Sentiment_Analisis_Twitter1mil

October 28, 2023

Estefanía Pérez Yeo A01639270

---

1 Actividad NLP - 2.0 - Aplicación de análisis de sentimientos

1.1 Carga de dataset

La carga de datos a utilizar será un dataset de Twitter proveniente de Sentiment140 el cual se menciona que cuenta con 1 millón de tweets clasificados, donde también se pueden encontrar en Google Drive y en la página de Stanford provenientes de la liga de Sentiment140.

La estructura dentro de las columnas del dataset se ve representada de la siguiente manera:

1. Polaridad del tweet (negativo, positivo).
2. ID del tweet.
3. Fecha.
4. La consulta, o en caso contrario se utiliza el valor 'NO_QUERY'.
5. Usuario.
6. Texto del tweet.

[96]:
```
import pandas as pd
```

[97]:
```
from google.colab import drive
drive.mount('/content/drive')
```

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

[98]:
```
#Carga de datos en análisis de sentimientos con dataset de Twitter
column_names = ["polarity", "tweet_id", "date", "query", "user", "tweet_text"]
df = pd.read_csv('/content/drive/MyDrive/AI datos/trainingandtestdata/training.
↪1600000.processed.noemoticon.csv', encoding='ISO-8859-1', header=None,␣
↪names=column_names)
```

[99]:
```
df.head()
```

[99]:

	polarity	tweet_id	date	query	\
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	

```
2         0  1467810917  Mon Apr 06 22:19:53 PDT 2009  NO_QUERY
3         0  1467811184  Mon Apr 06 22:19:57 PDT 2009  NO_QUERY
4         0  1467811193  Mon Apr 06 22:19:57 PDT 2009  NO_QUERY

              user                                         tweet_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...
```

[100]:
```
df.shape
```

[100]:
```
(1600000, 6)
```

Eliminar filas de forma aleatoria ya que son muchos datos

[101]:
```
# Número de filas que quieres eliminar
n = 1600000 - 5000

# Selecciona 'n' filas aleatorias
drop_indices = df.sample(n).index

# Elimina las filas seleccionadas
df = df.drop(drop_indices)
```

[102]:
```
df.shape
```

[102]:
```
(5000, 6)
```

1.2 Procesamiento

A partir de la biblioteca de 'trnasformers' de Hugging Face, se utilizará el modelo pre entrenado de BertTokenizer, el cual divide el texto en secciones más pequeñas de acuerdo con las especificaciones del modelo BERT, el cual no distingue entre mayúsculas y minúsculas.

[103]:
```
!pip install transformers
from transformers import BertTokenizer
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-
packages (4.34.1)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-
packages (from transformers) (3.12.4)
Requirement already satisfied: huggingface-hub<1.0,>=0.16.4 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.17.3)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-
packages (from transformers) (1.23.5)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from transformers) (23.2)
```

```
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-
packages (from transformers) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.10/dist-packages (from transformers) (2023.6.3)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-
packages (from transformers) (2.31.0)
Requirement already satisfied: tokenizers<0.15,>=0.14 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.14.1)
Requirement already satisfied: safetensors>=0.3.1 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.4.0)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-
packages (from transformers) (4.66.1)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages
(from huggingface-hub<1.0,>=0.16.4->transformers) (2023.6.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.10/dist-packages (from huggingface-
hub<1.0,>=0.16.4->transformers) (4.5.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.3.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests->transformers) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers)
(2023.7.22)
```

[104]:
```
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
```

Gracias al 'truncation=True' aquellos tokens para un tweet mayores al valor de 'max_lenght' que en este caso es 128, serán eliminados debido a que excede la longitud máxima.

[105]:
```
df['tokens'] = df['tweet_text'].apply(lambda x: tokenizer.encode(x,␣
↪add_special_tokens=True, truncation=True, max_length=128))
```

1.3 Aplicación del modelo

Para obtener resultados más precisos, se utiliza el modelo BERT.

Cada polaridad es representada por los siguientes números: (0 = negative, 4 = positive); cabe mencionar que en el caso de BERT este hace sus predicciones siguiendo el formato 0, 1 para "negativo" y "positivo"; para solucionar esto se crea la siguiente función que ayudar en las predicciones.

[106]:
```
def map_sentiment(output):
    sentiment_map = {0: 0, 1: 4}
    return sentiment_map.get(output, 1)  # Default si hay algún valor inesperado
```

[107]:
```
import torch
from transformers import BertForSequenceClassification
```

```
model = BertForSequenceClassification.from_pretrained('bert-base-uncased')
model.eval()
```

```
Some weights of BertForSequenceClassification were not initialized from the
model checkpoint at bert-base-uncased and are newly initialized:
['classifier.weight', 'classifier.bias']
You should probably TRAIN this model on a down-stream task to be able to use it
for predictions and inference.
```

[107]:
```
BertForSequenceClassification(
  (bert): BertModel(
    (embeddings): BertEmbeddings(
      (word_embeddings): Embedding(30522, 768, padding_idx=0)
      (position_embeddings): Embedding(512, 768)
      (token_type_embeddings): Embedding(2, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (encoder): BertEncoder(
      (layer): ModuleList(
        (0-11): 12 x BertLayer(
          (attention): BertAttention(
            (self): BertSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
            (output): BertSelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
          (intermediate): BertIntermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): BertOutput(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
      )
    )
```

```
    (pooler): BertPooler(
      (dense): Linear(in_features=768, out_features=768, bias=True)
      (activation): Tanh()
    )
  )
  (dropout): Dropout(p=0.1, inplace=False)
  (classifier): Linear(in_features=768, out_features=2, bias=True)
)
```

[108]:
```
from tqdm import tqdm
tqdm.pandas()

# Función para inferir el sentimiento
def infer_sentiment(tokens):
    with torch.no_grad():
        outputs = model(torch.tensor([tokens]))
        logits = outputs[0]
        return logits.argmax().item()

# Usar tqdm para mostrar una barra de progreso
df['predicted_sentiment'] = df['tokens'].progress_apply(lambda x:␣
 ↪infer_sentiment(x))
```

```
100%|| 5000/5000 [14:47<00:00,  5.63it/s]
```

Ajuste de predicciones por cuestiones de formato

[109]:
```
df['predicted_sentiment'] = df['predicted_sentiment'].apply(map_sentiment)
```

1.4 Análisis de resultados

[110]:
```
print(df['polarity'])
```

```
177        0
680        0
807        0
1484       0
1795       0
          ..
1599147    4
1599153    4
1599356    4
1599399    4
1599670    4
Name: polarity, Length: 5000, dtype: int64
```

[111]:
```
print(df['predicted_sentiment'])
```

```
177        4
```

```
680         0
807         4
1484        0
1795        4
           ..
1599147     4
1599153     4
1599356     4
1599399     4
1599670     4
Name: predicted_sentiment, Length: 5000, dtype: int64
```

Una vez obtenidos los resultos, estos pueden ser analizados con el original

Recordando que en df['polarity'] + negativo = 0 + positivo = 4

A continuación se pueden observar 5 ejemplos correspondientes a cada polaridad del dataframe

[116]:
```
positive_samples = df[df['polarity'] == 4].head(5)
negative_samples = df[df['polarity'] == 0].head(5)
```

Positivas

[117]:
```
positive_samples
```

[117]:
```
        polarity    tweet_id                          date     query  \
800650         4  1467936634  Mon Apr 06 22:53:43 PDT 2009  NO_QUERY
801385         4  1468069464  Mon Apr 06 23:32:56 PDT 2009  NO_QUERY
802119         4  1468180213  Tue Apr 07 00:09:06 PDT 2009  NO_QUERY
802292         4  1468209775  Tue Apr 07 00:18:49 PDT 2009  NO_QUERY
802341         4  1468221326  Tue Apr 07 00:22:46 PDT 2009  NO_QUERY

              user                                         tweet_text  \
800650  piajimenez             Uploading photos first before leaving
801385    killemil  Oh my. My new logo has been featured at http:/...
802119   renhuijun  RULE OF LIFE#1: STOP SAYING NO WHEN OFFERED CO...
802292  MyChelle22  @souljaboytellem http://twitpic.com/2y506 - aw...
802341        Yema                www.toutlemondesurcf.blogspot.com

                                                   tokens  predicted_sentiment
800650    [101, 2039, 18570, 7760, 2034, 2077, 2975, 102]                    4
801385  [101, 2821, 2026, 1012, 2026, 2047, 8154, 2038...                    4
802119  [101, 3627, 1997, 2166, 1001, 1015, 1024, 2644...                    4
802292  [101, 1030, 3969, 3900, 11097, 23567, 6633, 82...                    0
802341  [101, 7479, 1012, 2000, 4904, 16930, 15422, 22...                    4
```

Negativas

[118]:
```
negative_samples
```

[118]:

	polarity	tweet_id	date	query	\
177	0	1467857221	Mon Apr 06 22:31:54 PDT 2009	NO_QUERY	
680	0	1467983247	Mon Apr 06 23:06:50 PDT 2009	NO_QUERY	
807	0	1468011315	Mon Apr 06 23:15:02 PDT 2009	NO_QUERY	
1484	0	1468162941	Tue Apr 07 00:03:32 PDT 2009	NO_QUERY	
1795	0	1468234554	Tue Apr 07 00:27:20 PDT 2009	NO_QUERY	

	user	tweet_text	\
177	sarah_katie	I'm not still up I swear. Why do I keep losing...	
680	MichaelPe	@FollowSavvy I never found her. everytime I cl...	
807	carebearCC26	getting ready to clean the house from top to b...	
1484	Hollywood_Trey	@shalinique For saying 2 may change up ur twit...	
1795	donaji23	@Jamzeee I knowwwwww I sukkkk !!... .. Take...	

	tokens	predicted_sentiment
177	[101, 1045, 1005, 1049, 2025, 2145, 2039, 1045...	4
680	[101, 1030, 4076, 11431, 10736, 1045, 2196, 21...	0
807	[101, 2893, 3201, 2000, 4550, 1996, 2160, 2013...	4
1484	[101, 1030, 21146, 22153, 4226, 2005, 3038, 10...	0
1795	[101, 1030, 9389, 23940, 2063, 1045, 2113, 286...	4