Actividad NLP - 2.0 - Aplicación de análisis de sentimientos

Mario Alberto Castañeda Martínez - A01640152

Para esta actividad, se tiene como objetivo utilizar una arquitectura pre-entrenada de sentiment-analysis para saber cómo puede aplicarse en el análisis de sentimientos en texto. Por lo que se utilizará un dataset de Tweets para que el modelo pueda clasificar los tweets como "Positivo", "Negativo" y "Neutral". También se aplicará un código para saber el número de oraciones en cada tweet.

Primero se importan las librerías necesarias:

```
from google.colab import drive
drive.mount('/content/drive')
%cd "/content/drive/MyDrive/CONCENTRACION AI/data"
```

```
Drive already mounted at /content/drive; to attempt to forcibly
remount, call drive.mount("/content/drive", force_remount=True).
/content/drive/MyDrive/CONCENTRACION AI/data
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from textblob import TextBlob

import nltk
from nltk.tokenize import word_tokenize
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!

True
```

Se lee el archivo csv, en este caso, un dataset de diferentes tweets, 1,600,000 tweets.

En el dataset, hay una columna llamada "target_original" el cual ya tiene clasificado a los tweets, como positivo, neutral y negativo. Con esto se puede comparar el dataset original con el modelo creado a continuación:

```
df = pd.read_csv('training.1600000.processed.noemoticon.csv',
encoding='latin-1', header=None)

nombres_columnas = ['target_original', 'ids', 'date',
'flag','user','text']
df.columns = nombres_columnas
```

```
mapeo = {0: "negativo", 2: "neutral", 4: "positivo"}
df['target_original'] = df['target_original'].map(mapeo)

df.head()
```

```
  target_original         ids                              date      flag
\
0        negativo  1467810369  Mon Apr 06 22:19:45 PDT 2009  NO_QUERY

1        negativo  1467810672  Mon Apr 06 22:19:49 PDT 2009  NO_QUERY

2        negativo  1467810917  Mon Apr 06 22:19:53 PDT 2009  NO_QUERY

3        negativo  1467811184  Mon Apr 06 22:19:57 PDT 2009  NO_QUERY

4        negativo  1467811193  Mon Apr 06 22:19:57 PDT 2009  NO_QUERY


              user                                                text

0  _TheSpecialOne_  @switchfoot http://twitpic.com/2y1zl - Awww, t...

1    scotthamilton  is upset that he can't update his Facebook by ...

2         mattycus  @Kenichan I dived many times for the ball. Man...

3          ElleCTF    my whole body feels itchy and like its on fire

4           Karoli  @nationwideclass no, it's not behaving at all....
```

Se procede a generar el modelo de sentiment analysis utilizando la librería TextBlob:

```
def analizar_sentimiento_textblob(texto):
    analysis = TextBlob(texto)
    if analysis.sentiment.polarity > 0:
        return 'positivo'
    elif analysis.sentiment.polarity < 0:
        return 'negativo'
    else:
        return 'neutral'

# se aplica el modelo al dataset, específicamente a la parte del texto
de los tweets:
df['sentimiento'] = df['text'].apply(analizar_sentimiento_textblob)
```

Se visualizan los tweets con su interpretación del "sentiment analysis" a través de la columna "sentimiento", es importante recordar que la columna "target_original" es una columna que ya venía con el dataset:

```
df = df.drop(['ids','date','flag','user'], axis=1)
```

```
df.head(10)
```

```
  target_original
text  \
0        negativo  @switchfoot http://twitpic.com/2y1zl - Awww, t...

1        negativo  is upset that he can't update his Facebook by ...

2        negativo  @Kenichan I dived many times for the ball. Man...

3        negativo    my whole body feels itchy and like its on fire

4        negativo  @nationwideclass no, it's not behaving at all....

5        negativo                      @Kwesidei not the whole crew

6        negativo                                        Need a hug

7        negativo  @LOLTrish hey  long time no see! Yes.. Rains a...

8        negativo               @Tatiana_K nope they didn't have it

9        negativo                          @twittera que me muera ?


  sentimiento
0    positivo
1     neutral
2    positivo
3    positivo
4    negativo
5    positivo
6     neutral
7    positivo
8     neutral
9     neutral
```

De acuerdo con la columna "sentimiento" que utiliza el modelo generado de TextBlob, considero que se puede obsevar que hay ciertas palabras que determinan si el texto es positivo, negativo o neutral. Creo que el clasificador se enfoca más en palabras individuales que en toda la oración, pero en general creo que el clasificador es bueno.

Número de oraciones positivas, negativas o neutrales:

```
df['target_original'].value_counts()
```

```
negativo    800000
positivo    800000
Name: target_original, dtype: int64
```

Para la clasificación que ya viene en el dataset, se obtiene que la proporción es mitad y mitad, no hay neutrales.

```
df['sentimiento'].value_counts()
```

```
positivo    698007
neutral     568723
negativo    333270
Name: sentimiento, dtype: int64
```

Con el modelo generado de sentiment analysis, se observa que hay diferentes proporciones de los tipos de tweets.

Después de clasificar a los tweets de acuerdo si son positivos, negativos o neutrales, ahora se procede con obtener el número de oraciones en cada tweet, cada oración se cuenta después de cada punto:

```
def contar_oraciones(texto):
    oraciones = texto.split('.')
    oraciones = [oracion.strip() for oracion in oraciones if
oracion.strip()]
    return len(oraciones)

df['num_oraciones'] = df['text'].apply(contar_oraciones)

df.head(10)
```

```
  target_original
text  \
0        negativo  @switchfoot http://twitpic.com/2y1zl - Awww, t...

1        negativo  is upset that he can't update his Facebook by ...

2        negativo  @Kenichan I dived many times for the ball. Man...

3        negativo    my whole body feels itchy and like its on fire

4        negativo  @nationwideclass no, it's not behaving at all....

5        negativo                      @Kwesidei not the whole crew

6        negativo                                        Need a hug

7        negativo  @LOLTrish hey  long time no see! Yes.. Rains a...

8        negativo               @Tatiana_K nope they didn't have it

9        negativo                          @twittera que me muera ?


  sentimiento  num_oraciones
```

```
0    positivo              4
1     neutral              3
2    positivo              2
3    positivo              1
4    negativo              3
5    positivo              1
6     neutral              1
7    positivo              2
8     neutral              1
9     neutral              1
```

Como se muestra en el dataset generado con la columna "num_oraciones", se crea una columna donde se observan la oraciones en cada tweet o texto, la manera en que se hizo el modelo, es que después de cada punto, significa una oración. En los tweets se observa que en la mayoría de los casos se contabilizan correctamente el número de oraciones. Se puede decir que en el primer tweet tiene un poco de confusión ya que toma el punto en el link como una oración. Sin embargo a parte de esto se observa que el modelo trabaja bien.

Ahora se va a comparar la columna "target_original" con la columna "sentimiento" y obtener una precisión entre los datos originales y los predichos por el modelo:

```
from sklearn.metrics import accuracy_score

precision = accuracy_score(df['target_original'], df['sentimiento'])

print(f'Precisión del modelo: {precision:.2f}')
```

```
Precisión del modelo: 0.44
```

Se tiene un accuracy del 44%, no existe un parecido fuerte de sentimientos de datos originales con los sentimientos generados por el modelo.

Ejemplo sencillo de cómo funcionan los modelos generados:

```
#Texto con dos oraciones positivas:
prueba = 'I would rather be an ant that a human. That would be
awesome.'
#Aplicación de modelo de sentiment analysis
analizar_sentimiento_textblob(prueba)
```

```
{"type":"string"}
```

```
# Aplicación de conteo de oraciones:
contar_oraciones(prueba)
```

```
2
```

A manera de conclusión, puedo considerar que la forma en que funciona el modelo de sentiment analysis que se generó es muy basada en las palabras del texto, por lo que puede que no se tome en cuenta el contexto a la hora de clasificar el tweet, puede que por eso, el

parecido del modelo y los sentimientos originales en el dataset no sea tan fuerte. También es necesario decir que el dataset original no tiene a ningún tweet clasificado como "neutral" por lo que esto puede afectar a la manera en que se clasificaron los tweets originalmente. También, se genera el número de oraciones de forma correcta en la mayoría de ocasiones, salvo por páginas web en el texto, donde existe un poco confusión.

Finalmente, se genera un pequeño ejemplo de cómo funciona el modelo de sentiment analysis y de conteo de oraciones, para probar la efectividad del modelo.