APLICACIÓN DE UNA RED NEURONAL ARTIFICIAL PARA LA CLASIFICACIÓN AUTOMÁTICA DE TUITS EN ESPAÑOL

ARTIFICIAL NEURAL NETWORK APPLICATION FOR AUTOMATIC CLASSIFICATION OF TWEETS IN SPANISH

Andrea Gidalti García Pérez
Universidad Autónoma de Tamaulipas
a2173018004@alumnos.uat.edu.mx

Ana Bertha Ríos Alvarado
Universidad Autónoma de Tamaulipas
arios@docentes.uat.edu.mx

Edgar Tello Leal
Universidad Autónoma de Tamaulipas
etello@docentes.uat.edu.mx

José Hugo Barrón Zambrano
Universidad Autónoma de Tamaulipas
hbarron@docentes.uat.edu.mx

Alan Díaz Manríquez
Universidad Autónoma de Tamaulipas
amanriquez@docentes.uat.edu.mx

Resumen

Las plataformas sociales como Twitter se han convertido en formas muy populares de transmitir información. Los usuarios de Twitter crean y usan *hashtags* en sus tuits para categorizarlos de acuerdo a un tema y dar su opinión al respecto, permitiendo crear tendencias mediante *hashtags,* así como agrupar y vincular la información con otros usuarios a través de búsquedas. En este trabajo se propone un modelo de clasificación de tuits en español para tres clases: 1) Situación de riesgo (SDR), 2) Servicio de agua y 3) Política, mediante la implementación de una red neuronal de tipo perceptrón multicapa. Para la recolección de tuits se han utilizado las etiquetas o palabras clave que representan los temas mencionados. Adicionalmente, se implementó un modelo de clasificación bayesiano para evaluar

y comparar su desempeño mediante validación cruzada de k grupos. Los resultados muestran que la red neuronal presenta mejor exactitud en la clasificación de tuits en español.

Palabras Claves: Clasificación de texto, red neuronal artificial, tuits.

Abstract

Social platforms *like Twitter have become very popular ways of transmitting information. Twitter users create and use hashtags in their tweets to categorize them according to a topic and give their opinions about it, allowing to create trends through hashtags, as well as grouping and linking the information with other users through searches. In this paper we propose a classification model for Twitter publications in Spanish about three classes: 1) Risk situation, 2) Water service and 3) Politics through the implementation of a multilayer perceptron, a type of artificial neural network. For collecting tweets, the hashtags or keywords that represents mentioned topics were used. Additionally, a classification model based on the Bayes theorem was implemented to evaluate and compare its performance by k-fold cross-validation. The results show that the neural network presents better accuracy in the classification of Spanish tweets.*

Keywords: *Artificial neural network, text classification, tweets.*

1. Introducción

Las redes sociales se han vuelto un medio imprescindible para compartir y consultar información de temas de interés en tiempo real. De acuerdo con un estudio realizado en 2017 por la Asociación Mexicana de Internet[1] sobre los hábitos de los usuarios de Internet en México, se reporta que la principal actividad en línea es el acceso a redes sociales. En particular, a través de redes sociales se accede a información relacionada con los actores políticos y sus propuestas, así como información de eventos de interés social como el acceso a servicios públicos y de seguridad. Twitter[2] es una de las redes sociales con mayor actividad entre los

[1] http://www.asociaciondeinternet.mx
[2] https://twitter.com

usuarios de Internet, pero aún no hay soluciones contundentes para la verificación de cuentas de usuarios que se apeguen a los lineamientos de legitimidad y legalidad. En consecuencia, es posible publicar información ambigua o errónea acompañada de las etiquetas comunes llamadas *hashtags* (por ejemplo, #política, #SDR). En temas como seguridad esto deriva en falsas alarmas e incertidumbre entre la sociedad. En este documento se propone una aplicación que verifique el tema en tuits con la cual se pueda tener una certeza de que el tuit está asociado al tema de la etiqueta (*hashtag*) que contiene ese tuit.

Entre las técnicas de aprendizaje automático que se han usado para el tratamiento de tuits están las Redes Neuronales Artificiales (RNA). Las RNA son una técnica que ha generado gran interés desde su aparición debido a que pueden ser aplicadas para resolver problemas relacionados con las tareas de clasificación, reconocimiento de imágenes, reconocimiento de patrones, reconocimiento de voz, entre otras [Haykin, 1998] [Cheng, 1994]. Para el tratamiento de tuits, debido al gran volumen de información que se genera en cada momento, las redes neuronales han sido usadas en tareas como análisis de sentimientos [Duncan, 2015], [Jianqiang, 2018], [Rosenthal, 2017] y se han presentado estudios que abordan la diversidad en técnicas y herramientas de acceso a los tuits [Rosá, 2017], [Pla, 2013]. Otro de los intereses entre la comunidad ha sido la detección del lenguaje de los tuits [Wehrmann, 2018], y la identificación del perfil del autor [Shrestha, 2017], su género, edad, entre otras características [Martinc, 2017], [Rangel, 2013]. En algunos casos se han enfocado en el análisis de tuits de un tema en particular como en el dominio del cuidado de la salud [Kuang, 2017] donde se hace una clasificación sobre si un tuit está relacionado o no con ese tema. Una de las ventajas de usar una RNA como modelo de clasificación es que ha mostrado buenos resultados en la clasificación multiclase.

En este trabajo se implementa una RNA como modelo de clasificación para asignar una clase a un tuit, se ha considerado como caso de estudio una muestra de las publicaciones de los temas: 1) SDR, 2) Servicio de agua y 3) Política.

Dicho tuit puede contener alguna palabra clave, palabra relacionada o un *hashtag*. Las cuentas de donde se han recolectado los tuits pertenecen a usuarios con

cuentas habilitadas en el estado de Tamaulipas, México. Además, se ha construido una aplicación web que permite la clasificación automática de un tuit escrito por un usuario común.

En este trabajo se utiliza un perceptrón multicapa como clasificador principal, además, dicho modelo es comparado con un clasificador bayesiano ingenuo. Los modelos mencionados tienen algunas características que los diferencian, los clasificadores bayesianos son conocidos por tener un buen desempeño con conjuntos pequeños de datos, son más sencillos de comprender e implementar, es por eso que son populares y en muchas ocasiones son utilizados para medir el desempeño de otros clasificadores (como en este caso). Por otra parte, las redes neuronales debido a su complejidad se adaptan muy bien a conjuntos grandes y son utilizadas para una amplia variedad de tareas, además de tener la capacidad de modelar problemas no lineales y complejos, realizar inferencias a partir de la información proporcionada y trabajar sin importar la distribución de los datos. Una de las principales razones por la que se seleccionó una RNA como modelo principal para clasificación de tuits, es que presta gran atención a la correlación entre variables, en este caso el vocabulario de los tuits, a diferencia del clasificador bayesiano, en el que las variables son tratadas de manera independiente. Ambas técnicas fueron evaluadas mediante la estrategia de validación cruzada de *k* grupos, con $k = 10$, calculando exactitud y error promedio para los k grupos y promedio de precisión, exhaustividad y medida F para cada iteración.

2. Métodos

El método propuesto se divide en cinco etapas, dentro de las cuales se recuperan los tuits y, se implementan y se evalúan dos clasificadores. Además, se presenta una aplicación web con la que es posible asignar el tema o clase al tuit según el entrenamiento y predicción del clasificador seleccionado.

Recuperación de tuits

Los métodos de aprendizaje supervisado, como la clasificación, requieren dos conjuntos de datos, un conjunto de datos para la etapa de entrenamiento y otro

para la etapa de pruebas. En este caso, los datos son tuits, textos cortos de máximo 280 caracteres. Para conformar el conjunto de datos se recuperaron tuits de tres clases distintas en un periodo de abril a mayo del 2018. Las clases o temas corresponden a sucesos de interés en el estado de Tamaulipas, por lo tanto, los tuits fueron filtrados con las palabras clave o *hashtags*: #Tamaulipas, Tamaulipas, #Tamps o Tamps*;* y para cada clase las siguientes palabras clave:

- Situación de riesgo (SDR)**:**
  - ✓ #SDR, #SituaciónDeRiesgo, #SituacionDeRiesgo, #SituacionRiesgo, #SituaciónRiesgo.
  - ✓ #Victoria, #CdVictoria, #Reynosa, #Matamoros, #NuevoLaredo, #Tampico, #Madero, #CdMadero, #Altamira, #RíoBravo, #RioBravo o #Mante.
- Servicio de agua:
  - ✓ Falta de agua, no hay agua, problema del agua, sin agua, fuga o suministro de agua.
- Política
  - ✓ #Elecciones, Elecciones, #Politica, #Política, Politica, Política, votar, votos, eleccion o elección.

Para el caso de la clase SDR se consideraron las tendencias más comunes para identificar una situación de riesgo, aunque se consideró todo lo etiquetado como #Tamaulipas también se agregaron algunos nombres de los municipios con mayor índice de SDR, esto para el caso de que el tuit no tuviera una de las etiquetas del nombre del estado. Además, al recuperar los tuits se tomaron en cuenta solamente aquellas publicaciones originales, es decir, se eliminaron los retuits, obteniendo 1,399 tuits, distribuidos por clase según las cantidades mostradas en la tabla 1.

Tabla 1 Tuits recuperados por clase.

Clase	Tuits
SDR	157
Servicio de agua	231
Política	1,011

En los tuits recuperados se observa una alta presencia de la clase Política dada la cercanía de elecciones presidenciales respecto al periodo de recuperación. Además, el caso del servicio de agua y situaciones de riesgo son problemas que afectan a varios municipios del estado de manera simultánea.

Pre-procesamiento

En Twitter, las publicaciones presentan variabilidad debido al tipo de usuario (persona o institución), género, edad, grado de estudio o trabajo, intereses, lugar de residencia, el tema o tendencia de la que se esté publicando y el vocabulario particular de dichos temas. Por esto, el procesamiento inicial o tratamiento de los tuits es una de las etapas más importantes para lograr que un modelo de clasificación esté bien entrenado.

En nuestro caso el pre-procesamiento se dividió en dos etapas:

- Etapa de limpieza de los tuits**:** los tuits tienen características muy diferentes a otras fuentes de información, como textos extraídos de páginas web o de conjuntos de datos diseñados específicamente para la clasificación, por esto, en la etapa de limpieza se realizó lo siguiente:
  - ✓ Eliminar URLs a otros sitios web.
  - ✓ Eliminar referencias a usuarios (@usuario).
  - ✓ Eliminar tendencias tomadas en cuenta para la recuperación (#tendencia).
  - ✓ Eliminar emoticones/emojis.
  - ✓ Eliminar caracteres especiales (¿, ¡,*, {,}, entre otros).
- Etapa de eliminación de palabras vacías[3] y reducción de palabras: Existen listas de palabras vacías para cada idioma, las cuales pueden ser modificadas para adaptarse a las necesidades del usuario o aplicación. En este caso se utilizó una lista de palabras definida para el idioma español, a la cual se le agregaron algunos de los modismos detectados y/o abreviaturas comúnmente usadas, consideradas como faltas de ortografía (ke, k, x (por), + (más), entre otras):

[3] Una palabra vacía es aquella que carece de significado por sí sola, generalmente son artículos, preposiciones, conjunciones, entre otras.

- ✓ Reducción de palabras a su raíz léxica: al tratarse de texto en español se trabajó con una herramienta que trata este lenguaje, por lo que se utilizó la implementación del algoritmo Snowball incluida en la librería NLTK (*Natural Language Tool Kit*) para Python. Este algoritmo reduce las palabras a su raíz léxica, por ejemplo, para las palabras azulado, azules, azuloso y azular, su raíz léxica es azul.

Construcción del conjunto de datos

Una vez que los tuits recuperados fueron pre-procesados, se etiquetaron con las tres categorías disponibles, para esto se hizo primero una revisión manual de los tuits. En el caso de la clase SDR se creó un diccionario de forma manual con los términos más comunes para referirse a estos acontecimientos ya que en ocasiones la cadena #SDR fue utilizada de manera incorrecta. Además de esto, se observó que algunas palabras del vocabulario de las clases SDR y Política se compartían, por lo que los tuits recuperados para Política que pertenecieran a SDR (según el diccionario) fueron etiquetados como correctos para la clase SDR.
Posteriormente, para la clase SDR se hizo un etiquetado automático usando como base el diccionario definido, por lo que aquellos tuits en los que no se incluyó una palabra característica, sino que se utilizó sin razón la etiqueta, fueron descartados. En el caso de la clase de Política, después de filtrar aquellos con términos de SDR, los restantes se etiquetaron de manera directa, al igual que en el caso de la clase Servicio de agua.

Extracción de características

Una vez que el conjunto de tuits ha sido etiquetado se procede a realizar la extracción de características, como se muestra en la figura 1. Este proceso da como resultado un modelo que será utilizado para predecir la clase a la que pertenecen los tuits de prueba. Se usará un algoritmo de aprendizaje automático, en este caso se usará una red neuronal de tipo perceptrón multicapa. Para trabajar con una técnica como red neuronal para texto debe generarse una representación vectorial, también conocida como modelo bolsa de palabras [Carrillo, 2002]. En el

modelo bolsa de palabras se toman todos los textos de la colección, en este caso todos los tuits recuperados y se obtiene el vocabulario, es decir todas las palabras que aparecen en el conjunto (sin repetición). Posteriormente, cada tuit se representa como un vector en donde cada palabra tiene una ponderación.

Figura 1 Proceso de extracción de características.

Existen distintas formas de ponderar los términos, la más básica es el modelo booleano que indica la ocurrencia o ausencia de un término del vocabulario en un tuit. En la figura 2, se muestra un ejemplo de este tipo de representación para una colección de tres tuits, donde cada una de las palabras está presente en un tuit, ya sea T1, T2 o T3.

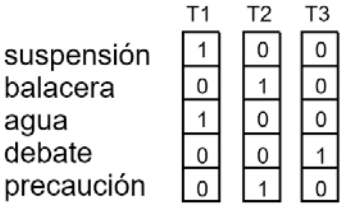

Figura 2 Ejemplo del modelo bolsa de palabras.

En nuestro caso, se utiliza la ponderación TF-IDF (ecuación 1), donde se calcula IDF con la ecuación 2. Se utiliza esta ponderación debido a que aporta más información del tuit de manera individual y respecto a su presencia en el conjunto total de tuits. Para obtener la ponderación de cada tuit en el conjunto se utilizaron las herramientas de la librería Scikit-learn de Python, como se describe a continuación:

- Tomar el conjunto de datos y transformarlo a una representación con frecuencia de ocurrencia del término por tuit.

- Posteriormente se transforma a la ponderación $TF\text{-}IDF$ por término, calculada en este caso con la ecuación 1.

$$TF - IDF(d,t) = TF(t) * IDF(d,t) \quad (1)$$

$$IDF = log\left[\frac{1+n}{1+DF(d,T)}\right] + 1 \quad (2)$$

Donde:

$TF(t)$: Frecuencia del término

$DF(d,t)$: Frecuencia de término en la colección

n: número total de documentos

Utilizar $TF - DF$ permite que se reduzca el impacto de términos que ocurren con mucha frecuencia en el conjunto de tuits, ya que estos son menos representativos para la extracción de características.

Configuración y entrenamiento del clasificador

El clasificador principal de esta propuesta es un perceptrón multicapa, además, se compara su desempeño con un clasificador bayesiano, en ambos casos se utilizó la implementación disponible en la librería Scikit-learn de Python.

El perceptrón multicapa es una de las RNA más utilizadas, se caracteriza porque en su estructura se distinguen tres tipos de capas. La capa de entrada, en la que se reciben las variables de entrada y se traspasan a la o las capas ocultas, en estas últimas se concentra el procesamiento, sus entradas corresponden a salidas de capas anteriores y sus salidas se utilizan para el procesamiento de capas posteriores, y por último la capa de salida la cual se encarga de proporcionar la salida de la red (figura 3).

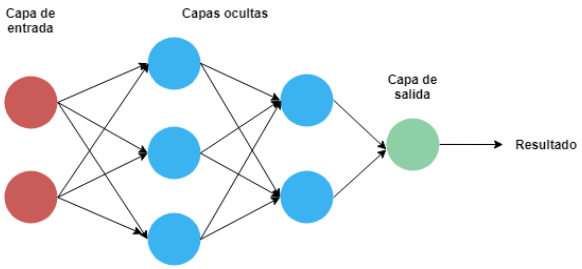

Figura 3 Estructura de un perceptrón multicapa.

Cada una de estas capas está formada por una o más neuronas con conexiones entrantes de capas anteriores y conexiones salientes a capas posteriores, con lo que es posible resolver problemas que no son linealmente separables por un solo plano, es decir, por un perceptrón simple.

La configuración de los parámetros para el perceptrón multicapa fue la siguiente:

- $tol = 0.0001$: este valor determina el umbral de error con el cual se detendrá la actualización de los pesos de la red.
- $learning\_rate\_init = 0.001$: indica la velocidad de aprendizaje.
- $max\_iter = 10000000$: máximo número de iteraciones en las que se detendrá el entrenamiento si no se consigue llegar a un error menos al umbral definido.
- $hidden\_layer\_sizes = (150, 50)$: con cada valor se indica una capa oculta, por lo que se tienen dos capas ocultas, la primera con 150 neuronas y la segunda con 50.
- $activation = 'relu'$: función de activación.
- $solver = 'adam'$**:** optimizador.

Se realizaron pruebas con distintos valores para los parámetros, al disminuir el valor de tolerancia se obtuvieron más errores en muchas de las pruebas y al cambiar la cantidad de neuronas no se observaron cambios considerables. Respecto a la función de activación se entrenó el clasificador con la función tangente, sin embargo, se detectaron más tuits etiquetados de manera incorrecta por lo que se optó por usar la función por defecto. Debido a que el tamaño del conjunto puede cambiar al recuperar más tuits se eligieron los parámetros presentados previamente ya que el modelo generado fue estable.

En el caso del clasificador bayesiano ingenuo, que es de tipo probabilístico y está basado en el Teorema de Bayes (ecuación 3) propuesto por Thomas Bayes.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \quad (3)$$

Donde:

$P(A|B)$ Indica la probabilidad de A dado que B es verdadero.

$P(B\|A)$	Probabilidad de ocurrencia del evento B dado que A es verdadero.
$P(A)$	Probabilidad del suceso A.
$P(B)$	Probabilidad del suceso B.

El teorema de Thomas Bayes permite inferir la probabilidad de un suceso con base en la probabilidad o conocimiento que se tiene de sucesos conocidos, se usa también una representación del texto en bolsa de palabras.

En la distribución multinomial las frecuencias de términos u otra ponderación se representan con el modelo bolsa de palabras en donde los documentos en cada clase son modelados como muestras extraídas de una distribución de palabras multinomial, por lo que la probabilidad de que un documento pertenezca a una clase está dada por la probabilidad de cada palabra que se conoce que pertenece a dicha clase [Aggarwal, 2012].

En este clasificador se trabajó con los valores por defecto que asigna la librería Scikit-learn, ya que al probar varias combinaciones estos dieron los mejores resultados. Entre algunos de los parámetros que es posible configurar es $alpha = 1.0$, el cual hace referencia al suavizado de Laplace, que se usa principalmente para evitar multiplicar las probabilidades por cero, con este suavizado las probabilidades son inicializadas de manera uniforme, posteriormente son modificadas con los datos por clasificar, es decir que inicialmente un elemento tiene la misma probabilidad de pertenecer a cualquier clase.

Una vez entrenado el clasificador, se realizan las pruebas, es decir, se recibe un tuit sin clasificar el cual pasa por una etapa de extracción de características y posteriormente se hace la consulta al clasificador para obtener su etiqueta o clase. Este proceso se muestra en la figura 4.

Figura 4 Proceso de predicción para los tuits de prueba.

3. Resultados

Para evaluar el desempeño de los clasificadores, el conjunto de tuits recuperados fue dividido por medio de la técnica de validación cruzada de k grupos, en este tipo de validación cruzada se realizan k iteraciones y los datos se dividen en k subconjuntos, en cada iteración una parte del conjunto se utiliza como datos de prueba y el resto como datos de entrenamiento $(k-1)$, el proceso de validación se repite la cantidad de veces definida por k con cada uno de los conjuntos generados.

Para calcular el error de los distintos conjuntos se calcula la media aritmética de los errores de cada iteración y se obtiene un único resultado, esto se hace de la misma manera con el valor de la exactitud [Wong, 2017].

Para estos experimentos se tomó $k = 10$, por lo que en cada iteración se asignó el 90% de los tuits para el entrenamiento y 10% para las pruebas, en las tablas 2 y 3 se presentan los resultados de precisión, exhaustividad y medida F para cada clase en una ejecución. Además, en la tabla 4 se presenta la exactitud y error promedio de la misma ejecución.

Tabla 2 Promedio de precisión, exhaustividad y medida F.

Clase	Precisión	Exhaustividad	Medida F
Política	0.824	1.000	0.903
SDR	1.000	0.304	0.459
Servicio de agua	0.952	0.513	0.658

Tabla 3 Promedio de precisión, exhaustividad y medida F.

Clase	Precisión	Exhaustividad	Medida F
Política	0.917	0.990	0.952
SDR	0.914	0.645	0.752
Servicio de agua	0.951	0.812	0.875

Tabla 4 Exactitud y error promedio.

Clasificador	Exactitud	Error
Bayesiano ingenuo	0.841	0.158
Perceptrón multicapa	0.921	0.078

Se ejecutaron múltiples iteraciones, y se comprobó que el perceptrón multicapa se mantuvo con mejores resultados que el clasificador bayesiano. En la tabla 5 se presentan los resultados de exactitud y error para la validación cruzada de 10 grupos en 10 iteraciones, como se observa, en todos los casos el perceptrón multicapa tiene mayor exactitud y menor error.

Tabla 5 Exactitud y error promedio por ejecución para los clasificadores.

	Clasificador Bayesiano		**Red Neuronal**	
Ejecución	**Exactitud**	**Error**	**Exactitud**	**Error**
1	0.842	0.157	0.917	0.082
2	0.842	0.157	0.915	0.084
3	0.841	0.158	0.922	0.077
4	0.841	0.158	0.919	0.080
5	0.848	0.151	0.920	0.079
6	0.842	0.157	0.919	0.080
7	0.843	0.156	0.914	0.085
8	0.841	0.158	0.917	0.082
9	0.842	0.157	0.915	0.084
10	0.836	0.163	0.917	0.082

Aplicación web

Para que el clasificador generado pudiera ser utilizado por un usuario final y así éste pudiera comprobar si un tuit pertenece o no a los temas de SDR, Política o Servicio de agua, se diseñó una aplicación web con las siguientes funcionalidades:

- **Vista principal:** en la figura 5 se muestra la interfaz en la que el usuario puede escribir un tuit, seleccionar el clasificador entre los dos disponibles y obtener la categoría a la que pertenece.

Figura 5 Vista principal de la interfaz para usar el clasificador.

- **Cargar tuit:** si el usuario no conoce el vocabulario o tuits de las clases disponibles puede cargar uno del conjunto de manera aleatoria para ver cómo funciona el clasificador, como se presenta en la figura 6.

Figura 6 Botón para cargar tuit aleatorio.

- **Obtener la categoría:** en la figura 7 se visualiza la categoría a la que un tuit pertenece una vez que ha sido ingresado y se ha seleccionado un clasificador.

Figura 7 Categoría asignada por el clasificador.

- **Retroalimentación:** el usuario tiene la opción de agregar nuevos tuits y su clasificación manual directamente al conjunto de datos con el objetivo de que estos sean incluidos en el modelo. En este caso, el modelo deberá ser reentrenado para que incluya el tuit y la clase asignada por el usuario, esto se realiza mediante la interfaz de la figura 8.

Figura 8 Vista para agregar tuits al modelo.

4. Discusión

La tarea de clasificación se encuentra presente en muchas actividades, tal es el caso de la clasificación de texto, en donde el desempeño de los modelos de clasificación depende en gran parte de la calidad de los datos con los que se esté trabajando, ya que si se trata de un conjunto artificial se espera que se encuentren correctamente etiquetados, a diferencia de un conjunto de datos construido con datos reales y etiquetado manualmente, que podría tener algunos errores por las técnicas de etiquetado que se hayan usado.

Además de lo anterior, es muy importante la manera en la que los datos son tratados, ya que la selección de técnicas de pre-procesamiento cambiará las características obtenidas durante la extracción y por lo tanto el proceso de entrenamiento y el valor de la exactitud en los resultados.

Respecto al desempeño de los clasificadores, RNA obtuvo valores de error menores que 0.09, a diferencia del clasificador bayesiano, que obtuvo valores superiores a 0.15. Lo anterior indica que para RNA la mayoría de los tuits de prueba fueron etiquetados correctamente en las 10 ejecuciones realizadas, esto se debe en gran parte a la manera en la que trabaja RNA respecto a la correlación entre variables, permitiendo que se mantenga esa relación entre palabras que forman frases características de una clase u otra, con lo que se mantiene el contexto del tuit y aumenta la exactitud del clasificador. Se espera que al reunir una mayor cantidad de tuits la exactitud de los modelos aumente, sin embargo, esto requiere un periodo más amplio de recolección de tuits y, en consecuencia, un mayor tiempo de entrenamiento de los modelos.

A través de la aplicación web propuesta es posible que los usuarios comunes participen e interactúen de una forma más fácil con el etiquetado y pruebas de los clasificadores automáticos.

5. Conclusiones

En este trabajo se ha presentado un enfoque de clasificación de tuits en múltiples clases, incluyendo todos los aspectos a considerar para su recolección, procesamiento y pruebas. En particular, procesar texto en español representa un reto por la diversidad de léxico y el tipo de usuario que escribe los tuits. En este caso para la adquisición de los tuits y el pre-procesamiento solo se consideraron palabras asociadas a los temas de interés, sin embargo, pueden asociarse otras características de los usuarios para poder clasificar otros temas.

Se ha evaluado el desempeño de un perceptrón multicapa como clasificador respecto a un clasificador probabilístico como el clasificador Bayesiano ingenuo y, ambos clasificadores fueron evaluados con el método de validación cruzada de 10 grupos. A partir de esto se observó que para ambos clasificadores se obtuvieron buenos resultados, siendo la red neuronal la que alcanzó los valores más altos con resultados superiores a 0.90 de exactitud promedio. Sin embargo, el clasificador Bayesiano no podría ser descartado ya que su exactitud promedio superó el 0.80 en todas las pruebas realizadas.

Es importante resaltar que la metodología propuesta puede trabajar con un número de clases superior o inferior y también con distintos contextos, solo deberá modificarse el vocabulario particular en la etapa de recolección de tuits.

Este trabajo ha sido financiado por la beca 27194 del proyecto de Investigación de Ciencia Básica SEP-CONACYT CB_2015/256922.

6. Bibliografía y Referencias

[1] Aggarwal C, Zhai C. 2012. A Survey of Text Classification Algorithms. Min. Text Data: 163–222.

[2] Cheng B, Titterington DM. 1994. Neural Networks: A Review from a Statistical Perspective. Stat. Sci. 9: 2–30.

[3] Duncan B, Zhang Y. 2015. Neural networks for sentiment analysis on Twitter. 2015 IEEE 14th Int. Conf. Cogn. Informatics Cogn. Comput.: 275–278.

[4] Haykin S. 1998. Neural networks: A Comprehensive Foundation, 2nde. Upper Saddle River, NJ, USA: Prentice Hall PTR. S0269888998214044 p.

[5] Jianqiang Z, Xiaolin G. 2018. Deep Convolution Neural Networks for Twitter Sentiment Analysis. IEEE Access 3536.

[6] Kuang S, Davison B. 2017. Learning Word Embeddings with Chi-Square Weights for Healthcare Tweet Classification. Appl. Sci. 7: 846.

[7] Carrillo Ruiz, M., López López, A. Una Representación Vectorial para Contenido de Textos en Tratamiento de Información, Servidor y biblioteca de modelos de recuperacion de informacion, 2002.

[8] Martinc M, Škrjanec I, Zupan K, Pollak S. 2017. PAN 2017: Author profiling - Gender and language variety prediction: Notebook for PAN at CLEF 2017. CEUR Workshop Proc. 1866.

[9] Pla F, Hurtado L-F. 2013. TASS-2013: Análisis de Sentimientos en Twitter. Proc. TASS Work. SEPLN 2013: 1–8.

[10] Rangel F, Rosso P. 2013. Use of Language and Author Profiling: Identification of Gender and Age. Proc. 10th Work. Nat. Lang. Process. Cogn. Sci.: 177.

[11] Rosá A, Chiruzzo L, Etcheverry M, Castro S. 2017. RETUYT in TASS 2017: Sentiment Analysis for Spanish Tweets using SVM and CNN. 2017.

[12] Rosenthal S, Farra N, Nakov P. 2017. SemEval-2017 Task 4: Sentiment Analysis in Twitter. Proc. 11th Int. Work. Semant. Eval.: 502–518.

[13] Shrestha P, Sierra S, González FA, Rosso P, Montes-Y-Gómez M, Solorio T. 2017. Convolutional Neural Networks for Authorship Attribution of Short Texts. Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguist. 2: 669–674.

[14] Wehrmann J, Becker WE, Barros RC. 2018. A Multi - Task Neural Network for Multilingual Sentiment Classification and Language Detection on Twitter. 8.

[15] Wong TT, Yang NY. 2017. Dependency Analysis of Accuracy Estimates in k-Fold Cross Validation. IEEE Trans. Knowl. Data Eng. 29: 2417–2427.