

La cara oculta de las opiniones

Guillermo Alonso Pacheco Rodrigues
dpto. Ciencias de la Computación e Inteligencia
Artificial
Universidad de Sevilla
Sevilla, España
guipacrod@alum.us.es

Antonio Barea Jiménez
dpto. Ciencias de la Computación e Inteligencia
Artificial
Universidad de Sevilla
Sevilla, España
antbarjim1@alum.us.es

El objetivo principal de este trabajo es predecir el sentimiento de un conjunto de tweets mediante un algoritmo de aprendizaje automático y comprobar y analizar la respuesta de este. Para ello, se elige el algoritmo que presenta las mejores prestaciones entre 3 posibles y se entrena con un conjunto de tweets de prueba los cuales han sido alterados por un grupo de técnicas de preprocesado para que su análisis sea lo más sencillo posible.

Tras el análisis, hemos podido observar que las respuestas no son del todo las esperadas debido a que la herramienta para analizar el sentimiento no toma la subjetividad y el contexto del tweet en cuenta para establecer la predicción.

Palabras Clave—Aprendizaje Automático, Naive Bayes, TextBlob, KNN, Preprocesamiento, Árboles de decisión, NLTK

I. INTRODUCCIÓN

El rápido desarrollo de las redes sociales y el aumento de la cantidad de opiniones en estas han creado la necesidad de desarrollar métodos eficientes y efectivos para conocer las connotaciones de los mensajes de los usuarios. En este contexto, surge el análisis de sentimientos o también denominado opinion mining el cual ya cuenta con diferentes librerías en Internet que se encargan de realizar dicho análisis, como es el caso de TextBlob. En este trabajo, nos centraremos en analizar los sentimientos de un conjunto de tweets utilizando TextBlob y posteriormente mediante un

método de aprendizaje automático predecir la connotación de varios tweets de personajes conocidos en internet. Nuestro objetivo es analizar cómo se comportan estos métodos de aprendizaje automático. Comenzaremos con una descripción general de la librería TextBlob.

A continuación, describimos el procedimiento de recopilación de datos y los métodos de pretratamiento utilizados en nuestro estudio. Luego presentaremos los métodos de aprendizaje automático utilizados y como hemos medido sus prestaciones para poder seleccionar 1 de ellos.

Finalmente, discutimos los resultados y conclusiones de nuestro estudio, destacando las capacidades y limitaciones del análisis de sentimientos usando la librería TextBlob.

II. PRELIMINARES

TextBlob es una librería de Python para el procesamiento del lenguaje natural [2]. Proporciona una interfaz API sencilla con un conjunto de características y funcionalidades como las siguientes:

- **Análisis de sentimientos.** Permite obtener la objetividad y subjetividad de un texto. Este método devuelve una tupla con dos valores numéricos, la polaridad y la subjetividad respectivamente. Mediante la polaridad podemos saber si el texto es positivo, neutro o negativo mientras que con la subjetividad podemos conocer si un texto es subjetivo u objetivo.

- Tokenización. Permite dividir un texto frases o palabra.
- Corrección ortográfica. Permite corregir errores ortográficos en un texto.
- Extracción de frases relevante. Permite extraer las frases más relevantes de un texto. Esto puede facilitar el resumen de un texto o extraer información clave.
- Obtener frecuencia de palabras y frases sustantivas. Permite obtener la cantidad de veces que aparece una frase sustantiva o una palabra en un texto.

Además de estas funcionalidades y características, TextBlob ofrece otras funcionalidades como la lematización de palabras, la detección de idiomas, entre otras.

III. METODOLOGÍA

Para realizar este estudio, hemos tomado un conjunto de tweets de la plataforma Kaggle. Este grupo, contiene diversas características adicionales a las que queremos como por ejemplo “target”. Esta columna nos indica que, si el tweet esta etiquetado con un 0 es negativo, con un 2 es neutral y con un 4 es positivo. Debido a que el conjunto de tweets original es demasiado grande (1.6 millones de tweets) hemos optado por seleccionar 500 tweets etiquetados como positivos y 500 tweets etiquetados como negativos. [1]

Para realizar las diversas tareas de preprocesamiento hemos utilizado las herramientas:

- Pandas. Herramienta utilizada para realizar tareas de manipulación y análisis de datos. [4]

- NLTK (Natural Language Toolkit). Herramienta utilizada para realizar tareas de procesamiento del lenguaje natural. [5]
- Re. Herramienta utilizada que proporciona funciones para trabajar con expresiones regulares.

Tras eliminar, las columnas que no nos sirven, definimos la función de limpiar texto. Esta función nos permite eliminar información poco relevante para el análisis de sentimientos de nuestros tweets como pueden ser las menciones, los RTs, los hashtags y los enlaces adjuntos al texto mediante el uso de la herramienta Re. Además, usando las funcionalidades que nos proporciona NLTK tokenizamos el texto y eliminamos las palabras comunes y poco relevantes conocidas como stop-words. Por último, lematizamos el texto y creamos una nueva columna en el archivo que almacena nuestros tweets con el texto de estos ya preprocesado.

Posteriormente, mediante una función llamada analisis_sentiment, que utiliza la herramienta TextBlob [2], obtenemos el valor numérico referente a la polaridad del tweet el cual nos indica si el texto analizado es positivo, neutro o negativo.

Una vez obtenida la polaridad de los tweets, definimos una función que divide el rango [-1,1] en 5 partes las cuáles serán las etiquetas que asignaremos a los tweets. Los rangos quedarían de la siguiente forma:

- [-1, -0.6); Hater.
- [-0.6, -0.2); Molesto
- [-0.2, 0.2); Neutro
- [0.2, 0.6); Contento

- [0.6, 1]; Muy feliz

Tras etiquetar cada tweet con su respectivo sentimiento, definimos dos funciones que se encargaran de comprobar que todos los tweets se han etiquetado correctamente y en el caso de que haya alguno sin sentimiento eliminarlo de nuestro conjunto de prueba.

Finalmente, con la ayuda de la librería Scikit-Learn [3], la cual nos proporciona herramientas para realizar algoritmos de aprendizaje automático y, además, nos proporciona herramientas para medir las prestaciones de estos algoritmos, calculamos qué método entre los 3 que elegimos que son: Naïve Bayes, KNN y Árboles de decisión, es el que mejores prestaciones nos proporciona para nuestro conjunto de pruebas. Para realizar la medición usamos el método de la validación cruzada de k iteraciones (4 iteraciones en nuestro caso) [7]. La validación cruzada toma como entrada un grupo de datos que ya han sido etiquetados junto con un modelo de clasificación. Divide el conjunto en k partes, en nuestro caso 4 partes, y crea un modelo para cada una de las partes que predice los resultados para comprobar si coinciden con las etiquetas. Finalmente, devuelve la precisión de cada una de las partes.

Una vez comprado que método era el mejor, procedimos a entrenar el modelo y posteriormente a realizar la predicción de los tweets de nuestros famosos seleccionados, en nuestro caso:

- Andrew Tate: Kickboxer retirado que trabaja en el mundo de las criptomonedas. Destaca por sus comentarios machistas, homofóbicos, misóginos, entre otros.
- Dwayne Johnson (The Rock): Actor y exluchador profesional. Destaca por su humildad y por la ausencia de polémicas que envuelven su persona.

Una vez obtenidos los resultados, los mostramos por pantalla mediante el uso de la librería matplotlib

[6] en forma de gráfico para una mejor visualización de estos.

IV. RESULTADOS

	tweet_original	tweet_transformado	etiqueta
0	is upset that he can't update his Facebook by ...	upset update Facebook texting might cry resu...	Neutro
1	@Kenichan I dived many times for the ball. Man...	dived many time ball Managed save 50 rest go ...	Contento
2	my whole body feels itchy and like its on fire	whole body feel itchy like fire	Contento
3	@Kwsidei not the whole crew	whole crew	Contento
4	Need a hug	Need hug	Neutro
5	@Tatiana_K nope they didn't have it	K nope	Neutro
6	@twittera que me muera ?	que muera	Neutro
7	spring break in plain city... it's snowing	spring break plain city snowing	Molesto
8	I just re-pierced my ears	repierced ear	Neutro
9	@caregiving I couldn't bear to watch it. And ...	bear watch thought UA loss embarrassing	Neutro

Fig. 1. Resultados tras preprocesar los tweets

En la figura 1 podemos observar cómo quedaría nuestro conjunto de prueba tras realizar todas las tareas de preprocesamiento, aplicar el análisis de sentimientos de TextBlob y realizar la asignación de la etiqueta correspondiente según la polaridad obtenida en el análisis.

```
Precisión de la iteración 1: 56.84%
Precisión de la iteración 2: 56.84%
Precisión de la iteración 3: 58.20%
Precisión de la iteración 4: 57.67%
Precisión media: 57.39%
```

Fig 2. Prestaciones del modelo Naïve Bayes

```
Precisión de la iteración 1: 57.89%
Precisión de la iteración 2: 60.00%
Precisión de la iteración 3: 58.20%
Precisión de la iteración 4: 59.26%
Precisión media: 58.84%
```

Fig 3. Prestaciones del modelo KNN

```

Precisión de la iteración 1: 62.11%
Precisión de la iteración 2: 61.05%
Precisión de la iteración 3: 61.90%
Precisión de la iteración 4: 58.73%
Precisión media: 60.95%

```

Fig 4. Prestaciones del modelo Árboles de decisión.

En las figuras 2, 3 y 4 podemos observar las prestaciones de cada modelo una vez realizada la validación cruzada. Las características para destacar de los modelos son:

- Para KNN hemos tomado 55 vecinos a considerar debido a que realizando diversas pruebas comprobamos que era el que mejor porcentaje obtenida para ese modelo tras aplicar la validación.
- Para los árboles de decisión hemos declarado que la profundidad máxima del árbol es 5.
- Para Naive Bayes hemos usado el modelo Multinomial.

Tras evaluar cada uno de los métodos decidimos entrenar el modelo de Arboles de decisión debido a que es el que mejor porcentaje medio presenta con respecto a los otros.

	tweet_andre_tate	tweets_transform	prediccion
0	If The Matrix has proof of you doing something...	Matrix proof something evidence charge punishm...	Neutro
1	I would never kill myself	would never kill	Neutro
2	The level of stress I tolerate daily would be ...	level stress tolerate daily would much 99 men ...	Neutro
3	Neva catch mi with a fassy headset pon mi face...	Neva catch mi fassy headset pon mi face chek b...	Neutro
4	You can give most men in life a guarantee of v...	give men life guarantee victory literal cheat ...	Neutro
5	The Matrix wants the best for you. The rules a...	Matrix want best rule made benefit people mak...	Neutro
6	A lot of people comment on the fact I have a s...	lot people comment fact swollen face black eye...	Neutro
7	The reason I work so hard when I make over 10 ...	reason work hard make 10 million dollar month ...	Contento
8	Hello @hollywills . You fooled absolutely nobo...	Hello fooled absolutely nobody guilty sexual ...	Neutro
9	Alex Jones was right. But its not the frogs th...	Alex Jones right frog want kid	Neutro
10	Linked in? Oh you mean... Gay?	Linked Oh mean Gay	Neutro

Fig 5. Tweets Andrew Tate

En la figura 5 podemos observar los tweets originales de Andrew Tate, los tweets transformados tras realizar el preprocesado y finalmente la predicción que realizado nuestro modelo entrenado a estos.

	tweet_therock	prediccion
0	Hope you've got your funderwear on... HOBBS IS B...	Neutro
1	Rest in Power Uncle Sheiky. Thank you for pavi...	Contento
2	After a successful XFL2023 season we look back...	Neutro
3	THANK YOU to our Mana Ohana for joining in on ...	Neutro
4	attention Orlando FL Bentonville AR Houston TX...	Neutro
5	The little Queen's have spoken - we LOVED the ...	Neutro
6	Dropped these dope new 305's inspired from my ...	Neutro
7	Grateful for the team we're assembling for MOA...	Neutro
8	94 days! Canes countdown!!	Neutro
9	The context of our @Teremana growth compared t...	Neutro
10	It's ironic his new comedy movie is called #Th...	Neutro

Fig 6. Tweets The Rock

En la figura 6 podemos observar los tweets originales de The Rock, junto con la predicción que realiza nuestro modelo a los tweets que ya han sido transformados, aunque estos no se puedan visualizar.

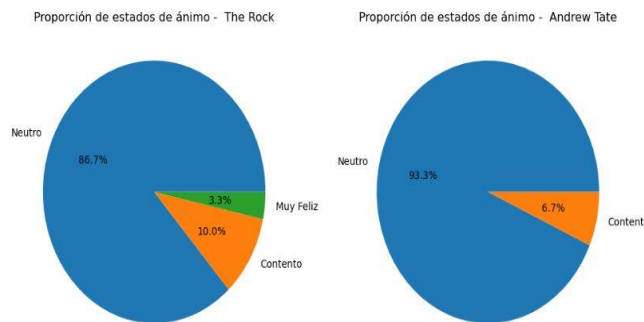


Fig 7. Resultados

En la figura 7 podemos observar los resultados obtenidos de los dos personajes públicos. Destacamos la gran cantidad de tweets neutros. Esto se debe principalmente a que TextBlob es una herramienta sencilla de clasificación debido a que solo etiqueta las palabras que se encuentran en un diccionario. Cada palabra tiene una cierta polaridad asignada la cual posteriormente nos indicara que etiqueta que debemos asignarle. El problema reside en que el diccionario no toma todas las palabras que existen, solo algunas, por lo que si un tweet aleatorio de un personaje público no contiene ninguna de estas palabras se clasificará como neutro. Además, tampoco toma la intención con la que van dirigidos ciertos tweets los cuales, aunque se clasifican como neutros, pueden ser perfectamente asignados con las etiquetas “Muy feliz” o en contra posición “Hater”. También, podemos observar que, aunque la mayoría de los tweets de nuestros seleccionados sean neutros podemos destacar que nuestro famoso elegido como positivo, tiene algunos tweets que son etiquetados como “Muy Feliz”.

Por último, aunque nosotros hayamos etiquetado a Andrew Tate con la etiqueta de “Hater” la predicción obtenida no concuerda con nuestra asignación. Esto esta relacionado con lo explicado anteriormente, debido a que, TextBlob no analiza la intencionalidad de los textos ni la subjetividad de estos, solo analiza un conjunto de palabras dentro de el por lo que, aunque nosotros sepamos que es sus tweets pueden llegar a ser dañinos la herramienta no es capaz de reconocerlo.

V. CONCLUSIONES

Como conclusión, en este estudio hemos realizado un análisis de sentimientos utilizando la herramienta TextBlob y diferentes algoritmos de aprendizaje automático. Hemos observado el funcionamiento de TextBlob y su capacidad para asignar polaridades a textos, así como la eficacia de los algoritmos de aprendizaje automático en la clasificación de sentimientos en un conjunto de tweets de famosos.

Sin embargo, hemos identificado áreas de mejora en nuestro estudio. En primer lugar, consideramos que mejorar nuestro dataset sería beneficioso para obtener una mayor variedad de valores de polaridad. Esto implica recopilar más ejemplos de textos con diferentes matices de sentimientos, lo que permitiría una asignación de etiquetas más precisa y detallada por parte de TextBlob.

Además, también hemos señalado la posibilidad de mejorar la herramienta TextBlob en sí misma. Una mejora sugerida es aumentar el tamaño del diccionario utilizado por TextBlob, lo que podría ampliar la cobertura de palabras y expresiones, y mejorar la asignación de polaridades en frases que actualmente se etiquetan como neutrales, aunque tengan un tono positivo o negativo.

Estas mejoras contribuirían a enriquecer la capacidad de análisis de sentimientos, permitiendo una mejor interpretación y clasificación de los textos. No obstante, es importante tener en cuenta que el análisis de sentimientos sigue siendo un desafío complejo debido a la subjetividad y la variabilidad en la expresión emocional, por lo que incluso con mejoras, es posible que existan limitaciones en la precisión y el alcance del análisis.

En conclusión, este estudio nos ha brindado una comprensión valiosa sobre el funcionamiento de TextBlob y los algoritmos de aprendizaje

automático en el análisis de sentimientos. Además, hemos identificado áreas específicas de mejora tanto en el dataset utilizado como en la herramienta TextBlob misma, que podrían potencialmente mejorar la precisión y la cobertura del análisis de sentimientos en futuros trabajos.

REFERENCIAS

- [1] Enlace de el dataset utilizado en el estudio. [Sentiment140 dataset with 1.6 million tweets | Kaggle](#)
- [2] Enlace a la documentación de TextBlob. [TextBlob: Simplified Text Processing — TextBlob 0.16.0 documentation](#)
- [3] Enlace a la documentación de Scikit-Learn. [scikit-learn: machine learning in Python — scikit-learn 1.2.2 documentation](#)
- [4] Enlace a la documentación de Pandas. [pandas - Python Data Analysis Library \(pydata.org\)](#)
- [5] Enlace a la documentación de NLTK. [NLTK :: Natural Language Toolkit](#)
- [6] Enlace a la documentación de Matplotlib. [Matplotlib — Visualization with Python](#)
- [7] Descripción del modelo de validación cruzada. [Modelo de validación cruzada: referencia del componente - Azure Machine Learning | Microsoft Learn](#)