

# Análisis de Sentimientos en texto mediante SVM y Redes Neuronales Convolucionales

Jhonathan Daniel Abreu Luque

*Escuela de Ingeniería de Sistemas*

*Universidad de Los Andes*

Mérida, Venezuela

jd.abreu25@gmail.com

**Resumen**—Las emociones son un componente semántico primario en la interacción humano-humano y juegan un papel sumamente importante en el comportamiento humano. Por tal motivo, la presente investigación se enfoca en el estudio de diferentes técnicas de representación numérica de documentos de texto y de clasificación de los mismos en base al sentimiento de evocar. Se comparan dos técnicas de vectorización, Word2Vec y el modelo clásico Bag-Of-Words para preparar la data que alimenta a clasificadores de tipo SVM y Redes Neuronales Convolucionales. Los datos utilizados para el entrenamiento y prueba fueron recolectados como parte de la investigación. Los resultados del estudio muestran que, para cantidades pequeñas de datos, modelos sencillos de vectorización, como Bag-Of-Words, se desempeña mejor que otros más complejos. Los clasificador que generó los mejores resultados fueron las Redes Neuronales Convolucionales (con exactitud de pruebas de 81.3%), sobre todo cuando se toma en cuenta mayor cantidad de información contextual en el proceso de vectorización de los documentos.

**Palabras clave**—Análisis de sentimientos, vectorización, clasificación, Redes Neuronales Convolucionales, SVM, Bag-Of-Words.

## I. INTRODUCCIÓN

Imagina un mundo en el que los humanos solo tengan las palabras como medio de comunicarse; un mundo en el que las expresiones faciales y corporales no jueguen ningún papel en la expresividad de las personas. Ahora intenta ponerte en el lugar de aquellas personas que viven con limitaciones motoras y cognitivas que les impiden expresar lo que piensan y sienten más allá de las palabras.

Las emociones son un componente semántico primario en la interacción humano-humano y juegan un papel sumamente importante en el comportamiento humano. Las emociones, al igual que el pensamiento, guían las acciones y los diálogos. Es por esto que una de las aplicaciones de gran importancia que puede darse al análisis de sentimiento es la robótica social, en la que se necesitan robots que sirvan como extensiones para los cuerpos de las personas con limitaciones motoras que les impidan expresarse por canales diferentes al habla; módulos que permitan a los robots detectar las emociones de los usuarios, de modo que puedan responder a los mismos y adaptarse a la interacción; robots que puedan expresar emociones para generar empatía. Sin embargo, no es el único campo de acción de esta disciplina.

Analizar la creciente cantidad de reseñas de productos permitiría a las empresas detectar potenciales clientes y sus

preferencias; analizar reseñas de películas permitiría la automatización de la clasificación de grandes bases de datos de cine internacional; detectar el sentimiento de una conversación permitiría que los avatares en juegos en línea muestren emociones en tiempo real, aumentando así la expresividad de las conversaciones humanas, incluso en ámbitos sociales remotos. Estas y muchas otras áreas forman el amplio campo de aplicación del análisis de sentimientos.

El presente artículo pretende sentar las bases para el desarrollo de un framework de detección de emociones de texto para una futura integración en un avatar robótico, de modo que se vea dotado de nuevos canales de expresión e interactividad. Para esto, se requiere analizar métodos de representación de las oraciones de modo que puedan ser alimentadas a un clasificador y, además, realizar un estudio comparativo que permita detectar el clasificador más apto para este fin entre las SVM y las Redes Neuronales Convolucionales.

El resto de este artículo, entonces, se organiza de la siguiente manera: en la Sección II se presenta el trabajo relacionado que ya ha sido realizado en el área del análisis de sentimientos de texto. La Sección III describe la metodología utilizada, con una breve introducción a los vectorizadores y los clasificadores a estudiar. La Sección IV muestra las pruebas realizadas y el análisis de los resultados obtenidos. Por último, la Sección V presenta las conclusiones de la investigación y los pasos siguientes para la continuación del proyecto.

## II. TRABAJO RELACIONADO

El análisis de sentimientos juega un papel realmente importante en la robótica social, puesto que en la interacción humano-robot es necesario que las máquinas tengan la capacidad de detectar y expresar emociones y/o sentimientos para mejorar la interacción. Sin embargo, la interacción humano-robot no es el único campo de aplicación del análisis de sentimientos, puesto que actualmente se ha convertido en uno de los campos de investigación más activos en las áreas de los negocios, marketing, toma de decisiones, campañas políticas y otras afines [1].

Debido a que la emoción es un componente semántico primario en la comunicación humana, los robots sociales deben ser capaces de expresar emociones en la comunicación interactiva. Se han desarrollado métodos de generación de expresiones faciales a partir de texto en los que se han

utilizado SVM para clasificar o mapear oraciones a la emoción que esta representa [2] — siendo estas solo cuatro de las seis emociones básicas [3]. Para la representación vectorial de las palabras utilizaron la herramienta Word2Vec, la cual toma un corpus grande como entrada y produce vectores de palabras como salida. Proponen, además, un método para utilizar la representación vectorial de las palabras para obtener la representación vectorial de las oraciones y, por motivo de comparación, utilizan también el modelo de aprendizaje automático Latent Dirichlet Allocation (LDA). Obtuvieron los mejores resultados cuando la dimensión de los vectores en la salida del modelo Word2Vec era 400 pero, sin embargo, no obtuvieron buenos resultados de las SVM (33% de precisión), alegando esto a la dispersión de los datos de alta dimensión. Según los resultados de dicha investigación, es posible que el desempeño de vectorizadores de tipo Word2Vec no sea el mejor para bases de datos pequeñas, siendo preferible elegir métodos menos complejos.

Lejos del campo de la robótica social, investigadores han utilizado las SVM para realizar análisis de sentimientos en el campo de la minería de opiniones, específicamente en microblogs de finanzas de China [4]. Las SVM, en este caso, son utilizadas para extraer la polaridad de las opiniones referentes a un mismo tópico, las cuales son representadas de forma vectorial según un lexicón de palabras características de sentimientos. Para el caso de microblogs de diferentes tópicos, proponen un método basado en reglas. La principal conclusión de dicha investigación es que, como lo indicaba su hipótesis, las SVM son más aptas para clasificar microblogs de un mismo tópico, alcanzando hasta un 81.75% de precisión. Un hecho que se destaca y puede tomarse de la investigación anteriormente mencionada, es que las SVM se desempeñan mejor cuando los datos provienen de un mismo tópico, por lo cual, es posible que sea necesario estudiar otros clasificadores para datos mas generalizados.

Las SVM también son utilizadas para detectar la orientación del sentimiento (positivo, negativo o neutral) de opiniones o reseñas de usuario sobre películas. Un estudio comparativo mostró que, en este caso, las SVM lineales proveen un mejor desempeño que los clasificadores Bayesianos Ingenuos, alcanzando hasta un 87.5% de precisión [5] en opiniones sobre películas de la categoría drama. Una cosa interesante de esta investigación es el uso del *stemming* (conversión de las palabras a su raíz) como parte del procesamiento de los datos previo al entrenamiento del clasificador; vale la pena estudiar este procedimiento y su utilidad en el Procesamiento de Lenguaje Natural y en análisis de sentimientos.

Cabe mencionar que, además del gran potencial de las SVM para el análisis de sentimientos, la literatura muestra que las Redes Neuronales Convolucionales son otro método de aprendizaje supervisado altamente utilizado en la actualidad para este fin. Se han alcanzado porcentajes de precisión de 80.69% para análisis de sentimientos en texto [6], 85.2% utilizando el método fastText de Facebook para obtener la representación vectorial de las palabras [7] y 99.87% para texto escrito en lenguaje Bangla [8].

Lo que indican las investigaciones ya realizadas en este campo es que no existe un único tipo de clasificador óptimo para realizar análisis de sentimientos, pues puede también depender del origen de los datos y la forma en la que estos se representan. Esto apunta a la necesidad de, probablemente, realizar estudios comparativos sobre los posibles clasificadores posibles que puedan ser utilizado, principalmente debido a los limitados conocimientos que se tienen en el área de la computación inteligente. Adicionalmente, destaca también el hecho de que la vectorización del texto puede realizarse de muchas maneras, también dependiendo de la naturaleza de los datos. Debido a la poca cantidad de datos disponibles para esta investigación, es posible que la utilización de vectorizadores simples como Bag-Of-Words se imponga sobre vectorizadores más complejos como Word2Vec o fastText.

### III. METODOLOGÍA

Con la finalidad de clasificar o *mapear* una frase u oración a un sentimiento, se propone un conjunto de operaciones o un método para llevar las frases a un dominio apto para la alimentación de un modelo de aprendizaje automatizado que clasifique cada oración en una clase particular. En lugar de emociones específicas, en este proyecto se trata con la polaridad de los sentimientos, agregando además la posible *neutralidad* de una oración, quedando entonces tres clases puntuales: *positivo* (felicidad, sorpresa, emoción, entre otros), *negativo* (tristeza, enojo, miedo, entre otros) y *neutral*.

#### A. Limpieza de la oración

El proceso para transformar las oraciones a un dominio apto para los clasificadores comienza con su limpieza. Este procedimiento depende del idioma que se esté procesando y para esto se utilizan algunas técnicas del Procesamiento de Lenguaje Natural (NLP). El proceso se resume a continuación:

1) *Tokenización*: Esta etapa consiste sencillamente en llevar una cadena de caracteres a una lista o arreglo de cadenas, siendo cada una de estas una palabra de la oración. Esto es necesario porque para llevar las oraciones a un dominio vectorial, primero se debe realizar este procedimiento para cada palabra para luego utilizar algún método aritmético o algebraico para unir las en la oración que estas conforman.

2) *Normalización*: En esta etapa, se llevan a cabo tres sub-tareas principales: la conversión de cada letra de los *tokens* a minúsculas, la eliminación de todos los acentos (incluyendo las virgulillas de las letras ñ) y la exclusión de los símbolos de puntuación y dígitos numéricos. Este procedimiento se realiza con el fin de estandarizar el corpus de entrenamiento y, de cierta forma, lidiar con los errores de puntuación y acentuación.

3) *Eliminación de palabras vacías*: Se denominan palabras vacías o *stop-words* a un conjunto de palabras que no tienen un significado semántico importante a la hora de realizar el análisis de sentimientos y procesamiento de lenguaje

TABLA I  
REPRESENTACIÓN VECTORIAL DEL CONJUNTO DE DOCUMENTOS U  
ORACIONES

|       | $t_1$         | $t_2$         | ... | $t_j$         | ... | $t_n$         |
|-------|---------------|---------------|-----|---------------|-----|---------------|
| $d_1$ | $\alpha_{11}$ | $\alpha_{12}$ | ... | $\alpha_{1j}$ | ... | $\alpha_{1n}$ |
| $d_2$ | $\alpha_{21}$ | $\alpha_{22}$ | ... | $\alpha_{2j}$ | ... | $\alpha_{2n}$ |
| ...   | ...           | ...           | ... | ...           | ... | ...           |
| $d_i$ | $\alpha_{i1}$ | $\alpha_{i2}$ | ... | $\alpha_{ij}$ | ... | $\alpha_{in}$ |
| ...   | ...           | ...           | ... | ...           | ... | ...           |
| $d_m$ | $\alpha_{m1}$ | $\alpha_{m2}$ | ... | $\alpha_{mj}$ | ... | $\alpha_{mn}$ |

natural en general. Se incluyen en este conjunto los artículos, pronombres, preposiciones, etc. Cabe destacar que no existe una lista definitiva de palabras vacías y se puede utilizar cualquiera disponible en corpus o bibliotecas y frameworks de NLP.

4) *Stemming*: El proceso de *stemming* finaliza la limpieza de las oraciones y consiste en la reducción de las palabras (flexionadas o derivadas) a su raíz o *stem*. A modo de ilustración, este proceso permite que se puedan identificar las palabras “bibliotecas”, “bibliotecario” y “bibliotecarias” a partir de su raíz “biblioteca” o simplemente “bibliotec”. De este modo, es posible ampliar el rango de identificación de las palabras sin necesidad de construir un vocabulario con todas las palabras y sus posibles formas.

#### B. Representación vectorial de las oraciones

La extracción de características o *feature extraction* consiste en extraer las características de las oraciones en vectores numéricos, pues la mayoría de los algoritmos esperan vectores de características numéricas de tamaños fijos en lugar de documentos en crudo de longitud variable.

En este contexto, se presenta una estrategia de extracción de características útil para vocabularios pequeños. En este, se genera un vocabulario con, posiblemente, todas las palabras que aparecen en los documentos de entrada. Es posible limitar el vocabulario a aquellos *tokens* con frecuencias de aparición en cierto intervalo. La representación de una lista de oraciones será, entonces, una matriz de unos y ceros de en la que las filas representan las oraciones y las columnas los *tokens* del vocabulario, como se muestra en la Tabla I. De esta manera, un  $\alpha_{ij} = 1$  representa que la oración o documento  $i$  contiene el *token*  $j$ .

De esta manera, si se tiene un el vocabulario {“oración”, “primera”, “es”, “frase”, “esta”, “una”, “escribe”}, los documentos “Esta es una pequeña oración” y “Escribe dos frases cortas”, se representarían con los vectores [1, 1, 1, 0, 1] y [1, 0, 1, 0], respectivamente. No obstante, la oración “¿Es esta una oración pequeña?”, se representa de con el mismo vector que “Esta es una pequeña oración”, pues no se considera ningún tipo de orden relativo entre las palabras y, por consiguiente, se pierde cierta información contextual.

En este sentido, es posible entonces utilizar el concepto de bigrama, con el cual se puede conservar parte del sentido contextual de los documentos al considerar *tokens* de longitud menor o igual que 2. Como ejemplo, si el vocabulario anterior

se construyó con un único documento “esta es una de las primeras oraciones o frases que escribe” (siendo el caso hipotético en que las palabras “de”, “las”, “o” y “que” fueron eliminadas por ser consideradas palabras vacías), entonces el nuevo vocabulario, al utilizar bigramas, sería {“esta”, “es”, “una”, “primera”, “oracion”, “frase”, “escribe”, “esta es”, “es una”, “una primera”, “primera oración”, “oración frase”, “frase escribe”} y los vectores tendrían dimensión 13. Utilizando este nuevo vocabulario, las oraciones “Esta es una pequeña oración” y “¿Es esta una oración pequeña?” tendrían representaciones vectoriales diferenciadas. Este concepto puede expandirse y considerarse n-gramas en lugar de bigramas.

Existen, además, otros métodos de extracción de características para texto, como lo son el *tf-idf* y el *Word2Vec*, que permiten asociar palabras con significados semánticos similares en “nubes” de vectores o dar pesos a las palabras con más importancia en cada documento. No obstante, estos son útiles y más aptos para conjuntos de datos bastante amplios que contengan vocabularios de mayor tamaño.

#### C. Clasificación de las oraciones

Existen diversos tipos de clasificadores en el mundo del aprendizaje automático. Serán considerados, en este caso, dos de estos, por ser muy utilizados en aplicaciones de clasificación de texto y análisis de sentimientos. Las Máquinas de Vectores de Soporte (SVM) y las Redes Neuronales Convolucionales (RNC) son clasificadores totalmente distintos, siendo el segundo muy utilizado para el reconocimiento y análisis de imágenes, pues se basa, muy someramente, en la búsqueda de patrones bidimensionales; sin embargo, este concepto se puede abstraer para análisis en vectores unidimensionales. Las SVM, por otro lado, se basan en conceptos matemáticos e intentan encontrar el hiperplano que mejor separe las clases a clasificar. Se está frente a dos formas totalmente diferentes de clasificación y se esperarían, entonces, resultados bien diferenciados.

### IV. EXPERIMENTOS

Los experimentos fueron llevados a cabo utilizando la plataforma Colaboratory de Google [9], en un entorno que utiliza GPU y Python 3. Los clasificadores utilizados se encuentran de forma pública en bibliotecas de Python y se utilizaron bases de datos públicas o construidas.

#### A. Base de datos

Ningún tipo de base de datos existente fue utilizado para el desarrollo de este proyecto. En su lugar, se llevó a cabo un proceso de captura de datos diseñado para ser lo más general posible — es decir, útil para representar frases típicas de una conversación común — para lo cual se creó un formulario de Google Forms, en el cual se solicitaba la información requerida mediante las siguientes solicitudes:

- 1) Escriba una frase que exprese un sentimiento positivo.
- 2) Escriba una frase que exprese un sentimiento negativo.
- 3) Escriba una frase que exprese un sentimiento neutral.

El formulario fue distribuido a diversas personas, no solo de la comunidad estudiantil, sino también a personas ajenas al proyecto, como familiares y amigos. De esta manera, se pudo obtener una base de datos compuesta de 600 frases u oraciones, distribuidas equitativamente entre *Positivas*, *Negativas* y *Neutrales*.

### B. Medida de rendimiento

Para realizar la comparación entre los clasificadores y obtener los hiper-parámetros óptimos, se utilizó la métrica denominada precisión de la validación, la cual indica el porcentaje de predicciones correctas echas sobre la data de prueba y apunta a optimizar la generalización de las predicciones echas por el clasificador en datos completamente diferentes a los utilizados para su entrenamiento. Optimizar esta métrica es útil en casos en los que se desea acertar la mayor cantidad de veces que sea posible, como es requerido en el presente proyecto.

### C. Resultados de los experimentos

Se realizaron implementaciones con SVM y RNC, utilizando vectorizadores de tipo Bag-Of-Words y Word2Vec, para hacer comparaciones. Para clasificar en tres clases diferentes (positivo, negativo y neutral), se espera que el mínimo porcentaje de exactitud sea 66,66%, de modo que las predicciones sean mejor que aleatorias. Los resultados indican que las SVM con el vectorizador Word2Vec no logran el desempeño mínimo requerido (ver Figura 1), incluso para distintos tamaños de los vectores que representan los documentos. Una de las razones que podrían constatar estos resultados es el tamaño del dataset utilizado, pues es requerido, para los SVM, que la cantidad de datos sea la suficiente para obtener del modelo una generosa habilidad para generalizar, sobre todo para los modelos no lineales. Otro factor importante es el uso de Word2Vec para vectorizar los documentos, pues este modelo también requiere de mayor cantidad de datos para mejorar su rendimiento.

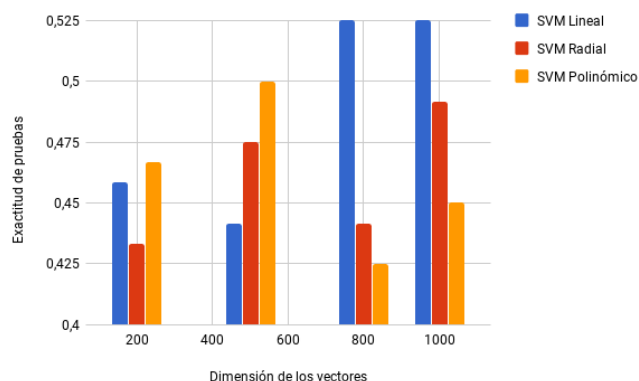


Fig. 1. Comparación del desempeño de los SVM con Word2Vec como vectorizador con distintas dimensiones de vectores.

Del mismo modo, el uso de un vectorizador de tipo Bag-of-Words no logró mucho como intento de mejorar el desempeño

de las SVM (Figura 2). Esto apunta a que el pobre desempeño de las SVM como clasificador para los datos siendo utilizados puede deberse al tamaño del dataset más que al tipo de vectorizador en uso.

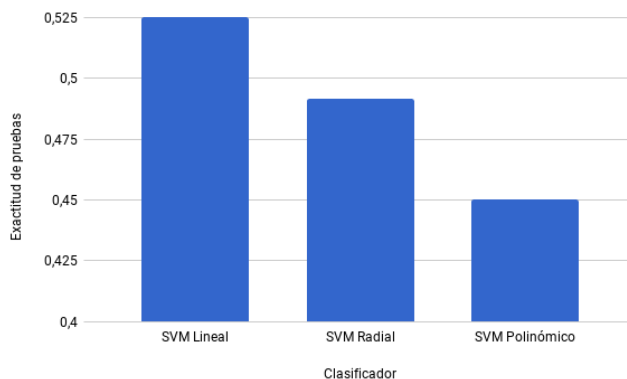


Fig. 2. Comparación del desempeño de los SVM con vectorizador de tipo Bag-Of-Words.

Por otro lado, la Red Neuronal Convolutiva alcanzó la exactitud mínima en todos los casos (Figura 3), tanto para el vectorizador Word2Vec con distintos tamaños de vectores, como con el vectorizador Bag-Of-Words, con el cual se alcanzó el mejor desempeño (78,9%), una diferencia bastante marcada respecto al uso de Word2Vec. Con lo anterior se puede inferir, entonces, que el vectorizador Word2Vec podría desempeñarse mejor si, tal como se expresa en la literatura, se utiliza un dataset con una cantidad de datos suficiente tanto para el vectorizador como para el clasificador.

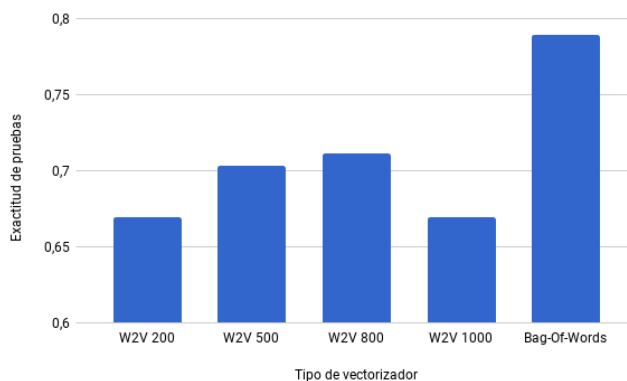


Fig. 3. Comparación del desempeño de los vectorizadores con el clasificador de tipo Red Neuronal Convolutiva.

Adicionalmente, las pruebas realizadas para comparar el desempeño del vectorizador Bag-Of-Words con la Red Neuronal Convolutiva, para palabras, bigramas, 3-gramas, 4-gramas y 5-gramas, muestran una ligera mejora en el desempeño del clasificador cuando se toma más información contextual de las oraciones para su vectorización — específicamente para 4-gramas (Figura 4). Sin embargo, el comportamiento es similar,

en el caso del dataset disponible, para todos los  $n$ -gramas tomados en consideración, lo cual puede deberse a la longitud limitada de las oraciones en la base de datos.

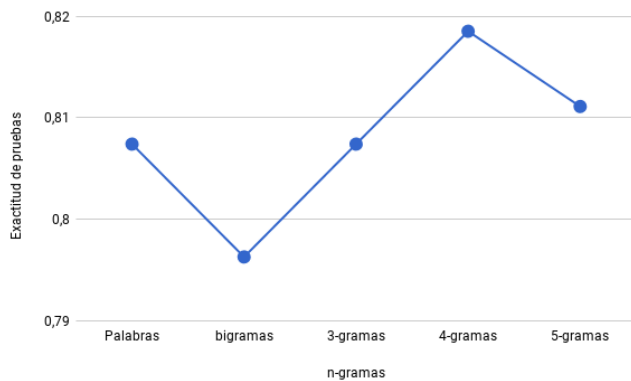


Fig. 4. Comparación del desempeño de las RNC con vectorizador Bag-Of-Words para distintos  $n$ -gramas.

La Figura 5 muestra la matriz de confusión de las pruebas de la Red Neuronal Convolutiva más apropiada que pudo encontrarse, utilizando 4-gramas, la cual indica, entre otras cosas, la cantidad de falsos positivos de un conjunto de clasificaciones. En esta se puede observar que, para la clase de oraciones con sentimiento *neutral*, existe una discrepancia considerable, con solo un 50% de oraciones correctamente clasificadas, lo cual puede deberse a que las personas participantes de la recolección de datos no entendieran exactamente lo que se considera un sentimiento neutral.

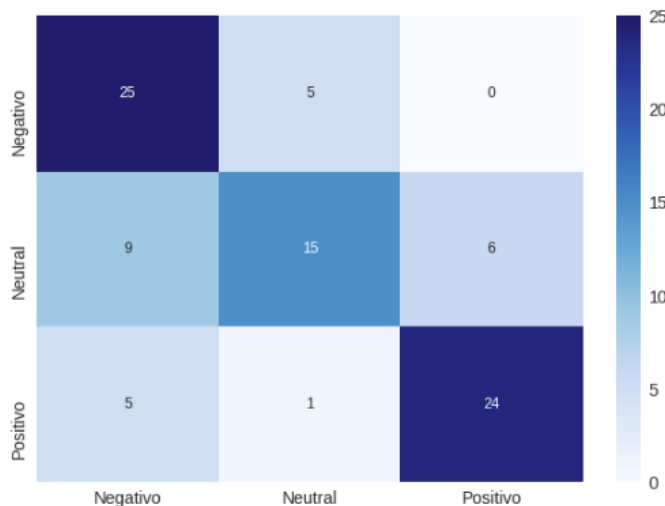


Fig. 5. Matriz de confusión de la evaluación del modelo de Red Neuronal Convolutiva con 4-gramas.

## V. CONCLUSIONES

El objetivo de esta investigación es estudiar y comparar dos tipos de vectorizadores y dos tipos de clasificadores para la identificación de la orientación del sentimiento de documentos

(texto). Las aplicaciones son bastas, siendo una de estas la implementación de detección de emociones, en un canal, para un robot social. Este canal puede ser el audio, el cual, luego se ser procesado y transformado a texto, puede ser clasificado por el módulo en cuestión.

Debido a la poca cantidad de datos disponibles, fue necesario realizar pruebas con distintos tipos de vectorizadores, las cuales apuntan a que un modelo simple, como el Bag-Of-Words, se desempeña mejor para pocos datos que vectorizadores más complejos, como el Word2Vec. Del mismo modo y, probablemente por el mismo motivo, las RNC mostraron un rendimiento superior a las SVM, sobre todo cuando se toma en cuenta más información contextual en el proceso de vectorización.

Se cree que, si se procede a la recaudación de más datos, siguiendo el mismo procedimiento para la recolección de los actualmente disponibles, se pueda mejorar el desempeño de los clasificadores para obtener mejores resultados. Sin embargo, se debe también estudiar a fondo la teoría detrás de las RNC para, de este modo, generar una arquitectura más adecuada para este tipo de aplicación. Se delega como trabajo futuro el proceso de expansión de la base de datos y el desarrollo de una aplicación que permita realizar la conversión audio-texto para sentar las bases para un módulo integrable de detección de emociones para un robot social.

## REFERENCIAS

- [1] S. M. Alzahrani, "Development of IoT mining machine for Twitter sentiment analysis: Mining in the cloud and results on the mirror," in *2018 15th Learning and Technology Conference (L T)*, Feb 2018, pp. 86–95.
- [2] Y. Bai, X. Tang, Y. Zhu, and D. Cai, "A Sentiment Analysis Method for Facial Expression Generation in Human-Robot Interactive Communication," in *2014 International Conference on Virtual Reality and Visualization*, Aug 2014, pp. 97–102.
- [3] K. R. Scherer, "Nonlinguistic vocal indicators of emotion and psychopathology," in *Emotions in personality and psychopathology*. Springer, 1979, pp. 493–529.
- [4] D. Yan, B. Hu, and J. Qin, "Sentiment Analysis for Microblog Related to Finance Based on Rules and Classification," in *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Jan 2018, pp. 119–126.
- [5] S. Rana and A. Singh, "Comparative analysis of sentiment orientation using SVM and Naive Bayes techniques," in *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, Oct 2016, pp. 106–111.
- [6] A. Chachra, P. Mehndiratta, and M. Gupta, "Sentiment analysis of text using Deep Convolution Neural Networks," in *2017 Tenth International Conference on Contemporary Computing (IC3)*, Aug 2017, pp. 1–6.
- [7] I. Santos, N. Nedjah, and L. de Macedo Mourelle, "Sentiment analysis using Convolutional Neural Network with fastText embeddings," in *2017 IEEE Latin American Conference on Computational Intelligence (LACCI)*, Nov 2017, pp. 1–5.
- [8] M. H. Alam, M. M. Rahoman, and M. A. K. Azad, "Sentiment analysis for Bangla sentences using Convolutional Neural Network," in *2017 20th International Conference of Computer and Information Technology (ICCIT)*, Dec 2017, pp. 1–6.
- [9] Google, "Colaboratory," <https://research.google.com/colaboratory/>.