Reconocer si un tweet está o no relacionado con el béisbol venezolano mediante el uso de redes neuronales.

Andrea Centeno
Universidad Simón Bolívar
10-10138@usb.ve

Isaac Gonzalez
Universidad Simón Bolívar
11-10396@usb.ve

Resumen
Análisis de tweets para definir si están relacionados con beisbol. Para el experimento se implementó un preceptrón multicapas con la idea idea de observar si el algoritmo era capaz de aprender y predecir si el tweet estaba relacionado con dicho deporte o no.

Keywords
Beisbol, redes neuronales, aprendizaje de máquina.

1. INTRODUCCIÓN
El béisbol llegó a Venezuela hace más de 100 años y en 1927 se fundó la Federación Venezolana de Béisbol. Desde entonces, este se ha vuelto el deporte de preferencia de gran parte de la población venezolana. Con este proyecto se busca reconocer si un mensaje de twitter esta relacionado con un tema en especifico, en este caso, béisbol, con el objetivo de observar si es posible realizar una clasificación de tweets con alto nivel de aciertos con respecto a su contenido.
Para el trabajo, se recopilaron 2397 tweets, los cuales pasaron por un proceso de filtro y limpieza para luego usarse para entrenar, por medio de backpropagation, el perceptrón multicapa implementado.

2. MARCO TEÓRICO
2.1 Aprendizaje supervisado
El aprendizaje supervisado se caracteriza porque el proceso de aprendizaje se realiza mediante un entrenamiento controlado por un agente externo que determina la respuesta que debería generar la red a partir de una entrada determinada. Esta técnica deduce una función a partir de datos de entrenamiento, dichos datos consisten de pares de objetos, una componente son los datos de entrada y el otro, el resultado deseado.

2.2 Redes neuronales
Las redes neuronales son un modelo computacional cuyo objetivo es imitar el comportamiento y funcionamiento de las neuronas de los organismos vivos. Estos sistemas aprenden y se forman a sí mismos, en lugar de ser programados de forma explícita, tal como un conjunto de neuronas de los cerebros biológicos que están conectadas entre sí y trabajan en conjunto, sin que haya una tarea concreta para cada una. Con la experiencia, las neuronas van creando y reforzando ciertas conexiones para ”aprender” algo que se queda fijo en el tejido.

2.3 Perceptrón multicapa
Es uno de los tipos de redes neuronales más comunes, se basa en otra más simple llamada perceptrón simple con la diferencia de que está formada por múltiples capas, esto le permite resolver problemas que no son linealmente separables, lo cual es la principal limitación del perceptrón simple.

2.4 Backpropagation
Es un algoritmo que se usa para entrenar una red neuronal. Emplea un ciclo de propagación y adaptación en dos fases: En la primera, la señal se propaga hacia la última capa al presentar la entrada, en la segunda fase se compara la salida obtenida con el valor deseado, se calcula el error y se propaga hacia atrás modificando los pesos.
La red reproduce una representación interna que le permite generar las salidas deseadas cuando se le presenta una entrada con características similares. [6]

2.5 Limpieza de datos
Cuando se realiza la limpieza sobre un conjunto de datos, hay dos definiciones que son importantes, estas son:

- Stopwords: son todas aquellas palabras que carecen de un significado por sí solas. Suelen ser artículos, preposiciones, conjunciones, pronombres, etc.
- Stemming: Método para reducir una palabra a su raíz.

3. DISEÑO DE LA SOLUCIÓN
3.1 Recolección de datos y limpieza
Se recolectaron 2397 tweets, los cuales fueron hechos entre el rango de fechas 01/10/2016 y 01/01/2017, estos fueron almacenados en un archivo csv. Para esta recolección se usaron dos herramientas:

- Tweepy[5]: Librería desarrollada en python para acceder a API de twitter.

- Get Old Tweets Programatically (Got)[4]: Librería que permite obtener tweets viejos.

La última liberia presentada, se usó debido a que el API de twitter presenta limitaciones con respecto a los rangos de fechas con los que se pueden realizar búsquedas, dicha API no permite obtener tweets anteriores a dos semanas. Con GOT, se pudo resolver este inconveniente.
Solo se recolectaron tweets en español. Para detectar el idioma de un tweet se usó una librería de Python llamada langdetect[2].
Se hicieron dos tipos de búsqueda, la primera relativa a béisbol, para esta se usaron palabras claves tales como Magallanes, Leones del Caracas, Cardenales de Lara, beisbol venezolano, LVBP, entre otras. La segunda búsqueda se hizo con la intención de conseguir tweets que no estuvieran relacionados con béisbol. Luego de la recolección los datos pasaron por un proceso de limpieza, aquí se realizaron las siguientes acciones:

- Eliminación de acentos.
- Transformación a minúsculas.
- Eliminación de nombres de usuarios.
- Eliminación de URLS.
- Eliminación de hashtags.

Cada tweet pasó de ser un string, a convertirse en una lista de palabras. De aquí se pasó a eliminar las stopwords, haciendo uso de la librería NLTK[3] de Python. De la lista de palabras obtenidas, se hizo el proceso de stemming, de igual manera se hizo uso de una librería del mismo lenguaje llamada PyStemmer[1].
Por último, se eliminaron los términos pocos frecuentes, ya que estos no aportaban nada relevante y existía la posibilidad de que fueran palabras mal escritas.

3.2 Implementación

La implementación se realizó en Python 3.5.2. Se implementó un diccionario, el cual contiene todas las palabras presentes en los tweets encontradas después del proceso de limpieza de datos, cada tweet es representado con dos vectores, el primero de n casillas, siendo n la cantidad de palabras del diccionario, donde cada casilla representa la presencia de una palabra del diccionario en el tweet, es decir, si la palabra se encuentra en el tweet tendrá un valor igual a 1, de lo contrario 0. El segundo vector es de tamaño 1 y representa si el tweet esta relacionado con baseball. Esta implementación ignora el orden de las palabras en los tweets.
Desarrollamos dos tipos de redes neuronales multicapa feedfoward con backpropagations, el primer tipo posee una capa oculta y el segundo tipo posee dos, las redes poseen una cantidad de nodos de entrada igual a las palabras presentes en el diccionario, de 4 a 8 neuronas internas y un nodo de salida.

4. EXPERIMENTO Y RESULTADOS

Para el experimento se tomaron aleatoriamente del conjunto de datos 500 tweets relacionados con baseball y 500 tweets no relacionados, para un total de 1000 tweets. Realizamos pruebas sobre estos datos en ambas redes neuronales dividiendo los mismos en 50, 60 y 70 porciento entre el conjunto de entrenamiento y el conjunto de prueba para ambas redes neuronales. Además realizamos construimos varias redes neuronales con diferentes cantidades de neuronas en las capas ocultas, con un rango de 4 a 8.
Una vez realizado el entrenamiento procedimos a evaluar la red con el conjunto de datos de prueba, los resultados arrojados por la red fueron redondeados, el dominio de los mismos quedó restringido a 0 y 1 donde 1 significa que esta relacionado con baseball y 0 que no lo está.

Neuro.	Capas	Iter.	Aciertos	Aciertos(%)
4	1	209	466	93 %
	2	164	472	94 %
5	1	61	458	91 %
	2	115	441	88 %
6	1	54	424	84 %
	2	38	462	92 %
7	1	248	470	94 %
	2	93	456	91 %
8	1	70	414	82 %
	2	78	467	93 %

Table 1: Entrenamiento: 500 - Prueba: 500

Neuro.	Capas	Iter.	Aciertos	Aciertos(%)
4	1	111	351	87 %
	2	439	377	94 %
5	1	201	379	94 %
	2	61	359	89 %
6	1	67	357	89 %
	2	43	372	93 %
7	1	102	356	89 %
	2	121	371	92 %
8	1	127	355	88 %
	2	98	381	95 %

Table 2: Entrenamiento: 600 - Prueba: 400

Neuro.	Capas	Iter.	Aciertos	Aciertos(%)
4	1	192	288	96 %
	2	167	281	93 %
5	1	51	278	92 %
	2	56	286	95 %
6	1	120	233	77 %
	2	40	282	94 %
7	1	67	289	96 %
	2	72	280	93 %
8	1	197	291	97 %
	2	75	284	94 %

Table 3: Entrenamiento: 700 - Prueba: 300

El comportamiento de la red neuronal con dos capas ocultas fue superior a la red neuronal con una sola capa oculta, como se puede observar en los resultados las redes con dos capas convergieron mas rápido, además poseen una media mas elevada para todas las proporciones del conjunto de

datos que tomamos y una desviación estándar menor con respecto a la red neuronal de una capa oculta, el aumento en el numero de neuronas en las redes no proporcionó mejoras significativas, los mejores resultados para la red de una capa se obtuvieron con la combinación de 8 neuronas con 70 % de datos de entrenamiento sin embargo vale la pena destacar que el peor resultado para esta red también se obtuvo con con 70 % de entrenamiento.

5. CONCLUSIONES

La selección y limpieza de datos fue un factor determinante para obtener buenos resultados en los experimentos, gracias a esto y el uso de redes neuronales logramos clasificar mensajes con respecto a la relación que poseen con un tema. Comparamos el desempeño de las redes de una capa oculta y dos capas ocultas con diferentes números de neuronas frente a varias situaciones lo que nos llevo a concluir que para este tipo de problemas las redes neuronales de dos capas presentan una ventaja frente a las de una capa.

6. REFERENCIAS

[1] Boulton, R. Pystemmer, 2013.

[2] Danilak, M. Langdetect repository, 2014-2015. [Online; accedido 28-Marzo-2017].

[3] Ewan˜Klein, E. L. y. S.˜B. Nltk 3.0 documentation, 2015.

[4] Henriquen, J. Get old tweets programatically, 2016. [Online; accedido 29-Marzo-2017].

[5] Roesslein, J. Tweepy documentation, 2017. [Online; accedido 28-Marzo-2017].

[6] y˜Manuel˜Ortega, J.˜B. Lineas de invetigacion en informatica. *Ediciones de la Universidad de Castilla-La Mancha* (2000), 29.