Detección de comentarios que violan pautas de comportamiento mediante clasificación de texto

Luis Enrique López Nerio

Universidad Autónoma de Nuevo León
Facultad de Ciencias Físico Matemáticas

July 23, 2022

El procesamiento de lenguaje natural es un área de investigación dentro del aprendizaje automático que ha cobrado vital importancia en años recientes, esto gracias a los constantes avances en sus diferentes acercamientos para resolver problemas relacionados con datos textuales. Uno de los principales problemas dentro del procesamiento de lenguaje natural está el de clasificación de textos, dentro de la literatura se cuentan con una gran variedad de metodologías para resolver este problema.
En este artículo se hará una análisis de 2 formas de extracción de características en mensajes de texto para comparar su desempeño en la clasificación de comentarios realizados en la comunidad de editores de Wikipedia.

1 Introducción y antecedentes

Desde que nacemos tenemos el deseo de comunicarnos y, a medida que nos desarrollamos, expresamos nuestras ideas y pensamientos. Hoy en día las redes sociales, los foros de discusión de páginas de difusión son el medio de comunicación más atractivo y accesible con el que contamos para compartir nuestro parecer sobre algún tema.

Sin embargo, no siempre estos espacios son seguros, la amenaza de abuso y acoso en línea hace que muchas personas dejen de expresarse y renuncien a buscar opiniones diferentes. Las plataformas luchan por facilitar las conversaciones de manera efectiva, lo que lleva a muchas comunidades a limitar o cerrar por completo los comentarios de los usuarios.

Con el propósito de poder identificar y censurar publicaciones que violen pautas de comunicación cordial y pongan en riesgo la sana convivencia dentro del Internet, crear un algoritmo aprendizaje supervisado que sea capaz de identificar mensajes que entren dentro de la categoría de mensajes no deseados, es de gran utilidad para poder fomentar una sana convivencia, tener discusiones constructivas entre los participantes y el crecimiento de la comunidad, evitando la fuga de usuarios por malas experiencias.
Identificar los comentarios que pueden que violan las pautas de comunicación, además de ser interés para la plataforma que busca eliminarlos, está comenzando a ser un tema de regulación en algunos países de Europa (“Endurecimiento frente a discurso de odio” 2019), como se menciona en este articulo la carga de moderar el contenido y retirarlo en tiempo y forma recaerá en las empresas, a su vez, el gobierno puede poner multas de hasta 50 millones de euros si las plataformas no cumplen con las solicitudes validas hechas por los usuarios o autoridades.

En este artículo se busca encontrar una forma atacar el problema de clasificación de mensajes de texto,

esta problemática es común en el procesamiento de lenguaje natural, como lo menciona Van Aken et al., 2018, algunos retos abiertos dentro de la clasificación de textos son las palabras con escritura incorrecta, uso de neologismos, lenguaje idiosincrásico.

La metodología clásica que siguen los problemas de clasificación de texto como enfatiza Kowsari et al., 2019 es un proceso de 4 pasos básicos:

1. Preprocesamiento y extracción de características
2. Reducción de dimensional dad
3. Modelo de clasificación
4. Evaluación

1.1 Preprocesamiento y extracción de características

En general la forma en que se tienen los datos es un conjunto de mensajes $M = \{m_1, m_2 \dots m_n\}$ donde cada uno de estos mensajes tiene una etiqueta binaria, en la parte de preprocesamiento se busca hacer una limpieza que incluye los siguientes pasos:

- Tokenización del mensaje en palabras
- Conversión de mayúsculas a minúsculas
- Eliminar signos de puntuación
- Eliminación de stop words
- Lemmatización o steming

Una vez que se hace este preprocesamiento a cada mensaje, se realiza la extracción de características o vectorización, en este paso lo que se busca es convertir cada mensaje m_i el cual está compuesto de diferentes palabras en una representación numérica.

En grandes rasgos existen dos formas de realizar la extracción de características o vectorización de un mensaje m_i:

1. Enfoque frequentista. Este enfoque se puede realizar de dos formas:
   Bolsa de palabras. Se construye un vocabulario del conjunto de mensajes M y para cada mensaje m_i, se cuenta el número de ocurrencias que cada término del vocabulario tiene en el mensaje, cada término en el vocabulario es una característica y el mensaje se convierte en un vector de longitud igual a la longitud del vocabulario. En la tabla 2 se ve un ejemplo con tres mensajes: Algunas

Table 1: *Vectorización, bolsa de palabras*

	comer	esta	gusta	me	pelicula	serie
Me gusta esta pelicula	0	1	1	1	1	0
Me gusta esta serie	0	1	1	1	0	1
Me gusta comer	1	0	1	1	0	0

   desventajas de este enfoque es que mapea cada mensaje a un vector de longitud igual a la del vocabulario, si se tiene con una gran cantidad de mensajes entonces la longitud del vector puede llegar a tener una gran longitud.

   TF-IDF El siguiente método que toma en cuenta la frecuencia de los términos es el TF-IDF,(Term Frequency- Inverse Document Frequency), en este método al igual que en el anterior, se toma el vocabulario de los términos presentes en los mensaje M, cada mensaje m_i es mapeado a un vector de longitud igual al número elementos en el vocabulario, cada palabra del vocabulario es una característica, primeramente se calcula la frecuencia que cada término del vocabulario tiene en el mensaje y además se multiplica por el logaritmo de la frecuencia inversa que el término tiene en todos los mensajes m_i.

$$tf - idf(m_i, t) = TF(m_i, t) * log\left(\frac{N}{df_M(t)}\right)$$

   m_i es el mensaje i, t es el termino, N es el número de mensajes y $df_M(t)$ es el número de documentos que tiene el término t presente. A continuación se muestra un ejemplo con tres mensajes diferentes: La ventaja de esta forma de extraer características

Table 2: *Vectorización, bolsa de palabras*

	comer	esta	gusta	me	pelicula	serie
Me gusta esta pelicula	0.	0.504	0.39	0.39	0.66	0.
Me gusta esta serie	0.	0.50	0.39	0.39	0.	0.66
Me gusta comer	0.767	0.	0.453	0.453	0.	0.

   de un mensaje es que no le da mucho peso a términos que se pueden presentar con gran frecuencia y no ayudan a diferenciar un mensaje de otro al estár presente en una gran cantidad de mensajes.
2. Word Embeding La vectorización mediante word embeding busca atacar una problemática a la que se enfrentan los enfoques frecuentistas, está problemática es que estas representaciones no logran capturar el significado semántico de las palabras, por ejemplo, las palabras avion, aeronave, aeroplano son palabras que son utilizadas con el mismo significado y en el mismo contexto, mediante los enfoques frecuentitas, la representación vectorial de estos términos es ortogonal. La extracción de características mediante word embeding busca asignar a cada término del vocabulario una representación vectorial real de longitud N, esta vectorización se hace mediante redes neuronales, el método más utilizado en la literatura es el de *Word2Vec* propuesto por Church, 2017.

1.2 Reducción de dimensionalidad

En general la extracción de características de un mensaje nos da como resultado un vector de gran cardinalidad, esto es una limitante ya que se incrementa la complejidad al manejar esta cantidad de datos y por ende el consumo de recursos computacionales también incrementa.

Una forma de solucionar esta problemática es aplicar un método de reducción de dimensionalidad a nuestros

vectores de características, algunos de estos métodos son:

- Análisis de componentes principales
- Análisis de discriminante lineal

1.3 Modelo de clasificación

Una vez que se tiene la matriz con los vectores que representan cada mensaje m_i se procede a elegir un modelo de clasificación, entre los modelos clásicos y más utilizados están:

- Regresión Logística
- Naive-Bayes
- K-Vecinos más cercanos
- Random Forest
- Redes Neuronales Profundas

En general para la clasificación de texto se sule usar modelos clásicos como Naive-Bayes, sin embargo en los últimos años ha incrementado el uso de redenes neuronales Arif, 2021 y ensambladores Ikonomakis, Kotsiantis, and Tampakas, 2005.

1.4 Evaluación

El último paso es la evaluación del modelo, en esta parte se busca evaluar el performance que tuvo el modelo para clasificar mensajes que el modelo no ha visto, se clasifican utilizando el modelo en cuestión y se compara la clasificación contra la etiqueta real del mensaje. Algunos ejemplos de métricas son:

- Exactitud $Acc = \frac{TP+TN}{TP+TN+FN+FP}$
- Tasa de Verdaderos Positivos $TPR = \frac{TP}{TP+FN}$
- Tasa de Falsos Positivos $FPR = \frac{FP}{FP+TN}$
- ROC. La curva ROC busca graficar la TPR y FPR del clasificador utilizando diferentes umbrales de clasificación
- AUC. El AUC(Area Under ROC Curve) mide el area debajo de la curva ROC, puede tomar valores desde $[0, 1]$, siendo $.5$ el valor que toma un clasificador con un desempeño igual a lanzar una moneda.

2 Datos

Para el presente artículo se utilizara un conjunta de mensajes publicados en la comunidad de editores de wikipedia, (AI, 2018), cada uno de estos mensajes puede tener las etiquetas de la tabla 3, estas etiquetas no son mutuamente excluyentes, es decir, un mensaje puede ser etiquetado tanto como tóxico, severamente tóxico y discurso de odio.

El conjunto de mensajes se separa en test y entrenamiento, donde el conjunto de entrenamiento cuenta con $159,571$ mensajes y el set de test cuenta con 63978

Table 3: *Etiquetas*

Etiqueta
Tóxico
Severamente Tóxicos
Obscenidades
Amenazas
Insulto
Discurso de odio

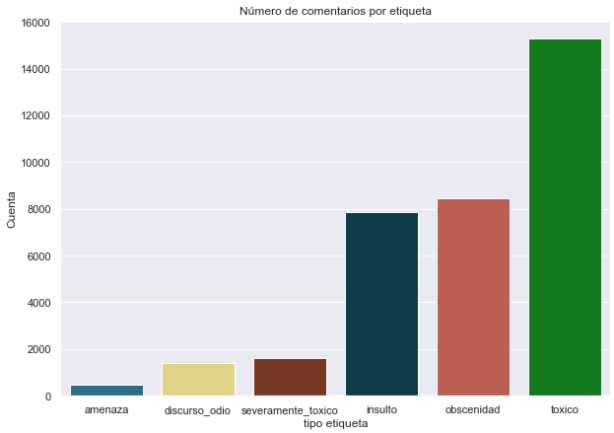

Figure 1: *Frecuencia etiquetas*

mensajes, como se mencionaba anteriormente, los mensajes pueden tener más de una etiqueta o ninguna. La frecuencia de las etiquetas se muestra en la figura 1, como podemos ver la etiqueta que más se presenta es la de *tóxico* y la que tiene menos frecuencia es la de *amenaza*, en general en los mensajes del conjunto de entrenamiento el 89% de los mensajes no tiene ninguna de las etiquetas, por lo que tenemos un conjunto de datos desbalanceados, en la figura 2 observamos que hay mensajes que pueden llegar a tener más de una etiqueta.

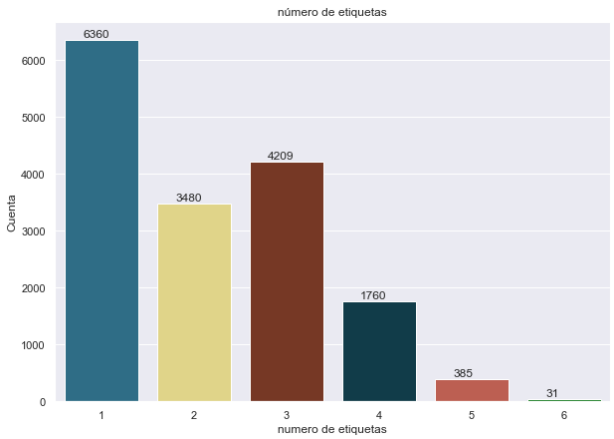

Figure 2: *Número de etiquetas*

3 Metodología

Para la clasificación se procede a seguir la metodología propuesta por Kowsari et al., 2019 utilizando dos formas de extraer las características de los mensajes y modelos de clasificación, se tomara metodología base la extracción de características por TF-IDF con un modelo de clasificación de regresión logística y extracción de características con la metodología *word2vec* y una red neuronal profunda con dos capas ocultas de 15 y 30 neuronas respectivamente.

Como se tienen 6 diferentes etiquetas que no son mutuamente excluyentes en los mensajes, para cada etiqueta se tiene que construir un clasificador binario, para cada uno de los clasificadores se calcula la exactitud y el AUC-Score, para poder comparar los dos modelos propuestos se utilizara el promedio de los AUC-Scores.

$$PromeidoAUC = \frac{1}{n}\sum_{i=1}^{n} AUC_i$$

donde n es el número de clasificadores que en nuestro caso son 6. Nuestra hipotesis es que el modelo con la extracción de características mediante el enfoque *word2vec* y una red neuronal profunda nos dará un mejor resultado que utilizando el TF-IDF.

4 Resultados

Una vez que se entrenan los modelos se hace la clasificación sobre el conjunto de evaluación, en la tabla 4 se puede observar el desempeño de los modelos de regresión logistica, en general se tiene un valor alto de exactitud para todas las etiquetas y el valor AUC promedio es de .9608, en la tabla 5 se observa el desemepeño del modelo de Red Neuronal, el valor AUC promedio es de .9602

Table 4: *Regresión Logistica|TF-IDF*

	Exactitud	AUC-Score
obscenidad	0.963	0.959
insulto	0.961	0.948
toxico	0.924	0.947
Severamente toxico	0.992	0.976
Discurso de odio	0.99	0.961
Amenaza	0.995	0.974

5 Discusión

En general se observa que la extracción de caractestiscas mediante en enfoque word2vec y red neuronal no ofrece un desempeño significativamente mejor que un modelo clásico de TF-IDF y regresión logistica, sin embargo, el desempeño en general de los dos modelos

Table 5: *Desempeño:Red Neuronal|Word2Vec*

	Exactitud	AUC-Score
obscenidad	0.957	0.956
insulto	0.955	0.949
toxico	0.928	0.943
Severamente toxico	0.99	0.977
Discurso de odio	0.987	0.958
Amenaza	0.995	0.976

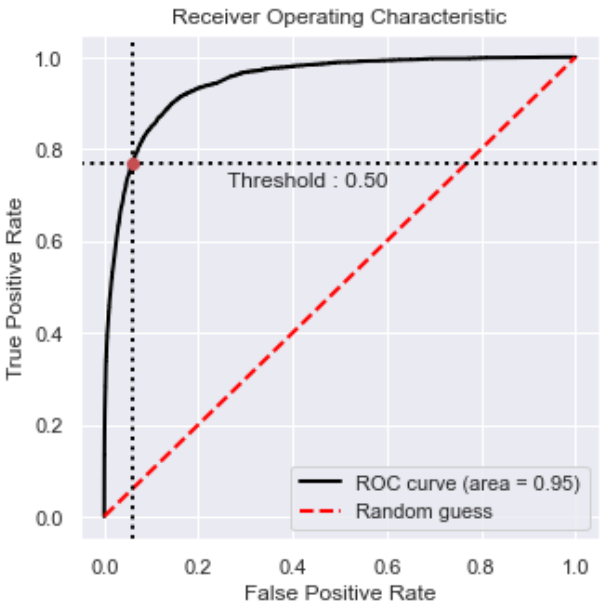

Figure 3: *Curva ROC-Etiqueta Tóxico-Regresión Logística*

tiene una alta exactitud y un valor alto de AUC score. Vemos que la etiqueta que tuve un menor desempeño es la de tóxico en ambos enfoques, con una exactitud del .924 y .928 respectivamente, para analizar con mas detalle esta etiqueta podemos observar en las figuras 3 4 que el valor del AUC Score es alto, por lo que podemos cambiar el umbral de decisión y no perderíamos mucha exactitud.

6 Conclusión

Podemos concluir con el trabajo realizado en este artículo que a pesar de que se utiliza un método de extracción de características más sofisticado, el desempeño de nuestro modelo no muestra una mejora significativa respecto al modelo base de regresión logística y extracción de características mediante TF-IDF, a su vez encontramos que ambos modelos tienen un desempeño sobresaliente, sin embargo no es significativamente mejor al obtenido en otros trabajos. Como futuro trabajo queda realizar un análisis de errores más detallado y experimentación de hiperparametros en el modelo de clasificación seleccionado para mejorar el desempeño del modelo elegido.

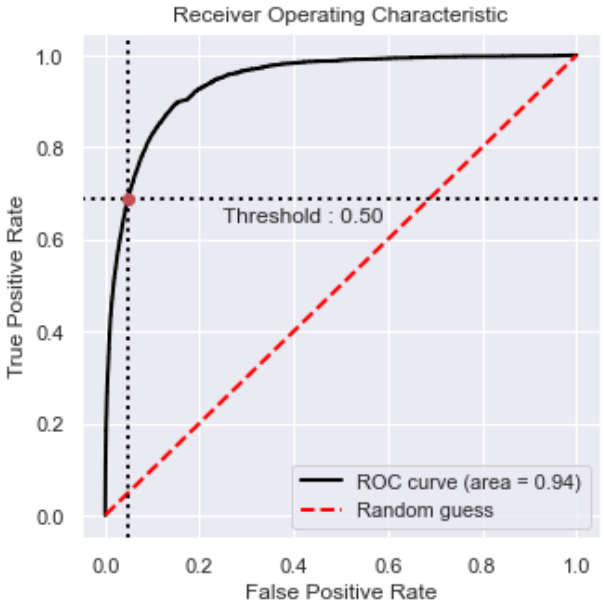

Figure 4: *Curva ROC-Etiqueta Tóxico-Red Neuronal*

Bibliografía

AI, Conversation (Jan. 2018). *Toxic Comment Classification Challenge*. URL: https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/overview.

Arif, Muhammad (2021). "A systematic review of machine learning algorithms in cyberbullying detection: future directions and challenges". In: *Journal of Information Security and Cybercrimes Research* 4.1, pp. 01–26.

Church, Kenneth Ward (2017). "Word2Vec". In: *Natural Language Engineering* 23.1, pp. 155–162.

"Endurecimiento frente a discurso de odio" (Apr. 1, 2019). In: *Internet Health Report*. URL: https://internethealthreport.org/2019/un-vistazo-al-endurecimiento-de-alemania-frente-al-discurso-de-odio/?lang=es (visited on 03/12/2017).

Ikonomakis, M, Sotiris Kotsiantis, and V Tampakas (2005). "Text classification using machine learning techniques." In: *WSEAS transactions on computers* 4.8, pp. 966–974.

Kowsari, Kamran et al. (2019). "Text classification algorithms: A survey". In: *Information* 10.4, p. 150.

Van Aken, Betty et al. (2018). "Challenges for toxic comment classification: An in-depth error analysis". In: *arXiv preprint arXiv:1809.07572*.