Aprendizaje Automático en la clasificación de Sentimientos en Textos: Comparación de los Algoritmos de LSTM Bidireccional, Regresión Logística y Naïve Bayes

Paula Monge Valverde, Emmanuel Zúñiga Chaves
Escuela de Ciencias de la Computación e Informática, Universidad de Costa Rica
San Pedro de Montes de Oca, Costa Rica
mariapaula.monge@ucr.ac.cr
emmanuel.zunigachaves@ucr.ac.cr

***Abstract*—**

I. INTRODUCCIÓN

El dar una opinión nunca había sido tan sencillo como lo es ahora con las redes sociales, estas le dan una facilidad a los usuario para comentar sobre cualquier tema que deseen puede abarcar desde productos, películas, política, entre otros [1], es por esto que se vuelve de interés analizar qué es lo que se está expresando a través de las redes, esta temática ha inspirado la creación de modelos que analizan los sentimientos a través de los textos que son compartidos.

Como se puede ver en el apartado II Trabajos Relacionados, la cantidad de investigaciones alrededor del análisis de sentimientos ha ido en aumento, por ende, existe una enorme variedad de algoritmos que se pueden utilizar para la clasificación de las cargas emocionales asociadas a dichos textos.

La aplicación de estos modelos ha dado buenos resultados en un alto porcentaje de los casos, sin embargo, la mayoría de la información proporcionada y de algoritmos creados se basan en un corpus en inglés (textos escritos en inglés). No obstante, cada vez son más los textos que utilizan otros idiomas. Si bien el inglés es la lengua predominante en Internet, hay otros idiomas como el español que cada vez tiene más presencia en las redes sociales, siendo este específicamente el tercer idioma con mayor presencia en la red [2]. Ante esta situación, nos planteamos las siguientes preguntas: ¿La clasificación de los modelos aplicados a corpus tanto en inglés como español es igual? ¿Alguno de los corpus da mejores resultados que el otro?

Es de gran relevancia comprobar cómo se comportan diversos clasificadores a la hora de ser entrenados con un corpus en español, puesto que, además de querer saber que se expresa en este idioma es importante medir su capacidad de clasificación cuando se cambia de lenguaje ya que se ha evidenciado que algoritmos como **Máquinas de Soporte Vectorial (SVM)** proporcionan resultados fiables en textos en inglés, mientras que en textos en español el máximo nivel de aciertos apenas supera el 72%. Se trata de una cifra que todavía no resulta aceptable para la investigación [3].

Este artículo explora y evidencia cómo se comportan los algoritmos: **LSTM Bidireccional, Regresión Logística y Naïve Bayes** usando dos corpus: uno en español y otro en inglés con el fin de exponer cuáles de ellos tienen un mejor rendimiento, tomando en cuenta las diversas métricas de clasificación, las cuales se explicaran más adelante en el artículo, así como también verificar si el funcionamiento de los modelos mencionados se ve afectado por el idioma con el que se entrena.

El funcionamiento de dichos algoritmos, se centra en la clasificación de los sentimientos, a partir de un análisis de las características sintácticas y semánticas del texto en cuestión [4]. La razón de dicha selección, recae en el amplio uso que se les da en dicha área, es así como surge la necesidad de identificar cuál de ellos genera los mejores resultados de clasificación con un corpus

II. TRABAJOS RELACIONADOS

Internet es un amplio entorno en el cual se puede recopilar una gran cantidad de información [5], la cual enriquece la capacidad de análisis de diferentes patrones de comportamiento, opinión y sentimientos de los usuarios en

diferentes plataformas. A consecuencia de esto surgen distintas alternativas en cuanto a opciones para la clasificación de sentimientos en textos, por lo tanto, gran parte de los trabajos relacionados se enfocan en ofrecer un análisis sobre los resultados obtenidos al aplicar dichas técnicas.

En su trabajo teórico el autor H. Khandewal [6], realiza una explicación sobre cómo el algoritmo de **Naïve Bayes** puede ser implementado para clasificar los sentimientos expresados en los tweets. Según Khandewal, Naïve Bayes permite determinar la contribución de cada palabra al sentimiento asociado, y calcular la relación entre la probabilidad de aparición de una palabra y las etiquetas asociadas a la clasificación. Similarmente, Ulfa, Irmawati y Husudo [7] efectúan una implementación del modelo basado en **Naïve Bayes**. Para ello realizan la experimentación con y sin la selección de características de información mutua (MI) para escoger las más relevantes dentro del conjunto de datos de tweets, obteniendo resultados de accuracy de entre el 96.2% y el 97.9% en el experimento implementado con MI.

Por otro lado, otras investigaciones se centran más en hacer un análisis del corpus dado un contexto. Chiorrini, Diamantini, Mircoli y Potena [8] realizan una implementación de un modelo de clasificación basado en **BERT**. Para ello, se recurrió a un conjunto de datos con más de un millón de tweets, clasificados según el sentimiento (positivo, negativo, neutral), obteniendo un accuracy del 92%, del cual se deduce que los modelos del lenguaje de **BERT** contribuyen a la obtención de buenos resultados en la clasificación de textos.

III. METODOLOGÍA

A continuación, se presenta la metodología utilizada en la presente investigación. Describiremos detalladamente el conjunto de datos a utilizar, y su debido tratamiento, así como los algoritmos utilizados para la clasificación binaria en el análisis de sentimientos.

III.I DATOS

Para efectos de investigación, se utiliza el conjunto de datos “IMDB Dataset of 50K Movie Reviews (Spanish)” el cual, posee más de 50 mil reseñas en inglés con su respectiva traducción en español [10], clasificadas como positivo o negativo, según el sentimiento asociado al texto. El conjunto se encuentra excepcionalmente equilibrado con respecto a las dos clases existentes. Esto significa que la distribución de instancias marcadas como positivas y negativas es muy similar, lo que permite un análisis imparcial y robusto de los resultados.

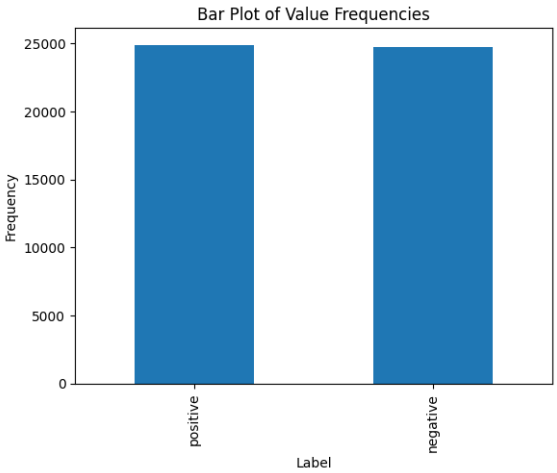

Fig 1. Distribución de frecuencias para cada clase en el conjunto de datos.

En este estudio, todos los datos utilizados se sometieron a un minucioso tratamiento previo. Este proceso incluyó la eliminación de datos duplicados para evitar la duplicación de información y garantizar la correcta representación del texto. Además, se aplicó una limpieza de los textos para eliminar caracteres especiales, palabras vacías y otros símbolos que no eran relevantes para la clasificación. Este tratamiento previo garantiza la integridad de los datos y permitió un análisis más preciso y fiable de los textos en términos de contenido emocional.

TABLA 1
EJEMPLOS DE TEXTOS PREPROCESADOS

Texto original en inglés	Texto original en español
Probably my all-time favorite movie, a story of selflessness, sacrifice and dedication to a noble cause, but it's not preachy or boring.	Probablemente mi película favorita de todos los tiempos, una historia de desinterés, sacrificio y dedicación a una causa noble, pero no es predicada ni aburrida.
Texto limpio en inglés	Texto limpio en español
probably time favorite movie story selflessness sacrifice dedication noble cause preachy boring	probablemente pelicula favorita tiempos historia desinteres sacrificio dedicacion causa noble predicada aburrida simplemente nunca envejece pesar haberlo visto unas 15 mas veces ultimos 25 anos

Posteriormente, se procede a la creación de conjuntos de entrenamiento y prueba, para lo cual se utiliza una proporción del 80:20, si bien, existen otros métodos de división de conjuntos más sotifisticadas, 80:20 suele ser suficiente para modelos de aprendizaje automático [11].

III.II MODELOS UTILIZADOS

III.II.I REGRESIÓN LOGÍSTICA

En un inicio uno de los algoritmos seleccionados fue "BERT", que como se mencionó anteriormente, obtenía buenos resultados de clasificación. No obstante, a la hora de realizar la experimentación con el dataset seleccionado se notó que el entrenamiento de dicho algoritmo tomaba mucho tiempo, lo cual no era conveniente para la investigación. Ante esta situación, se opta por usar como reemplazo el algoritmo de Regresión Logística, el cual sirve para resolver problemas de clasificación binaria [9].

Con regresión logística se mide la relación entre la clasificación del texto (variable objetivo) y una o varias variables independientes, mediante una función logística que calcula la probabilidad de la variable objetivo utilizando un conjunto de características del texto [12]. La implementación del modelo se realizó en Python, utilizando la librería de scikit-learn: Logistic Regression.

III.II.II LSTM BIDIRECCIONAL

Corresponde a una arquitectura de redes neuronales, utilizada comúnmente en el procesamiento del lenguaje natural. Mediante LSTM se asegura que algunos datos de la secuencia de entrada sean "almacenados" durante algún periodo de tiempo. En esta arquitectura, el procesamiento de los datos de entrada se realiza en ambas direcciones, lo cual, permite analizar el contexto de los datos a partir de la información en ambas secuencias [13].

Para la implementación de un clasificador basado en LSTM Bidireccional, se emplean las librerías LSTM y Bidirectional, del framework TensorFlow.

III.II.III NAIVE BAYES

El algoritmo de Naive Bayes se trata de aplicar la **Regla de Bayes,** la cual describe la probabilidad de que ocurra un evento, con el conocimiento previo de la ocurrencia de otro evento relacionado con él.

En Naive Bayes, encontraremos cómo cada palabra está contribuyendo al sentimiento, que se puede calcular mediante la relación de la probabilidad de aparición de la palabra para la clase positiva y negativa [6].

Para su implementación, se utiliza la librería multinomialNB, de scikit-learn.naive_bayes.

III.III RENDIMIENTO DE LOS MODELOS

Cuando se quiere evaluar el rendimiento de un *programa de clasificación* se suelen usar las métricas de precision, recall, F1-Score y accuracy. Cada una de ellas aporta distintos datos que permiten analizar el funcionamiento de la clasificación; **precision** indica el porcentaje de todas las predicciones positivas cuáles de ellas son correctas, **recall** indica el porcentaje de todos los positivos cuáles son reales (TP), **F1-Score** se trata de una mezcla de las dos métricas mencionadas anteriormente, haciéndolas funcionar de manera armónica/equilibrada y también **accuracy**, que nos brinda el porcentaje de casos que el modelo ha acertado, es decir, nos indica que tan exacta fue la clasificación de los datos. Si bien esta métrica es conocida por ser "engañosa", esta característica se da cuando se poseen datos con clases desbalanceadas. Con el fin de evaluar el desempeño de los algoritmos, se tomarán en cuenta las cuatro métricas mencionadas anteriormente.

Una vez creados los conjuntos, se procede al entrenamiento y ajuste de los modelos seleccionados, para ello se recurre a la combinación de diferentes hiperparámetros y posteriormente, la evaluación de los resultados de clasificación, lo cual será analizado en el apartado IV Análisis de resultados.

IV. ANÁLISIS DE RESULTADOS

A continuación, se presentan diversos resultados obtenidos por los diferentes algoritmos. Para ello, se presentan los valores para las métricas mencionadas en el apartado III.III. La selección de los valores para los hiperparamétros utilizados en cada modelo fue realizada mediante fuerza bruta, es decir, intentar encontrar aquella combinación que brinde mejores resultados.

Igualmente durante la experimentación con los tres algoritmos se realizaron múltiples ejecuciones, para así tener más seguridad acerca de los resultados.

IV.I RESULTADOS DE REGRESIÓN LOGÍSTICA

TABLA 2
MÉTRICAS DE REGRESIÓN LOGÍSTICA PARA TEXTOS EN ESPAÑOL

	Precision	**Recall**	**F1-Score**
negative	0.9	0.88	0.89
positive	0.88	0.9	0.89
Accuracy	0.89		

TABLA 3
MÉTRICAS DE REGRESIÓN LOGÍSTICA PARA TEXTOS EN INGLÉS

	Precision	**Recall**	**F1-Score**
negative	0.91	0.86	0.85
positive	0.86	0.84	0.85
Accuracy	0.9		

Para ambos resultados se utilizaron los siguientes hiperparametros:

max_iters: cantidad de iteraciones que hará el modelo para realizar la clasificación.

ngram_range: el rango de n-gramas se refieren a una secuencia de N palabras o caracteres. Dichas secuencias pueden ser bigramas (2 unidades), trigramas (3 unidades) o más generalmente como n-gramas.

max_features: cantidad de características del texto de entrada, ejemplo: palabras clave, tamaño, etc.

C: Inverso de la fuerza de regularización, está regularización aplica una penalización al aumento de la magnitud de los valores de los parámetros con el fin de reducir el sobreajuste.

IV.II LSTM BIDIRECCIONAL

TABLA 4
MÉTRICAS DE LSTM BIDIRECCIONAL PARA TEXTOS EN ESPAÑOL

	Precision	**Recall**	**F1-Score**
negative	0.79	0.86	0.82
positive	0.84	0.77	0.81
Accuracy	0.81		

TABLA 5
MÉTRICAS DE LSTM BIDIRECCIONAL PARA TEXTOS EN ESPAÑOL

	Precision	**Recall**	**F1-Score**
negative	0.85	0.86	0.85
positive	0.86	0.85	0.85
Accuracy	0.85		

Para ambos resultados se utilizaron los siguientes hiperparametros:

learning_rate: regula los pesos del modelo respecto al gradiente de pérdida. Indica con qué frecuencia se actualizan las nociones que ha aprendido.

activation: función que transmite la información generada por la combinación lineal de los pesos y las entradas

epochs: cantidad de iteraciones a realizar durante el entrenamiento.

batch_size: cantidad de ejemplos de entrenamiento utilizados en cada época,

IV.III NAIVE BAYES

TABLA 6
MÉTRICAS DE NAIVE BAYES PARA TEXTOS EN ESPAÑOL

	Precision	**Recall**	**F1-Score**
negative	0.84	0.86	0.85
positive	0.86	0.84	0.85
Accuracy	0.85		

TABLA 7
MÉTRICAS DE NAIVE BAYES PARA TEXTOS EN INGLÉS

	Precision	**Recall**	**F1-Score**
negative	0.84	0.88	0.86
positive	0.88	0.83	0.85
Accuracy	0.86		

Para ambos resultados se utilizaron los siguientes hiperparametros:

alpha: controla el nivel de suavizado aplicado a las estimaciones de probabilidad para evitar problemas de probabilidad de cero en características no observadas

fit_prior: determina si se deben estimar las probabilidades a priori de las clases a partir de los datos de entrenamiento.

V. DISCUSIONES Y CONCLUSIONES

REFERENCIAS

[1] L. Montesinos Garcia, “Análisis de Sentimientos y Predicción de Eventos en Twitter,” Memoria para optar al título de Ingeniero Civil Eléctrico, Universidad de Chile, Santiago de Chile, 2014.

https://repositorio.uchile.cl/bitstream/handle/2250/130479/cf-montesinos_lg.pdf

[2] Instituto Cervantes, “Anuario 2019. El español en internet y en las redes sociales,” CVC, 2019. https://cvc.cervantes.es/lengua/anuario/anuario_19/informes_ic/p04.htm

[3] T. Baviera, “Técnicas para el análisis del sentimiento en Twitter: Aprendizaje Automático Supervisado y SentiStrength”, *Dígitos*, pp. (38-40), Dic, 2017

[4] P. Mollah, “An LSTM model for Twitter Sentiment Analysis”, 2022, https://arxiv.org/abs/2212.01791

[5] 0N. A. Awad and A. Mahmoud, “Analyzing customer reviews on social media via applying association rule,” Computers, Materials and Continua, vol. 68, no. 2, pp. 1519–1530, Apr. 2021, doi: 10.32604/CMC.2021.016974.

[6] H. Khandewal. "Sentiment Analysis of a Tweet With Naive Bayes". Medium. https://towardsdatascience.com/sentiment-analysis-of-a-tweet-with-naive-bayes-ff9bdb2949c7

[7] M. A. Ulfa, B. Irmawati, and A. Y. Husodo, "Twitter Sentiment Analysis using Naive Bayes Classifier with Mutual Information Feature Selection," J. Comput. Sci. Inform. Eng. (J-Cosine), vol. 2, no. 2, pp. 106-111, Dec. 2018, doi: https://doi.org/10.29303/jcosine.v2i2.120

[8] A. Chiorrini, C. Diamantini, A. Mircoli, and D. Potena, "Emotion and sentiment analysis of tweets using BERT," in Proceedings of the EDBT/ICDT Workshops, Nicosia, Cyprus, Mar. 23-26, 2021.

[9] “Análisis de sentimiento con regresión logística,” ICHI.PRO. https://ichi.pro/es/analisis-de-sentimiento-con-regresion-logistica-173659997957551 (accessed May 12, 2023).

[10] L. D. Fernandez, “IMDB Dataset of 50K Movie Reviews (Spanish),” Kaggle, [En línea]. Disponible en: https://www.kaggle.com/code/lakshmi25npathi/sentiment-analysis-of-imdb-movie-reviews. (accessed May 15, 2023).

[11] “Aprendizaje automático y datos de entrenamiento: lo que debes saber,” Ciberseguridad. https://ciberseguridad.com/guias/nuevas-tecnologias/machine-learning/datos-entrenamiento/ (accessed May 12, 2023).

[12] D. Rodríguez, “La regresión logística,” Analytics Lane, 23-Jul-2018. [En línea]. Disponible en: https://www.analyticslane.com/2018/07/23/la-regresion-logistica/. (accessed May 8, 2023).

[13] Differences Between Bidirectional and Unidirectional LSTM | Baeldung on Computer Science. (s.f.). Baeldung on Computer Science. https://www.baeldung.com/cs/bidirectional-vs-unidirectional-lstm. (accessed May 8, 2023)