

Análisis de Sentimientos sobre Reseñas de Películas

Fernando, Doddoli Lankenau. Alejandra G., Olivas Altamirano. Ruben A., Robles Leal. Adrián A., Salgado Martínez.

Instituto Tecnológico y de Estudios Superiores de Monterrey

ABSTRACT Hoy en día, las empresas manejan enormes cantidades de información, las cuales, por su volumen, permiten la creación de modelos de inteligencia artificial con aprendizaje de máquina para realizar funciones valiosas como clasificación y predicción de nueva información. A través de la implementación de una red neuronal recurrente Long Short Term Memory, esta investigación destaca la relevancia del uso de la inteligencia artificial en el procesamiento de lenguaje natural, y el análisis y clasificación de sentimiento, herramienta que puede proporcionar información valiosa para la toma de decisiones en la industria donde se manejen grandes cantidades de datos de texto y se necesite hacer una clasificación de sentimiento. Las redes LSTM son una variante de las redes neuronales recurrentes (RNN) que han demostrado ser efectivas en el procesamiento de secuencias de datos, como la clasificación de sentimiento de texto.

En el presente trabajo se realizó un modelo de machine learning a partir de una base de datos de reseñas de películas que considera opiniones positivas y negativas. Se muestra el análisis exploratorio y resultados preliminares. Posteriormente se propone un método que incluye la preparación previa de los datos, arquitectura, configuración y capas del modelo y se definen sus alcances. De esta manera, después se presenta la definición de las pruebas a realizar para optimizar o ajustar los parámetros, a su vez se establecen pruebas para las restricciones del modelo, tales como una reseña en español y con palabras mal escritas. Finalmente se exponen los resultados del desempeño del modelo considerando las etapas de entrenamiento, validación y predicción. En esta última etapa, además de analizar la precisión y pérdida del modelo, se determinan sus limitantes a partir de una matriz de confusión. Por último, se reflexiona sobre el impacto de los múltiples parámetros sobre el modelo.

Esta investigación demuestra cómo, con el uso de una un modelo de inteligencia artificial con aprendizaje de máquina a través de una red neuronal con arquitectura Long Short Term Memory se logra clasificar el sentimiento de una reseña como positiva o negativa con una precisión del 85%.

KEYWORDS algoritmo adam, análisis de sentimientos, binary cross entropy, clasificación de sentimientos, descenso de gradiente estocástico, inteligencia artificial, long short term memory, machine learning, procesamiento de lenguaje natural, redes neuronales, redes neuronales recurrentes.

I. INTRODUCCIÓN

Considerando la creciente demanda de aplicaciones de inteligencia artificial en el mundo empresarial, la inteligencia artificial (IA) ha experimentado un rápido crecimiento y ha ganado una importancia cada vez mayor en la investigación y la industria. En el año 1943 se propuso el primer modelo matemático para la creación de una red neuronal, y desde entonces, la IA ha evolucionado de manera significativa gracias a los nuevos algoritmos que han ido surgiendo de aprendizaje automático, y la creciente capacidad de procesamiento de las computadoras. [1]

Actualmente existen múltiples técnicas de machine learning, procesamiento de lenguaje natural y análisis de sentimientos, que son aplicadas en diversos campos, principalmente en el ámbito empresarial donde la IA ha demostrado ser útil en la toma de decisiones, el análisis de datos y la automatización de procesos.

En el campo específico de la clasificación de sentimientos, varias técnicas de aprendizaje automático y procesamiento de lenguaje natural se utilizan para analizar y categorizar el lenguaje humano en textos escritos, incluyendo la clasificación de textos, el modelado de lenguaje y la traducción automática. [2]

En este artículo, se propone un modelo de IA para la clasificación de sentimientos en reseñas de películas utilizando una red neuronal con arquitectura Long Short Term Memory. La finalidad del modelo es clasificar reseñas de películas como positivas o negativas con una precisión de al menos el 80% de los casos.

Se discutirán las diferentes capas que conforman a nuestro modelo LSTM y su función en el procesamiento de los datos de entrada.

II. FUNDAMENTO

A. RED NEURONAL

Las redes neuronales utilizan nodos o neuronas artificiales interconectados en capas para enseñar a computadores a procesar datos—aprendizaje profundo [3].

La Fig. 1 muestra el funcionamiento de una neurona, se observa como recibe una entrada x y una salida a (activación)

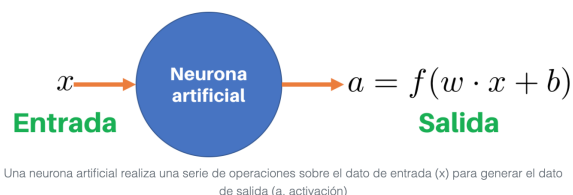


FIGURA 1. Diagrama de Funcionamiento de una Neurona [4].

A cada entrada se le asigna un peso. La salida de la neurona es el resultado de dos operaciones. La primera es la transformación, que toma el dato de entrada y aplica la

operación $wx + b$. Los parámetros de w y b se calculan usando el algoritmo del Gradiente Descendente durante el entrenamiento. La segunda, aplica una función de activación, una función matemática que puede variar, la más común es la función sigmoide, representada de la siguiente manera.

$$f(x) = \frac{1}{1 + \exp(-\sum_i w_i x_i + w_0)}$$

ECUACIÓN 1. Función sigmoide usada por el algoritmo de Gradiente Descendente.

La arquitectura de una red neuronal se divide en tres capas: entrada, oculta, y salida.

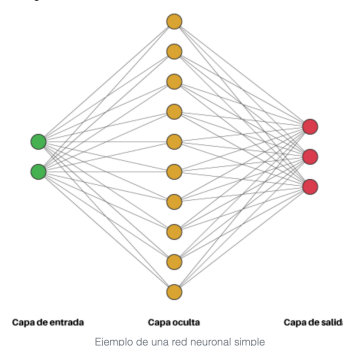


FIGURA 2. Diagrama de Funcionamiento de una Red Neuronal [5]

Capa de entrada: Los nodos en la capa de entrada toman los datos, los procesan y los pasan a la siguiente capa. Cada neurona de la capa de entrada pasa la información a todas las neuronas de la siguiente capa, que es una capa oculta.

Capa oculta: Las neuronas de las capas ocultas toman su entrada de la capa de entrada o de otras capas ocultas. Un modelo de red neuronal puede tener múltiples capas ocultas, con un número de neuronas diferente por capa. Cada capa tiene una función de activación, por ejemplo, las redes LSTM suelen tener la función logística en las conexiones de recurrencia y la función tanh en la salida.

Capa de salida: La capa de salida da el resultado de todo el procesamiento de datos que hace la red neuronal. Esta puede tener uno o varios nodos, dependiendo de la salida que estemos buscando. Por ejemplo, para una clasificación binaria, como es el caso de este reporte, la capa de salida tendrá un nodo que dará como resultado 1 o 0.

Es importante mencionar que las redes neuronales también tienen una función de costo, que ayuda a ajustar el modelo en cada iteración y minimizar el error. Para problemas de regresión estas son: Mean Absolute Error, Mean Squared Error, Root Mean Squared Error. Para problemas de clasificación estas son: Binary Cross-Entropy, y Categorical Cross-Entropy. Para este reporte utilizaremos Binary Cross-Entropy, Esta función de costo mide qué tan distante es el resultado de la clase binaria a la que

corresponde, 0 o 1. Su función se muestra a continuación, donde la y_k es el valor de la etiqueta del dato x_k

$$L = -\frac{1}{N} \sum_{k=1}^N [y_k \cdot \log(\hat{y}_k) + (1 - y_k) \cdot \log(1 - \hat{y}_k)]$$

ECUACIÓN 2. Función de costo Binary Cross-Entropy.

B. REDES NEURONALES RECURRENTE CON ARQUITECTURA LSTM

Partiendo del funcionamiento de una red neuronal, una red neuronal recurrente tiene dos entradas: el dato de entrada y el estado anterior—este estado anterior está oculto. Y se proporcionan dos salidas: la predicción y el valor ajustado del estado oculto [6]. Una red neuronal recurrente con arquitectura LSTM se diferencia al poder mantener una memoria de entradas de corto y largo plazo.

Además, a diferencia de una red neuronal básica, al implementarse, una arquitectura LSTM utiliza una capa de embedding para procesar los datos. Siendo la primera capa del modelo y la que hace el preprocesamiento de datos. Esta capa transforma las palabras en vectores de números, a estos vectores se les llama embeddings. El proceso para llevar esto a cabo implica tomar un texto y utilizando una matriz de pesos, asigna cada palabra de entrada a su correspondiente vector. Produciendo una matriz de vectores, un vector para cada palabra en el vocabulario. El resultado de hacer este preprocesamiento es lograr que el modelo LSTM entienda mejor el significado de las palabras en su debido contexto [7].

C. BASE DE DATOS

Tras la obtención de los datos, los cuales se encuentran disponibles de manera abierta como parte de los datasets accesibles en Kaggle, nos percatamos de que estos están compuestos por 50,000 reseñas escritas en inglés almacenadas en un archivo CSV. Asimismo, cada fila perteneciente a esta base de datos está compuesta por una tupla con el texto de la reseña y la cadena de texto “positive” o “negative” de acuerdo al sentimiento asociado a dicha reseña.

Cuantitativamente, como se ve en la Fig. 3 las reseñas tienen una longitud promedio de 1,309 caracteres, y una mediana de 970. Además, la reseña más corta tiene una longitud de tan solo 32 caracteres, mientras que la más larga es de más de 13,700.

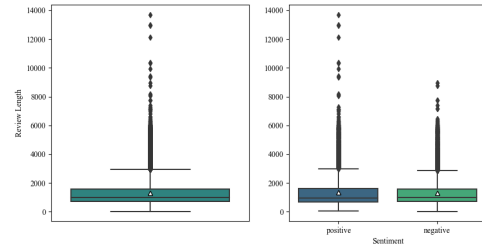


FIGURA 3. Medidas de distribución de longitudes de las reseñas globales, y por sentimiento.

La Fig. 4 muestra que las reseñas se encuentran distribuidas de forma perfectamente uniforme, en otras palabras el 50% de estas son de sentimiento positivo y el otro 50% negativo.

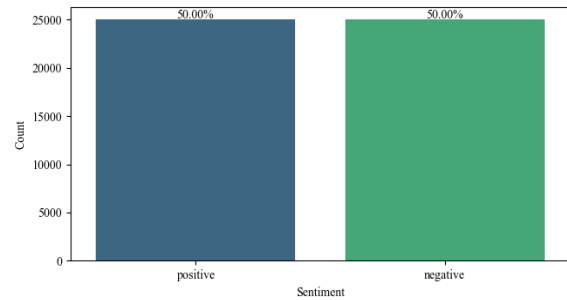


FIGURA 4. Histograma con distribución de sentimiento de las reseñas.

Analizando más a fondo el contenido de texto de las reseñas, buscamos explorar la frecuencia del uso de ciertas palabras en las reseñas, por lo que se utilizaron herramientas de análisis de texto para remover las palabras más comunes del idioma inglés y contar sus apariciones en todas las reseñas, y en reseñas de cada uno de los dos sentimientos.

Un aspecto interesante acerca de estas palabras más utilizadas en el total de las reseñas es que al cuantificar su uso por sentimiento nos encontramos que palabras tales como “good” aparecen en casi misma medida tanto en reseñas positivas como negativas, pero palabras como “story” y “even” tienden a aparecer casi en su totalidad exclusivamente en reseñas positivas o negativas de manera exclusiva, tal y como se muestra en la Fig. 5.

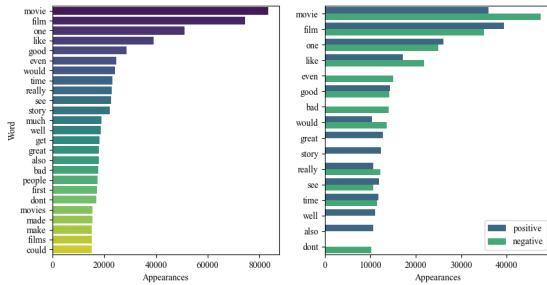


FIGURA 5. Palabras más usadas en reseñas globales, y uso de algunas de las mismas palabras por sentimiento.

Observando entonces el uso de palabras por sentimiento de reseña por separado, nos damos cuenta en la Fig. 6 que palabras como “movie” y “film” se usan muy frecuentemente tanto en positivas como en negativas, pero palabras como “plot” o “acting” se usan mucho más frecuentemente en reseñas negativas, mientras que “time” y “love” se usan más en positivas.

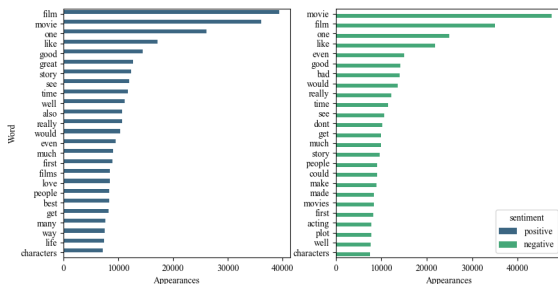


FIGURA 6. Palabras más usadas en reseñas por sentimiento.

III. MÉTODO PROPUESTO

A. DESCRIPCIÓN GENERAL

Se soluciona la problemática de clasificar una secuencia de palabras como positiva o negativa utilizando un modelo de inteligencia artificial con aprendizaje de máquina a través de una red neuronal Long Short Term Memory (LSTM).

Dado que es muy complicado definir una fórmula que haga un análisis de sentimiento efectivo en cada token de palabra, un modelo de aprendizaje de máquina es una excelente herramienta para resolver el problema [8]. Así mismo, se utiliza una red neuronal por sus capacidades de aprendizaje no lineal y generalización. La capacidad de una red neuronal de aprender relaciones no lineales permite aprender patrones y relaciones complejas entre palabras y sentimientos expresados en una reseña. La capacidad de generalización de redes neuronales permiten generalizar bien nuevos datos de entrada. Estas capacidades mencionadas significan que el modelo podrá aprender y

clasificar correctamente nuevos textos de reseña que no ha visto antes [9].

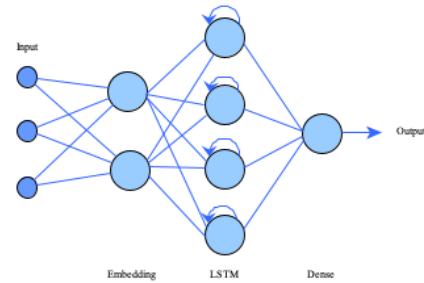


FIGURA 7. Ilustración de modelo.

A. DATOS DE ENTRADA

Para la fase de entrenamiento, se utiliza el 50% del conjunto de datos originales, de los cuales el 20% de ellos se utilizan para las validaciones. El 50% restante de ellos se utiliza en la fase de pruebas.

B. PREPARACIÓN DE DATOS

Los datos de entrada se encuentran estandarizados al ser escritos en inglés y estar codificados en UTF-8, por lo que no se requiere hacer alguna transformación o filtramiento en cuanto al texto de las reseñas para homogeneizarlas. Asimismo, no se encuentra ninguna tupla con información faltante. Sin embargo, una excepción al texto natural que compone al texto de las reseñas es la inclusión en numerosas ocasiones, incluso en la misma reseña, de la cadena “
”, que indica un salto de línea en HTML.

Tomando esto en cuenta, con el propósito de aumentar la probabilidad de obtener un mejor modelo, se sigue una práctica mencionada en literatura relacionada al análisis de lenguaje natural, que consiste en hacer una “limpieza” previa de los datos para eliminar palabras vacías y puntuación. Dichas palabras vacías (o stopwords en inglés) frecuentemente no proveen de información útil, por lo que “son comúnmente eliminadas del texto antes de aplicar técnicas de análisis de texto” [10]. Cabe destacar que un factor importante por el que decidimos eliminar a estas palabras es porque no se busca replicar o analizar la forma y estructura del texto original, que es en la situación en la que se destaca la importancia de mantener estas palabras [11].

La preparación de datos consistió en filtrar las palabras determinadas como stop words definidas en la librería Natural Language Toolkit (NLTK) de Python, que incluye palabras como “you”, “or”, “is” y “the”, así como la eliminación de símbolos de puntuación definidos en la librería string, que forma parte de las librerías estándares de Python, adicionalmente eliminando el código HTML restante por la cadena “
”.

C. ARQUITECTURA DEL MODELO

El modelo de red neuronal recurrente (RNN) utiliza una arquitectura Long Short Term Memory (LSTM).

D. CONFIGURACIÓN DEL MODELO

El modelo se configura con una función de pérdida Binary Cross Entropy porque es adecuada para resolver problemas de clasificación. Por otro lado, utilizamos el optimizador adam, que está basado en una optimización del algoritmo de Descenso de Gradiente Estocástico (SGD). Finalmente, tomamos como valor inicial un número de epochs de 5. Se decide utilizar estos parámetros como partida de forma arbitraria, sin embargo dentro de la sección de parámetros a ajustar/optimizar probamos con diferentes valores y compararemos los resultados para encontrar los parámetros más adecuados.

E. CAPAS DEL MODELO

La primera capa de nuestro modelo es una capa de embedding que pre-procesa los datos de entrada para convertir las reseñas en vectores numéricos. Esta capa toma como parámetros un vocab size de 10,000 y un embed_size de 32. La segunda capa de nuestro modelo es una capa LSTM, con 128 neuronas, donde se procesarán los vectores de entrada. Finalmente, la tercera capa de nuestro modelo es una capa densa de salida con función de activación sigmoide para hacer la clasificación binaria. Los valores de los parámetros utilizados en las capas mencionadas son iniciales y elegidos de forma arbitraria. Sin embargo, dentro de la etapa de prueba de parámetros, se probará el modelo con diferentes valores con el objetivo de comparar los resultados y encontrar las parámetros más adecuados. En la Fig. 8 se muestra una visualización de las capas que constituyen el modelo.

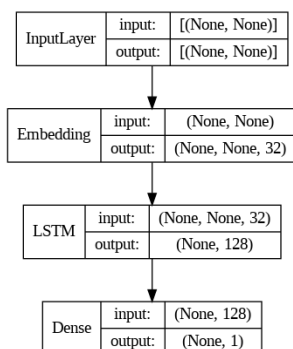


FIGURA 8. Ilustración de capas del modelo.

F. ALCANCE DEL MODELO

El alcance del modelo propuesto se define como: dado una reseña que no ha visto antes, el modelo clasificará correctamente la reseña como 0 o 1, representando si la reseña es negativa o positiva con una precisión de al menos el 80%. De forma arbitraria e inicial, esperamos que el modelo está limitado a alcanzar este nivel de precisión en

casos donde reseñas nuevas contengan: una longitud de ~1000 caracteres, sólo palabras en inglés, y ninguna palabra mal escrita.

Estas limitaciones se basan en cómo se conforman los datos de entrenamiento. Las reseñas que se utilizarán para entrenar el modelo tiene una longitud promedio de 1,309 caracteres, y una mediana de 970.

Validamos las restricciones definidas probando el modelo con reseñas que caben dentro de estas limitaciones. Por ejemplo, una reseña de una palabra, una reseña en español, y una reseña con palabras mal escritas. Estas pruebas se encuentran en la sección de pruebas de restricciones.

IV. METODOLOGÍA

A. MODELO INICIAL

Como fue mencionado en la sección de método propuesto, iniciaremos el análisis de nuestro modelo con los siguientes parámetros. Posteriormente, en la sección de resultados realizaremos una serie de pruebas con diferentes valores y compararemos los datos.

I. Valores iniciales del modelo propuesto

Parámetro	Valor inicial
epochs	5
batch_size	512
vocab_size	10,000
embed_size	32
lstm_units	128

B. TRANSPARENCIA DE PRUEBAS

Se podrán visualizar las pruebas realizadas en la siguiente liga:

https://drive.google.com/file/d/13R7WGkxDMSWbjnsnTHb1LfUeXKxnN7Z/view?usp=share_link

C. PARÁMETRO 1: ÉPOCAS

Probaremos el modelo con un número de épocas de 5 y 8 y 15, donde 5 representa al modelo con los valores iniciales.

II. Parámetros de optimización de épocas

Parámetro	Valores
-----------	---------

Épocas	5 , 10, 15
--------	------------

D. **PARÁMETRO 2: TAMAÑO DEL LOTE**

Probaremos el modelo con un número de tamaño del lote de 256 y 512, donde 512 representa al modelo con los valores iniciales.

III. Parámetros de optimización de tamaño del lote

Parámetro	Valores
Tamaño de Lote	256, 512

E. **CAPA 1: EMBEDDING**

Con base en el análisis de diversos modelos, definimos que se experimentará con dos valores, el vocab_size que representa el tamaño del vocabulario que utiliza la capa de embedding como entrada y el embed_size, donde se experimentará con vectores de dimensión 32 y 64.

IV. Parámetros de optimización para la capa Embedding

Parámetro	Valores
Tamaño del vocabulario (vocab_size)	5000, 10000
Tamaño del vector (embed_size)	32, 64

F. **CAPA 2: LSTM**

Con base en los parámetros y resultados observados en diversos modelos, establecimos los rangos a seguir para la cantidad de neuronas a utilizar. Un ejemplo es el modelo presentado en [3] donde se eligen 64 neuronas.

V. Parámetros de optimización para la capa LSTM

Parámetro	Valores
Neuronas LSTM (lstm_units)	32 , 64 y 128

G. **CAPA 3: DENSA DE SALIDA**

Por último, debido a la naturaleza de las capas anteriores y al resultado esperado por el problema, se utiliza una capa densa con una unidad de salida con una función de activación sigmoide. Debido a que el resultado es binario, sólo se necesita de una una neurona por lo que no es necesario incluir más neuronas en las pruebas. Por otro lado, la función de activación a utilizar es una función sigmoide, ideal para una clasificación binaria.

VI. Parámetros de optimización para la capa de embedding.

Parámetro	Valores
output	1
activation	sigmoid

H. **PRUEBAS DE RESTRICCIONES**

Con el objetivo de validar las restricciones del modelo definidas anteriormente y asumidas de forma arbitraria, realizamos las siguientes pruebas:

VII. Tabla de restricciones del modelo.

Tipo de Restricción	Reseña	Resultado Esperado
Reseña en Español	La película estuvo excelente, disfruté mucho de verla con mi familia.	Negativa
Reseña de 1 Palabra	Enjoyed.	Negativa
Reseña con Palabras Mal Escritas	I truly haaated dis movie, its te-rrri-ble. Bye.	Positiva

V. RESULTADOS

A. **PRUEBA CON VALORES INICIALES DE MODELO INICIAL**

VII. Pérdida y precisión de la prueba.

Pérdida	Precisión
0.4701	0.8613

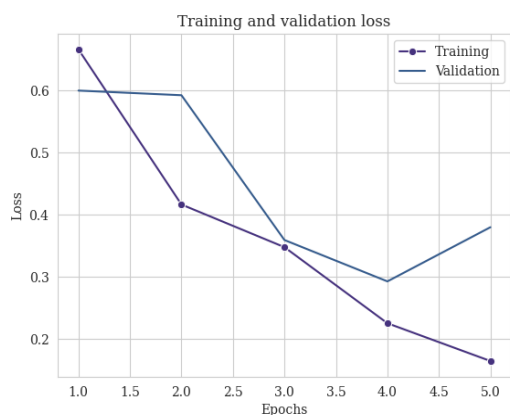


FIGURA 9. Pérdida durante el entrenamiento y validación del modelo propuesto.

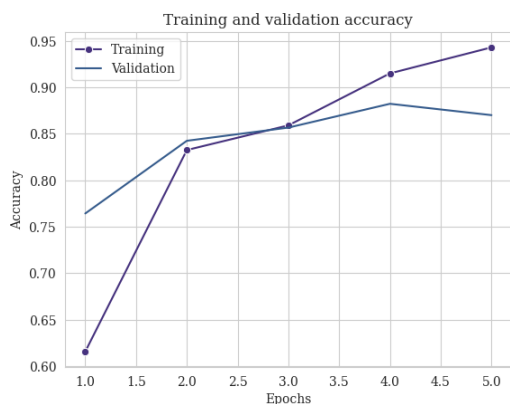


FIGURA 10. Precisión durante el entrenamiento y validación del modelo propuesto.

IX. Matriz de confusión del modelo propuesto.

	Predicción negativa	Predicción positiva
Valor real negativo	10,933	1,567
Valor real positivo	1,900	10,600

B. PRUEBA PARÁMETRO 1: ÉPOCAS

X. Pérdida y precisión de la prueba del modelo al aumentar el número de épocas.

Épocas	Pérdida	Precisión
8	0.4089	0.8562

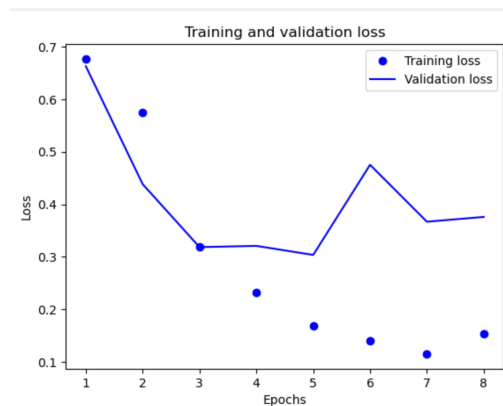


FIGURA 11. Pérdida durante el entrenamiento y validación del modelo inicial al variar el número de épocas.

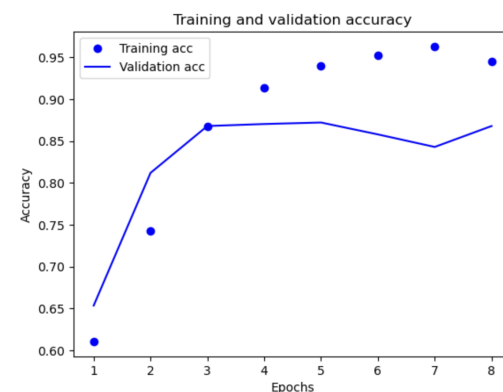


FIGURA 12. Precisión durante el entrenamiento y validación del modelo inicial al variar el número de épocas.

C. PRUEBA PARÁMETRO 2: TAMAÑO DEL LOTE

XI. Pérdida y precisión de la prueba del modelo al disminuir el tamaño del lote.

Tamaño del lote	Pérdida	Precisión
256	0.3919	0.8618

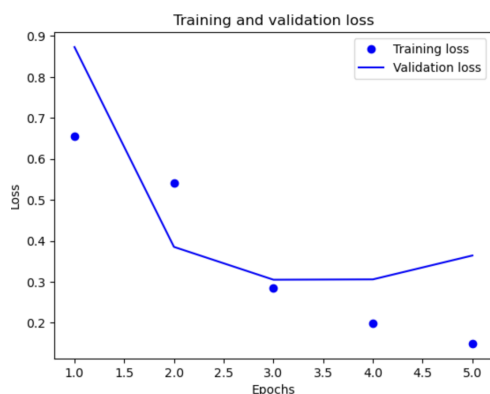


FIGURA 13. Pérdida durante el entrenamiento y validación del modelo con un tamaño del lote de 256.

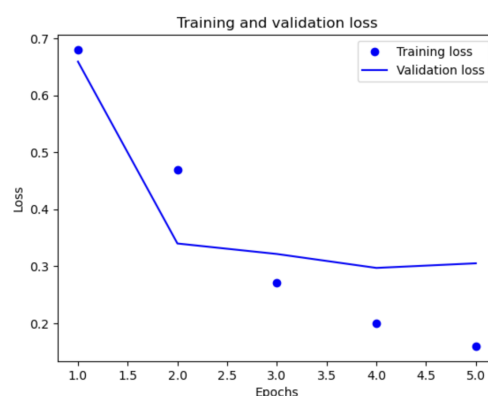


FIGURA 15. Precisión durante el entrenamiento y validación del modelo con tamaño de vocabulario de 5,000.

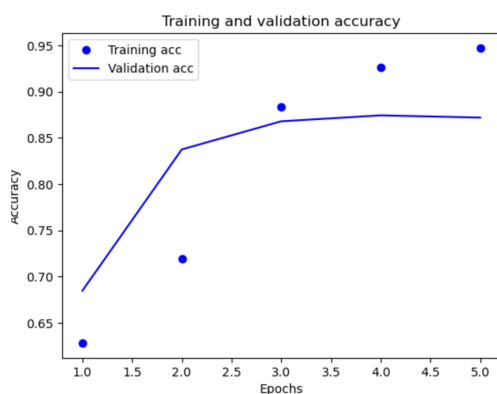


FIGURA 14. Precisión durante el entrenamiento y validación del modelo con un tamaño del lote de 256.

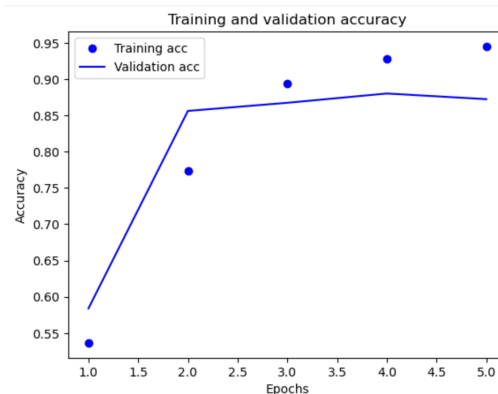


FIGURA 16. Precisión durante el entrenamiento y validación del modelo con un tamaño de vocabulario de 5,000.

D. PRUEBA CAPA 1: EMBEDDING

XII. Pérdida y precisión de la prueba del modelo al disminuir el tamaño del vocabulario.

Tamaño del vocabulario	Pérdida	Precisión
5,000	0.3232	0.8626

XIII. Pérdida y precisión de la prueba del modelo al aumentar el tamaño del vector.

Tamaño del vector	Pérdida	Precisión
64	0.3427	0.8604

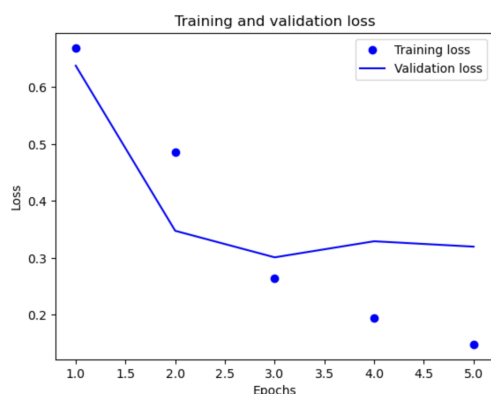


FIGURA 17. Pérdida durante el entrenamiento y validación del modelo con un tamaño del vector de 64.

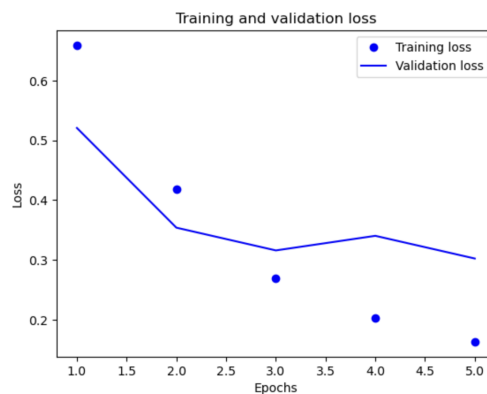


FIGURA 19. Pérdida durante el entrenamiento y validación del modelo con 32 neuronas LSTM.

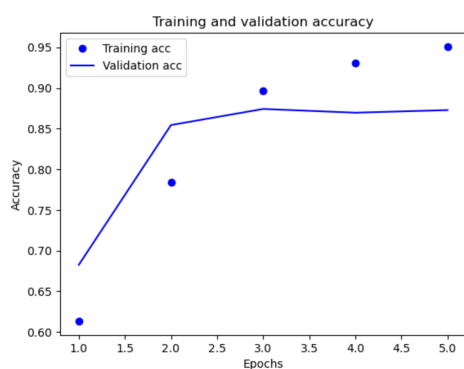


FIGURA 18. Precisión durante el entrenamiento y validación del modelo con un tamaño del vector de 64.

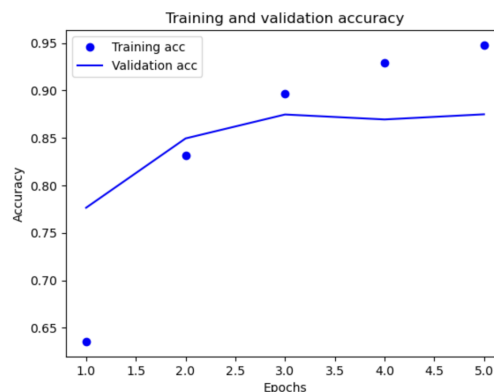


FIGURA 20. Precisión durante el entrenamiento y validación del modelo con 32 neuronas LSTM.

E. CAPA 2: LSTM

XIV. Pérdida y precisión de la prueba del modelo al disminuir en gran manera el número de neuronas LSTM.

Neuronas LSTM	Pérdida	Precisión
32	0.3328	0.8635

XV. Pérdida y precisión de la prueba del modelo al disminuir ligeramente el número de neuronas LSTM.

Neuronas LSTM	Pérdida	Precisión
64	0.3427	0.8604

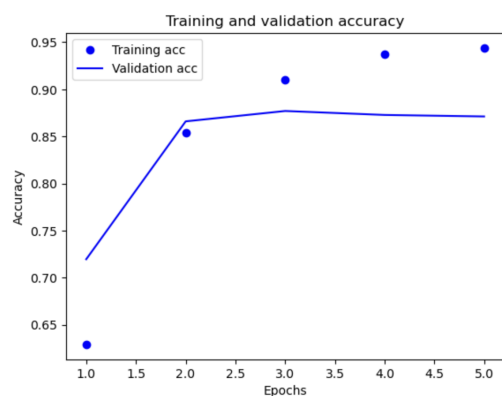


FIGURA 21. Pérdida durante el entrenamiento y validación del modelo con 64 neuronas LSTM.

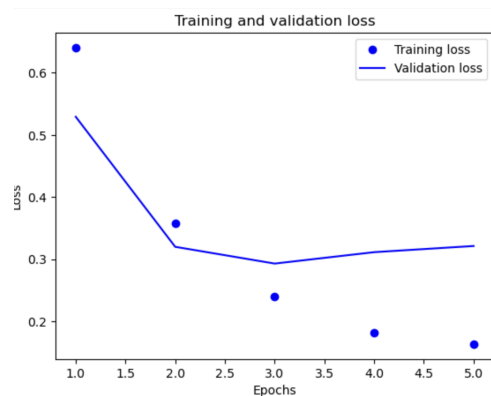


FIGURA 22. Precisión durante el entrenamiento y validación del modelo con 64 neuronas LSTM.

F. RESULTADOS DE PRUEBAS DE RESTRICCIÓN

XVI. Predicciones y probabilidades de pruebas de restricción.

Tipo de Restricción	Reseña	Resultado Esperado	Resultado	Probabilidad
Reseña Positiva en Español	La película estuvo excelente, disfruté mucho verla con mi familia.	Negativa	Positiva	63.7%
Reseña Positiva de 1 Palabra	Enjoyed	Negativa	Positiva	65.57%
Reseña Negativa con Palabras Mal Escritas	I truly haaated dis movie, its te-rr-ible. Bye.	Positiva	Positiva	76.6%

VI. CONCLUSIONES

El desarrollo del presente trabajo permitió explorar los diversos parámetros que componen un modelo de machine learning basado en una red neuronal y cómo el valor de estos impacta en el desempeño del mismo. Los hallazgos son los siguientes: Al aumentar el número de iteraciones (*epochs*) se logra que mejore la precisión hasta 8 iteraciones, después de este, la validación demuestra que el desempeño del entrenamiento empeora, por lo que no es beneficioso para el modelo realizar demasiadas iteraciones. Por ejemplo, al ingresar 10 épocas como parámetro, el modelo automáticamente para en 8 porque nota que empieza a emporar. Similarmente, si dejamos correr el modelo a 15 épocas observamos que la precisión baja a

Adicionalmente, al entrenar el modelo con un *batch_size* reducido, la precisión del modelo aumenta y la pérdida disminuye pero el tiempo de entrenamiento se eleva significativamente al necesitar un mayor número de iteraciones para lograr dichos resultados, a su vez el tiempo de ejecución de cada iteración se eleva.

Por otro lado, en la capa embedding, los parámetros *vocab_size* y *embed_size* impactan positivamente en el desempeño a medida que estos aumentan, sin embargo, requiere más recursos.

Es importante mencionar, que en la capa LSTM, al aumentar la cantidad de neuronas (*lstm_units*) de 32 a 64 a 128, observamos que la precisión disminuye. Por ejemplo, la precisión en 32 neuronas fue 86.7%, mientras que en 128 neuronas fue 84%.

Considerando todas las pruebas hechas y el análisis anterior, es evidente que el modelo propuesto con los parámetros iniciales se puede mejorar. Un modelo que utiliza los mejores parámetros encontrados: 5 épocas, 256 para el tamaño del lote, 5,000 para *vocab_size*, 64 para *embed_size*, y un número de 32 neuronas para la capa LSTM resulta en una precisión de 86.01%. Comparando esta precisión con la inicial de 85% encontramos que aunque sí es mejor, el incremento en precisión es mínimo.

Próximamente, los aspectos que pudieran mejorarse del modelo incluyen abordar más aspectos del lenguaje como el sarcasmo. Así mismo, dada las restricciones actuales, se puede explorar la opción de incrementar la base de datos para incluir palabras de otros lenguajes y así sobrepasar la restricción de sólo poder clasificar con precisión reseñas en inglés.

El modelo realizado, presenta gran significancia para futuras aplicaciones ya que podría utilizarse para clasificar texto en base a su sentimiento. Por ejemplo, puede tener importantes aplicaciones en redes sociales para detectar odio y acoso contra personas. Para la industria las aplicaciones del análisis de sentimientos también es de mucho valor, la importancia de contar con un modelo de alta precisión, reside en que estos métodos han mostrado ser herramientas competitivas, ya que las reseñas de consumidores contienen información relevante para las

partes interesadas [11], por lo que analizar grandes cantidades de esta información de manera precisa y sin intervención humana de forma eficiente. Yendo hacia adelante, las redes neuronales LSTM tendrán un impacto importante en nuestras vidas, dado que su aplicación es directa en el procesamiento de lenguaje natural, algo que es parte fundamental de la vida de cada ser humano.

REFERENCIAS

- [1] DataScientest, “Inteligencia artificial : Definición, Historia, usos, peligros,” April 21, 2023. [Online]. Available: <https://datascientest.com/es/inteligencia-artificial-definicion>
- [2] Manning, C. D., & Schütze, H. (1999), “Foundations of Statistical Natural Language Processing”. MIT Press.
- [3] Lahtela, M., Lamp; Kaplan, P., Amazon, May 3, 2023. [Online]. Available: <https://aws.amazon.com/es/what-is/neural-network/>
- [4] Codificando Bits, “¿Qué es una red neuronal?”, May 3, 2023. [Online]. Available: <https://www.codificandobits.com/blog/que-es-una-red-neuronal/>
- [5] Codificando Bits, “¿Qué es una red neuronal?”, May 3, 2023. [Online]. Available: <https://www.codificandobits.com/blog/que-es-una-red-neuronal/>
- [6] Codificando Bits, “¿Qué son las redes LSTM?”, May 3, 2023. [Online]. Available: <https://www.codificandobits.com/blog/redes-lstm>
- [7] Jordi TORRES.AI, “Redes Neuronales Recurrentes”, May 3, 2023. [Online]. Available: <https://torres.ai/redes-neuronales-recurrentes>
- [8] Arce-García, S., Díaz-Campo, J. & Cambronero-Saiz, B., “Online hate speech and emotions on Twitter: a case study of Greta Thunberg at the UN Climate Change Conference COP25 in 2019”, *Soc. Netw. Anal. Min.* 13, 48 (2023). [Online]. Available: <https://doi.org/10.1007/s13278-023-01052-5>
- [9] Li, R., Liu, M., Xu, D., Gao, J., Wu, F., & Zhu, L., “A review of machine learning algorithms for text classification,” SpringerLink, April 6, 2023. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-16-9229-1_14
- [10] Srinivasa-Desikan, B. “Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras”. 2018. Packt Publishing Ltd.
- [11] Rolland, E. Fundamentals of Sentiment Analysis and Its Applications. In *Sentiment Analysis in Social Networks* (pp. 1-25). Springer International Publishing. 2016. [Online]. Available: https://www.researchgate.net/profile/Erik-Rolland/publication/300965436_Fundamentals_of_Sentiment_Analysis_and_Its_Applications/links/5b8438fea6fdcc5f8b6b4f39/Fundamentals-of-Sentiment-Analysis-and-Its-Aplications.pdf

FERNANDO DODDOLI LANKENAU

Nació en Monterrey, Nuevo León, México en el año 2000. Se espera recibir como ingeniero en tecnologías computacionales del Tecnológico de Monterrey en Junio 2023.

Del 2021 al 2022 trabajó en Kanguru.mx como Director General y encargado de desarrollo de producto. Sus intereses incluyen valuación de empresas, desarrollo de software, blockchain, y emprendimiento.

ALEJANDRA G., OLIVAS ALTAMIRANO

Nació en Culiacán, Sinaloa, México en el año 2001. Se espera recibir como ingeniero en tecnologías computacionales del Tecnológico de Monterrey en Junio 2023.

Previamente trabajó como iterna de desarrolladora de software en Microsoft.

RUBEN A., ROBLES LEAL

Nació en Monterrey, Nuevo León, México en el año 2000. Se espera recibir como ingeniero en tecnologías computacionales del Tecnológico de Monterrey en Diciembre 2023.

Actualmente trabaja como desarrollador de aplicaciones .NET y PowerPlatform en CEMEX. Adicionalmente, se desempeña en el área de soporte a productiva para Servicios de CEMEX Online Stores.

ADRIÁN A., SALGADO MARTINEZ

Nació en Escuinapa, Sinaloa, México en el año 2001. Se espera recibir como ingeniero en tecnologías computacionales del Tecnológico de Monterrey en Junio 2023.

Previamente trabajó como interno de desarrollador de software en Meta. Actualmente se desempeña como desarrollador de aplicaciones de Android en una startup de medio tiempo.