuciones semánticas similares poseen significados similares. Esto quiere decir, teniendo dos palabras W1 y W2, las mismas que aparecen o tienden a aparecer cerca de una W3, en

distintos textos, quiere decir que tienen distribuciones similares, por lo tanto, se podría postular que tanto W1 como W2 tienen un significado similar.

Con base a esta hipótesis se han propuesto varios métodos para representar de manera computacional las palabras, uno de estos métodos es la representación mediante vectores, en la que cada dimensión es una medida de asociación entre palabras y un tipo de información en particular, como los documentos o textos en los que aparece o palabras junto a las que aparece.

El mayor avance de estos métodos de representación de palabras llega con el trabajo realizado en 2013 (Mikolov et al., 2013), llamados modelos predictivos. Estos modelos tratan de predecir palabras a partir de las palabras que están cercanas a estas en términos de vectores más pequeños y densos, y basan su concepto en que si se puede predecir el contexto en el cual aparece una palabra, entonces se entiende el significado de esta en su contexto. Por lo que palabras semánticamente similares estarán cerca entre sí en sus representaciones de espacios vectoriales. A estos métodos se los denomina *Word Embeddings*.

La figura a continuación es una representación de varias palabras en el espacio multidimensional, para este ejemplo en dos dimensiones, en donde las palabras con mayor relación semántica a la palabra objetivo "***sarscov2***", se encuentran más cerca de esta.

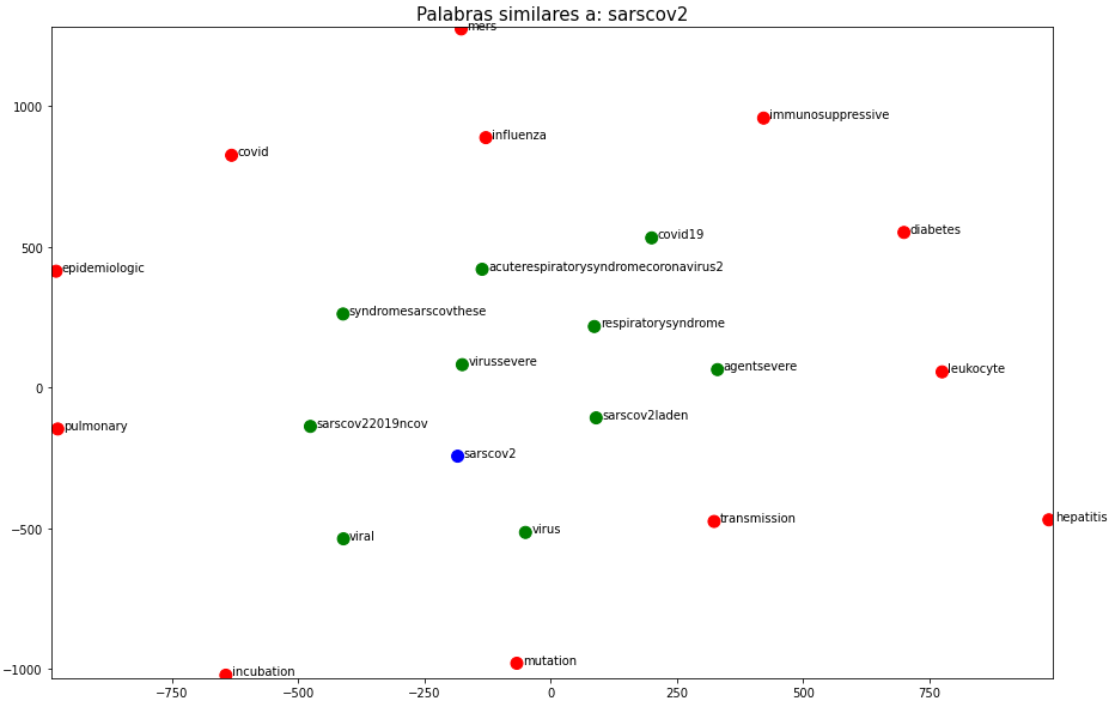


Figura 5: Representación de palabras por Word Embedding en el espacio Multidimensional

Word Embedding es un enfoque de la semántica distribucional, en el cual se representa a las palabras en vectores de números reales, esta representación brinda propiedades de agrupamiento, de manera que palabras que son tanto semánticamente y sintácticamente similares estarán agrupadas o cercanas entre sí. Estos enfoques se basan en redes neuronales que representan el texto en vectores, capturando la similitud semántica entre las palabras. Estas técnicas o modelos pueden realizar inferencias o analogías tales como "*El hombre es para la mujer como el rey es para la reina*", mediante el análisis de adyacencia de las palabras.

Esto se logra mediante el establecimiento de una ventana de contexto, esta ventana no es más que un conjunto o una cadena de palabras tanto antes y después de una palabra focal. Esta palabra focal es utilizada para el entrenamiento del modelo de *Word Embedding*, en donde tanto la palabra focal como la cadena de palabras antes y después (ventana), son representadas por vectores numéricos con el propósito de identificar los patrones y

regularidades lingüísticas. Por lo tanto, *Word Embedding* representa las palabras y conjunto de palabras adyacentes en el espacio multidimensional mediante un modelo de aprendizaje profundo, en donde las palabras cuyos pesos sean similares aparecerán cercanas entre sí en dicho espacio

De esta manera se puede decir que: *Word Embedding* representa las palabras como vectores de números reales en los que las dimensiones del vector capturan los significados de la palabra, lo que conlleva a qué, palabras que sean semánticamente similares posean vectores similares.

Para identificar la similitud entre palabras, este tipo de arquitecturas utilizan la medida de similitud del coseno, esta medida consiste en determinar la similitud que existe entre dos vectores en el que se evalúa el valor del coseno del ángulo comprendido entre estos vectores. Esta medida se abordará a detalle más adelante.

Las técnicas de *Word Embedding* se han convertido en las principales herramientas dentro de los modelos de PLN, capturando el significado de las palabras y convirtiéndolas a una codificación que puede ser utilizada para todo tipo de redes neuronales. Entre las principales aplicaciones de esta técnica son:

- **Sistemas de traducción.-** Este tipo de sistemas están conformados por redes neuronales, una como codificador y otra como decodificador. Tanto en la entrada como en la salida de esta red neuronal son secuencias de palabras que son representadas por *Word Embeddings*. Uno de los más reconocidos ejemplos es el traductor de Google.
- **Análisis de opinión de textos.-** Con el crecimiento de las redes sociales, hoy en día es muy requerido disponer de sistemas que sean capaces de realizar un análisis de opiniones de la gente sobre productos, aceptación de

partidos políticos, etc. Para ello, los sistemas implementan modelos con el uso de redes neuronales convolucionales cuya entrada se emplea *Word Embeddings*.

- **Generación de textos.-** Estos sistemas son empleados para describir automáticamente imágenes y videos.
- **Chatbox.-** Estos sistemas han ganado gran popularidad hoy en día y varias empresas los han implementado en teléfonos y dispositivos personales como lo son el asistente de Google, o Alexa de Amazon.

Estas son algunas de las aplicaciones que tiene Word Embedding, sin embargo, pueden ser utilizados en tareas de distintos ámbitos debido a la capacidad que poseen de capturar el significado de las palabras y la relación entre ellas.

2.4.3. Similitud del Coseno

Como se ha visto hasta ahora, los enfoques de *Word Embedding* representan las palabras en el espacio vectorial multidimensional, por lo que al ser cada palabra una representación en forma de vector, se conserva las propiedades que contienen estos modelos, es decir, se pueden aplicar operaciones vectoriales sobre las representaciones de cada palabra.

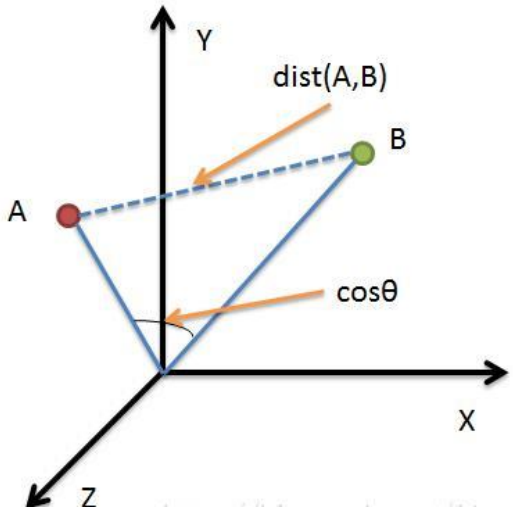


Figura 6: Representación de distancia del coseno entre dos vectores

En la figura anterior se tiene los vectores $\hat{A}$ y $\hat{B}$, representan dos palabras dentro del corpus, para determinar si estas palabras mantienen relación semántica entre ellas se aplica la operación de distancia euclidiana que está determinada por:

$$Cos\ \theta = \frac{\sum_{i=1}^{n} (Ai * Bi)}{\sqrt{\sum_{i=1}^{n} Ai^2} * \sum_{i=1}^{n} Bi^2}$$

El resultado de dicha operación determina la relación entre estos vectores, mientras más cercano el valor a 1, mayor relación tienen los mismos.

Una vez revisadas las definiciones sobre *Word Embeddings,* y su funcionamiento, se presenta a continuación algunas de las tecnologías desarrolladas hasta el momento sobre las cuales se basa el presente estudio.

2.4.4. Word2Vec

Word2Vec genera representaciones de palabras en vectores, los cuales almacenan la relación semántica entre las mismas, estos vectores resultantes son empleados en distintas tareas de PLN, por lo general estos vectores tienen cientos de dimensiones para

cada una de las palabras en el corpus. Word2Vec utiliza un modelo de redes neuronales en el cual se determinan asociaciones o relaciones de palabras en cuerpos de texto de gran tamaño. Una vez que el modelo se ha entrenado, este puede detectar sinónimos de palabras o sugerencias de las mismas para una oración. Este tipo de enfoque implementa dos modelos neuronales, CBOW y Skip-Gram. CBOW, considera el contexto de la palabra objetivo y así lograr su predicción. Por su parte Skip-gram trata de predecir el contexto, dada la palabra. Las capas redes neuronales internas codifican la palabra de destino a su representación vectorial (*Word Embedding*). Word2Vec está disponible para su descarga desde la página web: https://code.google.com/archive/p/word2vec/. A continuación se describen estas dos arquitecturas.

2.4.4.1. CBOW

Una de las arquitecturas pioneras que se desarrollaron para reducir los costes computacionales para la obtención de *embeddings* es la arquitectura llamada "*Continuos Bag Of Words*" (CBOW), en la cual se elimina la capa oculta no lineal que se utilizan en los modelos de redes neuronales NNLM. Como se puede observar en la ilustración wt es el constituyente o el componente del documento (palabras, caracteres, etc.), la red neuronal recibe como entrada a: $w_{t-k}, \dots\dots\dots w_{t-1}, w_{t+1}$, en donde k es el tamaño de la ventana y la salida w_t, siendo cada uno de estos elementos vectores one-hot o vectores de índices. El trabajo de la red neuronal consiste en predecir el constituyente w_t de un documento conociendo su contexto.

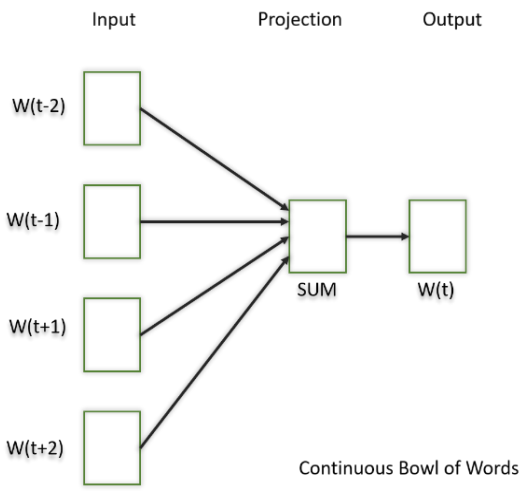


Figura 7: Modelo CBOW (https://arxiv.org/pdf/1301.3781.pdf Mikolov el al.)

Con respecto a la complejidad, si *N* es la dimensión de los elementos de la secuencia de entrada, *D* la dimensionalidad de los *Embeddings* (filas o columnas) y *V* el tamaño del vocabulario, el coste temporal *Q*, de entrenar el modelo CBOW está determinado por la expresión:

$$Q = N * D + D * log_2 V$$

2.4.4.2. Skip-Gram

El modelo Skip-Gram, por el contrario, de CBOW trata de predecir el contexto del constituyente w_t dado, esto es, dada una secuencia de constituyentes de entrenamiento $w_1, w_2, \ldots\ldots, w_t$, el objetivo es maximizar el logaritmo de la probabilidad $\frac{1}{T}\sum_{t=1}^{T} - k \leq j \leq k, j <> 0\ log\ (w_{t+j} \mid w_t)$, donde *T* es la longitud de la secuencia y *k* el tamaño del contexto, en donde mientras más grande el tamaño de *k*, serán mejores las representaciones de vectores debido a la cantidad de constituyentes o elementos lingüísticos.

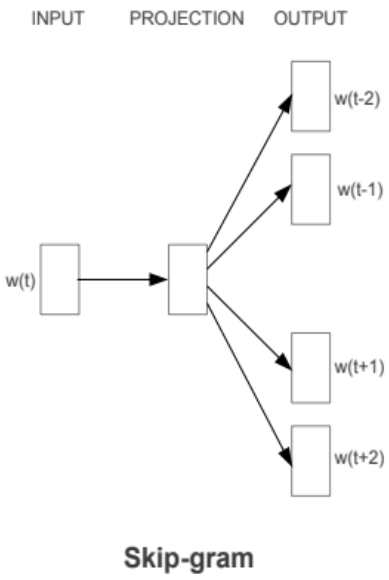


Figura 8: Modelo Skip-Gram (https://arxiv.org/pdf/1301.3781.pdf Mikolov el al.)

2.4.5. FastText.

Representa una palabra mediante la suma de sus composiciones de caracteres llamados n-grams. Por ejemplo, el vector de la palabra "apple" consiste en la suma de los vectores n-gram “<ap, app, appl, apple, apple>, ppl, pple, pple>, ple, ple>, le>”. En consecuencia, aplicando esta técnica, se obtiene una mejor representación de las palabras "raras" que pocas veces aparecen en el cuerpo del texto, y así generar vectores para palabras que no existen en el vocabulario de los *Word Embeddings* (Bojanowski et al., 2017).

2.4.6. Glove.

Es un modelo basado en conteo, en el cual se genera una matriz de gran tamaño que almacena la información de la concurrencia entre palabras y contextos. Es decir, para cada palabra se realiza un conteo de las veces que esta aparece en algún contexto. El objetivo de entrenamiento de dicha matriz es aprender vectores de forma que el producto

escalar entre las palabras sea igual al logaritmo de la probabilidad de co-ocurrencia entre las palabras. El número de contextos es muy alto, por lo tanto, se realiza una factorización de dicha matriz para obtener una de menores dimensiones, dando como resultado mejores representaciones de palabras o *Word Embeddings* (Pennington et al., 2014).

Esto se puede entender mejor de la siguiente manera: se define una matriz de conteo de co-ocurrencia de palabra-palabra, denotada por X, donde la entrada X_{ij}, tabula el número de veces en las que la palabra j aparece en el contexto de la palabra i .

Entonces $X_i = \sum_k \ X_{ik}$, será el número de veces que cualquier palabra aparece en el contexto de la palabra i.

Por último $P_{ij} = P(i) = X_{ij}/X_i$, será la probabilidad en la que la palabra j, aparece en el contexto de la palabra *i*.

2.5. Clasificación de Texto.

La clasificación automática de elementos cualquiera que sea su naturaleza, ya sean imágenes, documentos, formas, etc., consiste en aplicar técnicas computacionales que permitan ordenar estos elementos en determinadas clases o categorías. Esta clasificación automática puede ser de dos formas:

- No supervisada o clustering.- para este caso los objetos se agrupan o clasifican por sí mismos en función de su contenido, de ahí su nombre como clasificación automática, ya que no requiere asistencia o supervisión manual alguna.
- Supervisada.- para este caso se necesita una serie de clases o categorías elaboradas definidas manualmente, para ello, la técnica empleada debe

generar modelos de las categorías definidas manualmente, esta fase del proceso se la conoce como etapa o fase de entrenamiento.

La clasificación de documentos de texto puede llegar a ser una tarea que conlleva un gran esfuerzo en el caso en donde el universo de documentos a clasificar sea de gran tamaño, hoy en día casi toda la información documental es mayormente digital, es aquí donde tareas como la clasificación automática de documentos puede llegar a ser una herramienta de gran importancia para la extracción de conocimiento y búsqueda de información específica.

La clasificación de texto se caracteriza por asignar una clase o varias clases a un documento en cuestión. Esto puede interpretarse de la siguiente manera:

d = Documento

C = Conjunto de Clases {c1, c2, c2, ……., cn}

f = Clasificador.

El clasificador f, debe asignar el documento d, a su clase o clases respectivas.

$$f(d) = C_d$$

Para lograr dicha clasificación se emplean enfoques supervisados en los que dado un conjunto de entrenamiento de documentos etiquetados, se aprende un clasificador (González Barba, 2017).

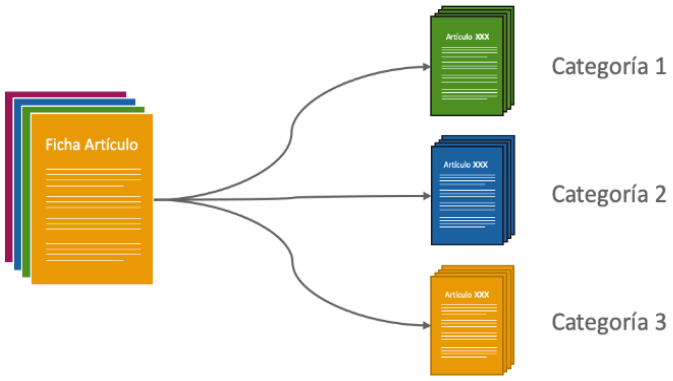


Figura 9: Proceso de clasificación de documentos.

La figura anterior muestra como es el proceso por el cual un conjunto de documentos de texto deben ser clasificados en sus distintas categorías según corresponda el criterio de clasificación.

La clasificación automática de texto es un problema de PLN, que consiste en emplear los mecanismos para etiquetar unidades textuales, ya sean oraciones, consultas, párrafos o documentos, para clasificarlos en categorías o clases dependiendo su naturaleza o contenido, esta área de la PLN tiene una gran aplicación en sistemas de Respuesta de Preguntas o QA (Question Answer), detección de spam, análisis de sentimiento, categorización de noticias, clasificación de intención de usuario, moderación de contenido, etc. El texto es una gran fuente de información, pero extraer conocimientos de él puede ser un desafío y llevar mucho tiempo, debido a su naturaleza no estructurada. Entre las principales tareas de clasificación de texto se destacan las siguientes:

Análisis de sentimiento.- tiene como objetivo analizar las opiniones textuales de las personas sobre productos, películas, tweets. Estas opiniones pueden ser de tipo binaria, es decir clasificar las opiniones como positivas o negativas, así como también puede ser de clases múltiples en las que se realiza una clasificación con base a niveles de intensidad.

Clasificador de noticias.- las noticias son una importante fuente de información que tiene una fuerte influencia sobre las personas, el clasificador de noticias puede ayudar a los usuarios a obtener información relevante o de interés en tiempo real, así como realizar recomendaciones de interés con base a las preferencias del usuario.

Topic Analysis.- o análisis de temas realiza la extracción del significado del texto identificando sus temas, su objetivo es asignar uno o varios temas a cada documento para simplificar su análisis.

Respuestas a Preguntas (QA)

- Extractivos, en el cual dada cierta pregunta y un conjunto de respuestas, el objetivo es clasificar cada respuesta como correcta o incorrecta.
- Generativos, aprende a generar respuestas desde cero.

Inferencia de Lenguaje Natural.- tiene como objetivo predecir el significado de un texto es inferido de otro, como ejemplo el parafraseado es una forma de NLI que también es conocida como comparación entre pares de texto, en el cual se identifica la similitud semántica entre un par de oraciones para determinar si son parafraseo una de la otra.

Las tareas de clasificación de texto pueden realizarse a través de métodos de anotación manual o automáticos. Con la creciente información textual en aplicaciones industriales, la clasificación por medio de métodos automáticos es cada vez más importante y necesaria. Las metodologías de clasificación pueden ser consideradas en las siguientes categorías:

- Métodos basados reglas
- Métodos basados en aprendizaje automático (*Machine Learning*).
- Métodos híbridos.

Los métodos basados en reglas realizan la clasificación del texto basándose en un conjunto de reglas predefinidas, estos métodos son difíciles de mantener y requieren un amplio conocimiento del dominio, por ejemplo, si se quiere clasificar los artículos de noticias sean temas de política y deportes, primero se debe definir dos listas de palabras que sean representativas y caractericen a cada clase es decir para el caso de **Política** se pueden tener una lista de palabras relacionadas como: "*Presidente, legislación, Asamblea Nacional, Plan de Gobierno, Corrupción, etc.*" Para el caso de **Deportes** se deberá contar con una lista como: "*Fútbol, baloncesto, Copa Libertadores, etc.*" Una vez definidas estas listas, para cada uno de los artículos a clasificar el sistema debe ser capaz de realizar un conteo de las palabras relacionadas con cada una de las listas, en donde si el número de palabras relacionadas con política es mayor que deportes, entonces el texto se clasifica como política y viceversa.

Los métodos de aprendizaje automático clasifican el texto mediante observaciones anteriores de los datos usando ejemplos pre-etiquetados o datos de entrenamiento, el algoritmo de aprendizaje automático identifica las relaciones entre fragmentos de texto y las etiquetas, estas etiquetas son las categorías o la clasificación en la que estaría el texto.

Este método en primer lugar extrae las características que consiste en convertir el texto en representaciones de vectores de números.

Luego el modelo es alimentado con datos de entrenamiento que son pares de conjuntos de características y etiquetas para obtener el modelo de clasificación. Por último, una vez que el modelo ha realizado suficientes entrenamientos, ahora es capaz de realizar predicciones precisas.

Por su parte, los métodos híbridos, como su nombre lo indica, utiliza la combinación de los métodos tanto basados en reglas como de aprendizaje automático para realizar la clasificación.

2.6. Algoritmos de Clasificación de Texto

Entre los principales algoritmos de clasificación de texto cabe mencionar los siguientes: *Naive Bayes, Support Vector Machine, Decision Trees y Deep Learning*.

2.6.1. Naive Bayes.

Es uno de los algoritmos más comúnmente utilizados para la clasificación de texto, teniendo buenos resultados aun cuando los conjuntos de datos no son muy amplio y no se posee una gran capacidad de procesamiento.

Este algoritmo se basa en el Teorema de Bayes, cuyo objetivo es calcular la probabilidad condicional de ocurrencia de dos eventos, con base a la probabilidad de ocurrencia individual de cada evento (Cárdenas et al., 2014). Expresado esto de otra manera, lo que intenta el algoritmo es calcular la probabilidad de cada etiqueta para un texto dado y generar la etiqueta con probabilidad más alta.

$$P(A|B) = \frac{[P(B|A)\ x\ P(A)]}{P(B)}$$

En donde:

- $P(A|B) =$ Probabilidad de que A ocurra, una vez que el evento B haya ocurrido.
- $P(A)$ = Probabilidad de ocurrencia del evento A.
- $P(B)$ = Probabilidad de ocurrencia del evento B.

La ecuación anterior se interpreta como, dado un evento A y un evento B, el algoritmo trata de determinar la probabilidad de que ocurra el evento A dado el evento B.

El objetivo del algoritmo de Naive Bayes, es que el texto representado mediante un vector debe poseer información de las probabilidades de ocurrencia de ciertas palabras en los textos para una categoría determinada, para que el algoritmo calcule la probabilidad de que ese texto corresponda a la categoría, y así de esta manera clasificar el mismo, esto se puede representar mediante la siguiente fórmula:

$$P(y|x) = \frac{[P(x|y)\ x\ P(y)]}{P(x)}$$

En donde y corresponden a las etiquetas del texto y x son las variables o texto a clasificar.

2.6.2. Support Vector Machine.

El algoritmo *Support Vector Machine*, de igual manera que Naive Bayes no requiere de un conjunto de datos extensos para obtener buenos resultados, pero, por otra parte, si requiere más recursos de procesamiento.

Este tipo de algoritmo lo que hace es tomar los elementos a clasificar y los lleva al espacio vectorial multidimensional, en donde el algoritmo calcula un hiperplano óptimo, el mismo que sirve para separar los vectores que pertenecen a una clase y los que no pertenecen a esta (Joachims, 2002).

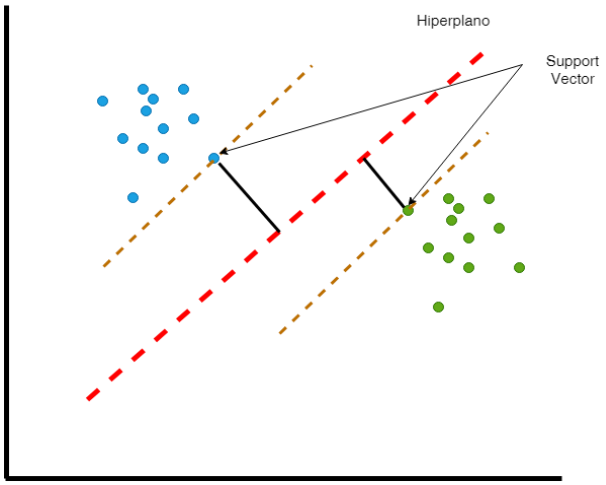


Figura 10: Clasificación SVM

La figura anterior es una representación de la clasificación realizada por SVM, en donde, el objetivo del algoritmo es calcular la distancia que mayor sea posible entre las observaciones y el hiperplano, a esta distancia se la conoce como margen, en donde, un clasificador óptimo obtiene un mayor valor de margen como sea posible.

2.6.3. Decision Tree

Los árboles de decisión se los puede interpretar como árboles binarios, donde cada nodo es una característica del conjunto de datos y las hojas las son las clases en las cuales se clasifican dichas características y las ramas son el conjunto de decisiones que se hacen sobre las características para llegar a obtener una clasificación u otra.

El principal algoritmo de este tipo de algoritmo es el denominado *Random Forest* o Bosques Aleatorios, este algoritmo aplica una serie de técnicas denominadas *bagging,* en donde los árboles con mayor profundidad se combinan para obtener una única salida con varianza reducida.

Cabe resaltar que este algoritmo para disminuir la correlación entre los árboles en las agrupaciones se utiliza muestras y las características de las muestras, por lo tanto, al momento de construir el árbol, este únicamente contiene un subconjunto aleatorio de los datos de entrenamiento y características (Khan et al., 2010).

La figura a continuación representa la clasificación mediante el algoritmo de *Decision Trees*, en el cual el dato de entrada X, cada ramificación del árbol representa la probabilidad que tiene el dato de ser una opción u otra, al final el algoritmo realiza operaciones estadísticas (+) para realizar la clasificación del dato en la salida Y.

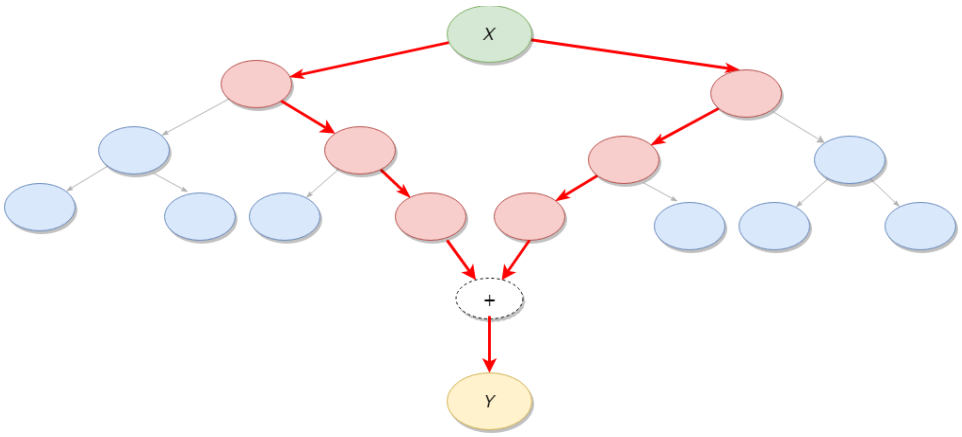


Figura 11: Clasificación por Decision Tree.

2.6.4. Deep Learning.

Los algoritmos de *Deep Learning* o Aprendizaje Profundo, están inspirados en el funcionamiento del cerebro humano, fundamentado en el funcionamiento de redes neuronales. Este tipo de tecnología es parte del área de estudio de la Inteligencia Artificial, y requiere mayor número de datos de entrenamiento en comparación con otros algoritmos tradicionales como SVM o Bayes.

Las dos arquitecturas predominantes basadas en *Deep Learning* para clasificación de texto son las Redes Neuronales Recurrentes – *Recurrent Neural Network* (RNN) y las

Redes Neuronales Convolucionales – *Convolutional Neural Network* (CNN) (Minaee et al., 2020).

Las redes neuronales han demostrado gran eficiencia en cuanto a tareas de PLN, los modelos basados en redes neuronales pueden realizar abstracciones de alto nivel y reducir las dimensiones haciendo uso de múltiples capas de procesamiento, estas capas son nada más que estructuras complejas o combinadas con transformaciones no lineales. Las Redes Neuronales Recurrentes son las más populares en la mayoría de las tareas de PLN. A continuación se realiza una descripción de cómo trabajan las redes neuronales RNN y las CNN.

2.6.4.1. Recurrent Neural Network.

Este tipo de redes utilizan para su procesamiento información secuencial/consecutiva y aumentar la salida mediante el almacenamiento de cálculos previos, esto es posible gracias a su estructura, la cual emplea una función de memoria que realiza dicha acción de almacenamiento. Las RNN tradicionales o básicas tienen una desventaja debido al desvanecimiento del gradiente y son incapaces de aprender dependencias a largo plazo. Debido a esta desventaja, las redes LSTM (*Long Short-Term Memory*) o redes neuronales de corto y largo plazo no son más que una ampliación del enfoque de las RNN, solventan la desventaja del desvanecimiento del gradiente ajustando la información en una celda de estado usando tres compuertas, una compuerta de entrada, una de salida y una de olvido, esta última compuerta determina qué información previa debe ser olvidada. La compuerta de entrada controla la nueva información que es almacenada en la celda de memoria, por último la compuerta de salida determina la cantidad de información de la celda de memoria que debe ser expuesta (De et al., 2017).

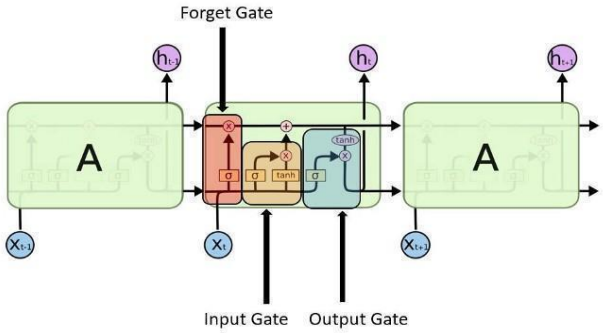


Figura 12: Red Neuronal LSTM (https://www.researchgate.net/).

La figura anterior representa la estructura de una red neuronal tipo LSTM, la cual está compuesta por sus compuertas de entrada, salida y olvido junto con cada uno de sus componentes que se describen a continuación:

- h_{t-1}, h_t, h_{t+1}: Son los estados de salida anteriores de las unidades LSTM.
- X_{t-1}, X_t, X_{t+1}: Son las entradas de las unidades LSTM.
- $\sigma, tanh$: Son funciones de activación no lineares, que realizan la decisión de que valores dejar pasar, así como de ponderar los valores que pasan según su nivel de importancia.

Como se menciona, este tipo de redes neuronales son excelentes para procesar secuencias de datos, siendo muy útiles para el procesamiento de videos o de texto, donde las secuencias de datos puede ser variable, ya que un video puede contener un número variable de *frames*, así como un texto escrito, o una conversación pueden tener un conjunto variable de palabras.

Para la clasificación de texto es necesario analizar las palabras como una secuencia, puesto que al analizar palabras individualmente por sí solas, estas carecen de sentido, sin embargo, al estar relacionadas dentro de un párrafo, el orden determinado de

cada palabra que compone el párrafo o el texto le da sentido al mismo. A continuación se describe los distintos tipos de RNN que existen:

- **One To Many.-** Este tipo de redes son utilizadas cuando la entrada de información es un dato y la salida es una secuencia de datos, un ejemplo de uso de estas redes es "***image captioning",*** en el cual la entrada puede ser una imagen y la salida puede ser un conjunto de palabras o texto de descripción de dicha imagen.

Figura 13: Red Neuronal RNN tipo One to Many.

- **Many to One.-** Al contrario del modelo anterior, las redes neuronales de tipo Many to One tienen un conjunto de datos de entrada y su salida es un solo dato, un ejemplo de aplicación de este tipo de redes neuronales son la clasificación de sentimientos, donde la entrada puede ser la reseña en texto de la sobre un producto o servicio, y la salida es una categoría sobre el gusto de la persona sobre dicho producto o servicio.

Figura 14: Red Neuronal RNN tipo Many to One.

- **Many to Many.-** Como su nombre lo indica, este tipo de redes tienen tanto a la entrada como a la salida secuencias de datos, un ejemplo de este tipo

de redes pueden ser los conversores de voz a texto que vienen en algunos dispositivos móviles que hoy en día son usados ampliamente. También se utilizan en los traductores de diferentes idiomas como el caso de la red neuronal de Google Translate.

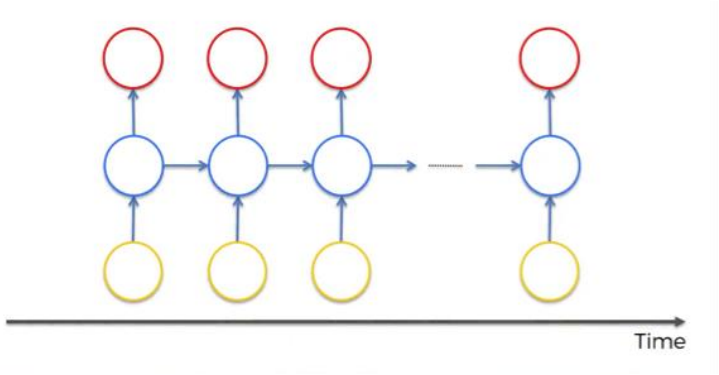


Figura 15: Red Neuronal RNN tipo Many to Many (https://sds-platform-private.s3-us-east-2.amazonaws.com)

2.6.4.2. Convolutional Neural Networks

Las redes de tipo CNN, por otra parte, su principal objetivo de este tipo de redes neuronales era el de reconocer patrones visuales en imágenes. Este tipo de redes emplean como su nombre lo indica operaciones convolucionales de ventanas móviles o tamaños de filtro que lo que hacen es analizar y reducir secciones superpuestas de una matriz para la extracción de características. La funcionalidad de extraer *embeddings* en este tipo de redes la hace en una excelente herramienta para extraer conocimiento y realizar funciones de clasificación de texto (Chatsiou, 2020).

Las redes convolucionales tienen gran éxito gracias a los filtros que se utilizan en cada capa de la cual se extraen distintas características, tal como se muestra en la figura a continuación, en donde la red neuronal durante la primera convolución se identifican características primarias como líneas o curvas y a medida que se realicen más operaciones de convolución la red neuronal es capaz de reconocer figuras de mayor complejidad. Cada

operación de convolución consiste en tomar grupos de píxeles e ir realizando el producto escalar con una matriz más pequeña llamada kernel.

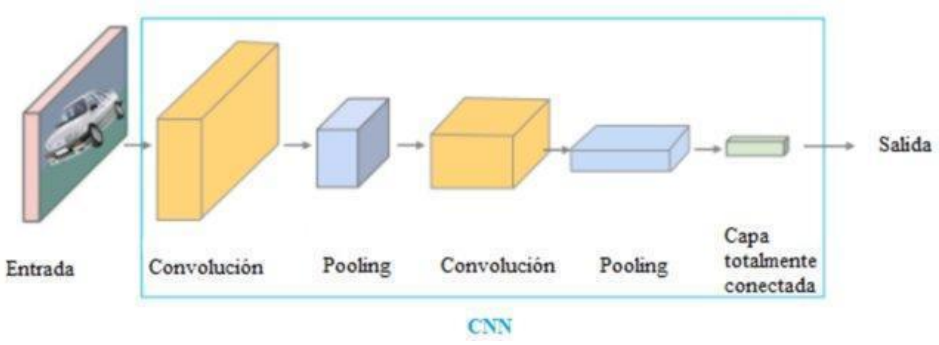


Figura 16: Red Neuronal Convolucional (https://www.researchgate.net/).

Este tipo de redes neuronales son excelentes para el procesamiento de imágenes, ya que en la entrada de la red neuronal se tiene la imagen a ser procesada, las primeras capas de la red convolucional se extraen patrones básicos como líneas, y bordes y mientras más profunda es la red todos estos elementos básicos se van combinando en formas más complejas hasta que al final la red es capaz de detectar la imagen completa.

2.7. Evaluación de Modelos de Machine Learning.

Existen diversos métodos de evaluación de modelos basados en ML, se presenta a continuación algunos de estos métodos que son los más comunes al momento de realizar dicho proceso de evaluación.

Para entender estas métricas o métodos de evaluación se debe conocer los siguientes términos que son necesarios al aplicar cualquiera de dichas métricas:

- **True Positives (TP):** Verdadero positivo cuando tanto la clase real y la predicha son verdaderas.
- **True Negative (TN):** Falso negativo, cuando tanto la clase real y la predicha son falsas.

- **False Positive (FP):** Falso positivo, cuando la clase predicha es verdadera, pero en realidad es falsa.
- **False Negative (FN):** Falso negativo, cuando para una determinada clase la predicción indica que es falsa, pero en realidad es verdadera.

Ahora que se ha revisado los términos necesarios que son empleados por las métricas de evaluación, se describen a continuación dichas métricas, las mismas que serán utilizadas para evaluar el modelo propuesto en el presente estudio.

- **Accuracy o exactitud.-** Es el porcentaje de predicciones realizadas correctamente, este método de evaluación es significativo únicamente si las muestras para cada una de las clases está balanceado.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precisión.-** Esta métrica corresponde al porcentaje de predicciones que se han realizado sobre una clase correctamente.

$$Precision = \frac{TP}{TP + FP}$$

- **Recall.-** Esta métrica está dada por el número de predicciones bien etiquetadas, entre un número de muestras relevantes. Este tipo de métrica es muy utilizada cuando se disponen de conjuntos de datos desbalanceados cuando existen datos de clases muy minoritarias.

$$Recall = \frac{TP}{TP + FN}$$

- **F1 Score.-** Esta métrica es la combinación de las métricas de precisión y *recall*, y puede darnos la relación que existe entre ellas, entre más cercano el valor a 1 tiene una mejor puntuación. Esta métrica es la media armónica, que tiende hacia los últimos elementos de una lista, es decir, a los valores en los que las clases son minoritarias, por lo contrario de la media aritmética, la cual es influenciada por los valores de clases mayoritarias.

$$f1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

- **Matriz de confusión.-** Permite visualizar de manera gráfica el comportamiento del modelo, en donde, uno de los ejes representa la clase real y el otro eje representa la clase predicha, en donde la relación entre *x* = *y,* se obtiene que la diagonal contiene los valore TP de cada una de las clases, los valores que se encuentran por su parte fuera de la diagonal y fuera de la fila corresponden a los valores TN, por otra parte, los valores FP son los valores de la fila menos los valores de la diagonal, por último los valores FN corresponden a los valores de la columna menos los valores de la diagonal.

3. Estado del Arte

En este capítulo se analizan distintos trabajos que abordan la clasificación de artículos acerca del COVID-19, y las distintas técnicas que emplean cada uno de ellos para realizar dicha clasificación.

3.1. COVIDScholar

El proyecto COVIDScholar, nace de un esfuerzo para afrontar los problemas de extraer conocimiento de las múltiples publicaciones realizadas en distintos servicios de repositorios de información mediante la aplicación de técnicas de PLN, para agregar, analizar y buscar literatura de investigación acerca del COVID-19 mediante la implementación de una infraestructura automatizada y escalable. El proyecto busca integrar investigaciones recientes tal como se van publicando, logrando así, levantar un corpus de más de 81,000 artículos científicos y demás documentos relacionados con el COVID-19 (Trewartha et al., 2020).

El núcleo de *COVIDScholar*, está compuesto por la canalización, procesamiento y entrada de información, las fuentes de información se verifican de manera continua en búsqueda de nuevo material, que luego es analizado por modelos de PLN. El corpus sobre el cual trabaja *COVIDScholar* consta de 14 fuentes de servicios de datos abiertos (Ver Tabla 1) en los que para cada una de estas fuentes de información, un web scraper,[5] comprueba periódicamente si existen nuevas publicaciones.

Tabla 1: Fuentes de Datos COVIDScholar

#	Fuente
1	preprins.org
2	osf.io
3	lens.or
4	SSRN

[5] Web scraping o raspador web, son un conjunto de programas de software que sirven para extraer información de sitios web.

5	Psyarxiv
6	CORD-19
7	Dimensions.ai
8	Elsevier
9	Chemrxiv
10	LitCovid
11	Biorxiv / Medrxiv
12	NBER.org
13	COVIDScholarUser
14	Submission

Luego las publicaciones ya recopiladas son analizadas y transformadas a un formato unificado, se depuran y se eliminan publicaciones duplicadas para el caso del mismo artículo en versión de preimpresión y final. Las publicaciones duplicadas se identifican cuando comparten características como el DOI[6], título, ID de publicación. Para el caso de ensayos clínicos se utiliza el título para identificar duplicados.

Los *abstracts* o resúmenes de estas fuentes de información son clasificados dependiendo de su relevancia con el tema COVID-19, disciplina y campo. Las publicaciones se clasifican en cinco disciplinas:

- Ciencias Biológicas y Químicas.
- Ciencias Médicas.
- Salud Pública.
- Ciencias Físicas.
- Ciencias Humanas y Sociales.

Un artículo o publicación puede pertenecer a varias disciplinas. Los algoritmos de aprendizaje automático pueden ser empleados para identificar tendencias emergentes en la literatura y correlacionarse con patrones similares de investigación. Es así que

[6] Un DOI (Digital Object Identifier) es una forma de identificar un objeto digital (por ejemplo un artículo electrónico de una revista, un capítulo de un libro electrónico) sin importar su URL, de forma que si ésta cambia, el objeto sigue teniendo la misma identificación.

COVIDScholar basa su *back-end* empleando un motor de búsqueda VESPA [7] (https://docs.vespa.ai/, s.f.), que integra modelos personalizados de aprendizaje automático.

COVIDScholar emplea técnicas no supervisadas de ***documents embeddings***, para la búsqueda de documentos relacionados de modo que se vinculan automáticamente investigaciones, ya sea por temas, métodos, medicamentos y otras piezas claves de información. La clasificación de documentos, se realiza empleando la distancia o similitud del coseno [6.3.3], entre cada *embedding* de documento generado, que posterior se combinan con una puntuación de clasificación de resultados, brindando a los usuarios la obtención de resultados con base a un dominio de búsqueda específico.

Para la clasificación de *abstracts* se emplea la técnica SciBERT (Beltagy et al., 2019) refinado, debido a que otros modelos BERT como son BioBERT (Lee et al., 2020), MedBERT (Rasmy et al., 2021) y ClinicalBERT (Mulyar et al., 2021), son modelos que se centran en una disciplina específica, sin embargo, SciBERT es un modelo de análisis de texto multidisciplinario tanto biomédico como otras ramas en general y tiene un gran rendimiento en tareas de clasificación de documentos de texto. A continuación la Figura 17 muestra la estructura del proyecto COVIDScholar.

[7] Vespa es un motor para el cálculo de baja latencia en grandes conjuntos de datos. Almacena e indexa sus datos para que las consultas, la selección y el procesamiento de los datos se puedan realizar en el momento de la entrega.

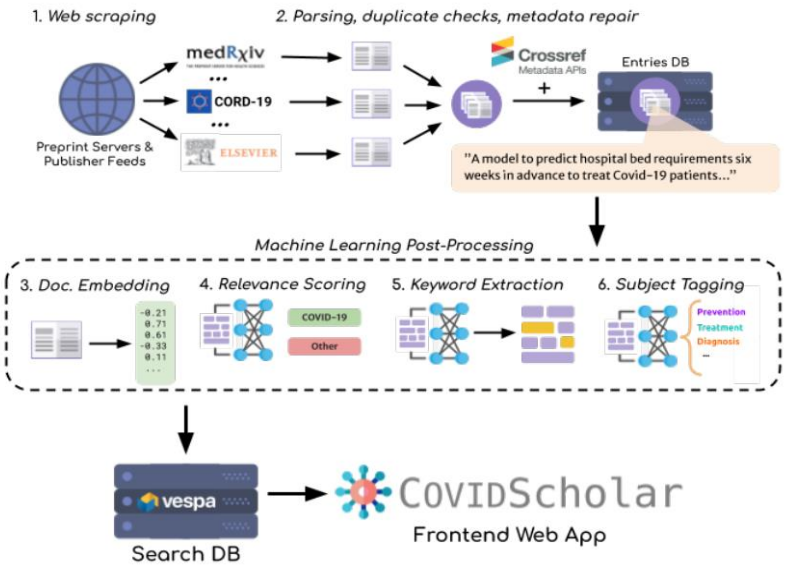


Figura 17: Estructura de COVIDScholar (Trewartha et al., 2020).

3.2. CovidNLP

Este estudio realizado por *Indraprastha Institute of Information Technology Delhi*, All *India Institute of Medical Sciences, New Delhi* y *CSIR-Institute of Genomics and Integrative Biology*, el cual emplea mecanismos de PLN y *Machine Learning* sobre los artículos de investigación de la OMS, con el fin de generar conocimiento que pueda guiar tanto las políticas del COVID-19, investigaciones y desarrollo (Awasthi et al., 2020). Se aplican enfoques de resumen de texto y los modelos entrenados de *Word Embeddings* para resumir la información publicada, dando como resultado la herramienta CovidNLP, la misma que está disponible en http://covidnlp.tavlab.iiitd.edu.in/.

CovidNLP emplea la arquitectura *Skip-Gram* de Word2Vec [6.3.4.2], para extraer el contexto de cada palabra, junto con una red neuronal, que se entrena con vectores codificados *One-Hot*, para predecir la palabra objetivo con un tamaño de ventana fijo, iterando a lo largo del corpus. En la última capa de la red neuronal, se emplea la función

Softmax[8], para el cálculo de las probabilidades y los errores contra las etiquetas de clase reales. La matriz de peso calculada genera *Word Embeddings* cuando se multiplica con los vectores *One-Hot*.

Por último, se realiza un Análisis Semántico Latente (*Latent Semantic Analysis*) LSA, que es un método de aprendizaje no supervisado, este procedimiento es utilizado para la extracción de los resúmenes de los *abstracts*. Además, se hace uso de una matriz de oraciones de término de *n x m* (*n* términos, *m* oraciones) con ponderaciones obtenidas a partir del método de Frecuencia de TF-IDF. La descomposición de valores singulares (SVD) se utilizó para descomponer la matriz en términos y temas, además de una matriz de frases temáticas. La matriz diagonal codifica el peso de los temas a lo largo de la diagonal y define en qué medida una oración se parece a un tema.

3.3. Document Classification for COVID-19 Literature.

Este estudio realizado por la Universidad de Ohaio (Jimenez Gutierrez et al., 2020), realiza una clasificación de los documentos acerca del COVID-19, aplicando distintas metodologías como modelos tradicionales de *Machine Learning* como *Support Vector Machine*, Regresión Logística, modelos de redes neuronales como CNN, y modelos de lenguajes pre-entrenados. Este estudio, realiza una comparativa de estas distintas metodologías mediante un análisis de rendimiento basándose en las métricas de rendimiento *f1-score* y *accuracy*.

La clasificación se realiza empleando los conjuntos de datos LitCovid y CORD-19, en donde, las métricas de rendimiento aplicadas a los modelos desarrollados obtuvieron mejores resultados con el modelo que emplea BioBERT (*Bidirectional Encoder*

[8] Una función softmax es una generalización de la función logística que se puede utilizar para clasificar múltiples tipos de datos. La función softmax toma valores reales de diferentes clases y devuelve una distribución de probabilidad.

Representations from Transformers for Biomedical Text Mining) (Lee et al., 2020). El método consiste en una representación de lenguaje entrenado específicamente en corpus biomédicos a gran escala. BioBERT supera en un gran nivel a su predecesor BERT. En términos generales, el algoritmo de BERT analiza tanto las palabras a la derecha como a la izquierda de una palabra focal, para de esta manera, relacionar todas las palabras de la consulta entre sí en vez de cada una de ellas, independientemente, así como de establecer las relaciones entre las palabras, pronombres y preposiciones, consiguiendo así captar el sentido y contexto de una frase en lenguaje natural tal como lo expresa el ser humano.

BioBERT ha sido entrenado en corpus biomédicos como *abstracts* de artículos de PubMed y textos completos de artículos de PMC, el mismo que ha demostrado una alta efectividad en tres tareas populares de minería de texto como son NER (*Named Entity Recognition*), RE (*Relation Extraction*) y QA (*Question Answer*).

Como se ha visto hasta ahora, los estudios realizados han empleado distintas metodologías, sin embargo, no se han encontrado trabajos que empleen metodologías de minería de datos aplicados a proyectos de minería de texto para la clasificación del mismo, así también los estudios realizados emplean distintos enfoques de representación del texto como SciBERT, BioBERT, *Documents Embeddings,* entre otros, y los métodos utilizados de clasificación del mismo van desde redes neuronales CNN, TF-IDF y LSA, es por ello que este estudio presenta un marco de referencia para la clasificación de artículos científicos, apoyándose en el marco metodológico CRISP-DM empleando los enfoques de PLN como es *Word Embedding*.

4. Marco Metodológico

Este capítulo describe la metodología adoptada para el desarrollo del presente estudio, a fin de conseguir el objetivo propuesto. La metodología CRISP-DM (*CRoss-Industry Standard Process for Data Mining*), se toma como base metodológica, debido a que tanto la minería de texto como la de datos, tratan de obtener el mayor conocimiento posible, basándose en el análisis de grandes volúmenes de texto y datos respectivamente. De la misma manera, cabe señalar, que el presente estudio se apoya en esta metodología, en vista que no se han encontrado en los estudios realizados hasta la actualidad trabajos que empleen metodologías de minería de datos para proyectos de minería de texto para su clasificación.

4.1. Metodología CRISP-DM

La metodología CRISP-DM (*CRoss-Industry Standard Process for Data Mining*), es un modelo el cual brinda una descripción de manera general el ciclo de vida de un proyecto de minería de datos, donde se detallan sus fases, tareas y resultados respectivos. Todo proceso en el cual se aplique ciencia de datos requiere seguir un conjunto de procedimientos estandarizados y, ya que tanto la minería de datos como la minería de texto tratan de obtener conocimiento del análisis de grandes volúmenes de información, se toma como base esta metodología en el presente estudio. Es importante señalar que en los casos de aplicación de minería de datos la metodología más aplicada ha sido CRISP-DM, tal como se observa en la Figura 16, publicada por www.datascience-pm.com, en donde se muestra que durante los últimos veinte años ha sido la metodología mayormente utilizada en este tipo de proyectos.

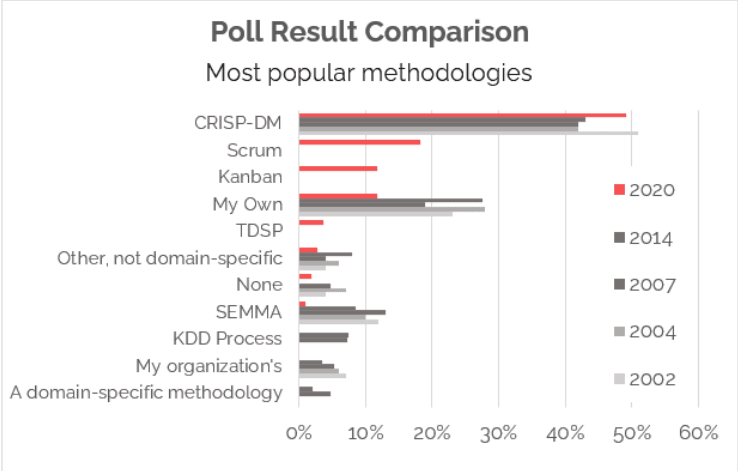


Figura 18: Metodologías más utilizadas de DM (www.datascience-pm.com)

CRISP-DM está compuesta por seis fases que en algunos casos son bidireccionales, es decir, en ciertos casos se puede regresar a la fase anterior, para revisar nuevamente si es necesario. Este modelo proporciona una visión acerca del ciclo de vida de un proyecto de minería de datos y como las fases se interrelacionan entre ellas (Wirth, 2000), tal como se muestra en la siguiente figura.

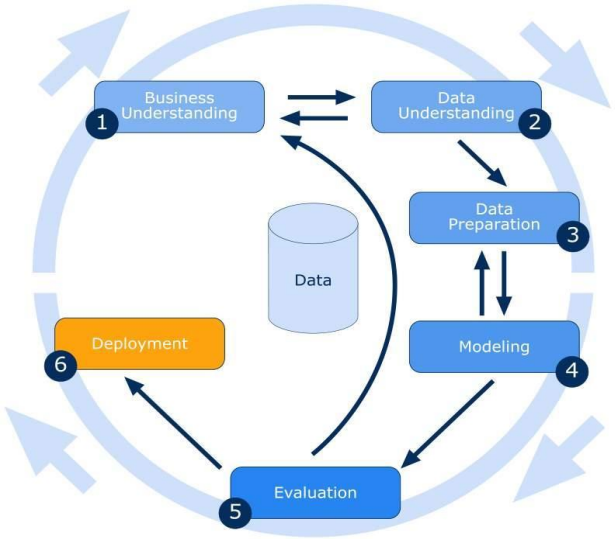


Figura 19: Fases de la metodología CRISP-DM (https://www.ukessays.com/)

Una vez presentada de manera general la base metodológica sobre la cual se apoya el presente estudio, se describe en las secciones a continuación, cada una de las fases y las tareas que componen cada una de ellas, para su entendimiento dentro en la aplicación en proyectos de minería de datos.

4.1.1. Conocimiento del Negocio

La primera fase de entendimiento o comprensión del negocio está compuesta por todas las tareas de comprensión de los objetivos y requisitos del proyecto, para de esta manera transformar estos objetivos y requisitos en objetivos técnicos para un plan de proyecto. Al igual que en todo proyecto, el entender completamente los requisitos iniciales permitirá obtener buenos resultados. Esta fase se debe conseguir un buen entendimiento del problema a resolver para de esta manera realizar una correcta recolección de los datos y consecuentemente obtener una buena interpretación de los mismos. A continuación, se describen las tareas que componen esta fase.

- Determinar objetivos del negocio.- La primera tarea consiste en identificar el problema que se necesita resolver, identificar por qué se requiere aplicar minería de datos, es común que tener al inicio muchos objetivos y limitaciones que se deben equilibrar en esta fase, por lo tanto, es necesario identificar los factores que pueden influir en los resultados finales del proyecto de *Data Mining*.
- Evaluación de la situación.- En esta tarea se analiza la situación actual antes de iniciar con el proyecto de *Data Mining*, es decir, se debe considerar los factores como: cuál es la información disponible, se cuenta con los datos suficientes para resolver el problema, disponibilidad de recursos

informáticos, etc., esta fase es determinante en donde se identifican los requisitos necesarios para la resolución del problema.

- Determinar los objetivos de *Data Mining*.- En esta tarea se establecen los objetivos de negocio en términos de objetivos técnicos o de *Data Mining*, esto quiere decir que aquí se definen los criterios técnicos para que lograr un resultado exitoso del proyecto.

  Realizar el plan del proyecto.- Se estructura en esta tarea, el plan que va a lograr los objetivos de *Data Mining* propuestos en la fase anterior. Este plan debe contener todos los pasos y técnicas que se ejecutarán durante el desarrollo del proyecto.

4.1.2. Comprensión de los datos

Esta fase tiene como objetivo el de tener contacto con los datos y poder familiarizarse con los mismos, determinar la calidad de los mismos. Esta fase está compuesta de las siguientes tareas:

- Recolección inicial de datos.- Es el proceso que lleva en recolectar los datos iniciales para que puedan ser procesados en el futuro, esta tarea conlleva el registro de la fuente de donde se consiguieron los datos, las técnicas empleadas para su recolección y posibles problemas que se hayan presentado durante este proceso.
- Descripción de datos.- Esta tarea consiste en realizar la descripción de cómo se encuentran los datos recolectados en la tarea anterior, esta tarea implica identificar el número de registros, campos por registro, formato de los datos y cualquier otra característica que se pueda identificar en esta exploración inicial.

- Exploración de los datos.- Esta tarea consiste en refinar la descripción ya realizada en la tarea anterior mediante técnicas de consulta, visualización e informes.
- Verificar la calidad de los datos.- Esta tarea determina la consistencia de los datos, en los que se obtiene si los mismos contienen valores nulos, distribución de los mismos y demás factores que pueden introducir ruido en el proceso.

4.1.3. Preparación de datos

Esta fase compone tareas como la exploración inicial de los datos, emplear herramientas para la visualización de los mismos, en los casos que se dispongan de varias fuentes de datos, se debe tratar el problema de integración de estas fuentes desde esta fase.

Como resultado de esta fase se obtiene un informe que contiene la lista de conjuntos de datos conjuntamente con las fuentes respectivas, así como las dificultades de extracción de los mismos. A continuación se detallan las distintas tareas que componen esta fase.

- Seleccionar datos.- Implica la acción de seleccionar los datos que van a ser objeto de estudio basándose en los criterios definidos en las fases anteriores.
- Limpieza de datos.- Esta tarea puede conllevar gran esfuerzo por las distintas técnicas a emplearse para mejorar la calidad de los datos para su procesamiento, entre las técnicas empleadas puede ser normalización de los datos, reducción del conjunto de datos mediante aplicación de filtros de interés, etc.

- Construir datos.- Esta tarea implica operaciones como generar nuevos atributos tomando valores ya existentes, transformación de valores.
- Integrar datos.- Consiste en combinar varios registros o valores para crear nuevos, la combinación de datos puede cubrir lo que se denomina como agregación, que consiste en generar nuevos valores al resumir información de varios registros o valores.
- Formateo de datos.- Consiste en aplicar técnicas para modificar de manera sintáctica los datos sin alterar de alguna manera su significado, para facilitar el uso de las técnicas de minería.

4.1.4. Modelado

Esta fase se selecciona la técnica o técnicas de modelado para ser aplicada al proyecto de *Data Mining*, todas las técnicas que se seleccionen en esta fase deben obedecer en función de los siguientes parámetros:

- Técnica apropiada para el problema.
- Disponer de datos adecuados.
- Cumplir con los requisitos del problema.
- Relación costo beneficio.
- Conocimiento de la técnica a emplear.

4.1.5. Evaluación

Esta fase constituye el grado en el cual el modelo empleado cumple con los requisitos y objetivos planteados, así como también el determinar las causas de falencia en caso de que el modelo sea deficiente.

4.1.6. Despliegue e Implantación

Esta fase, tal como su nombre lo indica, es el despliegue del modelo ya construido y validado para generar conocimiento en el proceso de negocio. Por lo general, los proyectos de minería de datos no concluyen en la implementación del modelo, sino en la documentación de los resultados con el propósito de generar conocimiento.

5. Aplicación de la metodología

Una vez revisada la base metodológica sobre la cual se apoya el desarrollo del presente estudio, se pone en práctica la aplicación de la misma, adaptándola según el objetivo a cumplir. Esta se focaliza en cinco fases principales: (i) Identificar Objetivos del Proyecto, (ii) Acceso a los Datos, (iii) Tratamiento de los Datos, (iv) Modelización, (v) Evaluación. La Figura 20 a continuación muestra un esquema de la aplicación de la metodología.

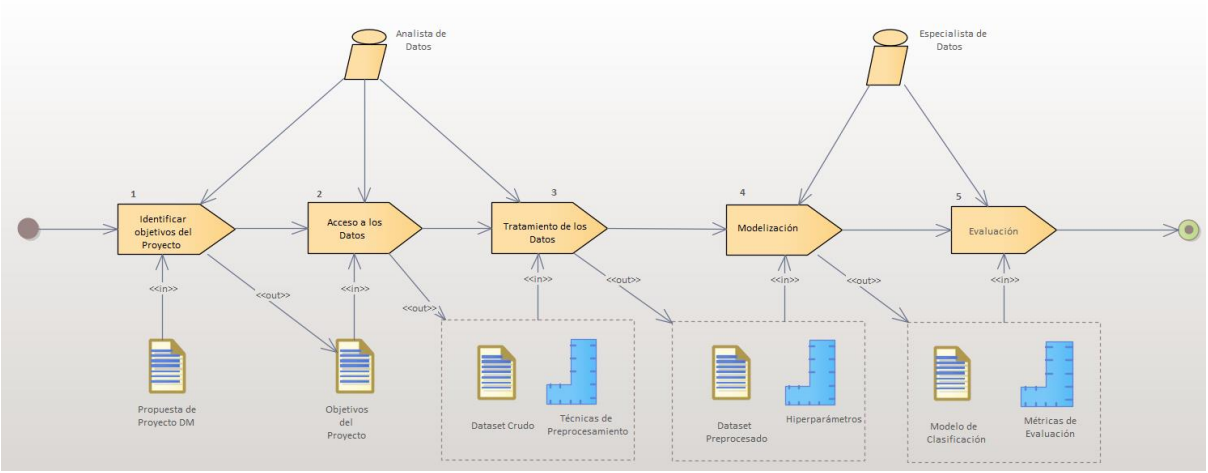


Figura 20: Metodología Experimental

5.1. Identificar Objetivos del Proyecto

El objetivo de clasificar los artículos académicos mediante la aplicación de técnicas de minería de texto es sin duda dar soporte a las distintas áreas de la investigación para extraer conocimiento sobre el gran volumen de artículos que han sido publicados desde el inicio de la enfermedad del COVID-19. Este conocimiento puede ser de ayuda en temas como tratamiento, prevención, desarrollo de vacunas, etc., que son prescindibles para hacer frente a esta pandemia.

La comunidad científica ha recopilado un sin número de documentación que van desde recomendaciones, publicaciones científicas, protocolos de investigación, toda esta información se encuentra publicada y disponible para las autoridades, investigadores, profesionales de la salud y público en general para apoyar en la toma de decisiones frente a la pandemia.

Hoy en día muchos de los artículos se publican casi a diario, bajo la modalidad de publicación continua, así como muchas de las publicaciones están disponibles en repositorios de libre acceso como *preprints*, es decir, previo de su revisión por pares.

De la misma manera, las revistas médicas, en sus versiones digitales, emplean en gran medida la publicación continua, para proveer casi en tiempo real los artículos que describen hallazgos que aparecen en distintas partes del mundo(Beldarraín, 2020).

También hay que señalar, que la clasificación de artículos sería un tipo de clasificación multiclase, esto permitirá más adelante identificar cuál sería la técnica a emplear para la implementación del modelo de clasificación.

Como parte de esta fase, hay que identificar los recursos disponibles, para lo cual, se considera el uso de la herramienta Google Colab, que es una herramienta de Google Research, la cual facilita la codificación y ejecución de código en lenguaje Python desde el navegador web, esta herramienta cuenta con todas las librerías que se requieren para realizar tareas de PLN.

5.2. Acceso a los Datos

Luego de realizar la búsqueda de los conjuntos de datos disponibles se han determinado varias fuentes, entre las más destacadas se puede mencionar a LitCovid y CORD-19, que son conjuntos de datos de libre acceso para desarrollar tareas de PLN, *Deep*

Learning y *Machine Learning.* Para el desarrollo del presente estudio se ha establecido a LitCovid como *dataset* para la clasificación, esto debido a que LitCovid es un conjunto de datos etiquetados y tal como se revisó en el capítulo 6, la técnica mayormente empleada para tareas de PLN es de tipo supervisada, por lo que las etiquetas asociadas a estos datos son de ayuda para aplicar este tipo de técnicas.

LitCovid es una recopilación de artículos recientemente publicados, cuyas temáticas están relacionadas con la literatura actual del Coronavirus. Este conjunto de datos contiene más de 23,000 artículos y en promedio se agregan 2,000 nuevos artículos semanalmente, siendo así un recurso integral para que la comunidad científica pueda actualizarse con información acerca de la crisis que ha provocado la pandemia de la COVID-19.

Cada uno de los artículos contenidos en el conjunto de datos de LitCovid, son etiquetados en una de las siguientes temáticas: (i) Prevención, (ii) Tratamiento, (iii) Diagnóstico, (iv) Mecanismo, (v) Reporte de casos, (vi) Transmisión, (vii) Pronóstico y (viii) General. La Figura 21 muestra una breve descripción de la estructura que tiene el *dataset*, en donde se puede observar que el mismo está compuesto por nueve columnas de la siguiente manera.

- **Pmid:** Identificador del artículo en el *dataset*
- **Journal:** Revista en la cual está publicado el artículo.
- **Tittle:** Título del artículo
- **Abstract:** Resumen o abstract del art
- **Keywords:** Palabras claves o términos de búsqueda relevantes
- **Pub_type:** Tipo de publicación que puede ser artículo de investigación, guía práctica, etc.
- **Authors:** Authores del artículo

- **Doi:** Identificador único para los documentos publicados de manera electrónica
- **Label:** Etiqueta que se ha asignado al artículo dependiendo de su temática.

	pmid	journal	title	abstract	keywords	pub_type	authors	doi	label
0	32519164	J Thromb Thrombolysis	Potential role for tissue factor in the pathog...	In December 2019, a new and highly contagious ...	covid-19;il-6;sars-cov-2;tnf-alpha;thrombosis;...	Journal Article;Review	Bautista-Vargas, Mario;Bonilla-Abadia, Fabio;C...	10.1007/s11239-020-02172-x	Treatment;Mechanism
1	32691006	J Tradit Complement Med	Dietary therapy and herbal medicine for COVID-...	A novel coronavirus disease (COVID-19), transm...	covid-19;coronavirus;dietary therapy;herbal me...	Journal Article;Review	Panyod, Suraphan;Ho, Chi-Tang;Sheen, Lee-Yan	10.1016/j.jtcme.2020.05.004	Treatment;Prevention
2	32858315	J Affect Disord	First report of manic-like symptoms in a COVID...	BACKGROUND: In December 2019, the novel corona...	cerebrospinal fluid;igg;manic-like symptoms;sa...	Case Reports;Journal Article	Lu, Shaojia;Wei, Ning;Jiang, Jiajun;Wu, Lingli...	10.1016/j.jad.2020.08.031	Case Report
3	32985329	J Dent Res	Epidemiological Investigation of OHCWs with CO...	During the coronavirus disease 2019 (COVID-19)...	dental education;dental public health;infectio...	Journal Article;Research Support, Non-U.S. Gov't	Meng, L;Ma, B;Cheng, Y;Bian, Z	10.1177/0022034520962087	Prevention
4	32812051	J Antimicrob Chemother	The impact of sofosbuvir/daclatasvir or ribavi...	OBJECTIVES: Sofosbuvir and daclatasvir are dir...	NaN	Journal Article;Randomized Controlled Trial;Re...	Eslami, Gholamali;Mousavisl, Sajedeh;Radmanes...	10.1093/jac/dkaa331	Treatment

Figura 21: Visualización del Dataset

La mayoría de estos artículos pueden ser etiquetados con varias de estas etiquetas, sin embargo, alrededor del 76% ha sido etiquetado solo con una. Además de ello, LitCovid se actualiza diariamente con nuevos artículos relacionados con COVID-19 identificados en PubMed siendo un conjunto de datos primero en su clase, que brinda varias características únicas que mejoran la capacidad de búsqueda e interpretación de la literatura y que diferencias de otras herramientas como BIP4COVID19, covidscholar (https://covidscholar.org/) , iSearch COVID-19 Portfolio (https://icite.od.nih.gov/covid19/search/), CORD-19 y la Literatura mundial de la OMS sobre la enfermedad por coronavirus (https://www.who.int/emergencies / enfermedades / nuevo-coronavirus-2019 / investigación-global-sobre-nuevo- coronavirus-2019-ncov).

En particular, LitCovid identifica artículos acerca del COVID-19 en PubMed, esto quiere decir que artículos que contengan otros tipos de coronavirus como SARS o MERS están fuera del criterio de selección. Los artículos se afinan o depuran a diario, permitiendo que los usuarios puedan navegar de manera rápida por el entorno de la investigación de

temas acerca del COVID-19 con un alto nivel, geolocalización y organizaciones relacionadas.

La información afinada integra la búsqueda entre datos y conocimiento, lo que facilita el descubrimiento de conocimientos en aplicaciones posteriores, como la síntesis de pruebas y la reutilización de fármacos. Cabe señalar que LitCovid es una fuente de datos abierta, por lo que se puede descargar libremente para la investigación, así como para tareas de *Machine Learning*.

Una vez revisada la fuente de datos a utilizar en el desarrollo de la experimentación, se describe a continuación el análisis inicial realizado a los mismos, con el propósito de conocer su estructura y prepararlos para la fase de experimentación.

EDA.- El Análisis Exploratorio de Datos o EDA por sus siglas en inglés (*Exploratory Data Analysis*), brinda un análisis inicial de cómo están los datos antes de crear el modelo, este paso es importante, ya que al realizar la inspección del conjunto de datos se puede identificar qué distribución tienen sobre ciertas características, si existen datos que aporten a la construcción del modelo o que deban ser descartados, normalizados, etc.

Si el EDA no se lleva a cabo adecuadamente, se presentan problemas o dificultades en las etapas o fases siguientes durante la construcción del modelo de ML. A continuación se describen las tareas realizadas durante este proceso.

- Revisión de la cantidad de datos; consiste en determinar si se cuenta con los suficientes recursos para el procesamiento de los mismos, al realizar un conteo de filas y columnas el *dataset*, este consta de un total de 24,960 registros divididos entre nueve columnas, siendo posible el procesamiento de esta cantidad de información empleando la herramienta Google Colab.

- Identificación de filas o columnas en blanco, ya que si estos datos son parte de la construcción del modelo podrían introducir ruido y afectar el cálculo del modelo. En este caso se han identificado que no existen columnas en blanco, sin embargo se observa que no todos los datos cuentan con el campo *keywords, authors* y *doi*, sin embargo esto no implica un problema en el desarrollo de la experimentación, ya que el análisis se lo realiza por el campo *abstract*.
- Identificación del tipo de datos, esto se lleva a cabo con el propósito de identificar si los datos a analizar son únicamente texto, números, alfanuméricos, etc., en donde el campo *abstract* contiene tanto datos de texto como números y caracteres especiales, como direcciones o enlaces web presentes dentro del *abstract* y términos correspondientes al ámbito médico relacionados al COVID-19.
- Se realiza una visualización del corpus en una nube de palabras (representación gráfica de la frecuencia de las palabras en un texto), esta representación gráfica provee una descripción general del corpus de texto, permitiendo visualizar si el mismo contiene palabras acorde la temática de interés.
- Se analiza la distribución de los datos, esto permite revisar cómo se distribuyen los mismos con relación a cierta característica a lo largo del *dataset*.

La Tabla 2 a continuación, muestra los resultados obtenidos al aplicar los procedimientos descritos en ésta fase, al analizar las columnas y el tipo de datos que componen cada una de ellas.

Tabla 2: Cantidad de Registros por Columnas

#	Columna	Cant. No-Null	Dtype
0	Pmid	24,960 non-null	Int 64
1	Journal	24,960 non-null	Object
2	Title	24,960 non-null	Object
3	Abstract	24,960 non-null	Object
4	Keywords	18,968 non-null	Object
5	pub_type	24,960 non-null	Object
6	Authors	24,859 non-null	Object
7	Doi	24,406 non-null	Object
8	Label	24,960 non-null	Object

Tal como se puede observar en la Tabla 2, el *dataset* se compone de un total de 24,960 registros, distribuidos en ocho columnas, en las cuales la columna "*Pmid"*, es el identificador de cada registro, mientras que las otras siete columnas son objetos que se componen de caracteres de texto y numéricos. A continuación se muestra en la figura 21, la distribución de los datos por cada columna, en donde se puede identificar que el 75.99% de los registros tiene posee el atributo "*keywords*", el 99.59% posee el atributo "*authors*" y el 97.78% posee el atributo "*doi*", esto se puede observar de mejor manera en la siguiente figura.

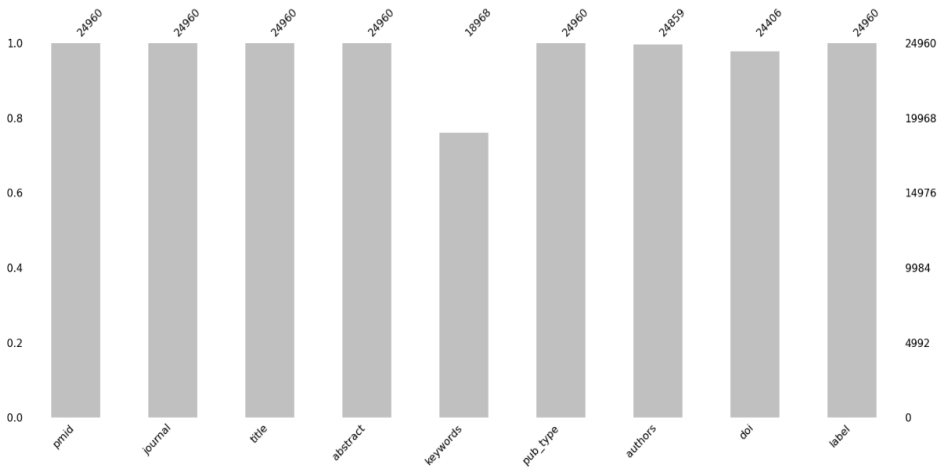


Figura 22: Distribución del dataset por columna.

Tal como se ha podido observar, todos los registros del *dataset* contienen el atributo *abstract* y el atributo *label* que son los atributos por los cuales se realizará la clasificación de los mismos, por tanto, se puede establecer a priori que se cuenta con la información necesaria para el desarrollo de la clasificación de los artículos.

La figura a continuación representa la nube de palabras de las que está compuesta el corpus de los artículos que se tiene en el *dataset,* en la cual se puede identificar por simple observación que la terminología más frecuente dentro del corpus a analizar está relacionada con palabras y términos empleados en la temática acerca del COVID-19.

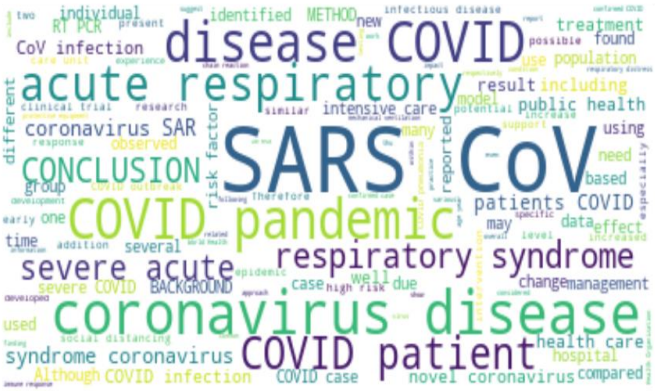


Figura 23: Representación del corpus en nube de palabras.

Realizando un análisis de frecuencia de palabras aplicado a la columna "*keywords*", se puede observar la frecuencia que tienen las palabras como "*coronavirus*", "*respiratory*", "*disease*", "*covid-19".*

Tabla 3. Frecuencia de Keywords

#	Palabra	Frecuencia
1	coronavirus	1501
2	respiratory	1305
3	acute	1244
4	disease	1226
5	syndrome	898
6	health	652
7	care	513
8	covid-19	492
9	2019	468
10	2	428

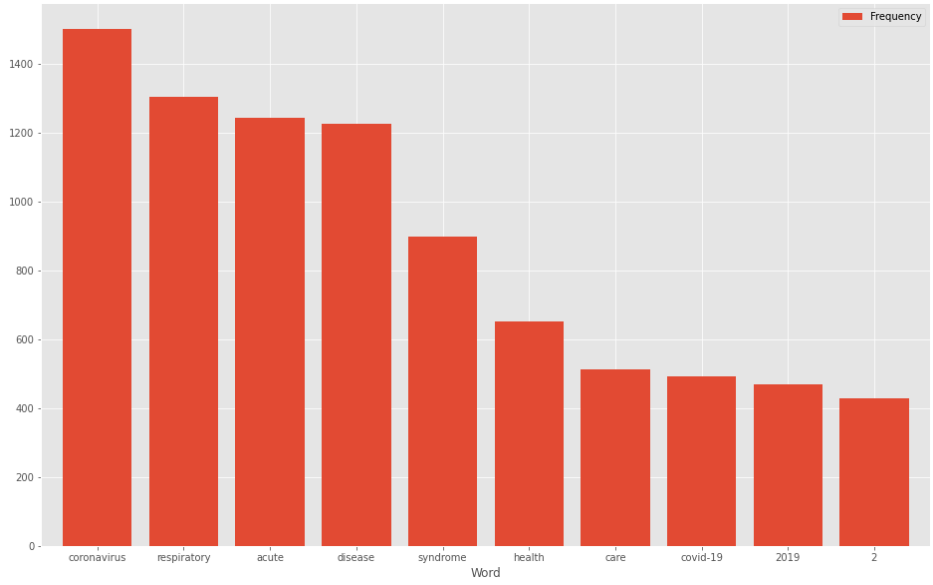


Figura 24: Representación Gráfica de Frecuencia de Palabras

Con esta información inicial, es posible continuar con la siguiente fase de tratamiento de los datos, esto con el fin de obtener los mejores resultados durante la experimentación, y evitar errores de cálculo del modelo a emplearse en las fases siguientes.

5.3. Tratamiento de los Datos

Dentro de esta fase se realizan las primeras operaciones sobre el *dataset* seleccionado, con el propósito de preparar los mismos para que el modelo de PLN a implementar obtenga mejores resultados. Es por ello que dentro de esta fase se lleva a cabo el preprocesamiento del *dataset*, este es un proceso importante antes de ejecutar cualquier tarea de PLN, para el presente estudio se abordan las siguientes subtareas de preprocesamiento: filtrado de datos, eliminación de *stopwords* o palabras vacías, eliminación de signos o caracteres especiales, y tokenización.

Debido a que el *dataset* se compone de información de varias fuentes, contiene diversas características, esto hace necesario estandarizar todas estas características, de manera que el modelo que va a realizar la predicción de clasificación de texto, contenga únicamente información que sea relevante.

Es fundamental destacar, que no existe un método estandarizado para llevar a cabo el preprocesamiento, ya que muchos de estos procedimientos pueden utilizarse dependiendo del tipo de tarea a realizar y del texto que vaya a ser analizado, puesto que podría ser el caso que para ciertas tareas de PLN, se requieran ciertos procedimientos de preprocesamiento y para otras tareas no. La Figura a continuación muestra de manera gráfica todo el proceso de tratamiento de los datos.

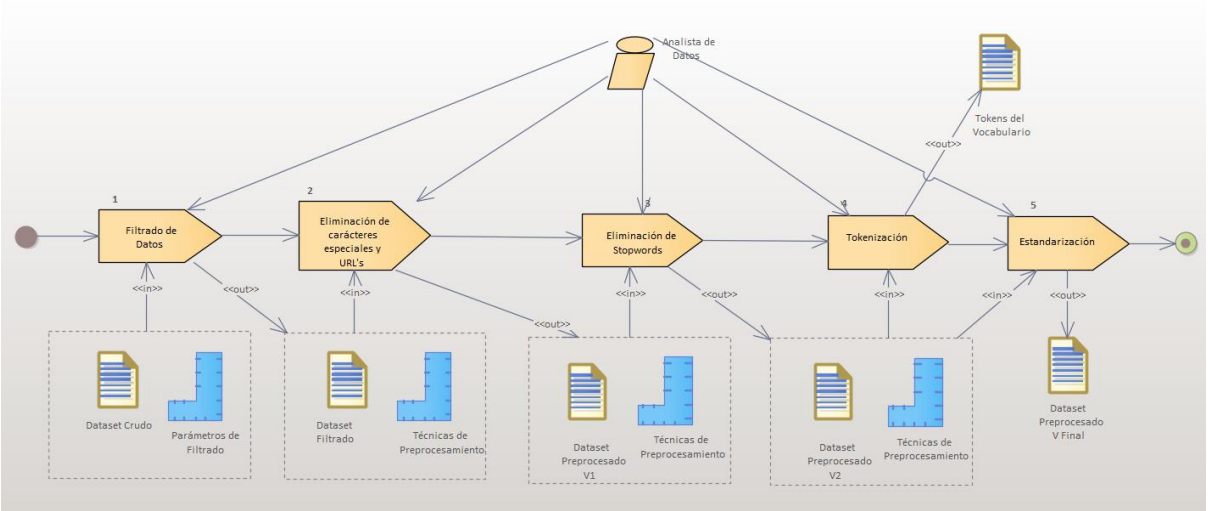


Figura 25: Tratamiento de los Datos

- **Filtrado de Datos.-** Para lograr el objetivo de realizar una clasificación se toma únicamente los artículos que contengan el *abstract* debidamente registrado, así como la etiqueta asignada al artículo de cada una de las clases, ya que pueden existir documentos que no contengan el *abstract*. El *dataset* está compuesto de 24,960 artículos tal como se mencionó en la fase anterior, una vez aplicado el filtro el resultante de artículos son 16,814, cuya distribución por etiqueta se muestra en la Figura 26 a continuación.

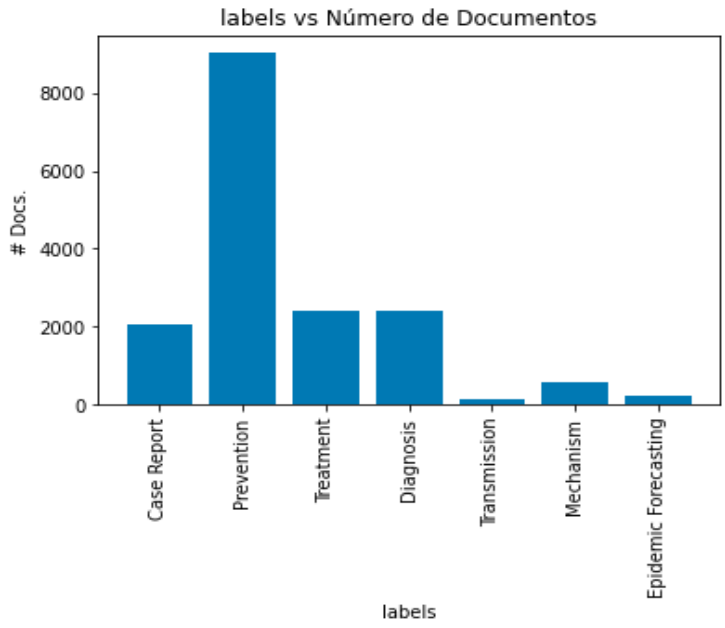


Figura 26: Distribución de los datos por etiquetas

La Tabla 4 a continuación muestra el porcentaje de las etiquetas que están distribuidas en el *dataset,* en donde se observa que existen clases muy minoritarias con respecto a otras.

Tabla 4: Distribución de los datos por etiqueta

Label	Cant.	%
Case Report	2,062	12.26%
Prevention	9,038	53.75%
Treatment	2,394	14.24%
Diagnosis	2,400	14.27%
Transmission	133	0.79%
Mechanism	585	3.48%
Epidemic Forecasting	202	1.20%

- **Eliminación de caracteres especiales y puntuación.-** La tarea de clasificación que se propone en el presente proyecto se basa en la aplicación de enfoque de *Word Embedding*, y en vista que estas representaciones vectoriales de texto no proporcionan representaciones para signos de puntuación y caracteres especiales, estos deben eliminarse.

  De la misma manera que los caracteres especiales, las direcciones web o *urls* no aportan información semántica o sintáctica para establecer relación del texto, por lo que debe eliminarse este tipo de contenido.

- **Eliminación de stopword o palabras vacías.-** El lenguaje natural está conformado de dos clases de palabras, las que contienen significado asociado entre ellas y palabras funcionales que no contienen ningún significado. Las *stopwords* o palabras vacías, son palabras utilizadas para identificar palabras funcionales y no necesitan ser parte del procesamiento de tareas de PLN por su bajo aporte al análisis. Las *stopwords* o palabras vacías son palabras funcionales que carecen de sentido en el contexto de tareas de clasificación de texto. Estas deben ser eliminadas con el propósito de reducir el tamaño del texto y analizar palabras que únicamente aportan al contexto dentro del corpus.

- **Tokenización.-** La tokenización no es más que el proceso de dividir el texto en unidades más pequeñas, se puede interpretar como dividir un conjunto de información en símbolos, es decir, los tokens o símbolos de una palabra son cada una de sus letras, de una frase un token sería una palabra, de un párrafo un símbolo o token podría ser toda una oración, etc. Esta tokenización, asigna un índice o identificador para cada uno de los elementos o unidades textuales, obteniendo de esta manera el diccionario

de cada palabra en el corpus. En la Figura 26 a continuación se muestra el proceso de tokenizar un texto, tal como se indica, cada palabra es identificada por un índice para identificarla dentro del diccionario de palabras dentro del texto.

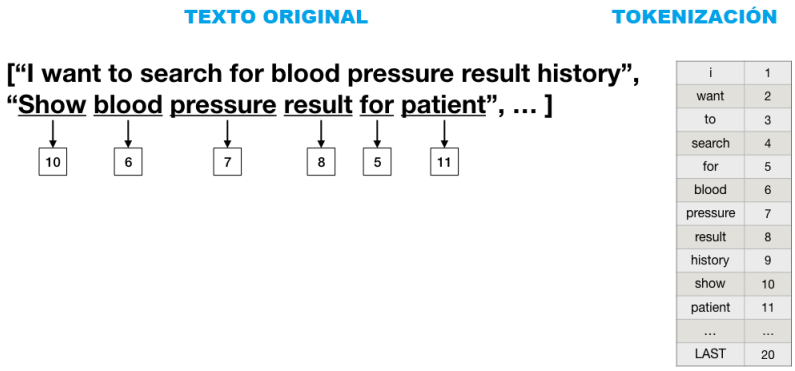


Figura 27: Proceso de tokenización del texto (https://www.researchgate.net/)

- **Estandarización del texto.-** Esta tarea consiste en dar un formato adecuado a todo el corpus de texto, para este caso se estandariza convirtiendo todas las palabras en minúsculas y evitar así diferenciar las mismas palabras escritas tanto en minúsculas como mayúsculas.

5.4. Modelado de los Datos

Dentro de esta fase se desarrollan los modelos de representación semántica de las palabras que componen el corpus de estudio, para este caso LitCovid. La representación del texto debe mantener la similitud semántica entre las palabras que componen el mismo, la representación por *Word Embedding* consiste en generar vectores que representen cada una de las palabras del corpus de manera que aquellas que sean similares entre sí semánticamente están cerca unas de las otras en el espacio vectorial.

Para que el texto pueda ser procesado computacionalmente, este debe ser transformado a una representación que el computador pueda entenderlo, es decir, de forma numérica, para ello, el texto debe pasar por un proceso para conseguir una representación lo más aceptable u óptima, con el propósito de que el modelo de *Deep Learning* realice de mejor manera las predicciones deseadas.

A pesar de que existen ya modelos pre-establecidos de vectores por *Word Embeddings* generalizados para tareas de PLN, en el presente estudio se construyen estos modelos a partir del corpus del *dataset* seleccionado, dichos modelos de contexto emplean las arquitecturas antes mencionadas: Word2Vec, FastText y Glove.

Durante la construcción de los mencionados modelos de contexto se tiene que establecer ciertos híper parámetros, los cuales afectan la calidad de entrenamiento así como la velocidad del mismo.

El caso de los modelos de Word2Vec y FastText se ha determinado los siguientes híper parámetros:

- **MIN_COUNT:** Este parámetro se utiliza para delimitar el número de veces que la palabra se repite dentro del corpus, este valor por defecto es 5, sin embargo, depende mucho del tamaño del conjunto de datos para entrenar.
- **SIZE:** Este parámetro determina el tamaño del vector resultante que va a representar cada palabra, para el presente estudio se configura con un tamaño de 300, ya que son los tamaños por defecto que maneja esta arquitectura, además hay que considerar que seleccionar un vector de mayor tamaño requerirá mayor cantidad de recursos y tiempo de procesamiento.

- **WINDOW:** La ventana o tamaño de ventana significa que la palabra del centro es la palabra objetivo y las demás son las palabras de contexto, para el presente estudio se ha considerado un valor de 5.
- **SG:** Este parámetro indica que arquitectura de Word2Vec se utiliza, para el caso de Skip-Gram es 1 y para CBOW es 0.

En el caso del modelo Glove, se define únicamente el parámetro NO_COMPONENT, el mismo que indica la dimensión que va a tener los vectores para cada palabra, equivale al híper parámetro SIZE, del modelo anterior.

Tal como se ha mencionado, los vectores de palabras resultantes tienen una dimensión 300, en donde cada una de las dimensiones del vector representa una relación que tiene la palabra con el resto de palabras del corpus, tal como se muestra en la siguiente figura, en donde se representan los vectores para las palabras "*pandemic*" y "*desease*":

	1	2	3	4	5	6							299	300
pandemic	0.68	-1.5	0.8	0,765	0.0034	-0.1	0.412	...	...	...	...	...	0,0056	0.8329
desease	-0.3467	0.8121	-0.0012	0.0038	0.665	-0.44567	-0.0867	...	...	...	...	...	0.4007	-0.0451

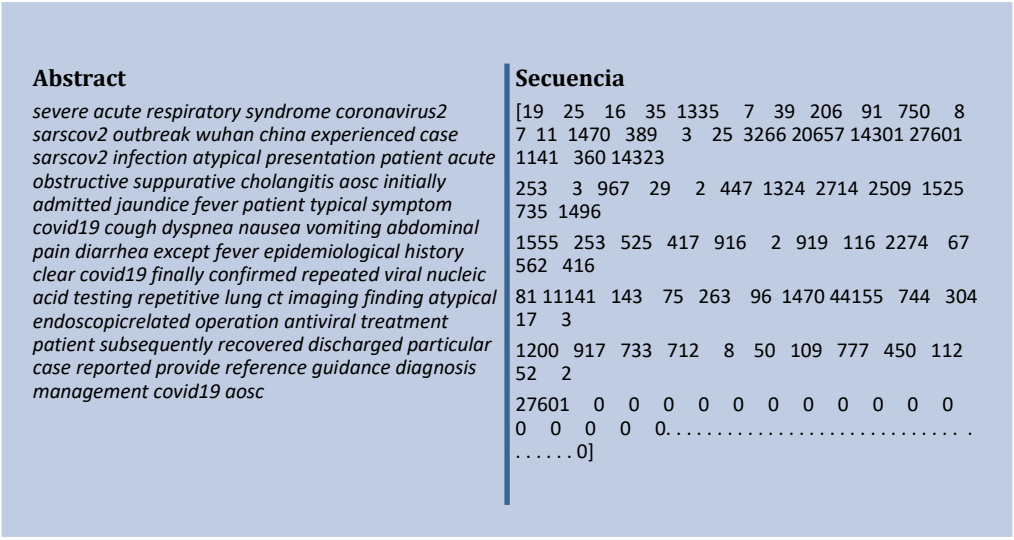
Figura 28: Representación de vectores de palabras

5.4.1. Generación de secuencias

La generación de secuencias consiste en transformar cada uno de los *abstracts* en un conjunto de secuencias del mismo tamaño. Para ello se transforma el corpus de texto en secuencias rellenas de identificadores de palabras, es decir, de los índices generados durante la tokenización. Cabe resaltar que el relleno de las secuencias se determina con base en el tamaño de la secuencia de mayor tamaño, por lo que, secuencias de menor tamaño son rellenadas con dígitos cero, hasta lograr un tamaño igual a la secuencia mayor, para el presente estudio, el artículo con mayor número de palabras dentro del *abstract* es

de 847 palabras o tokens, por lo que las secuencias de los *abstracts* de menor tamaño serán rellenados con ceros, hasta completar el tamaño mencionado.

En la siguiente figura se muestra un ejemplo de cómo se representa las secuencias para cada *abstract* del corpus.



Abstract	Secuencia
severe acute respiratory syndrome coronavirus2 sarscov2 outbreak wuhan china experienced case sarscov2 infection atypical presentation patient acute obstructive suppurative cholangitis aosc initially admitted jaundice fever patient typical symptom covid19 cough dyspnea nausea vomiting abdominal pain diarrhea except fever epidemiological history clear covid19 finally confirmed repeated viral nucleic acid testing repetitive lung ct imaging finding atypical endoscopicrelated operation antiviral treatment patient subsequently recovered discharged particular case reported provide reference guidance diagnosis management covid19 aosc	[19 25 16 35 1335 7 39 206 91 750 8 7 11 1470 389 3 25 3266 20657 14301 27601 1141 360 14323 253 3 967 29 2 447 1324 2714 2509 1525 735 1496 1555 253 525 417 916 2 919 116 2274 67 562 416 81 11141 143 75 263 96 1470 44155 744 304 17 3 1200 917 733 712 8 50 109 777 450 112 52 2 27601 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0. . . . . 0]

Figura 29: Ejemplo de representación de texto en secuencias

El ejemplo anterior muestra como el conjunto de palabras que componen uno de los *abstracts* del corpus, es representado mediante una secuencia de números o índices, que corresponden al token generado en el proceso de tokenizacion durante la fase anterior.

Los dígitos en cero corresponden al relleno, en donde habrá una cantidad de ceros hasta alcanzar el tamaño de la secuencia que contenga el mayor número de palabras dentro del *abstract*, tal como se mencionó anteriormente para el presente estudio se obtienen rellenos hasta completar un tamaño de secuencias igual a 847.

5.4.2. División de datos de prueba y entrenamiento

Una vez obtenidas las secuencias para cada uno de los *abstracts* del corpus, se procede a dividir el conjunto de datos de dichas secuencias obtenidas en datos de entrenamiento y prueba. Las secuencias o datos de entrenamiento son los que aportan a la identificación de patrones que se requieren para la clasificación, también en esta etapa se reducen las tasas de error para la etapa de prueba y evaluación del rendimiento del modelo. Algunos estudios, como (Khan et al., 2010), indican que para realizar el entrenamiento de modelos de ML, es necesario contar con un subconjunto representativo lo suficiente para evitar el sobre entrenamiento. Del conjunto de datos seleccionado, el 70% de ellos se consideran como datos de entrenamiento y el 30% como datos de prueba del modelo.

5.4.3. Matriz de Embeddings

Esta matriz actúa como una matriz de pesos, y se genera a partir de la representación semántica obtenida de las palabras mediante los modelos de *Word Embedding* previamente desarrollados, en donde el vector de la palabra n se ubica en la fila n de la matriz, mediante el índice generado en la tokenización. La figura 27 a continuación muestra de manera gráfica el proceso para generar la matriz de *embeddings*.

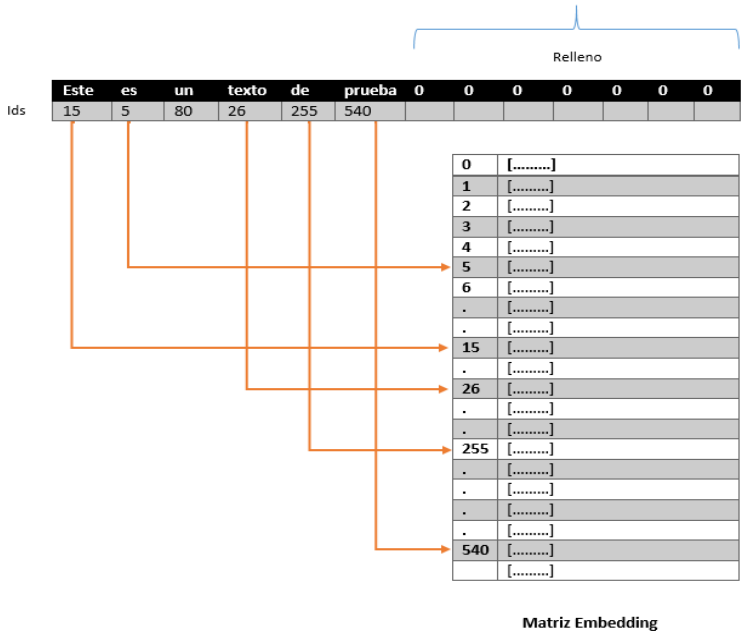


Figura 30: Matriz de Embeddings

Tal como se observa en la figura anterior, cada vector que representa a una palabra se ubica en la posición correspondiente al índice de esta palabra en la matriz, dando así una matriz de tamaño de 300 x 847.

5.4.4. Modelo de Clasificación

Por último, el modelo de red neuronal es creado con la matriz de *embeddings* que actúa como una matriz de pesos. El modelo de clasificación es desarrollado empleando la arquitectura de red neuronal LSTM, ya que como se mencionó anteriormente, este tipo de redes tienen un mejor desempeño al momento de procesar secuencias de datos, y predecir la salida, así como el de brindar mejores resultados en tareas de PLN.

La arquitectura de la red neuronal utilizada en el presente estudio consiste en una red neuronal recurrente LSTM bidireccional, la misma que consta de capas hacia adelante y hacia atrás que están conectadas juntas a la capa de salida, para así, conservar la

información contextual en ambas direcciones, lo que es precisamente útil para el caso de tareas de clasificación de texto.

Para entender de mejor manera la celda RNN toma como valor de entrada un estado oculto o vector, y un vector de la representación de la palabra, luego esta celda produce como salida el siguiente estado oculto, esta celda RNN tiene algunos pesos que se autoajustan mediante backpropagation de las pérdidas. Además, a todas las palabras se aplica la misma celda para que los pesos se compartan.

Una red neuronal RNN tradicional produce el mismo número de salidas para una secuencia de longitud determinada que se pueden vincular y luego esta pasarse a la capa de densidad hacia adelante. Por otra parte, la diferencia con las redes LSTM Bidireccionales es que toma la secuencia de entrada tanto en su forma inicial así como inversa (forward y backward), se aplica dos RNN en paralelo y se obtiene una salida del doble de tamaño de la entrada, una vez obtenida esta salida se envía a la capa de densidad para aquí aplicar una función softmax y obtener la clasificación mediante las probabilidades de cada clase o categoría (Abduljabbar et al., 2021).

Teniendo en cuenta estas consideraciones para este tipo de redes neuronales, se ha construido el modelo de clasificación para el presente estudio de la siguiente manera:

- La capa de *embedding* toma las secuencias como entrada y los vectores de palabras como pesos.
- Dos capas de red neuronal LSTM Bidireccional, que tiene como objetivo modelar el orden de palabras en una secuencia en ambas direcciones.
- Dos capas finales de densidad que lo que hacen es predecir la probabilidad de cada una de las distintas categorías.

- Debido a que es un problema multiclase, se emplea una función softmax, esta función devuelve valores entre 0 y 1, los cuales representan las probabilidades para cada categoría.

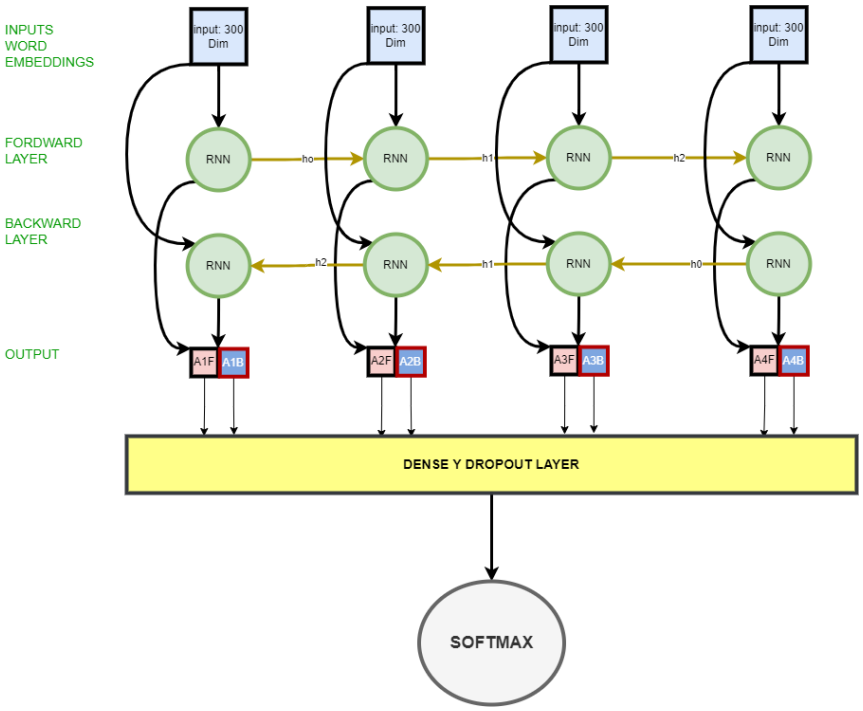


Figura 31: Modelo de Clasificación basado en una Red Neuronal LSTM Bidireccional.

De la misma manera que para la construcción de los modelos de contexto o *Word Embeddings* se establecieron ciertos hiperparámetros, así también se deben definir los hiperparámetros para el modelo de clasificación basado en redes neuronales. La siguiente tabla muestra los hiperparámetros utilizados en el modelo neuronal del presente estudio junto con una descripción de cada uno de ellos.

Tabla 5: Hiperparámetros del Modelo de Clasificación

Hyper Parámetro	Valor	Descripción
Neuronas en capas BiDirectional LSTM	32	Número de neuronas en cada una de las capas de la red neuronal
Número de capas	2	Número de capas ocultas de la red neuronal

Tamaño de vocabulario	83,439	Tamaño del vocabulario del corpus de texto, palabras únicas.
Tamaño de vectores	300	Tamaño del vector de cada palabra obtenido en el modelo
Dropout	0.2	Técnica para regularizar el sobreajuste en modelos de redes neuronales
Activación	Softmax	Función de activación brinda la probabilidad de cada clase en la salida

La Tabla anterior muestra los valores que se han tomado para la implementación del modelo, estos valores se tomaron en base a los recursos disponibles y complejidad de entrenamiento, por ejemplo se estableció un número de 32 neuronas en la red debido a que mientras más compleja es la red mayor cantidad de recursos se requiere llevando, a un mayor tiempo de procesamiento, con este valor se pudo obtener un modelo estable para la experimentación, ya que al tomar valores inferiores las métricas de evaluación caen por debajo del 55% de rendimiento.

El tamaño del vocabulario se define en base al proceso de tokenización, y está determinado por el número de palabras únicas de todo el corpus de texto a analizar.

El valor de *Dropout,* se estableció en 0.2, para optimizar el modelo durante el entrenamiento evitando un sobreajuste del mismo con un 20% de omisión en la activación de redes neuronales, ya que según estudios (Srivastava et al., 2014), los mejores valores para utilizar son de un 20% a un 50% en modelos de redes neuronales.

El tamaño de 300 dimensiones en los vectores se toma como tamaño óptimo en base al estudio realizado por (Pennington et al., 2014), en el cual se demuestra que los vectores con este tamaño, aportan mejores resultados al momento de capturar las relaciones semánticas entre las palabras.

La capa de activación emplea la función Softmax, que determina la salida, mediante la probabilidad de cada clase para la clasificación, esta función es propia para problemas de clasificación de tipo multiclase.

El modelo de clasificación se lo entrena con tres épocas, para identificar los patrones que el algoritmo debe entender y así lograr la clasificación deseada. Cabe señalar que se estableció el número de épocas en tres, debido al tiempo de procesamiento toma dicho entrenamiento por cada época, mientras más épocas se establezcan en el entrenamiento, mejores los resultados se pueden obtener, sin embargo, esto también implica mayor cantidad de tiempo y recursos.

5.5. Evaluación

En esta fase se evalúa los modelos generados desde el punto de vista de los resultados obtenidos en las pruebas realizadas a los modelos aplicados.

Aplicando las distintas métricas de evaluación a los modelos de *Deep Learning* vistos en el capítulo 6, se han obtenido los siguientes resultados sobre cada modelo, en los cuales se han aplicado las arquitecturas propuestas de *Word Embedding* mencionados en la fase de modelado, para los cuales el modelo que utiliza la arquitectura de FastText ha proporcionado ligeramente un mejor rendimiento una vez analizadas las métricas de *precision*, *recall* y *f1-score,* tal como se muestra en la figura a continuación.

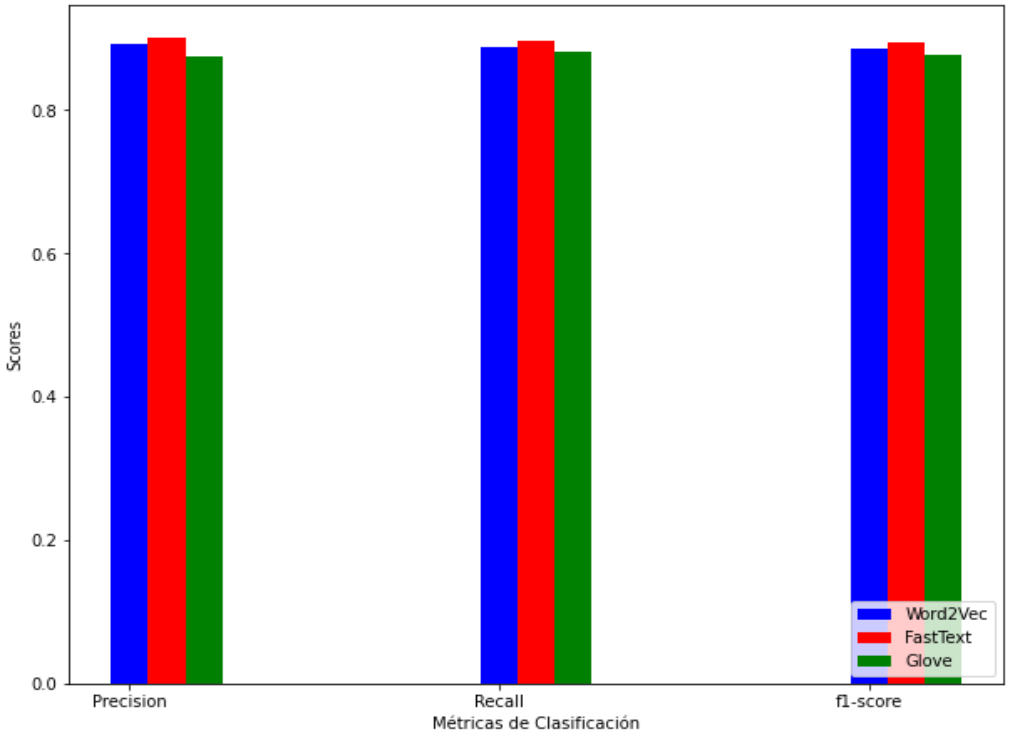


Figura 32: Métricas de evaluación de cada modelo

Como se observa en la figura 27, los tres modelos han logrado un rendimiento superior al 80%, esto se puede interpretar como un rendimiento aceptable en cuanto a las predicciones realizadas por cada modelo.

Analizando la exactitud o *accuracy* de los modelos que emplean Word2Vec y FastText está entre el 72% y 74% respectivamente, mientras que en el caso del modelo empleando Glove se encuentra en el 65%, siendo así el modelo FastText el que mejor rendimiento presenta al momento de realizar las predicciones de clasificación. La Figura 32 a continuación muestra el grado de exactitud de cada uno de los modelos.

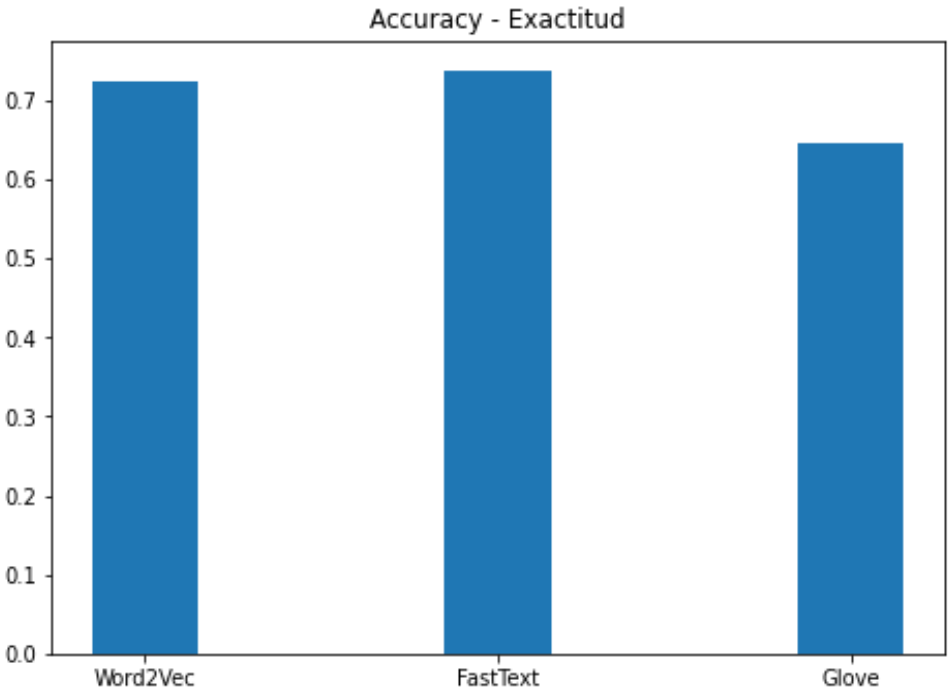


Figura 33: Evaluación de exactitud de cada modelo

Para determinar la exactitud o *accuracy* de cada uno de los modelos, se evalúa mediante la precisión equilibrada, que consiste en la media aritmética sobre la métrica *recall*, obtenida para cada clase, esta medida se aplica en vista de que se tiene el conjunto de datos desbalanceado, entonces la métrica *recall*, brinda el porcentaje de clasificaciones acertadas que el modelo es capaz de realizar.

Considerando que el problema de clasificación multiclase hay que tomar en cuenta las predicciones que los modelos realizan sobre cada una de las distintas clases, es por ello que la tabla a continuación presenta las métricas de evaluación que tiene cada modelo en cada clase.

Tabla 6: Métricas de evaluación por cada clase

Clase	precision			recall			f1-score		
	Word2Vec	FastText	Glove	Word2Vec	FastText	Glove	Word2Vec	FastText	Glove
Case Report	0,72	0,81	0,77	0,93	0,86	0,83	0,81	0,83	0,80
Diagnosis	0,88	0,87	0,83	0,82	0,89	0,85	0,85	0,88	0,84
Epidemic Forecasting	0,45	0,63	0,61	0,83	0,75	0,34	0,58	0,68	0,43
Mechanism	0,90	0,73	0,78	0,70	0,84	0,66	0,79	0,78	0,71
Prevention	0,97	0,96	0,95	0,93	0,94	0,94	0,95	0,95	0,94
Transmission	0,00	1,00	0,00	0,00	0,02	0,00	0,00	0,04	0,00
Treatment	0,83	0,84	0,81	0,86	0,87	0,90	0,85	0,86	0,85

Tal como se puede observar en la Tabla 6, las distintas métricas de evaluación para cada modelo en cada una de las clases, en donde se puede observar que la métrica de *precision,* del modelo FastText alcanza un 100% de predicciones para la clase *Transmission*, mientras que las métricas de *recall* y *f1-score* para esta misma clase alcanzan apenas el 2% y 4% respectivamente. Esto puede visualizarse de mejor manera a continuación en las siguientes figuras.

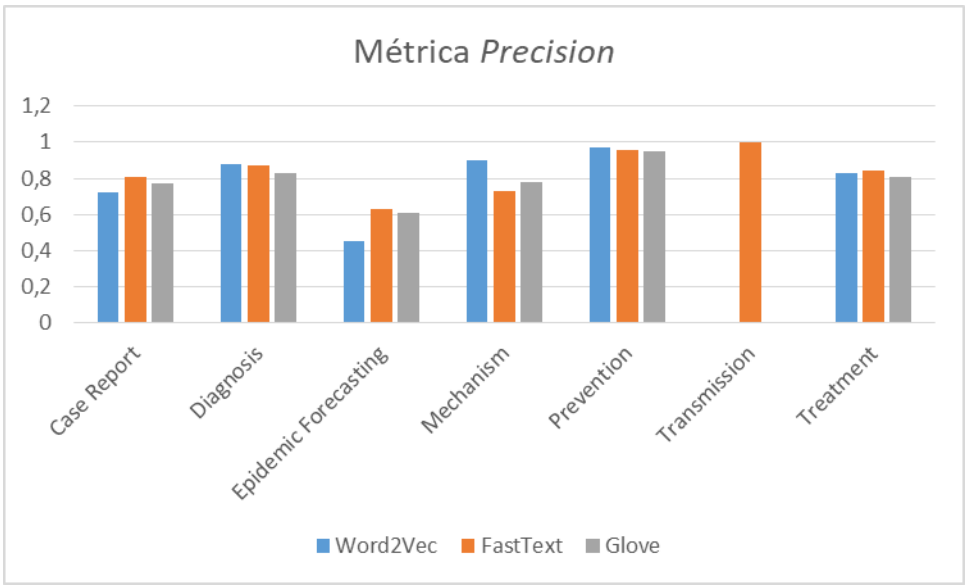


Figura 34: Métrica precision para cada clase

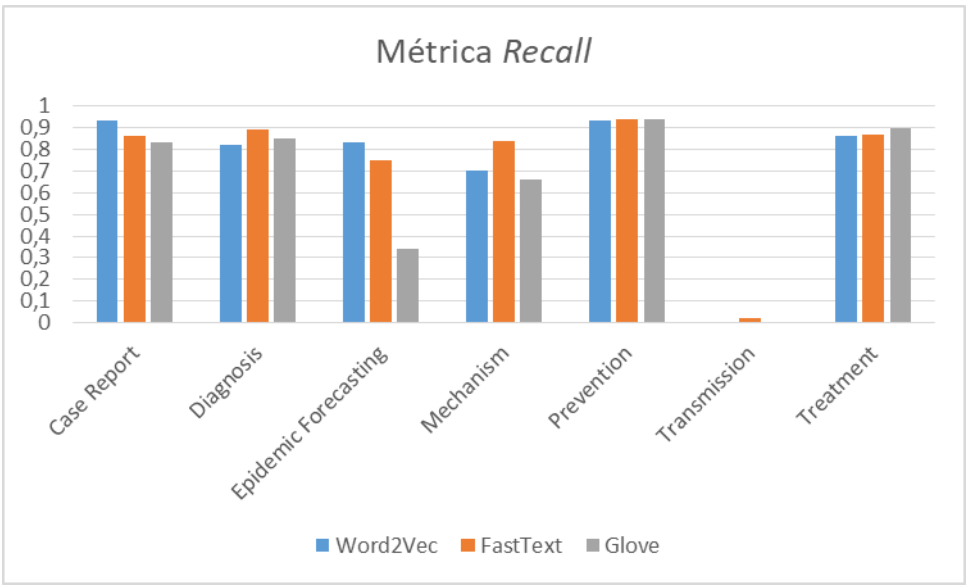


Figura 35: Métrica recall para cada clase

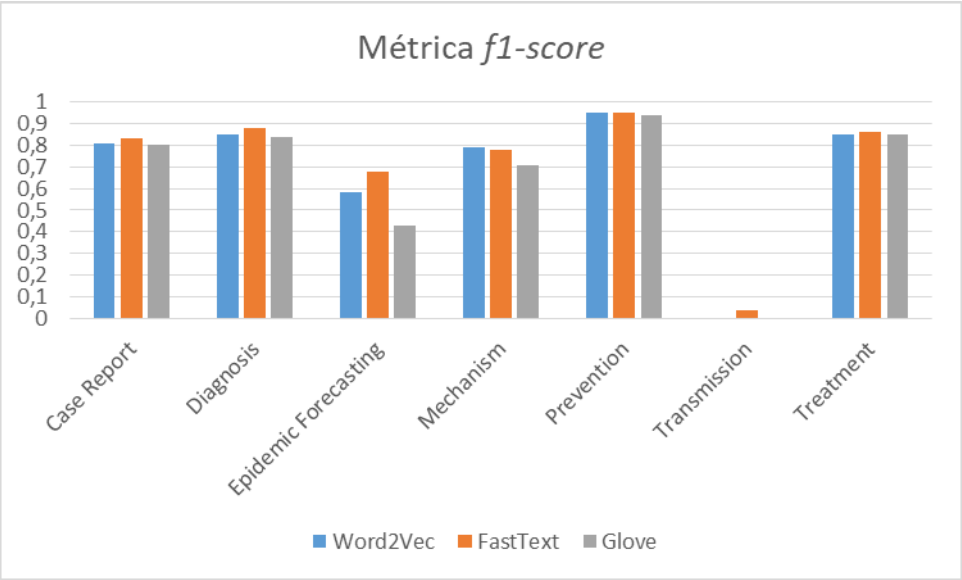


Figura 36: Métrica f1-score para cada clase

Como se observa en las figuras 34, 35 y 36, existen métricas de rendimiento con 0%, esto se debe a que el conjunto de datos está desbalanceado y existen clases muy mayoritarias en comparación con otras, por lo que los resultados de la clasificación realizada se ven afectados por este fenómeno. Tal como se observó en la etapa de preprocesamiento y análisis exploratorio de datos, la clase *Transmission* representa apenas el 0.79% de artículos etiquetados con esta clase, por lo que las predicciones al entrenar los modelos afectan a este tipo de clases.

Al analizar las matrices de confusión de cada modelo, en donde analizan los valores reales de cada clase vs. los valores predichos, se puede identificar nuevamente que las predicciones para la clase *Transmission,* obtiene un bajo grado de predicción en los tres modelos, esto debido al bajo porcentaje que tiene esta categoría dentro del *dataset*.

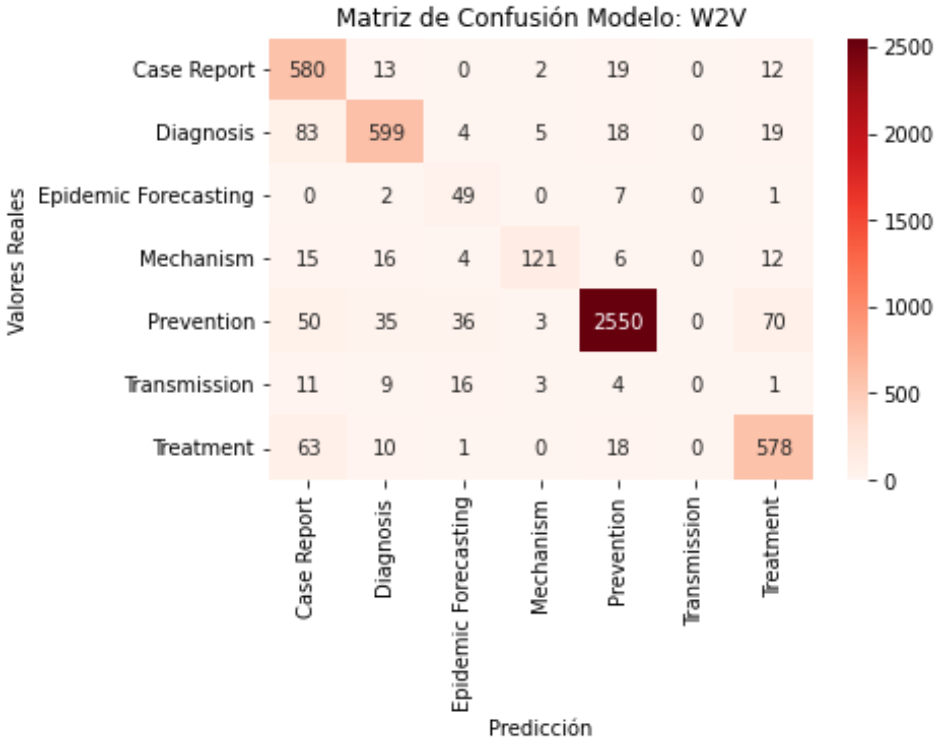


Figura 37: Matriz de Confusión - Modelo Word2Vec

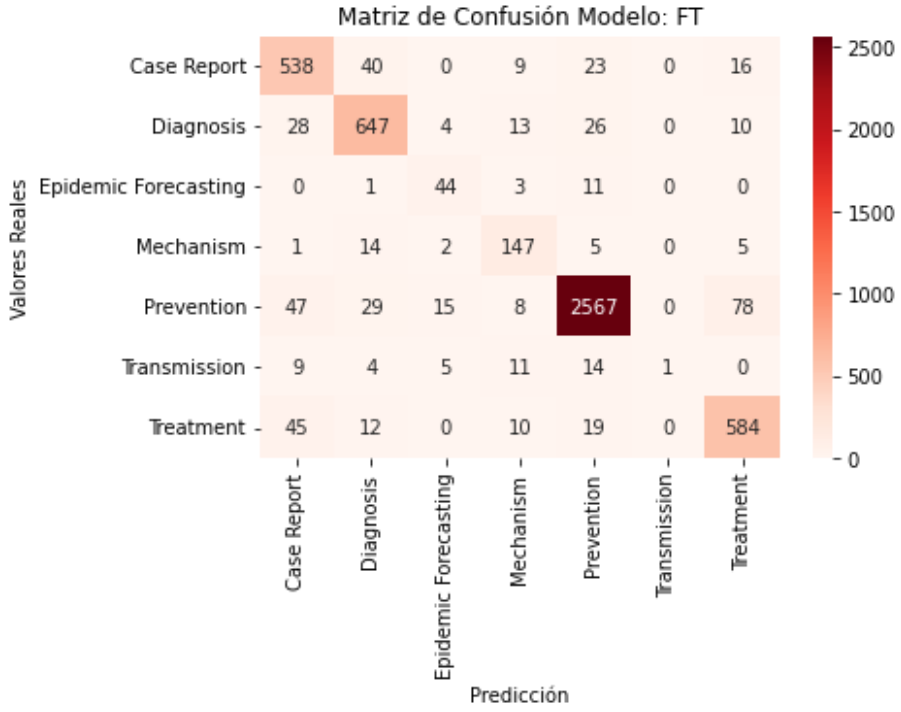


Figura 38: Matriz de Confusión - Modelo FastText

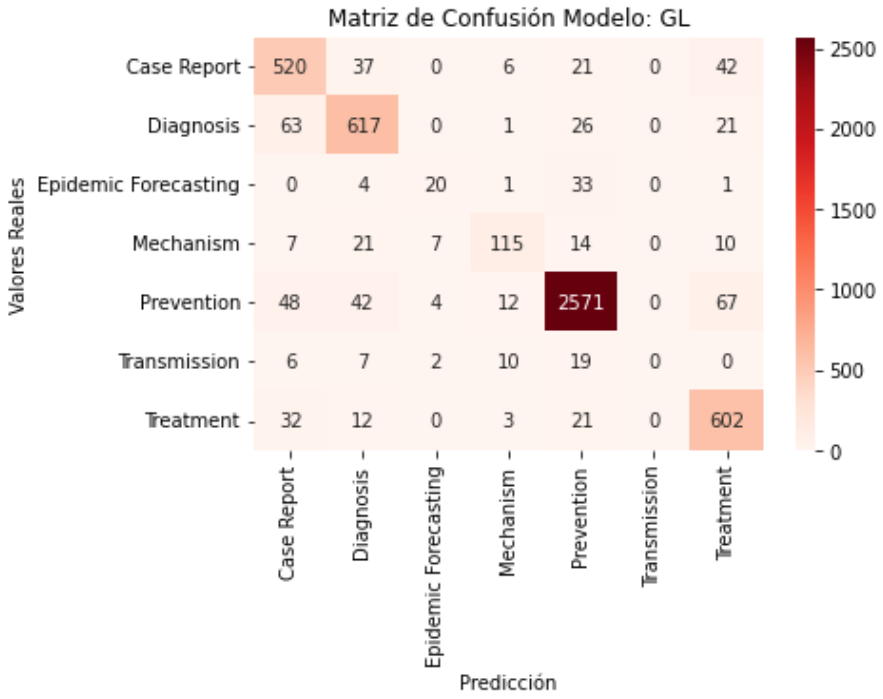


Figura 39: Matriz de Confusión - Modelo Glove

5.6. Despliegue e Implementación.

Hasta ahora, la experimentación desarrollada, ha cumplido con el objetivo de realizar una clasificación de los artículos que contienen temas relacionados con el COVID-19, por tanto, se deja como propuesta para una posible ampliación del trabajo, utilizar la técnica aplicada en el presente estudio de manera que pueda implementarse en una aplicación de minería de texto o afinar la técnica empleada.

6. Conclusiones.

El presente estudio mantuvo como objetivo principal el realizar una clasificación de artículos científicos con temáticas relacionadas con el COVID-19, aplicando enfoques de *Word Embeddings.* Luego de aplicar la metodología modelos propuestos en la experimentación se consigue cumplir con este objetivo propuesto.

Existen trabajos similares que han aplicado distintas metodologías y enfoques para realizar clasificación de contenidos acerca de la pandemia del COVID-19, sin embargo, no se encontraron trabajos en los cuales apliquen la metodología CRISP-DM, adaptándola a trabajos o estudios para *Text Mining*, como lo es aquí presentado, así como aplicar el enfoque de *Word Embedding* para representar el texto. Por lo que es destacable mencionar que es posible emplear metodologías de *Data Mining* en proyectos de *Text Mining*, consiguiendo resultados aceptables.

Se ha conseguido revisar el procedimiento y metodología para la clasificación de tipo multiclase de artículos científicos acerca del COVID-19, sobre el *dataset* LitCovid, aplicando el enfoque de *Word Embedding,* consiguiendo de esta manera representar las palabras y sus relaciones semánticas, dentro los *abstracts* en los artículos científicos. Si bien existen diversos clasificadores de texto vistos en el estado del arte como son LSA, Redes neuronales tipo CNN, TF-IDF, el presente estudio demuestra que el empleo de una representación del texto mediante *Word Embedding*, junto con un modelo de clasificación basado en redes neuronales tipo LSTM Bidireccionales que captura la información contextual completa tanto pasada como futura por su bidireccionalidad tanto hacia adelante como hacia atrás, proporcionando resultados aceptables en la clasificación del texto.

Luego de analizar los resultados al aplicar la metodología en cada uno de los modelos de clasificación propuestos, se obtuvo que la exactitud o *accuracy* se encuentra

entre el 65% al 74%, siendo el modelo que emplea FastText el que alcanzó el mayor porcentaje de exactitud mientras que el modelo que emplea Glove alcanzó la menor exactitud de los tres.

Cabe señalar, que si bien los resultados obtenidos demuestran que la clasificación de los artículos académicos de tipo multiclase es posible aplicando la metodología propuesta, podría ser factible mejorar el desempeño de los modelos implementados, aplicando otras técnicas de selección de datos para aminorar el problema que se presenta con el desbalance en la distribución de los mismos.

7. Recomendaciones.

Como recomendación a una posible ampliación del presente estudio, se podría considerar el analizar no solo el *abstract* del documento, sino secciones que contengan más contenido, como podría ser la introducción de cada artículo, el título del documento, entre otras, dependiendo la capacidad de procesamiento disponible, ya que como se ha indicado *Word Embedding* obtiene mejor relación semántica entre las palabras mientras mayor sea el corpus de texto sobre el cual se realice el análisis.

A su vez, tal como se observó que la clasificación se vio afectada debido al desbalance en la distribución de los datos, se plantea el aplicar otras técnicas de muestreo para afrontar este fenómeno, como puede ser técnicas de submuestreo o sobremuestreo o muestreo estratificado.

Bibliografía.

Abduljabbar, R. L., Dia, H., & Tsai, P.-W. (2021). *Unidirectional and Bidirectional LSTM Models for Short-Term Traffic Prediction*. https://doi.org/10.1155/2021/5589075

Adhanom Ghebreyesus, T. (2020). Alocución de apertura del Director General de la OMS en la rueda de prensa sobre la COVID-19 celebrada el 11 de marzo de 2020. In *Discursos del director General de la OMS* (Issue March 2020, pp. 1–4). https://www.who.int/es/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020

Aristovnik, A., Keržič, D., Ravšelj, D., Tomaževič, N., & Umek, L. (2020). Impacts of the COVID-19 pandemic on life of higher education students: A global perspective. *Sustainability (Switzerland)*, *12*(20), 1–34. https://doi.org/10.3390/SU12208438

Awasthi, R., Pal, R., Singh, P., Nagori, A., Reddy, S., Gulati, A., Kumaraguru, P., & Sethi, T. (2020). CovidNLP: A Web Application for Distilling Systemic Implications of COVID-19 Pandemic with Natural Language Processing. *MedRxiv*, 2020.04.25.20079129. https://doi.org/10.1101/2020.04.25.20079129

Beltagy, I., Lo, K., & Cohan, A. (2019). SCIBERT: A pretrained language model for scientific text. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 3615–3620. https://doi.org/10.18653/v1/d19-1371

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146. https://doi.org/10.1162/tacl_a_00051

Cárdenas, J. P., Olivares, G., & Alfaro, R. (2014). Clasificación automática de textos usando redes de palabras. *Revista Signos*, *47*(86), 346–364. https://doi.org/10.4067/S0718-09342014000300001

Chandrasekaran, B., & Fernandes, S. (2020). Target specific mining of COVID-19 scholarly articles using one-class approach. *Diabetes Metab Syndr.*, *14(4)*(January), 337–339.

Chatsiou, K. (2020). *Text Classification of Manifestos and COVID-19 Press Briefings using BERT and Convolutional Neural Networks*. http://arxiv.org/abs/2010.10267

Christopher D.Manning. (2021). Speech and Language Processing: An introduction to natural language processing. *SPEECH and LANGUAGE PROCESSING An Introduction to Natural Language Processing Computational Linguistics and Speech Recognition*, 1–18. http://www.cs.colorado.edu/~martin/slp.html

Daud, A., Khan, W., & Che, D. (2017). Urdu language processing: a survey. *Artificial Intelligence Review*, *47*(3), 279–311. https://doi.org/10.1007/s10462-016-9482-x

De, C., Guridi, G., Tutor, M., Barbero, Á., Ponente, J., Ramón, J., & Ibero, D. (2017). *MODELOS DE REDES NEURONALES RECURRENTES EN*.

Dynomant, E., Lelong, R., Dahamna, B., Massonnaud, C., Kerdelhué, G., Grosjean, J., Canu, S., & Darmoni, S. (2019). Word embedding for French natural language in healthcare: A comparative study. *Studies in Health Technology and Informatics*, *264*, 118–122. https://doi.org/10.3233/SHTI190195

González Barba, J. Á. (2017). *Aprendizaje profundo para el procesamiento del lenguaje natural*. https://riunet.upv.es/bitstream/handle/10251/86279/González - Aprendizaje profundo para el procesamiento del lenguaje natural.pdf?sequence=1

Harris, Z. S. (2015). Distributional Structure.

Http://Dx.Doi.Org/10.1080/00437956.1954.11659520, *10*(2–3), 146–162. https://doi.org/10.1080/00437956.1954.11659520

Jelodar, H., Wang, Y., Orji, R., & Huang, S. (2020). Deep Sentiment Classification and Topic Discovery on Novel Coronavirus or COVID-19 Online Discussions: NLP Using LSTM Recurrent Neural Network Approach. *IEEE Journal of Biomedical and Health Informatics*, *24*(10), 2733–2742. https://doi.org/10.1109/JBHI.2020.3001216

Jimenez Gutierrez, B., Zeng, J., Zhang, D., Zhang, P., & Su, Y. (2020). *Document Classification for COVID-19 Literature*. 3715–3722. https://doi.org/10.18653/v1/2020.findings-emnlp.332

Joachims, T. (2002). Learning to Classify Text Using Support Vector Machines. In *Learning to Classify Text Using Support Vector Machines*. Springer US. https://doi.org/10.1007/978-1-4615-0907-3

Khan, A., Baharudin, B., Hong Lee, L., & khan, K. (2010). *A Review of Machine Learning Algorithms for Text-Documents Classification*. https://doi.org/10.4304/jait.1.1.4-20

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, *36*(4), 1234–1240. https://doi.org/10.1093/BIOINFORMATICS/BTZ682

Maguiña Vargas, C., Gastelo Acosta, R., & Tequen Bernilla, A. (2020). El nuevo Coronavirus y la pandemia del Covid-19. *Revista Medica Herediana*, *31*(2), 125–131. https://doi.org/10.20453/rmh.v31i2.3776

Middle East respiratory syndrome coronavirus (MERS-CoV). (2022). https://www.who.int/health-topics/middle-east-respiratory-syndrome-coronavirus-mers#tab=tab_1

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013, January 16). Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*. https://arxiv.org/abs/1301.3781v3

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2020). *Deep Learning Based Text Classification: A Comprehensive Review*. *1*(1), 1–42. http://arxiv.org/abs/2004.03705

Mulyar, A., Uzuner, O., & McInnes, B. (2021). MT-clinical BERT: scaling clinical information extraction with multitask learning. *Journal of the American Medical Informatics Association : JAMIA*, *28*(10), 2108–2115. https://doi.org/10.1093/JAMIA/OCAB126

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1532–1543. https://doi.org/10.3115/v1/d14-1162

Rasmy, L., Xiang, Y., Xie, Z., Tao, C., & Zhi, D. (2021). Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *Npj Digital Medicine*, *4*(1). https://doi.org/10.1038/s41746-021-00455-y

Thompson, L. (2003). Inicio de una nueva epidemia, SARS. *5. Holmes K.U. SARS-Associated Coronavirus. N Eng J Med*, *14*(2), 1948–1951. www.who.int;

Torres-Salinas, D. (2020). Daily growth rate of scientific production on covid-19. Analysis in databases and open access repositories. *Profesional de La Informacion*, *29*(2). https://doi.org/10.3145/epi.2020.mar.15

Trewartha, A., Dagdelen, J., Huo, H., Cruse, K., Wang, Z., He, T., Subramanian, A., Fei, Y., Justus, B., Persson, K., & Ceder, G. (2020). *COVIDScholar: An automated COVID-19 research aggregation and analysis platform*. https://arxiv.org/abs/2012.03891v1

Wang, L. L., & Lo, K. (2021). Text mining approaches for dealing with the rapidly expanding literature on COVID-19. *Briefings in Bioinformatics*, *22*(2), 781–799. https://doi.org/10.1093/bib/bbaa296

Wirth, R. (2000). CRISP-DM : Towards a Standard Process Model for Data Mining. *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, *24959*, 29–39.

Beldarraín, R. E. C. (2020). La información científica confiable y la COVID-19 Reliable scientific information and COVID-19. *Revista Cubana de Información En Ciencias de La Salud*, *31*(3), 1–6. http://orcid.org/0000-0003-4448-8661

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1532–1543. https://doi.org/10.3115/v1/d14-1162

Srivastava, N., Hinton, G., Krizhevsky, A., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Beldarraín, R. E. C. (2020). La información científica confiable y la COVID-19 Reliable scientific information and COVID-19. *Revista Cubana de Información En Ciencias de La Salud*, *31*(3), 1–6. http://orcid.org/0000-0003-4448-8661