UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA:
INGENIERÍA DE SISTEMAS

Trabajo de titulación previo a la obtención del título de:
Ingeniero de Sistemas

TEMA:
MINERÍA DE OPINIÓN PARA TEXTOS EN ESPAÑOL USANDO PROCESAMIENTO NATURAL DEL LENGUAJE

AUTOR:
EDWIN FRANCISCO AUQUILLA MOROCHO

TUTOR:
JULIO RICARDO PROAÑO ORELLANA

Quito, agosto de 2021

CESIÓN DE DERECHOS DE AUTOR

Yo, Edwin Francisco Auquilla Morocho, con documento de identificación N° 1723555338, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación con el tema: MINERÍA DE OPINIÓN PARA TEXTOS EN ESPAÑOL USANDO PROCESAMIENTO NATURAL DEL LENGUAJE., mismo que ha sido desarrollado para optar por el título de INGENIERO DE SISTEMAS en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada.

En concordancia, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

…….

EDWIN FRANCISCO
AUQUILLA MOROCHO

CI: 1723555338

Quito, agosto de 2021

DECLARATORIA DE COAUTORÍA DE LA TUTORA

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Artículo Académico, con el tema: MINERÍA DE OPINIÓN PARA TEXTOS EN ESPAÑOL USANDO PROCESAMIENTO NATURAL DEL LENGUAJE, realizado por Edwin Francisco Auquilla Morocho, obteniendo un producto que cumple con todos los requisitos por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

Quito, agosto de 2021

……
JULIO RICARDO PROAÑO ORELLANA
CI: 0103909412

MINERÍA DE OPINIÓN PARA TEXTOS EN ESPAÑOL USANDO PROCESAMIENTO NATURAL DEL LENGUAJE

OPINION MINING FOR TEXTS IN SPANISH USING NATURAL LANGUAGE PROCESSING

Edwin Auquilla[1], Julio Proaño[2]

Resumen

Este artículo se propone el desarrollo de una herramienta que permite la minería de opiniones de textos exclusivamente en español. Para ello, se realizó la construcción de un corpus lingüístico a través de tweets. Este corpus se caracteriza por tener tweets en idioma español latinoamericano y español castellano. Los algoritmos de clasificación usados son: Naïve Bayes, SVM y LSTM. Para la evaluación de la propuesta se realizaron experimentos con un corpus de 14.666 con una clasificación en dos clases (positivo y negativo), luego en tres clases (positivo, negativo, neutro). Los Resultados obtenidos muestran que la clasificación binaria obtuvo mejores resultados que en la clasificación a tres clases en los tres algoritmos.

Palabras Clave:

NLP, SVM, LSTM, Naive Bayes, Tokenizar, TASS, Procesar, Corpus lingüístico, API, Twitter.

Abstract

This paper proposes development of a tool that allows opinion mining of texts exclusively in Spanish. For this, a linguistic corpus has been constructed through tweets. This corpus is characterized by having tweets in Latin American Spanish and Castilian Spanish. The evaluated classification algorithms were: Naive Bayes, SVM, and LSTM. The experiments were carried out with a corpus of 14,666 with a classification in two classes (positive and negative) in three classes (positive, negative, neutral). The results obtained show that the binary classification got better results than the three-class classification in the three algorithms.

***Keywords*:**

NLP, SVM, LSTM, Naïve Bayes, Tokenize, TASS, Process, Linguistic Corpus, API, Twitter.

[1]Estudiante de Ingeniería de Sistemas – Universidad Politécnica Salesiana, Egresado – UPS – sede Quito. Autor para correspondencia: eauquillam@est.ups.edu.ec

[2]Julio Proaño Docente de la Universidad Politécnica Salesiana – UPS - sede Quito
Email: jproanoo@ups.edu.ec

1. Introducción

El procesamiento natural del lenguaje tiene una gran cantidad de aplicaciones hoy en día como son: análisis de sentimientos, historias clínicas, clasificación de currículos, minería de opiniones, clasificación de incidencias, búsqueda en internet por voz, Traducción automática de comentarios en foros, etc [1].

En la actualidad existen varias herramientas o sistemas de Deep Learning, Machine Learning para minería de opiniones o análisis de sentimientos a través del procesamiento natural del lenguaje para textos en inglés.

Para NLP existe una gran variedad de algoritmos que se pueden utilizar. Para esta investigación se tomó como referencia a [2], [3], [4] para la selección de los algoritmos. Los artículos consultados tratan de aplicaciones de procesamiento de lenguaje natural, extracción de aspectos en opiniones textuales y análisis de sentimientos. De acuerdo [2] menciona que para análisis de sentimientos se usa algoritmos de aprendizaje supervisado que pueden ser: de regresión y de clasificación. Los primeros relacionan un cierto número de características y una variable objetiva continua. En cambio, los de clasificación se fundamentan en un conjunto finito de resultados, dicho resultado es una etiqueta discreta. Aunque ambos tipos de algoritmos pueden ser utilizados se centra en los siguientes: Naïve Bayes, Máquinas de Vectores de Soporte, K vecinos más cercanos y Arboles de Decisión.

En [3] el autor, se centra solamente en el uso del algoritmo Naïve Bayes consideran que el modelo representa un éxito considerable en aplicaciones de NLP debido a que es fácil y rápido predecir clases, para problemas de clasificación binarias y multiclase.

En [4] los autores proponen el uso de "Long Short-Term Memory" (LSTM) que son redes de memoria de a corto plazo o a largo plazo. y que ha obtenido resultados récord en comprensión de textos en lenguaje natural y reconocimiento de escritura manual entre otras aplicaciones.

Teniendo en cuenta lo descrito por dichos autores se llegó a la conclusión de usar tres algoritmos de aprendizaje supervisado estos son: Naïve Bayes, "Máquina de Soporte de Vectores" (SVM) y LSTM, se eligió estos tres algoritmos ya que se acoplan perfectamente con el procesamiento natural del lenguaje y proporcionan una integración con el texto de cada uno de los tweets recolectados.

"Natural Language Processing" (NLP) es un área de la lingüística computacional y de la inteligencia artificial, el termino lenguaje natural se usa para diferenciar los lenguajes humanos, de los lenguajes de programación [5]. Su primordial tarea es la de analizar los aspectos lingüísticos del texto a través de un programa informático [6].

La arquitectura de un sistema de NLP se sustenta en una definición del LN (lenguaje natural) por niveles estos son:

- **Nivel Fonológico:** se enfoca en cómo las palabras se relacionan con los sonidos que representan.
- **Nivel Morfológico:** se enfoca en cómo las palabras se construyen a partir de unas unidades de significado más pequeñas llamadas morfemas.
- **Nivel Sintáctico:** está enfocado en enlazar palabras que crean oraciones, teniendo en cuenta la estructura de la oración donde interviene cada palabra.
- **Nivel Semántico:** está enfocado en el significado que tiene las palabras y como estas palabras se pueden relacionar entre sí para formar una oración.
- **Nivel Pragmático:** está enfocado en el significado de las oraciones que se utiliza en diferentes ámbitos y que pueden

cambiar su significado dependiendo de la situación [7] .

Por otro lado, varios autores usan Naïve Bayes, este es un algoritmo de clasificación supervisada que se basa en el teorema de Bayes [8], menciona que es posible conocer la probabilidad de que ocurra un evento a partir de que ocurra otro evento, del cual depende el primero [3]. Se la denomina Nave (ingenua) debido a que asume que las palabras son condicionalmente independientes entre sí dada una cierta clase [9].

Proporciona una mejor precisión de clasificación en conjuntos de datos en tiempo real que cualquier otro clasificador y puede usar un conjunto de entrenamiento pequeño [4].

Este algoritmo usa tokens los cuales después de ser procesados se debe convertir a un diccionario ya que así entiende el clasificador esto se logra usando las palabras como claves seguidas de TRUE valores. [10], como se muestra en la Figura 2 y 3.

Type	Size	Value
dict	5	{'manutonic':True, 'buenotes':True, 'diotas':True, 'g 'abraz ...
str	8	Positiva

Figura 2. Diccionario de palabras.

Key	Type	Size	Value
abrazo	bool	1	True
buenotes	bool	1	True
diotas	bool	1	True
gran	bool	1	True
manutonic	bool	1	True

Figura 3. True Valores.

Otro algoritmo utilizado es LSTM que traducido al español significa unidades de memoria a corto plazo o a largo plazo. Son un subconjunto de las redes neuronales recurrentes que son mayormente conocidas y con una gran variedad de aplicaciones en Deep Learning [11], permiten distinguir y pronosticar una serie de datos a lo largo del tiempo, como, por ejemplo: textos, genomas, discurso hablado o series numéricas [12].

Las LSTM están diseñadas para evitar el problema de dependencia a largo plazo es decir que recordarán información durante largos periodos de tiempo [13] según [11] las redes neuronales recurrentes que son antecesoras a las LSTM tenían un problema que poseían un cierto tiempo de memoria y esto hacía que no fueran tan eficaces como las memorias LSTM.

En la Figura 3 muestra la arquitectura de LSTM que tienen una estructura tipo cadena lo que cambia de la arquitectura de “Redes Neuronales Recurrentes” (RNN) es el módulo de repetición el cual en vez de tener una sola capa de red neuronal hay cuatro que se relaciona entre sí [11].

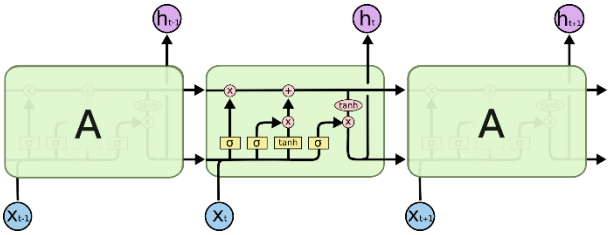

Figura 3. Arquitectura LSTM. [11]

Finalmente, las máquinas de soporte vectorial pertenecen al grupo de algoritmos de aprendizaje supervisado, se centra en la clasificación binaria o regresión de manera acelerada y efectiva, de acuerdo a [10] en términos geométricos se puede observar a las máquinas de soporte vectorial como un hiperplano como muestra la Figura 4 las instancias positivas y negativas están separadas por los vectores de soporte.

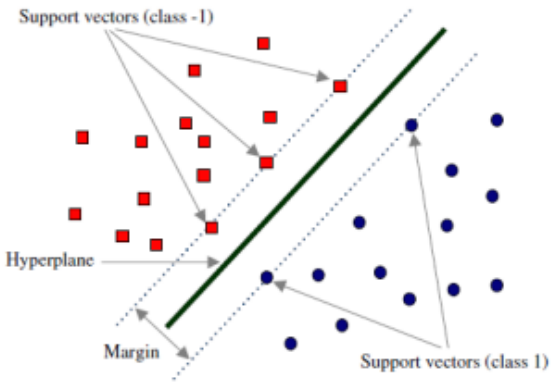

Figura 4. Máquina de soporte de vectores [2].

SVM cuenta con diferentes argumentos que se puede utilizar para la configuración y de esta forma tener mejores resultados de clasificación. Cuando no es posible clasificar de forma lineal, plano o hiperplano se usa el argumento kernel el cual brinda un nuevo espacio dimensional para obtener una clasificación eficaz y eficiente [2] [3] [14].

Sin embargo, para textos en español hay muy poca información o aplicaciones, esto debido a que el lenguaje español es más complejo en sus formas verbales, gramática, multitud de dialectos y variaciones semánticas de cada región.

De acuerdo a [15] los autores en su investigación de "Aprendizaje profundo para la extracción de aspectos en opiniones textuales", menciona que es una tarea muy importante en el análisis de sentimientos o minería de opiniones, ya que puede lograr una mayor precisión en el análisis de información y promover la toma de decisiones. El aprendizaje profundo combina varios algoritmos o estrategias que obtienen resultados relevantes en diversas tareas del procesamiento del lenguaje natural.

En [14] el autor de "Sistema Deep Learning para el análisis de sentimientos en opiniones de productos para la ordenación de resultados de un buscador semántico", aborda el problema de análisis de sentimientos, enmarcado dentro del área de estudio del Procesamiento de Lenguaje Natural, usando un conjunto de tweets en español filtrados por hashtag utilizando técnicas de aprendizaje automático aplicadas al campo de NLP en esta investigación el autor antes de usar algoritmos de aprendizaje supervisado usa algoritmos de aprendizaje automático con la intención de que su sistema se capaz de distinguir la estructura de textos y así lograr mejorar los resultados de clasificación obtenidos

En [2] el autor de "Análisis de sentimientos en Twitter" en su investigación crea un sistema que clasifica los tweets de un usuario en sentimiento positivo o negativo, utiliza algoritmos de aprendizaje supervisado y para la creación de su corpus usa los tweets en español que proporciona TASS(Taller de Análisis de Sentimientos) de los cuales toma unos 68.000 tweets, al usar ese corpus obtuvo buenos resultados en sus algoritmos de clasificación lo cual facilito que no expandiera más su corpus ya que los experimentos y resultados fueron los esperados.

De acuerdo a Alejando Pérez autor de la investigación "Aplicación para el análisis de sentimientos y tendencias en redes sociales" plantea el uso de redes neuronales recurrentes y LSTM para el desarrollo de una herramienta de análisis de sentimientos, al momento de realizar experimentos con estos algoritmos de clasificación toma en cuenta que la mejor opción es LSTM ya que no tiene problema del desvanecimiento del gradiente esto significa que los pesos y los sesgos no pueden actualizarse correctamente porque el gradiente que actúa de corrector se desvanece. El gradiente permite ajustar los parámetros de la red a través de caculos de tal manera que minimice su desviación a la salida [16]. Pero para que su algoritmo funcione correctamente debió usar 1.6 millones de tweets los cuales fueron proporcionados por la Universidad de

Stanford con estos datos el algoritmo obtuvo buenos resultados.

En este artículo se propone la construcción de una herramienta para minería de opiniones con su respectivo corpus lingüístico construido con tweets en español de varios países de Latinoamérica como: Ecuador, Costa Rica, Uruguay, México, Chile y España. Además, se propone una metodología específica para la construcción del corpus, mediante el uso de la API de Twitter y el procesamiento de estos tweets antes de ser enviados a los algoritmos de clasificación.

Este artículo está organizado de la siguiente manera: en la Sección II, describe la metodología empleada para la construcción del corpus mediante la API de Twitter para recolección de tweets, procesamiento de los tweets, selección del algoritmo de clasificación, entrenamiento y pruebas. La Sección III, se incluyen los experimentos realizados y sus respectivos resultados y discusión. Finalmente se presentan las conclusiones del trabajo.

2. Métodos y materiales

En la siguiente sección se describe la metodología usada en la investigación. Este consta de las siguientes fases: I) recolección de tweets, II) procesamiento tweets y limpieza de caracteres especiales, III) construcción del corpus, IV) entrenamiento y pruebas, V) expansión del corpus, VI) análisis de resultados

2.1. Recolección de tweets

Esta fase se realizó a través del uso de la API (Interfaz de programación de aplicaciones) de Twitter y de TASS-SEPLN (Taller de análisis semántico de la Sociedad Española de Procesamiento del Lenguaje Natural) [17].

2.1.1.1. API de Twitter

Twitter tiene una gran cantidad de API's las cuales son utilizadas mayormente por desarrolladores de software para realizar sistemas automatizados que interactúan con Twitter [18]. Para esta investigación se usó solamente la obtención de tweets.

Se empezó a descargar tweets usando usuarios como son: KfcEcuador, MacDonaldsEcuador, Carlos Vera, Erika Vélez, Ursula Strenge, Carolina Jaume, Tania Tinoco, Alfonso Laso, Teleamazonas, El Universo, El Comercio y hashtags, por ejemplo: @noticiaspositivas, @malasnoticias, @noticiasec, @ClaroEcua, se usó estos usuarios y hashtag ya que son exclusivamente del Ecuador, los usuarios o hashtag usados representan a periodistas, empresas de comida, telefónicas, presentadores o presentadoras de televisión, actores, canales de televisión y periódicos que cuando tuitean o retuitean algo , usan palabras, modismo que usamos habitualmente los ecuatorianos así se complementará con los tweets que proporciona TASS.

Se recolectó una cantidad de 1.000 tweets los cuales son almacenados en un archivo con extensión .csv siguiendo la siguiente estructura: tres columnas con etiquetas de id, texto o tweet y sentimiento.

2.1.1.2. Taller de Análisis Semántico en SEPLN(TASS)

TASS proporciona una gran cantidad de tweets exclusivamente para el idioma español. Estos tweets son obtenidos de países como España, México, Perú, Uruguay, Chile y Costa Rica [17]. Esta colección de tweets al ser descargados tiene una extensión de archivo .csv.

2.2. Procesamiento de Tweets

Para llevar a cabo el procesamiento de tweets se debe realizar los siguientes pasos:

- Tokenizar.
- Normalizar.
- Remover palabras, caracteres especiales no significativos [19].

Se debe considerar que este procesamiento se lo realiza para mantener un tweet más limpio entendible y que solo contenga texto.

2.2.1.1. Tokenización de Tweets

El proceso para realizar la tokenización de tweets es dividir el texto en oraciones y estas en palabras. Para NLP un token es la unidad mínima de procesamiento que puede ser términos o palabras [20].

Las palabras son separadas en tokens cada que exista un espacio en blanco entre ellas [21] como muestra la Figura 1, se puede observar como un texto u oración de entrada después de un proceso de Tokenización es dividido en términos independientes.

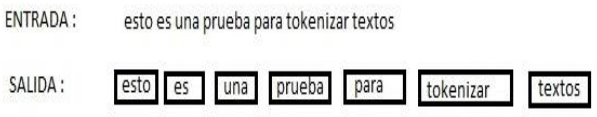

Figura 1. Proceso de Tokenización

La tokenización se aplica en el algoritmo de clasificación Naïve Bayes ya que es un clasificador de textos supervisado que entiende tokens [10] que serán las características de cada uno de los tweets.

2.2.1.2. Normalización de Tweets

Normalizar es una tarea en la cual se debe poner al texto en igualdad de condiciones [22]. En NLP la normalización es el proceso de convertir una palabra es su forma canónica esto permite que proceda el procesamiento de manera uniforme. [23], como muestra la Tabla 1 el texto original tiene palabras que no están bien escritas al ser normalizadas estas palabras ya tienen la forma estándar.

Tabla 1. Ejemplo de normalización de texto.

Texto Original	Texto Normalizado
holaa, le pregunto x el anunsio q tiene sobre el empleo	hola le pregunto por el anuncio que tiene sobre el empleo

2.2.1.3. Remoción de palabras o caracteres especiales no significativos

De acuerdo a [24] al remover cualquier palabra o carácter especial no significativo dentro de un tweet permite una mejor comprensión de los algoritmos de clasificación ya que se tendrá un mejor procesamiento del lenguaje natural.

Los elementos a remover son los siguientes:

- **Hipervínculos**: todos los hipervínculos en Twitter se convierten a la URL acortador.
- **Twitter identifica en las respuestas**: estos nombres de usuario de Twitter están precedidos por un @símbolo, que no transmite ningún significado.
- **Puntuación y caracteres especiales**: si bien estos suelen proporcionar contexto a los datos textuales, este contexto suele ser difícil de procesar. Para simplificar, removerá todos los signos de puntuación y caracteres especiales de los tweets [24].

2.3. Construcción del corpus lingüístico

El corpus lingüístico es un conjunto de textos que contiene un número considerable de textos, dichos textos comparten entre si varios rasgos definitorios [25] que en este caso para esta investigación serán divididos en textos que describen opiniones positivas, negativas y neutras.

Para la construcción se usó la data que nos proporciona TASS, esta data se divide en cinco archivos con extensión .csv de los países de Costa Rica, España, México, Perú y Uruguay esta data se la une en un solo archivo .csv con tweets obtenidos de Ecuador este conjunto de datos tiene 5.147 tweets con su respectivo sentimiento.

2.4. Entrenamiento y pruebas

Este apartado describe el procedimiento que se llevó a cabo para el entrenamiento y pruebas de los algoritmos de clasificación antes mencionados en los cuales se usó el corpus creado de tweets.

2.4.1. Entrenamiento

El hardware y software requerido para el entrenamiento de Naïve Bayes, SVM y LSTM se muestran en la Tabla 2 con la respectiva versión usada de Python y el IDE Spyder.

Tabla 2. Características de hardware y software.

Características	Especificaciones
Ram	8 GB, 2400 MHz
Cpu	4 núcleos, 4 hilos,2.4 MHz, 6MB en caché
Disco Duro	500 GB
SO	Windows 10 – 1809 x64
Python	V3.6
Spyder	V5.0
Nltk	V3.6.2

Para un correcto funcionamiento de “natural languaje tool kit” (nltk) y Python se usó la versión de Anaconda 1.10.0 con el respectivo IDE Spyder en su versión 5.0.

Luego de cumplir con los requisitos de hardware y software se procedió con el siguiente diagrama de la Figura 4 que esquematiza las etapas del entrenamiento.

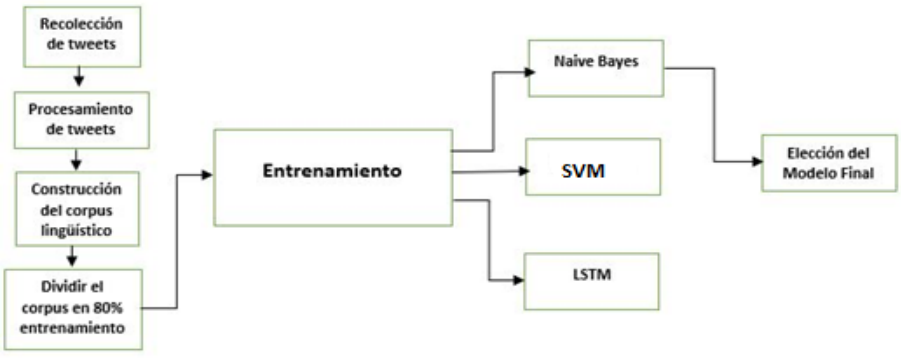

Figura 4. Etapas del Entrenamiento

La recolección de tweets se la realizo por medio de la API de Twitter y de los data sets proporcionados por TASS, luego se agrupo estos tweets que contaban con su sentimiento y se los envió a procesar para limpiarlos de caracteres especiales, links etc., a partir de esos tweets ya limpios se construyó el corpus con alrededor de 5.145 tweets con sui respectivo sentimiento: positivo, negativo y neutro. Por último, se usó el 80% para realizar el entrenamiento.

Los tweets de entrenamiento disponen de su sentimiento el cual al entrenarlo con SVM y LSTM se debe convertir a un valor numérico que entenderá cada clasificador, en esta investigación al usar SVM se asignó 1 para tweets con sentimiento positivo, 0 para tweets con sentimiento neutro y -1 para tweets con sentimiento negativo como muestra la Tabla 3.

Tabla 3. Datos para SVM.

Id	Texto	Sentimiento
1	Antojo de empanada colombiana	0
2	Me volvieron a dejar sola	-1
3	Buenos dias, un gran abrazo	1

En el algoritmo LSMT también se debe convertir a un valor numérico el sentimiento de cada tweet, esto de acuerdo a [26] donde menciona que LSTM se componen de una secuencia de unidades, o celdas, encadenadas, donde las unidades se almacenan en forma de vector. En esta investigación se usó 1 para sentimiento positivo, 0 para neutro y 2 para negativo como muestra la Tabla 4.

Tabla 4. Datos para LSTM.

Id	Texto	Sentimiento
1	Antojo de empanada colombiana	1
2	Me volvieron a dejar sola	2
3	Buenos dias, un gran abrazo	0

Para el algoritmo de Naïve Bayes no se debió convertir el sentimiento del tweet a un valor numérico ya que el algoritmo tomará en cuenta el texto del tweet, los tokens del tweet y que serán agrupados los tweets según el sentimiento: NEU(neutro), N(negativo) y P(positivo), como se muestra en la Tabla 5.

Tabla 5. Datos para Naïve Bayes.

Id	Texto	Sentimiento
1	Antojo de empanada colombiana	NEU
2	Me volvieron a dejar sola	N
3	Buenos dias, un gran abrazo	P

2.4.2. Pruebas

Las pruebas se realizaron para clasificación a dos clases y a tres clases. Para la clasificación a dos clases se utilizaron los tweets con sentimientos positivos, negativos y para clasificación a tres clases se integrará el sentimiento neutro con los dos anteriores. Para cada uno de los algoritmos seleccionados, se tomó un conjunto de datos del 20% del corpus de tweets que se dividieron de la siguiente manera: un 10% para prueba y el otro 10% para validación.

Para verificar que los algoritmos utilizados tengan un buen desempeño a la hora de clasificar, se necesita evaluarlos con una serie de métricas que toman en cuenta las muestras clasificadas como correctas, erróneas y además las muestras clasificadas como erróneas que pueden haberse etiquetado correctamente. Las métricas utilizadas son cuatro, que van de la mano de un conjunto de elementos, una clase X y un algoritmo que clasifica si el elemento pertenece o no a esa clase:

- **Verdaderos Positivos:** es la clasificación correcta de los elementos que pertenecen a la clase X
- **Falsos Positivos**: es la clasificación incorrecta de los elementos que pertenecen a la clase X pero que en realidad no lo son.
- **Verdaderos Negativos**: es la clasificación correcta de los elementos que no pertenecen a la clase X.
- **Falsos Negativos**: es la clasificación incorrecta de los elementos que no pertenecen a la clase X pero que en realidad si pertenecen [26].

Con los estados descritos anteriormente para esta investigación se definió tres medidas para los algoritmos de clasificación seleccionados que son:

- **Exactitud (del inglés Accuracy):** medida que representa el número de elementos clasificados correctamente sobre el número total de clasificaciones realizadas [2] como muestra la ecuación (1).

$$Acurrancy = \frac{TP+TN}{TP+FN+TN+FN} \quad (1)$$

- **Precisión (del inglés Precision):** medida que representa el número de elementos que han sido detectados por el modelo como pertenecientes a la clase X y que de hecho son clasificados correctamente [27] como de clase X, como muestra la ecuación (2).

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

- **Exhaustividad (del inglés Recall):** es la medida que representa el número de elementos de la clase X clasificados correctamente [27], como muestra la ecuación (3).

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

2.4.2.1. Expansión del corpus

Teniendo en cuenta las métricas o medidas mencionadas, se decidió expandir el corpus ya que de acuerdo a [3] y las pruebas realizadas, cuando el corpus de entrenamiento y prueba es pequeño, pueden producirse errores al estimar probabilidades, debido a esto se aumentó el corpus con los tweets proporcionados por TASS se aumentó a 14.666 el corpus.

3. Resultados y Discusiones

Esta sección describe los resultados a partir de los experimentos realizados. Los experimentos fueron: I) Primer corpus con clasificación a dos clases (positivo y negativo), II) Primer corpus con clasificación a tres clases (positivo, negativo, neutro), III) Corpus extendido con clasificación a dos clases y IV) Corpus extendido con clasificación a tres clases.

3.1. Primer corpus con clasificación a dos clases

Para este experimento se usó el primer corpus de 5.147 datos, como muestra la Tabla 6 son los resultados de los tres algoritmos usados para la clasificación binaria, se puede observar que el resultado en acierto es mayor en el algoritmo de Naïve Bayes que de los demás.

Tabla 6. Resultados de métricas en algoritmos de clasificación a dos clases con primer corpus.

Algoritmo	Acurrancy (Exactitud)	Precision (Precisión)	Recall (Exhaustividad)
Naïve Bayes	76.30 %	78.69 %	71.13 %
SVM	73.41 %	75.41 %	70.41 %
LSTM	71.68 %	73.18 %	70.25 %

3.2. Primer corpus con clasificación a tres clases

En este experimento se utilizó las tres clases de sentimiento (positivo, negativo y neutro) pero los resultados fueron bajos como muestra la Tabla 7 para cada uno de los algoritmos de clasificación, esto incide en que haya mayores errores cuando clasifique un texto por su sentimiento. De acuerdo a [3] menciona que para obtener mejores resultados se debe aumentar el corpus.

Tabla 7. Resultados de métricas en algoritmos de clasificación a tres clases con primer corpus.

Algoritmo	Acurrancy (Exactitud)	Precision (Precisión)	Recall (Exhaustividad)
Naïve Bayes	56.51 %	57.12 %	54.01 %
SVM	55.97 %	53.39 %	52.41 %
LSTM	50.00 %	52.39 %	51.68 %

3.3. Corpus extendido con clasificación a dos clases

Buscando obtener mejores resultados se usó el corpus extendido con 14.666 datos, al realizar el experimento de clasificación con los tres algoritmos se obtuvo una mejora significativa como muestra la Tabla 8, en la cual los mejores resultaos fueron del algoritmo Naïve Bayes ya que en su composición usa mejor los tokens de los textos de cada tweet.

Tabla 8. Resultados de métricas en algoritmos de clasificación a dos clases con corpus extendido.

Algoritmo	Acurrancy (Exactitud)	Precision (Precisión)	Recall (Exhaustividad)
Naïve Bayes	92.61 %	94.10 %	91.59 %
SVM	75.41 %	77.41 %	75.32 %
LSTM	71.68 %	73.24 %	70.12 %

3.4. Corpus extendido con clasificación a tres clases

La razón por la cual se extendió el corpus es porque en los artículos consultados para esta investigación el corpus usado para aplicaciones de procesamiento de lenguaje natural, extracción de aspectos en opiniones textuales y análisis de sentimientos constaban de una data de entre 66,000 y 1,6 millones de tweets, otra razón fue que cuando se realizó los experimentos en esta investigación con el primer corpus se obtuvo resultados bajos para la clasificación a tres clases.

Al usar el corpus extendido con los tres sentimientos se redujo los porcentajes de las métricas en todos los algoritmos como muestra la Tabla 9, esto se debe a que estos modelos se comportan de una mejor manera cuando solo se utiliza dos clases en este caso dos sentimiento positivo y negativo, sin embargo, el porcentaje no se redujo demasiado en Naïve Bayes y SVM esto se debe a que se usó un corpus más robusto y equilibrado en cuanto al número de tweets y su respectivo sentimiento, en cambio en LSTM si se redujo de manera significativa esto por la introducción de elementos neutros en el conjunto de datos produce picos de pérdida muy altos en el modelo.

Tabla 9. Resultados de métricas en algoritmos de clasificación a tres clases con corpus extendido.

Algoritmo	Acurrancy (Exactitud)	Precision (Precisión)	Recall (Exhaustividad)
Naïve Bayes	86.46 %	89.30 %	85.11 %
SVM	59.35 %	59.25 %	57.21 %
LSTM	50.00 %	52.68 %	51.00 %

3.5. Discusión de resultados

Con los experimentos y sus resultados se demostró que el algoritmo que obtuvo mejor desempeño en cuanto a métricas fue Naïve Bayes por esta razón se seleccionó este algoritmo para esta investigación además se destaca que este algoritmo de acuerdo a [14] es el más utilizado en cuanto a aplicaciones de NLP como la categorización de artículos de noticias, el filtrado de correo no deseado y el análisis de opiniones. Esto se debe a que Naïve Bayes no solo usa el texto o tweet procesado como los otros algoritmos, sino que además para una mejor clasificación se divide en tokens y de esos tokens se elige solo verbos, adverbios y adjetivos esta composición determina de mejor manera cuando existe un sentimiento positivo, negativo o neutro en el tweet.

Los resultados obtenidos en esta investigación al realizar el experimento de clasificación a tres clases con corpus extendido fueron mayores en los algoritmos de Naïve Bayes y SVM que al compararlos con las investigaciones de [26] y [3], son mejores que a las de dichos autores además solo usan la clasificación a dos clases. Hay que destacar que al usar el algoritmo de clasificación LSTM, aunque su resultado haya sido bajo en comparación de los otros dos algoritmos de clasificación al realizar algunas pruebas con un texto, tweet clasifica de manera correcta el sentimiento al que pertenece dicho tweet o texto.

En [28] la autora menciona que usa Naïve Bayes para su investigación ya que se puede entrenar muy rápido en comparación con otros clasificadores y que funcionan bien incluso cuando no se tienen suficientes datos de entrenamiento, en esta investigación esto se puede validar con el primer experimento realizado con una data de 5.147 tweets los resultados del algoritmo son buenos a comparación de los otros algoritmos.

Por otro lado, en [3] el autor usa un corpus de 7.000 tweets para su investigación en entrenamiento y prueba de los algoritmos Naïve Bayes y SVM esto produce errores al estimar probabilidades de tres clases al mismo tiempo y decide aumentar su data a 66.000 tweets de corpus eso produce una mejora muy significativa, esto permitió en esta investigación dar un sustento de por qué se expandió el corpus a 14.666 datos ya que al realizar el experimento usando el corpus normal y clasificación a tres clases los resultados obtenidos fueron muy bajos pero cuando se usó el corpus extendido y clasificación a dos o tres clases mejoro los resultados de los algoritmos Naïve Bayes y SVM.

Los resultados en el algoritmo de LSTM usando corpus normal y extendido a tres clases fue menor que los otros dos algoritmos, de acuerdo a [26] en su investigación usa este algoritmo pero para entrenar a la red neuronal usa 1,6 millones de tweets con su respectivo sentimiento y así obtiene resultados que se acercan al 90% en exactitud, precisión y exhaustividad esto hace referencia a que para el uso de este algoritmo en futuras investigaciones se debe contar con un corpus que al menos se acerque a los datos que uso el autor antes citado.

CONCLUSIONES

Este trabajo logró la construcción de una herramienta para la minería de opiniones junto con un corpus lingüístico con tweets del idioma español latinoamericano y español castellano junto con el uso del algoritmo de Deep Learning Naïve Bayes.

El procesamiento de los tweets es de suma importancia para que los algoritmos de clasificación entiendan el sentimiento ligado a un tweet. La efectividad del algoritmo Naïve Bayes se basa en los experimentos realizados ya que obtuvo los mejores resultados frente a los otros algoritmos de clasificación esto se debe a que se complementa de mejor manera con NLP que los otros algoritmos y también por que usa tokens de los tweets procesados no todo el tweet esto hace que tenga una mayor comprensión de cuando un texto este compuesto por un sentimiento positivo, negativo o neutro.

El algoritmo Naïve Bayes obtuvo mejores resultados cuando se extendió el corpus a los 14.666 datos esto permitió que al ingresar un texto clasifique de forma correcta el sentimiento.

Uno de los mayores retos fue construir el corpus ya que muchos de estos tweets que se recolecto constaban más de links, publicidad que en si no describía el sentimiento que se esperaba del tweet por tal razón el uso de TASS fue importante ya que proporcionaba una data de tweets que cuenta con su sentimiento esto permitió construir un corpus equilibrado en cuanto al número de tweets por sentimiento.

En la actualidad existen diversas investigaciones y aplicaciones comerciales cuya premisa principal es la minería de opiniones o análisis de sentimientos a través del procesamiento natural del lenguaje para textos en inglés, pero para la presente investigación logró desarrollar una herramienta para la minería de opinión para textos en español, que permitirá el procesamiento natural del lenguaje

La mayoría de los artículos consultados para referenciar esta investigación solo usaban la clasificación a dos clases, esta investigación por otro lado utiliza la clasificación a tres clases y con los experimentos realizados se obtuvo muy bueno resultados para dicha clasificación además cuando se probó ingresando un texto o tweet clasifico en su mayoría correctamente el sentimiento del texto o tweet.

Referencias

[1] A. Moreno, «Instituto de ingenieria del conocimiento,» 23 Febrero 2021. [En línea]. Available: https://www.iic.uam.es/inteligencia/aplicaciones-procesamiento-lenguaje-natural/. [Último acceso: 04 Junio 2021].

[2] J. C. S. Sandé, «Análisis de sentimientos en twitter,» *Universidad Oberta de Catalunya*, 2018.

[3] E. A. J. Cárdenas, «ANÁLISIS DE LA RED SOCIAL TWITTER PARA LA IDENTIFICACIÓN DE PATRONES QUE GENERAN OPORTUNIDADES DE NEGOCIO EN LA CIUDAD DE GUAYAQUIL UTILIZANDO EL ENTORNO DE TRABAJO JUPYTER NOTEBOOK Y EL LENGUAJE DE PROGRAMACIÓN PYTHON,» *Universidad de Guayaquil FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS*, 2019.

[4] «Máquinas de Soporte Vectorial, Clasificador Naïve Bayes y Algoritmos Genéticos para la Predicción de Riesgos Psicosociales en Docentes de Colegios Públicos Colombianos,» *Rodolfo Mosquera Omar D. Castrillón Liliana Parra*, 2018.

[5] E. Kumar, Natural Language Processing, India: I. K. International Pvt Ltd, 2011.

[6] I. G. L. y. J. V. R. MuÑoz, «EL PROCESAMIENTO DEL LENGUAJE NATURAL APLICADO AL ANÁLISIS DEL CONTENIDO DE LOS DOCUMENTOS,» *Universidad de Murcia*, 2018.

[7] M. A. C. Vásquez, M. H. V. Huerta y L. J. P. Quispe, «Procesamiento de lenguaje natural,» *Revista de Ingeniería de Sistemas e Informática*, 2009.

[8] L. Dubiau, «Procesamiento de Lenguaje Natural en Sistemas de Analisis de Sentimiento,» *Universidad de Buenos Aires- Facultad de Ingeniería*, 2013.

[9] M. E. C. RIVERA, «RECONOCIMIENTO DE AGRESIÓN VERBAL EN TWITTER CON EL USO DE PATRONES LENGUISTICOS,» *PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO FACULTAD DE INGENIERÍA*, 2017.

[10] M. B. Hernández y J. M. Gómez, «Aplicaciones de Procesamiento de Lenguaje Natural,» *Revista Politécnica*, 2013.

[11] A. Mañas, Notas sobre pronóstico del flujo de tráfico en la ciudad de Madrid, Madrid: BookDown, 2019.

[12] C. Nicholson, «Una guía para principiantes sobre LSTM y redes neuronales recurrentes,» *Wiki de IA*, 2020.

[13] S. Hochreiter, «Long Short-Term Memory.” Neural Computation,» 1997.

[14] A. I. Yañez, «Sistema Deep Learning para el análisis de sentimientos en opiniones de productos para la ordenación de resultados de un buscador semántico,» *Universidad de Coruña*, 2019.

[15] L. A. G. Dionis López Ramos, «Aprendizaje profundo para la extracción de aspectos en opiniones textuales,» *Revista Cubana de Ciencias Informáticas*, 2019.

[16] J. Durán, «Todo lo que Necesitas Saber sobre el Descenso del Gradiente Aplicado a Redes Neuronales,» MetaDatos, 04 Septiembre 2019. [En línea]. Available: https://medium.com/metadatos/todo-lo-que-necesitas-saber-sobre-el-descenso-del-gradiente-aplicado-a-redes-neuronales-19bdbb706a78.

[17] TASS 2020, «TASS 2020,» [En línea]. Available: http://tass.sepln.org/2020/. [Último acceso: 17 Mayo 2021].

[18] Pavloski, Mihajlo, «Accessing the Twitter API with Python,» stackabuse, 2019. [En línea]. Available: https://stackabuse.com/accessing-the-twitter-api-with-python/. [Último acceso: 17 Mayo 2021].

[19] M. Mayo, «Preprocesamiento de datos de texto: un tutorial en Python,» [En línea]. Available: https://medium.com/datos-y-ciencia/preprocesamiento-de-datos-de-texto-un-tutorial-en-python-5db5620f1767. [Último acceso: 18 Mayo 2021].

[20] F. Murzone, «Procesamiento de Lenguaje Natural: Stemming y Lemmas,» *Medium*, 2020.

[21] P. R. S. Christopher D. Manning, An Introduction to Information Retrieval, England: Cambridge UP, 2009.

[22] W. J. T. y. A. Crymble, «Normalizar datos de texto con Python,» 03 Julio 2012. [En

línea]. Available: https://programminghistorian.org/es/lecciones/normalizar-datos. [Último acceso: 21 Mayo 2021].

[23] A. Bracco, «Normalización de Texto en Español de Argentina,» *Universidad Nacional de Córdoba,* 2017.

[24] Samuel, «Introducción al procesamiento del lenguaje natural en Python – SitePoint,» 08 Abril 2019. [En línea]. Available: https://stips.wordpress.com/2019/04/08/introduccion-al-procesamiento-del-lenguaje-natural-en-python-sitepoint/. [Último acceso: 21 Mayo 2021].

[25] G. Parodi, «LINGÜISTICA DE CORPUS: UNA INTRODUCCION AL AMBITO,» *RLA. Revista de lingüística teórica y aplicada,* 2008.

[26] A. P. SanJuan, «Aplicación para el anáisis de sentimientos y tendencias en redes sociales,» *Universitat Politécnica de Valencia - Campus de Alcoi,* 2019.

[27] G. G. E. M. d. L. M. Fuentes y T. A. R. d. Real, «Un modelo basado en el Clasificador Naïve Bayes para la evaluación del desempeño docente,» *Iberoamaerica de educación a distancia,* 2017.

[28] V. L. C. Alvarado, «CLASIFICACIÓN DE TWEETS MEDIANTE MODELOS DE APRENDIZAJE SUPERVISADO,» *NIVERSIDAD COMPLUTENSE DE MADRID,* 2018.

[29] Twitter Developer, «Twitter Developer,» [En línea]. Available: https://developer.twitter.com/en/docs. [Último acceso: 17 Mayo 2021].

[30] eiki, «Natural Language Processing: Naive Bayes Classification in Python,» Marzo 2019. [En línea]. Available: https://medium.com/@eiki1212/natural-language-processing-naive-bayes-classification-in-python-e934365cf40c. [Último acceso: 24 Mayo 2021].

[31] D. J. &. J. H. Martin, «Naive Bayes and Sentiment,» *Speech and Language Processing,* 2020.

[32] A. Rai, «Text Classification in NLP — Naive Bayes,» Julio 2017. [En línea]. Available: https://theflyingmantis.medium.com/text-classification-in-nlp-naive-bayes-a606bf419f8c. [Último acceso: 25 Mayo 2021].

[33] Sitio Big data, «Sitio Big data,» Abril 2019. [En línea]. Available: https://sitiobigdata.com/2019/12/24/clasificacion-de-aprendizaje-automatico-supervisado/. [Último acceso: 25 Mayo 2021].

[34] B. Ripley, Pattern Recognition and Neural Networks, Cambridge, 2007.